

DYNAMIC ENGINEERING

150 DuBois St., Suite C, Santa Cruz, CA. 95060

831-457-8891, Fax 831-457-4793

sales@dyneng.com

www.dyneng.com

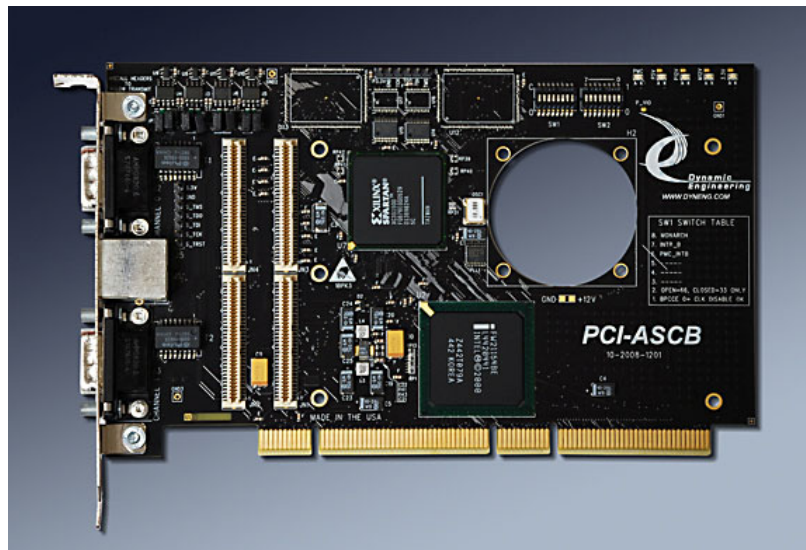
Est. 1988

User Manual

PCI-ASCB

Avionics Bus Tester

2-channel Bi-directional
Manchester Encoded
PCI Board and PMC Module Carrier



Revision B1

Corresponding Hardware: Revision B

10-2008-1202

Corresponding Firmware: Revision L

**PCI-ASCB - Avionics Standard
Communication Bus Rev. D Tester
2-Channel Dual-Bus Bi-Directional
Manchester Encoded PCI Board
and PMC Module Carrier**

Dynamic Engineering
150 DuBois St., Suite C
Santa Cruz, CA 95060
831-457-8891
FAX: 831-457-4793

©2009-2010 by Dynamic Engineering.
Other trademarks and registered trademarks are
owned by their respective manufactures.
Manual Revision B1. Revised March 16, 2010.

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

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

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

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

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



Table of Contents

PRODUCT DESCRIPTION	6
THEORY OF OPERATION	8
ADDRESS MAP	10
PROGRAMMING	11
REGISTER DEFINITIONS	12
BASE_CONTROL	12
BASE_STATUS	14
BASE_FRAME_END	15
BASE_FRAME_INT	15
BASE_FRAME_CNT	16
BASE_PKT_DELAY	16
CHAN_CONTROL	17
CHAN_STATUS	19
CHAN_TX_DMA	21
CHAN_RX_DMA	21
CHAN_TX_INDX	22
CHAN_RX_INDX	22
CHAN_RX_NEXT	23
JTAG_STATUS	24
JTAG_MODE/DATA	25
Dual-Port RAM Configuration and Packet Control/Status	26
TX_MEM_CONTROL	26
RX_MEM_STATUS	26
PCI-ASCB I/O CHANNEL PIN ASSIGNMENT	27
APPLICATIONS GUIDE	29
Interfacing	29
CONSTRUCTION AND RELIABILITY	30
THERMAL CONSIDERATIONS	30
WARRANTY AND REPAIR	31
Service Policy	31
Out of Warranty Repairs	31
For Service Contact:	31

SPECIFICATIONS

32

ORDER INFORMATION

33



List of Figures

FIGURE 1	PCI-ASCB BLOCK DIAGRAM	6
FIGURE 2	PCI-ASCB MANCHESTER TIMING DIAGRAM	8
FIGURE 3	PCI-ASCB INTERNAL ADDRESS MAP	10
FIGURE 4	PCI-ASCB BASE CONTROL REGISTER BIT MAP	12
FIGURE 5	PCI-ASCB STATUS REGISTER	14
FIGURE 6	PCI-ASCB FRAME END COUNT REGISTER	15
FIGURE 7	PCI-ASCB FRAME INTERRUPT COUNT REGISTER	15
FIGURE 8	PCI-ASCB CURRENT FRAME COUNT REGISTER	16
FIGURE 9	PCI-ASCB INTER-PACKET DELAY COUNT	16
FIGURE 10	PCI-ASCB CHANNEL CONTROL REGISTER	17
FIGURE 11	PCI-ASCB CHANNEL STATUS REGISTER	19
FIGURE 12	PCI-ASCB CHANNEL INPUT DMA POINTER PORT	21
FIGURE 13	PCI-ASCB CHANNEL OUTPUT DMA POINTER PORT	21
FIGURE 14	PCI-ASCB CHANNEL TRANSMIT INDEX REGISTER	22
FIGURE 15	PCI-ASCB CHANNEL RECEIVE INDEX REGISTER	22
FIGURE 16	PCI-ASCB CHANNEL RECEIVE INDEX REGISTER	23
FIGURE 17	PCI-ASCB JTAG STATUS REGISTER	24
FIGURE 18	PCI-ASCB JTAG FIFO PORTS	25
FIGURE 19	PCI-ASCB BUS INTERFACE PINOUTS	27
FIGURE 20	PCI-ASCB TRANSMITTER FAIL-SAFE SHUNTS	27
FIGURE 21	PCI-ASCB ETHERNET PINOUTS	28

Product Description

The PCI-ASCB is a special-purpose PCI board designed to interface with the Avionics Standard Communication Bus Revision D for testing avionics components.

