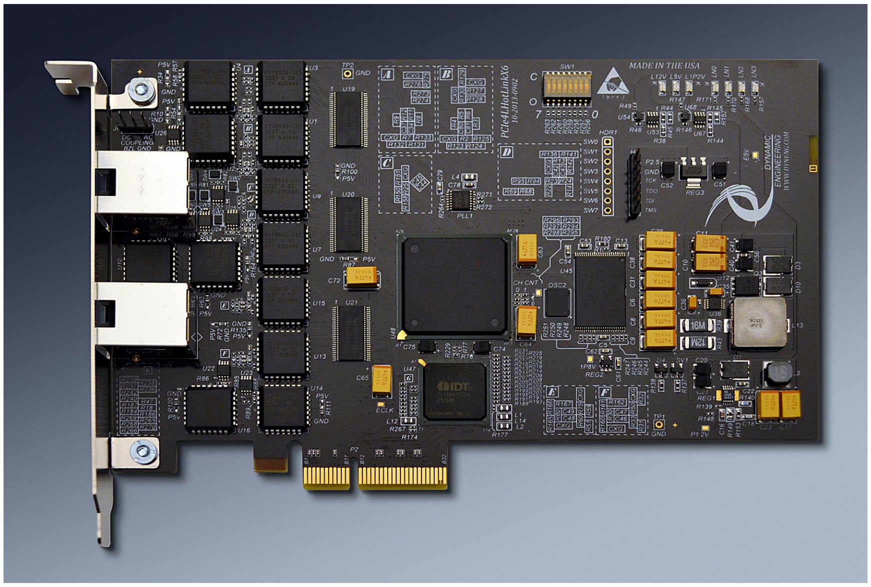
DYNAMIC ENGINEERING
150 DuBois St., Suite C Santa Cruz, CA 95060
(831) 457-8891   **Fax** (831) 457-4793
http://www.dyneng.com
sales@dyneng.com
Est. 1988

User Manual

PCIe-HOTLink

Six-Channel HOTLink® Interface



Revision D
Corresponding Firmware: Design ID 2, Revision C
Corresponding Hardware: 10-2013-0902

PCIe-HOTLink
PCIe Based Six-Channel
HOTLink® Interface

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

Manual Revision D. Revised September 1, 2016

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 Engineeringôs 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.

Connection of incompatible hardware is likely to cause serious damage.

Table of Contents

List of Figures

Product Description

The PCIe-HOTLink is part of the PCIe family of I/O components by Dynamic Engineering. It features the Cypress Semiconductor CY7B923/CY7B933 HOTLink® Transmitter/Receiver pair. The HOTLink devices use positive emitter-coupled logic (PECL) data inputs and outputs. These can be AC coupled or translated to LVDS before being connected to the I/O routing switches and RJ-45 connectors. See the block diagram below:

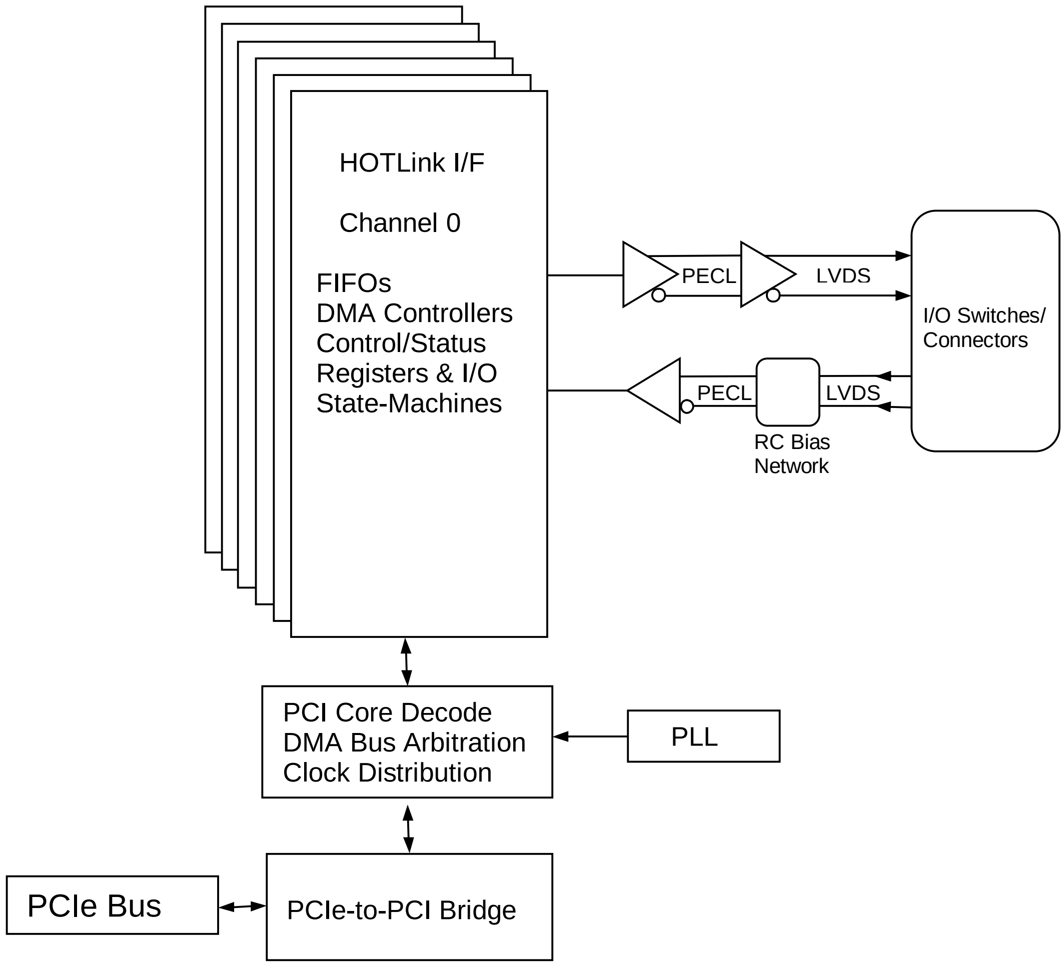


FIGURE 1 PCIE-HOTLINK BLOCK DIAGRAM

Up to six independent HOTLink channels are provided per card. Each HOTLink channel has two differential I/O signal pairs that can use either LVDS or PECL as a build option. When LVDS signaling is used, the input is connected across an equivalent 100 Ý shunt termination. The differential signals are then AC-coupled into the HOTLink receiver inputs referenced to 3.5 volts. The HOTLink differential output signals are biased to 3.3 volts and connected to a PECL-to-LVDS converter. When PECL signaling is used, each side of the differential input is terminated with 50 Ý to an equivalent 1.8 volts and then AC-coupled with 1000 pf to the HOTLink receiver referenced to 3.5 volts. The HOTLink differential output signals are biased to about 3 volts and AC-coupled with 1000 pf to the output switch/connector. A single-ended on-board signal-path is also provided between each channelôs HOTLink transmitter and receiver for built-in test capability.

The transmit byte-rate is determined by the programmed frequency of the PLL A clock output. This clock must be in the range of 16 to 32 MHz; it is multiplied by ten by the HOTLink transmitter which serializes and sends the transmitter byte-wide data-stream that has been expanded to 10 bits by the transmitterôs internal 8B/10B encoder. The PLL is programmed via software by a logic block in the FPGA which generates the waveform timing for the I²C interface to the device. Forty bytes or ten long-words are needed to program or read the PLL device.

Channels 0-3 are configured for half-duplex operation with high-speed multiplexors determining if the transmitter or receiver is connected to the I/O connector. Channels 4 and 5 are configured for full-duplex operation; both the receiver and the transmitter are always connected to the I/O connector.