The PCI-ASCB board has an on-board PCI bridge that creates a local two-slot PCI bus. Slot 0 is a PMC slot intended for an optional PrPMC (Processor PCI Mezzanine Card); slot 1 interfaces with a XC3S1500 Xilinx FPGA that decodes the PCI bus signals and implements a two-channel ASCB-D bus interface.

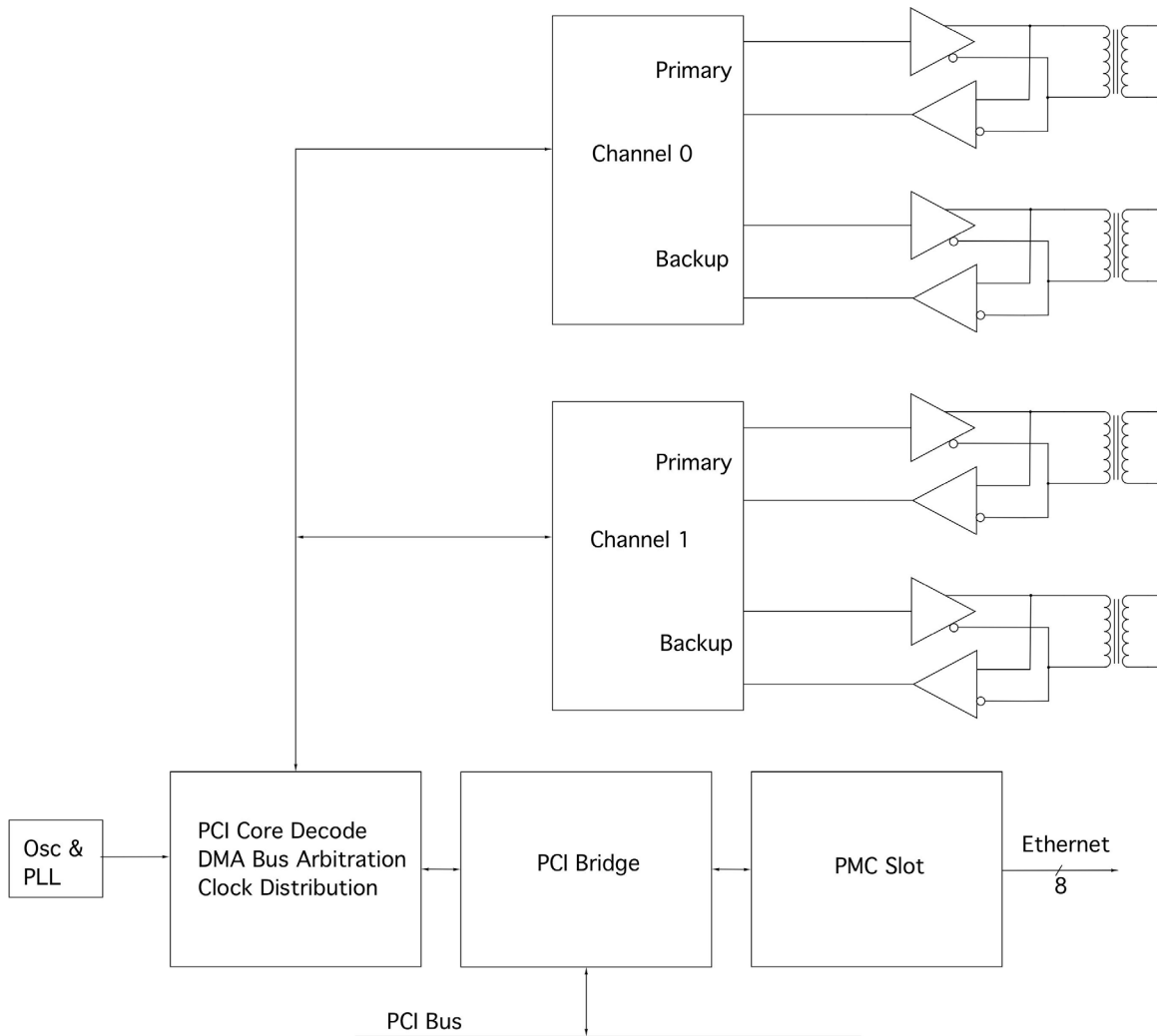


FIGURE 1

PCI-ASCB BLOCK DIAGRAM

Each ASCB-D bus connects to the board through a 9-pin D-connector in the PCI front panel. Along with the primary and back-up transformer-coupled 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.

Each transmitter differential output is connected to its transformer through a pair of fail-safe signal disconnect shunts to ensure that no unwanted bus transmissions can occur from the PCI-ASCB when the shunts are not installed. (See figure 20)

The Pn4 PMC connector connects to a front panel mounted RJ45 Ethernet connector for the use of the PrPMC. The ASCB interrupt can be configured to be serviced by either the local PrPMC processor or the PCI host computer.

The PCI-ASCB has an on-board oscillator and programmable PLL to supply the frame timing clock and the transmit and receive I/O clocks. Each ASCB-D channel has two separate 16 Kbyte frame data dual-port RAM memory blocks for transmit and receive data storage. Separate bus-master scatter-gather DMA is implemented to write to the transmit memory and read from the receive memory. Channel DMA access to the PCI bus is determined by a four-input arbiter block at the base level.

Theory of Operation

The PCI-ASCB features a Xilinx FPGA. 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. The PCI interface to the host CPU is controlled by a logic block within the Xilinx. The register and RAM-block addresses are all offsets from base addresses, which are assigned by the system when the PCI bus is configured.

The PCI-ASCB bus protocol is derived from 10base-T Ethernet with a 64-bit preamble/sync pattern preceding each data packet and a 32-bit Ethernet CRC word appended to the end. A delay of at least 9.6 microseconds separates data packets. Data is read from the transmit dual-port RAM in 32-bit words and shifted out serially least-significant-bit first, but most-significant-byte first i.e. bit 24, 25, 26,..., 31, 16,..., 23, 8,..., 15, 0,..., 7. The data is then Manchester encoded (see Figure 2) and sent to the output driver.

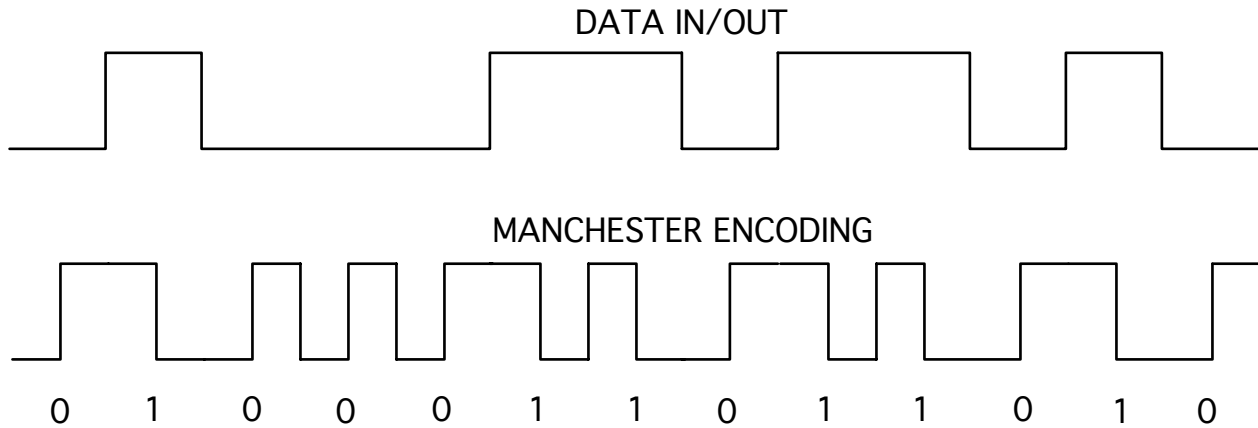


FIGURE 2 PCI-ASCB MANCHESTER TIMING DIAGRAM

Similarly, Manchester encoded data is received into the serial input where the clock is recovered and the data decoded. Once the sync pattern is recognized, the data is accumulated in a 32-bit shift-register and written into receive dual-port RAM with the bytes again reversed. As the data arrives, the CRC of the received data is calculated and when reception stops, this CRC is compared to the last 32-bit word received (the transmitted CRC). These CRC words should match if data was transferred error-free. The byte-swapping process was found to be necessary in order to successfully receive data from an actual 10base-T Ethernet connection with correct CRC values.

Each packet written do the receive dual-port RAM is preceded by two 32-bit status words. When a received packet begins, two 32-bit addresses in the receiver memory are skipped and data storage begins on the third word address. The fields in the status words are evaluated during the packet reception and after the packet has completed

the two status words are written to the addresses that were previously skipped.

Similarly, each packet stored in the transmit dual-port RAM is preceded by two 32-bit control words that determine how and when the data will be sent. Packet transmission is scheduled relative to a frame-clock counter running at 12.5 MHz. When the specified packet start-count is reached by the frame-clock counter, the packet transmission will begin, provided the start enable control bit is set. Data transmission will continue until the packet data is exhausted or the stop-count is reached. If the next packet has no valid start count and the previous stop-count has not been reached, data transmission will continue with the next packet after the inter-packet delay. This delay is nominally 9.6 usecs, but is adjustable by writing to the BASE_PKT_DELAY register. Packet transmission will continue until a new valid start-count is read, the previous stop-count is reached or transmit data is exhausted.

The frame-length timing is controlled with the BASE_FRAME_END register. This register contains the frame-clock count that causes the frame-clock counter to roll-over to zero. When this occurs, an 80 nanosecond pulse is output on the Frame signal of each 9-pin bus connector.

The transmitter and receiver dual-port RAMs are used as circular buffers to continuously send and receive data. When the end of the 16 Kbyte block is reached, reading or writing of data continues at address zero.

Scatter-gather DMA is provided for in this design. Once the physical address of the first chaining descriptor is written to the DMA pointer register, the interface will read a 16-byte block from this location. The first four bytes comprise a long-word indicating the physical address of the first block of the IO buffer passed to the read or write call. The next four bytes represent a long-word that is the local RAM address to start writing the memory block into. The next four bytes represent a long-word indicating the length of that block (only the lower 22 bits are valid). The final four bytes are a long-word indicating the physical address of the next chaining descriptor along with two flag bits, in bit position 0 and 1. Bit zero is set to one if this descriptor is the last in the chain. Bit one is set to one if the IO transfer is from the PCI-ASCB board to host memory, and zero if the transfer is from memory to the board. These bits are then replaced with zeros to determine the address of the next descriptor, if there is one.

Various interrupts are supported by the PCI-ASCB. An interrupt can be configured to occur at the end of a transmitted or received packet, when a write or read DMA transfer completes, or at a specific frame-clock count. All interrupts are individually maskable, and individual channel master interrupt enables are provided to disable transmit and receive interrupts simultaneously.