The óCô revision of the design allows selected channels to be synchronized so that they start transmitting data concurrently. A 20-bit counter running at a one MHz rate generates a trigger pulse to start the transmission. A start count register holds a count that can be used to initialize the counter. A trigger count register holds a count that generates the trigger when reached. An end count register holds a count that causes the counter to be re-initialized to the start count or alternatively to be cleared to zero. A bit in the channel control register selectively enables this feature for each channel.

The HOTLink receiver is supported by a 16k by 32-bit data FIFO; the HOTLink transmitter is supported by an 8k by 32-bit data FIFO. These FIFOs can be accessed by single-word read/writes as well as DMA burst transfers. The transmit FIFO supports write accesses only and the receive FIFO supports read accesses only, but a FIFO test bit in the channel control register enables the data to be routed from the transmit FIFO to the receive FIFO for a full 32-bit path for loop-back testing of the FIFOs.

The HOTLink board supports various interrupts. An interrupt can be configured to occur when the transmit FIFO is almost empty or the receive FIFO is almost full as well as other events and error conditions. All interrupt conditions can be individually masked and a channel master interrupt enable is provided to disable all interrupts on the channel simultaneously. The current real-time status is always available making it also possible to operate in polled mode.

Other custom interfaces are available. We will redesign the state machines and create a custom interface protocol. That protocol will then be offered as a ñstandardò special order product. Please see our web page for current protocols offered and feel free to contact Dynamic Engineering with your custom application.

Theory of Operation

The HOTLink board features a Xilinx Spartan-6-LX100 FPGA. The FPGA contains the PCI interface, all of the registers, FIFOs and protocol controlling elements for the HOTLink design. Only the PCI-to PCIe Bridge, HOTLink transceivers, I/O switches and connectors, PLL and clock circuitry are external to the Xilinx device.

A logic block within the Xilinx controls the PCI interface to the onboard PCI-to-PCIe Bridge. The HOTLink design requires one wait state for read or writes cycles to any address. The wait states refer to the number of clocks after the PCI core decode before the ñterminate with dataò state is reached. Two additional clock periods account for the 1 clock delay to decode the signals from the PCI bus and to convert the terminate-with-data state into the TRDY signal. The PCI bus clock has been increased to 50 MHz to improve the data throughput of the board.

An output and an input 32 by 32-bit FIFO are used to buffer the data into and out of the PLL. Once the data has been written into the input FIFO, the programmer is enabled and the FPGA generates the proper signal timing according to the I^{2}C and PLL device specifications. When a read of the PLL is requested, the proper signal timing for that operation is generated and the data written to the output FIFO.

Scatter-gather bus-master 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 12-byte block from this location. The first four bytes comprise a long-word that is 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 indicating the length, in bytes, of that block. The final four bytes are a long-word that is 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 the direction bit and is set to one if the transfer is from the HOTLink board to host memory, and zero if the transfer is from memory to the board. All descriptor pointers, including the initial descriptor address written to start the DMA, must have the correct direction bit set. These bits are then replaced with zeros to determine the address of the next descriptor, if there is one. This process continues automatically until the last chaining descriptor in the list is processed.

A PCIe-HOTLink channel has both a HOTLink receiver and transmitter. The transmitter is connected to the receiver input B by an always-on internal link from transmitter output C. Transmitter output A and Receiver input A are used for normal operation. Each channelôs input and output DMA engines interface with the channelôs HOTLink transmit and receive FIFOs respectively.

A control bit in the channel configuration register can be set to cause data written to the transmit FIFO to be immediately transferred to the receive FIFO where it can be read and verified to test the input and output FIFOs. The onboard HOTLink signal path can be enabled to test the HOTLink transceivers as well. Finally, I/O signals can be externally connected for a full loopback test of the entire circuit.