Address Map

BAR 0:	Offset	Reg num	Function
BASE_CNTRL	0x0000	0	Base control register
BASE_STATUS	0x0004	1	Base status register
BASE_FRAME_END	0x0008	2	Base frame clock end count register
BASE_FRAME_INT	0x000C	3	Base frame clock interrupt count register
BASE_FRAME_CNT	0x0010	4	Base current frame count read port
BASE_PKT_DELAY	0x0014	5	Base inter-packet delay count register
CHAN_0_CNTRL	0x0018	6	Channel 0 control register
CHAN_0_STATUS	0x001C	7	Channel 0 status register
CHAN_0_TX_DMA	0x0020	8	Channel 0 write DMA start register
CHAN_0_RX_DMA	0x0024	9	Channel 0 read DMA start register
CHAN_0_TX_INDXX	0x0028	10	Channel 0 Transmitter index read port
CHAN_0_RX_INDXX	0x002C	11	Channel 0 Receiver index read port
CHAN_0_RX_NEXT	0x0030	12	Channel 0 Receiver start next packet read port
CHAN_1_CNTRL	0x0034	13	Channel 1 control register
CHAN_1_STATUS	0x0038	14	Channel 1 status register
CHAN_1_TX_DMA	0x003C	15	Channel 1 write DMA start register
CHAN_1_RX_DMA	0x0040	16	Channel 1 read DMA start register
CHAN_1_TX_INDXX	0x0044	17	Channel 1 Transmitter index read port
CHAN_1_RX_INDXX	0x0048	18	Channel 1 Receiver index read port
CHAN_1_RX_NEXT	0x004C	19	Channel 1 Receiver start next packet read port
JTAG_STATUS	0x0050	20	JTAG status read
JTAG_MODE	0x0050	20	TMS FIFO write
JTAG_DATA	0x0054	21	TDI FIFO write/TDO FIFO read
BAR 1: I/O channel RAM blocks are mapped to this space for single-word accesses			
CHAN_0_TX_RAM	0x0000 – 0x3FFC		
CHAN_0_RX_RAM	0x4000 – 0x7FFC		
CHAN_1_TX_RAM	0x8000 – 0xBFFC		
CHAN_1_RX_RAM	0xC000 – 0xFFFF		

FIGURE 3

PCI-ASCB INTERNAL ADDRESS MAP

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

Programming

Programming the PCI-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. For example in WindowsNT there is a system registry, which is used to identify the resident hardware.

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 packet control and data into the appropriate channel dual-port RAM, configure the frame-clock counter 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 packet has been started and completed. If more than one interrupt is enabled, then the software 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 Status register definitions below for more information regarding the exact sequencing and interrupt definitions.

The VendorId = 0x10EE. The CardId = 0x0037. Current FLASH revision = 0x0C

Register Definitions

BASE_CONTROL

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

Control Register	
DATA BIT	DESCRIPTION
31-24	Spare
23	PLL Sdat output
22	PLL S2
21	PLL Sclk
20	PLL Enable
19-8	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 4 PCI-ASCB BASE CONTROL REGISTER BIT MAP

All bits are active high and are reset on power-up or reset command except the PLL enable bit which resets to a high state for PLL initialization.

PLL Sclk/Sdata output: These signals are used to program the PLL over the I2C serial interface. SCLK is always an output whereas sdata is bi-directional.

PLL S2 output: This is an additional control line to the PLL that can be used to select additional pre-programmed frequencies.

PLL Enable: When this bit is set to a one, the sdat output signal is enabled. When set to a zero the sdat signal is tri-stated by the Xilinx.

The register bits for PLL Enable, PLL S2 and PLL Sclk are unidirectional from the Xilinx to the PLL – always driven. PLL Sdat is a bi-directional signal (open drain). The Sdat register bit, when written low and enabled will be reflected with a low on the Sdat signal to the PLL. When Sdat is taken high or disabled the Sdat signal will be tri-stated by the Xilinx, and can be driven by the PLL. The Sdat register bit when read reflects the state of the internal Sdat bit but may not be in the same state as the signal between the Xilinx and PLL. To determine the state of the external Sdat signal, read the bit from the BASE_STATUS register described below.

The PLL is a separate device controlled by the Xilinx. The PLL has a fairly complex programming requirement which is simplified by using the Cypress® frequency descriptor software <http://www.dyneng.com/CyberClocks.zip>, and then programming the resulting control words into the PLL using the PLL Control ports. The interface can be further simplified by using the Dynamic Engineering Driver to take care of the programming requirements.

If you are writing your own driver, contact Dynamic Engineering and we can send you a file with code excerpts from our driver and test software that cover each step of the process from parsing the .jed file generated by CyberClocks to the low-level bit manipulation of the I²C bus.

Channel 0/1 Tx/Rx Interrupt Enable: These channel interrupt enable bits are replicated here for convenience to allow a single point of control for all user interrupts. When a one is written to one of these bits, the corresponding interrupt source is enabled to cause a system interrupt. When a zero is written, the corresponding interrupt is disabled.

Frame Count Loop Enable: Writing a one to this bit causes the frame-clock counter to roll-over to zero once it has reached the count specified in the BASE_FRAME_END register. Writing a zero to this register bit causes the frame-clock counter to stop when it reaches the specified end count.

Frame Interrupt Enable: Writing a one to this bit causes the frame-clock counter to cause an interrupt once it has reached the count specified in the BASE_FRAME_INT register. Writing a zero to this register bit disables the frame count interrupt.

Frame Count Clear: Writing a one to this bit clears the frame-clock counter to zero. A zero must be written to this register bit when normal frame-clock operation is desired.

Frame Count Enable: Writing a one to this bit allows the frame-clock counter to begin counting. When a zero is written to this register the frame-clock counter will stop counting.

BASE_STATUS

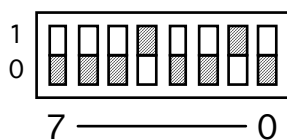
[0x04] ASCB-D Base Status Register (read status/write clear)

Status Register	
DATA BIT	DESCRIPTION
31	Interrupt Active (read only)
30-24	Spare
23	PLL Sdat input (read only)
22	Spare
21	JTAG Error (read/write clear)
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 5

PCI-ASCB STATUS REGISTER

The User Switch read port returns the board's user-specified identification bits. These bits are connected to an eight-position dip-switch. The switches allow individual boards to be distinguished for software or other configuration control purposes.



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.

Xilinx Flash Revision: The flash revision will be updated as features are added or revisions made. See the programming section for the current revision.

Channel 0/1 Tx/Rx Interrupt Status: Both channel's transmit and receive interrupt status bits are duplicated here for convenience. The interrupt status can be accessed here or from the appropriate channel status register. A one indicates that the interrupt is active. These bits are latched and can be cleared by writing back to this register as a one.

Frame Interrupt Status: This bit is latched when the frame-clock counter reaches the count specified in the BASE_FRAME_INT register. A one indicates that the interrupt is active. This bit can be cleared by writing back to this register as a one.

JTAG Error: This bit is latched when an error occurs in the JTAG programming subsystem. The most likely cause is an overflow condition in the TDO FIFO although a sequence error in the control state-machine will also produce an error. This bit can be cleared by writing it back to this register as a one.

PLL Sdat: This is where the Sdat bus value is read when data is returned from the PLL.

Interrupt Active: When this bit is read as a one, it indicates that a system interrupt has been asserted. When a zero is read the interrupt is not active.

BASE_FRAME_END

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

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

FIGURE 6 PCI-ASCB FRAME END COUNT REGISTER

This register controls the frame length for I/O scheduling. The Frame End Count field specifies the frame-clock counter end count. When this count is reached, if the Frame Count Loop Enable bit is set to a one, the counter will roll-over to zero and continue counting. If the Frame Count Loop Enable bit is zero, the counter will stop when it reaches this count.

BASE_FRAME_INT

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

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

FIGURE 7 PCI-ASCB FRAME INTERRUPT COUNT REGISTER

This register is where the Frame Interrupt Count is entered. When the frame-clock counter reaches the specified count and the frame interrupt is enabled, then the interrupt will be asserted.

BASE_FRAME_CNT

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

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

FIGURE 8 PCI-ASCB CURRENT FRAME COUNT REGISTER

The current frame-clock count can be read from this read-only port.

BASE_PKT_DELAY

[0x14] ASCB-D Inter-Packet Delay Count Register (read/write)

Inter-Packet Delay Count Register	
DATA BIT	DESCRIPTION
31-8	Spare
7-0	Delay Count

FIGURE 9 PCI-ASCB INTER-PACKET DELAY COUNT

The length of the inter-packet delay can be adjusted by writing a value to this register. After reset or any time the Delay Count field is equal to zero, a default value of 0x73 is passed to the transmit scheduler. This will result in the nominal inter-packet delay of 9.6 microseconds. If a value other than zero is written to the Delay Count field, then that value will be passed to the transmit scheduler. The resulting inter-packet delay is determined by multiplying the delay count value by 80 nanoseconds and adding 0.4 microseconds for processing delays.

CHAN_CONTROL

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

Channel Control Register	
DATA BIT	DESCRIPTION
31-14	Spare
13	Internal CRC Write Enable
12	Re-Initialize Channel
11-10	Spare
9	Force Interrupt
8	Master Interrupt Enable
7	Output DMA Enable
6	Input DMA Enable
5	Receive Interrupt Enable
4	Transmit Interrupt Enable
3	Manchester Data Invert
2	Receiver Input Select
1	Receive Enable
0	Transmit Enable

FIGURE 10

PCI-ASCB CHANNEL CONTROL REGISTER

Transmit Enable: When this bit is set to a one, the transmitter will read two control words from the dual-port RAM and proceed accordingly based on the values in the control-word fields. When this bit is a zero, the transmitter will be disabled and the transmit I/O RAM address will be reset to zero.

Receive Enable: When this bit is set to a one, the receiver will be enabled to begin looking for a sync pattern and storing I/O data in the dual-port RAM. When this bit is a zero, the receiver will be disabled and the receiver I/O RAM address will be reset to zero.

Receiver Input Select: When this bit is a zero, the receiver will monitor the Primary I/O line for data input. When this bit is set to a one, the receiver will monitor the Backup I/O line for data input.

Manchester Data Invert: When this bit is set to a one, a rising edge transition in the middle of the I/O bit period corresponds to a zero for both transmitter and receiver and a falling edge corresponds to a one. When this bit is a zero, a falling edge is a zero and a rising edge is a one.

Transmit Interrupt Enable: When a one is written to this bit, the transmit interrupt source is enabled to cause a system interrupt, provided the master interrupt enable is a one. When a zero is written, the corresponding interrupt is disabled.

Receive Interrupt Enable: When a one is written to this bit, the receiver interrupt source is enabled to cause a system interrupt, provided the master interrupt enable is a one. When a zero is written, the corresponding interrupt is disabled.

Input DMA Enable: This bit, when set to one, enables the interrupt for DMA write completion for the referenced channel. This interrupt cannot be disabled by the master interrupt enable.

Output DMA Enable: This bit, when set to one, enables the interrupt for DMA read completion for the referenced channel. This interrupt cannot be disabled by the master interrupt enable.