There are three storage modes used to write HOTLink data to the receive FIFO. In data-only mode all control characters are stripped from the data-stream and the received HOTLink data (bits 7-0) are written as long-words to the receive FIFO. The first byte received is loaded into byte position zero (bits 7-0), the second byte goes in byte one (bits 15-8) and so on. If the receiver is disabled when a long-word has not been completed, the remaining unfilled bytes are set to zero and the long-word is written to the receiver FIFO.

The other two data storage modes store control characters as well as data. The store-all mode stores all received characters, both data and control. This mode uses 10 bits per character, storing the control/data bit (bit 8) and the symbol violation bit (bit 9). In this mode there are only three bytes per long-word. Byte 0 is stored in bits 9-0, byte 1 is stored in bits 19-10, and byte 2 is stored in bits (29-20). Bits 30 and 31 are not used and are set to zero.

In the compact storage mode the same byte-to-bit mapping is used as the store-all mode, but while all data characters are stored, control characters are only stored if they differ from the character just received. Repeated received control characters are compacted to a single stored control character.

Programmable start and stop sequences of up to three characters and a byte count of up to 16 Mbytes can be specified to control the transmission of messages. The characters of the start sequence are prepended to the message data. After the specified number of bytes have been transmitted, the characters of the stop sequence are sent. See figures 17 and 18 for the details of these sequences. Character 0 is the first character sent or received in both the start and stop sequence, followed by character 1 and 2 as appropriate depending on the character count. The sequence characters can be any data character (control character/data select set to zero, byte set to 0-255) or control characters K28.0-K28.6, K23.7, K27.7, K29.7 or K30.7 (control character/data select set to one, byte set to 0-6, or 8-11). K28.7 transmits a deliberate code violation, so it cannot be used.

The HOTLink transmitter data is always stored four bytes to a long-word. The transmitter reads long-words from the transmit FIFO and sends byte zero (least significant) first followed by byte one, byte two and finally byte three. A transmit channel group sync function was added with the rev. C design. There are now four transmit modes, one video frame mode and three streaming modes.

They are selected by the existing Transmit Mode Select (bit 12) and a new channel control bit, Group Start Enable (bit 31). When both bits are low, the transmitter sends a video frame as described below; when the mode bit is high and the group start bit is low, the independent channel streaming mode is enabled. In addition two new streaming modes are implemented; when the group start bit is high and the mode bit is low, the initial frame is synched to the group start trigger; when both bits are high, all frames are synched to this trigger. The streaming modes send only data characters, aside from the start and stop sequences. As long as data is present in the transmit FIFO and the transmitter is enabled and optionally synched, data-frames will be sent. The video-frame mode replicates a typical target video-frame that is used for test purposes. See the description of the channel control register for information on how to select these different modes. The test video frame is described in figure 2 below:

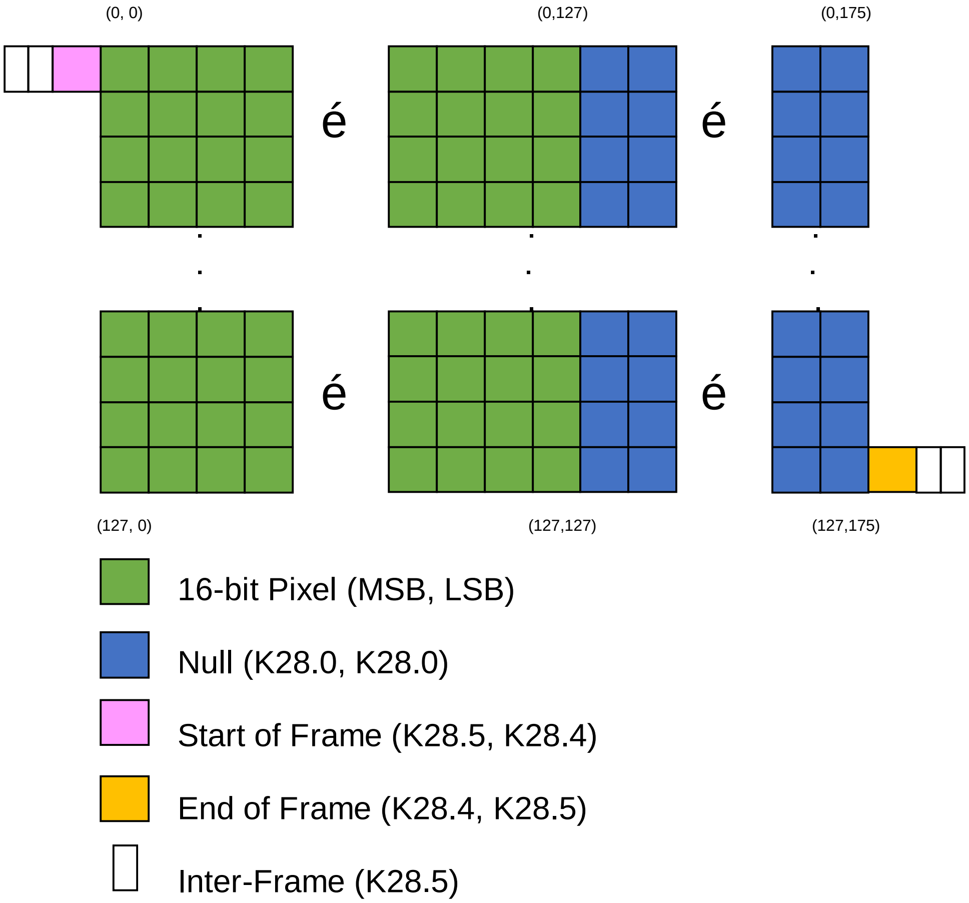


FIGURE 2 PCIE-HOTLINK TEST VIDEO FRAME

Programming

Programming the PCIe-HOTLink board requires only the ability to read and write data from the host. The base address is determined during system configuration of the PCI/PCIe subsystem. The base address refers to the first user address for the slot in which the board is installed. The VendorId = 0xDCBA. The CardId = 0x0054.

If DMA is to be used it will be necessary to acquire a block of non-paged memory that is accessible from the PCI bus in which to store chaining descriptor list entries. If the Dynamic Engineering device driver is used, the driver will handle all the DMA internal mechanics automatically.

In order to transmit or receive HOTLink data, the PLL must be programmed to the desired clock configuration. The PLL is connected to the Xilinx by an I^2C serial bus and its internal registers are loaded with 40 bytes of data that can be derived from a .jed file generated by the CyberClock utility from Cypress semiconductor http://www.dyneng.com/CyberClocks.zip. If you are using our driver, the PLL will be programmed to the default frequency settings when the driver initializes and can be read or re-programmed by an IOCTL call to the base driver.

Routines to use these calls to read and program the PLL are included in the UserApp code provided in the engineering kit for the board. 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 to controlling the logic that drives the I^2C bus.

Once the relevant enables and configuration options are set, the board will wait to receive HOTLink data and store it in a 16K by 32-bit FIFO. The FIFO almost full level is programmable to whatever level is desired and once that level is reached, an interrupt will be asserted if it has been enabled. The data can then be read from the FIFO using DMA in an efficient manner.

The receive FIFO almost full and transmit FIFO almost empty levels can also be used to control DMA channel preemption. When the receive or transmit preemption control bit is enabled, and the transmit FIFO is almost empty or the receive FIFO is almost full, other channels will be forced to relinquish the PCI bus so that the preempting channel can relieve the backlog.

The HOTLink receiver will continue to receive and store characters as long as it is enabled and there remains room in the receive FIFO. When the receiver is disabled, if less than four bytes have been received since the last FIFO write, the last partial word will be written to the FIFO with the unfilled bytes loaded with zeros.