Master Interrupt Enable: When this bit is set to a one, all enabled interrupts for the referenced channel (except the DMA interrupts) will be gated through to the PCI host; when this bit is a zero, the interrupts can be used for status without interrupting the host.

Force Interrupt: When this bit is set to a one, a system interrupt will occur provided the channel master interrupt enable is set. This is useful to test interrupt service routines.

Re-Initialize Channel: When this bit is set to a one, all channel state-machines and latches will be reset. This channel control register and the dual-port RAMs are excluded. When this bit is a zero, the channel will operate normally according to the active control bits.

Internal CRC Write Enable: When this bit is set to a one, the CRC calculated by the receive I/O block will be written to the receiver RAM in the location following the CRC that was received from the connected transmitter. When this bit is a zero, writing of the locally calculated CRC is suppressed. This is useful for diagnosing persistent CRC errors.

CHAN_STATUS

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

Channel Status Register	
DATA BIT	DESCRIPTION
31	Interrupt Active
30-18	Spare
11	Output DMA Ready
10	Input DMA Ready
9	Output DMA Error
8	Input DMA Error
7	Output DMA Interrupt
6	Input DMA Interrupt
5	Receive Interrupt Active (read/write clear)
4	Transmit Interrupt Active (read/write clear)
3	Spare
2	Receiver Packet
1	Receiver Active
0	Spare

FIGURE 11

PCI-ASCB CHANNEL STATUS REGISTER

Receiver Active: When a one is read, it indicates that the channel receiver state-machine is enabled. A zero indicates that the receiver state-machine is disabled.

Receiver Packet: When a one is read, it indicates that a received packet is currently in progress. This means that the channel receiver state-machine is enabled and has recognized a sync pattern. A zero indicates that a received packet is not currently in progress. This status bit will transition from a one to a zero when the I/O data bus becomes idle or the receiver state-machine is disabled.

Transmit Interrupt Active: When a one is read, it indicates that a transmit packet has completed since this bit was last cleared. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that a transmit packet has not completed.

Receive Interrupt Active: When a one is read, it indicates that a packet has been received since this bit was last cleared. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that a packet has not been received.

Input DMA Interrupt: When a one is read, it indicates that the current input DMA has completed. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that the input DMA has not completed.

Output DMA Interrupt: When a one is read, it indicates that the current output DMA has completed. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that the output DMA has not completed.

Input DMA Error: When a one is read, it indicates that an error has occurred while the input DMA was in progress. This could be a target or master abort or an incorrect direction bit in one of the DMA descriptors. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that no input DMA error has occurred.

Output DMA Error: When a one is read, it indicates that an error has occurred while the output DMA was in progress. This could be a target or master abort or an incorrect direction bit in one of the DMA descriptors. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that no output DMA error has occurred.

Input DMA Ready: This bit reports the input DMA state-machine status. If it is read as a one, the input DMA state-machine is idle and available to start a transfer. If the bit is read as a zero, the input DMA state-machine is currently processing a data transfer.

Output DMA Ready: This bit reports the output DMA state-machine status. If it is read as a one, the output DMA state-machine is idle and available to start a transfer. If the bit is read as a zero, the output DMA state-machine is currently processing a data transfer.

Interrupt Active: When a one is read, it indicates that the referenced channel interrupt is active. A zero indicates that the channel interrupt is not active.

CHAN_TX_DMA

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

Input DMA Pointer Address Port	
Data Bit	Description
31-0	First Chaining Descriptor Physical Address

FIGURE 12 PCI-ASCB CHANNEL INPUT DMA POINTER PORT

CHAN_RX_DMA

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

Output DMA Pointer Address Port	
Data Bit	Description
31-0	First Chaining Descriptor Physical Address

FIGURE 13 PCI-ASCB CHANNEL OUTPUT DMA POINTER PORT

These write-only ports are used to initiate scatter-gather DMA. When the physical address of the first chaining descriptor is written to one of these ports, the corresponding DMA engine reads four successive long-words beginning at that address. The first is the physical address of the first memory block of the DMA data buffer, the second is the local RAM address to start writing the memory block into, the third is the length in bytes of the block (only the lower 22 bits are valid), and the fourth is the physical address of the next chaining descriptor in the list of buffer memory blocks along with two flag bits, in bit position 0 and 1. Bit zero is the end-of-chain bit, which is set to one if its descriptor is the last in the chain. Bit one is the direction bit, which is set to zero for all input DMA descriptors and one for all output DMA descriptors (including the initial address descriptor that is written to the registers). These two bits are then replaced with zeros to determine the address of the next descriptor, if there is one. This process is continued until the end-of-chain bit is set in one of the next pointer values.

CHAN_TX_INDXX

[0x28, 0x44] ASCB-D Channel 0, 1 Transmit Index Register

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

FIGURE 14 PCI-ASCB CHANNEL TRANSMIT INDEX REGISTER

Current Transmit Access RAM Address: This 12-bit field is the address that the transmit state-machine is currently reading from the I/O side of the transmitter dual-port RAM.

Starting RAM Address of Next Input DMA: This 12-bit field is the address after the last DMA write to the PCI side of the transmitter dual-port RAM.

CHAN_RX_INDXX

[0x2C, 0x48] ASCB-D Channel 0, 1 Receive Index 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 15 PCI-ASCB CHANNEL RECEIVE INDEX REGISTER

Current Receive Access RAM Address: This 12-bit field is the address that the receive state-machine is currently writing to on the I/O side of the receiver dual-port RAM.

Starting RAM Address of Next Output DMA: This 12-bit field is the address after the last DMA read from the PCI side of the receiver dual-port RAM.

CHAN_RX_NEXT

[0x30, 0x4C] ASCB-D Channel 0, 1 Rx Next Packet Start Address (read only)

Channel Receive Next Packet Start Address Register	
DATA BIT	DESCRIPTION
31-12	Spare
11-0	Starting Address of Next Receive Packet

FIGURE 16

PCI-ASCB CHANNEL RECEIVE INDEX REGISTER

This 12-bit field represents the starting address of the next received packet. This value will be updated when the current received packet completes.

JTAG_STATUS

[0x50] ASCB-D JTAG status (read only)

JTAG Status Register	
DATA BIT	DESCRIPTION
31-28	Test-Access-Port State
27	TDO FIFO Data Valid
26	TDO FIFO Full
25-18	TDO FIFO Data Count
17	TDI FIFO Full
16-9	TDI FIFO Data Count
8	TMS FIFO Full
7-0	TMS FIFO Data Count

FIGURE 17

PCI-ASCB JTAG STATUS REGISTER

TMS FIFO Data Count: This eight-bit field contains the number of words currently in the TMS FIFO.

TMS FIFO Full: When this bit is read as a one, there is no more room in the TMS FIFO (256 words). When this bit is a zero, the FIFO has room for at least one more word.

TDI FIFO Data Count: This eight-bit field contains the number of words currently in the TDI FIFO.

TDI FIFO Full: When this bit is read as a one, there is no more room in the TDI FIFO (256 words). When this bit is a zero, the FIFO has room for at least one more word.

TDO FIFO Data Count: This eight-bit field contains the number of words currently in the TDO FIFO.

TDO FIFO Full: When this bit is read as a one, there is no more room in the TDO FIFO (256 words). When this bit is a zero, the FIFO has room for at least one more word.

TDO FIFO Data Valid: When data is initially written to the TDO FIFO, a read is issued to have the data immediately available for the PCI bus interface. This bit indicates that valid data is available. As TDO data is read over the bus, subsequent reads are issued to keep the TDO data ready for immediate access. When this bit is read as a one, the TDO FIFO count should be increased by one to reflect the actual number of valid data words, if the Valid and Full bits are set, the total number of words available is 257.

Test-Access-Port State: This four-bit field contains the pending state of the JTAG Test-Access-Port (TAP) interface. There are 16 possible states for this port that are traversed in response to the Test-Mode-Select (TMS) signal. There is a state-machine in the JTAG programming module that anticipates the state of the target device's TAP in order to have TDI data ready to be shifted out in the appropriate states without causing delays in the bit stream. This field reports the current position of that state-machine.

JTAG_MODE/DATA

[0x50, 0x54] ASCB-D TMS FIFO (write only)/TDI FIFO (write only)/TDO FIFO (read only)

JTAG FIFOs	
DATA BIT	DESCRIPTION
31-0	TMS/TDI/TDO FIFO Data

FIGURE 18

PCI-ASCB JTAG FIFO PORTS

The JTAG programming interface is accessed through these FIFO ports. Data written to the Test-Mode-Select (TMS) FIFO (0x50) and Test-Data-In (TDI) FIFO (0x54) are serialized and shifted out lsb first. The TMS data is always active while the TDI data is only sent in the Shift-Data-Register (SDR) and Shift-Instruction-Register (SIR) states of the Test-Access-Port (TAP) interface. When in the SDR state, Test-Data-Out (TDO) data will be captured lsb first and written to the TDO FIFO where it can be read and compared to expected data, possibly qualified with a TDO data mask.

The status register described above can be read to determine when new TMS and TDI data can be written to the FIFOs and when TDO data is available to be read. Partial words of TDO data will be written to the FIFO (right justified) when the SDR state is exited. Likewise, when the SDR or SIR state is entered, a new TDI word will be read from the TDI FIFO regardless of whether the last TDI word was completely shifted out in the previous SIR or SDR state.

Dual-Port RAM Configuration and Packet Control/Status

Data packets to be transmitted and packets that have been received are stored in their respective dual-port RAMs. Port A of each RAM is accessible only from the PCI bus and Port B is accessible only by the I/O sub-system. An input DMA transfer writes data to Port A of the transmitter RAM, while an output DMA transfer reads data from Port A of the receiver RAM. In addition, the second Base Address Register of the PCI configuration space maps both RAM blocks to a separate address space for single-word accesses in and out of either RAM block (see Figure 3). Each transmitted or received packet is preceded by two 32-bit control/status words as described below.

TX_MEM_CONTROL

The first two 32-bit words of a transmit packet are control words. The control fields are for the current packet only, except potentially for the stop_cnt.

Bit(s)	Field	Function
0 – 11	end_indx	First address of the next packet
12	pri_dis	When this bit is '1', disable the primary I/O bus
13	bkp_dis	When this bit is '1', disable the backup I/O bus
14	crc_dis	When this bit is '1', don't send the transmit CRC
15	start_en	When this bit is '1', use the start-count
16	stop_en	When this bit is '1', use the stop-count
17 – 34	strt_cnt	The frame-clock count to start sending this packet
35 – 52	stop_cnt	The frame-clock count to stop sending packets
53 – 63	spare	Unused

RX_MEM_STATUS

The first two 32-bit words of a receive packet are status words. The status fields refer to the current packet only.

Bit(s)	Field	Function
0 – 11	end_indx	First address for the next packet
12 – 16	num_bits	Excess bits received (S/B zero for normal transfer)
17	bus_sel	When this bit is '1', the backup line was selected
18	man_err	When this bit is '1', a Manchester error was seen
19	crc_err	When this bit is '1', a CRC error was seen
20 – 37	strt_cnt	The frame-clock count when the sync was detected
38 – 55	stop_cnt	The frame-clock count when the bus went inactive
56 – 63	0xAA	Fill pattern

The last word of the received packet will be the CRC word calculated and sent by the transmitter, if so enabled. The receiver calculates a CRC from the data received and compares this value to the one received in order to detect transmission errors.