Address Map

Register Name	Offset	Description
PHLNK_BASE_CNTRL	0x0000	// Base Control register
PHLNK_BASE_USER_INFO	0x0004	// Base User Info read port
PHLNK_BASE_INT_STATUS	0x0008	// Base Interrupt Status read port
PHLNK_BASE_PLL_FIFO	0x000C	// Base PLL FIFO port
PHLNK_BASE_CNT_CNTRL	0x0010	// Counter Control register
PHLNK_BASE_START_CNT	0x0014	// Start Count register
PHLNK_BASE_TRIG_CNT	0x0018	// Trigger Count register
PHLNK_BASE_END_CNT	0x001C	// End Count register
PHLNK_CHAN_CNTRL_0	0x0020	// Channel 0 Control register
PHLNK_CHAN_STATUS_0	0x0024	// Channel 0 Status register
PHLNK_CHAN_FIFO_0	0x0028	// Channel 0 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_0	0x002C	// Channel 0 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_0	0x002C	// Channel 0 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_0	0x0030	// Channel 0 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_0	0x0030	// Channel 0 Receive FIFO data count
PHLNK_CHAN_TX_AMT_0	0x0034	// Channel 0 TX almost empty level
PHLNK_CHAN_RX_AFL_0	0x0038	// Channel 0 RX almost full level
PHLNK_CHAN_START_SEQ_0	0x003C	// Channel 0 Frame start sequence
PHLNK_CHAN_STOP_SEQ_0	0x0040	// Channel 0 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_0	0x0044	// Channel 0 Frame byte count
PHLNK_CHAN_FRM_SPACER_0	0x0048	// Channel 0 Inter-Frame byte count
PHLNK_CHAN_CNTRL_1	0x004C	// Channel 1 Control register
PHLNK_CHAN_STATUS_1	0x0050	// Channel 1 Status register
PHLNK_CHAN_FIFO_1	0x0054	// Channel 1 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_1	0x0058	// Channel 1 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_1	0x0058	// Channel 1 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_1	0x005C	// Channel 1 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_1	0x005C	// Channel 1 Receive FIFO data count
PHLNK_CHAN_TX_AMT_1	0x0060	// Channel 1 TX almost empty level
PHLNK_CHAN_RX_AFL_1	0x0064	// Channel 1 RX almost full level
PHLNK_CHAN_START_SEQ_1	0x0068	// Channel 1 Frame start sequence
PHLNK_CHAN_STOP_SEQ_1	0x006C	// Channel 1 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_1	0x0070	// Channel 1 Frame byte count
PHLNK_CHAN_ FRM_SPACER _1	0x0074	// Channel 1 Inter-Frame byte count
PHLNK_CHAN_CNTRL_2	0x0078	// Channel 2 Control register
PHLNK_CHAN_STATUS_2	0x007C	// Channel 2 Status register
PHLNK_CHAN_FIFO_2	0x0080	// Channel 2 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_2	0x0084	// Channel 2 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_2	0x0084	// Channel 2 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_2	0x0088	// Channel 2 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_2	0x0088	// Channel 2 Receive FIFO data count
PHLNK_CHAN_TX_AMT_2	0x008C	// Channel 2 TX almost empty level
PHLNK_CHAN_RX_AFL_2	0x0090	// Channel 2 RX almost full level
PHLNK_CHAN_START_SEQ_2	0x0094	// Channel 2 Frame start sequence
PHLNK_CHAN_STOP_SEQ_2	0x0098	// Channel 2 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_2	0x009C	// Channel 2 Frame byte count
PHLNK_CHAN_ FRM_SPACER _2	0x00A0	// Channel 2 Inter-Frame byte count