PCI-ASCB I/O Channel Pin Assignment

9-Pin D-Connectors		
J2		
Signal	Pin #	Function
Pri 0 +	1	Channel 0 Primary +
Pri 0 -	6	Channel 0 Primary -
PriTxDis 0	2	Channel 0 Transmit Disable (Primary)
GND	7	Ground Reference
BackTxDis 0	3	Channel 0 Transmit Disable (Backup)
Back 0 -	8	Channel 0 Backup -
Back 0 +	4	Channel 0 Backup +
GND	9	Ground Reference
Frame 0	5	Channel 0 Frame Sync Pulse
J3		
Signal	Pin #	Function
Pri 1 +	1	Channel 1 Primary +
Pri 1 -	6	Channel 1 Primary -
PriTxDis 1	2	Channel 1 Transmit Disable (Primary)
GND	7	Ground Reference
BackTxDis 1	3	Channel 1 Transmit Disable (Backup)
Back 1 -	8	Channel 1 Backup -
Back 1 +	4	Channel 1 Backup +
GND	9	Ground Reference
Frame 1	5	Channel 1 Frame Sync Pulse

FIGURE 19

PCI-ASCB BUS INTERFACE PINOUTS

Transmitter Disconnect Fail-Safe Shunts	
Signal	Shunt
Primary_0_P	JP-1
Primary_0_N	JP 2
Backup_0_P	JP 3
Backup_0_N	JP 4
Primary_1_P	JP 5
Primary_1_N	JP 6
Backup_1_P	JP 7
Backup_1_N	JP 8

FIGURE 20

PCI-ASCB TRANSMITTER FAIL-SAFE SHUNTS

Ethernet Connector to PMC Slot

Signal	J1 Pin #	Standard	Alternate
E0_TRD0_P	1	Jn4-1	Jn4-45
E0_TRD0_N	2	Jn4-3	Jn4-47
E0_TRD1_P	3	Jn4-7	Jn4-46
E0_TRD2_P	4	Jn4-2	Jn4-49
E0_TRD2_N	5	Jn4-4	Jn4-51
E0_TRD1_N	6	Jn4-9	Jn4-48
E0_TRD3_P	7	Jn4-8	Jn4-50
E0_TRD3_N	8	Jn4-10	Jn4-52

FIGURE 21

PCI-ASCB ETHERNET PINOUTS

Applications Guide

Interfacing

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

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

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

Power all system power supplies from one switch. Connecting external voltage to 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. Alternatively, the use of OPTO-22 isolation panels is recommended.

Keep cables short. Flat cables, even with alternate ground lines, are not suitable for long distances. The twisted pairs are defined to match up with the pin definitions. It is suggested that this standard cable be used for most of the cable run.

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

The PCI-ASCB is constructed out of 0.062-inch thick FR4 material. Through-hole and surface-mount components are used. IC sockets use screw machine pins. High insertion and removal forces are required, which assists in the retention of components. If the application requires unusually high reliability or is in an environment subject to high vibration, the user may solder the corner pins of each socketed IC into the socket, using a grounded soldering iron.

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

The PMC is secured against the PCI-ASCB with four screws attached to the 2 stand-offs and 2 locations on the front panel. The four screws provide significant protection against shock, vibration, and incomplete insertion.

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

Thermal Considerations

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

Warranty and Repair

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

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

Service Policy

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

Out of Warranty Repairs

Out of warranty repairs will be billed on a material and labor basis. The current minimum repair charge is \$100. 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 Street, Suite C
Santa Cruz, CA 95060
831-457-8891
831-457-4793 fax
support@dyneng.com



Specifications

Host Interface (PCI):	PCI Interface 33 MHz. 32-bit
Serial Interfaces:	Two ASCB-D channels, One Ethernet connection Local and Remote JTAG bus-master ports
TX Bit-rates generated:	10 MHz for each channel (primary and backup busses)
Timing Reference:	12.5 MHz Frame Clock for Packet Scheduling
Software Interface:	Control Registers, Dual-port RAMs, and Status Ports
Initialization:	Hardware reset forces all registers to 0 except as noted
Access Modes:	LW boundary Space (see memory map)
Wait States:	One for all addresses
Interrupt:	A Single frame-clock count interrupt for both channels. Also, each channel has an interrupt for TX packet done and Rx packet done. Read and write DMA interrupts are also implemented for each channel.
DMA:	Independent input and output Scatter/Gather DMA support implemented for each channel
Onboard Options:	Fail-safe transmitter shunts (4 per channel) - all other options are software programmable
Interface Options (PCI):	Two 9-pin D-connectors for channel 0 and 1 only
Interface Options (PMC):	8-pin RJ-45 Ethernet connector to PMC Slot Pn4 Standard or Alternate Connections (See Figure 18)
Dimensions:	Dimensions: 6.600 inches by 4.200 inches
Construction:	FR4 Multi-Layer Printed Circuit, Through-Hole and Surface-Mount Components
Temperature Coefficient:	2.17 W/°C for uniform heat across PMC
Power:	Max. TBD mA @ 5V

Order Information

PCI-ASCB

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

Standard version with two 16KB Dual-port RAMs per channel,

PCI-ASCB-Eng-1

Engineering Kit for the PCI-ASCB
board-level schematics (PDF)

PCI-ASCB-Eng-2

Board-level schematics (PDF), Software Driver and
sample application

All information provided is Copyright Dynamic Engineering