Figure 3 PCIe-HOTLink Register Offset Address Map

Register Name	Offset	Description
PHLNK_CHAN_CNTRL_3	0x00A4	// Channel 3 Control register
PHLNK_CHAN_STATUS_3	0x00A8	// Channel 3 Status register
PHLNK_CHAN_FIFO_3	0x00AC	// Channel 3 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_3	0x00B0	// Channel 3 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_3	0x00B0	// Channel 3 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_3	0x00B4	// Channel 3 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_3	0x00B4	// Channel 3 Receive FIFO data count
PHLNK_CHAN_TX_AMT_3	0x00B8	// Channel 3 TX almost empty level
PHLNK_CHAN_RX_AFL_3	0x00BC	// Channel 3 RX almost full level
PHLNK_CHAN_START_SEQ_3	0x00C0	// Channel 3 Frame start sequence
PHLNK_CHAN_STOP_SEQ_3	0x00C4	// Channel 3 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_3	0x00C8	// Channel 3 Frame byte count
PHLNK_CHAN_ FRM_SPACER _3	0x00CC	// Channel 3 Inter-Frame byte count
PHLNK_CHAN_CNTRL_4	0x00D0	// Channel 4 Control register
PHLNK_CHAN_STATUS_4	0x00D4	// Channel 4 Status register
PHLNK_CHAN_FIFO_4	0x00D8	// Channel 4 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_4	0x00DC	// Channel 4 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_4	0x00DC	// Channel 4 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_4	0x00E0	// Channel 4 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_4	0x00E0	// Channel 4 Receive FIFO data count
PHLNK_CHAN_TX_AMT_4	0x00E4	// Channel 4 TX almost empty level
PHLNK_CHAN_RX_AFL_4	0x00E8	// Channel 4 RX almost full level
PHLNK_CHAN_START_SEQ_4	0x00EC	// Channel 4 Frame start sequence
PHLNK_CHAN_STOP_SEQ_4	0x00F0	// Channel 4 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_4	0x00F4	// Channel 4 Frame byte count
PHLNK_CHAN_ FRM_SPACER _4	0x00F8	// Channel 4 Inter-Frame byte count
PHLNK_CHAN_CNTRL_5	0x00FC	// Channel 5 Control register
PHLNK_CHAN_STATUS_5	0x0100	// Channel 5 Status register
PHLNK_CHAN_FIFO_5	0x0104	// Channel 5 TX/RX FIFOs single word access
PHLNK_CHAN_WR_DMA_PNTR_5	0x0108	// Channel 5 Write DMA physical PCI dpr address
PHLNK_CHAN_TX_FIFO_COUNT_5	0x0108	// Channel 5 Transmit FIFO data count
PHLNK_CHAN_RD_DMA_PNTR_5	0x010C	// Channel 5 Read DMA physical PCI dpr address
PHLNK_CHAN_RX_FIFO_COUNT_5	0x010C	// Channel 5 Receive FIFO data count
PHLNK_CHAN_TX_AMT_5	0x0110	// Channel 5 TX almost empty level
PHLNK_CHAN_RX_AFL_5	0x0114	// Channel 5 RX almost full level
PHLNK_CHAN_START_SEQ_5	0x0118	// Channel 5 Frame start sequence
PHLNK_CHAN_STOP_SEQ_5	0x011C	// Channel 5 Frame stop sequence
PHLNK_CHAN_BYTE_COUNT_5	0x0120	// Channel 5 Frame byte count
PHLNK_CHAN_ FRM_SPACER _5	0x0124	// Channel 5 Inter-Frame byte count

Figure 3(Continued) PCIe-HOTLink Register Offset Address Map

Register Definitions

PHLNK_BASE_CNTRL

[0x000] Base Control (read/write)

Base Control Register

Data Bit	**Description**
31-5	Spare
4	PLL Use Alternate ID
3	PLL Check ID
2	PLL Read Enable
1	PLL Reset
0	PLL Enable

FIGURE 4 PCIE-HOTLINK BASE CONTROL REGISTER

All bits are active high and are reset on system power-up or reset, except PLL enable, which defaults to enabled (high) on power-up or reset.

PLL Enable: When this bit is set to a one, the PLL programmer module, used to program and read the PLL, is enabled. When this bit is zero, the PLL programmer is disabled.

PLL Reset: When this bit is set to a one, the PLL programmer will stop processing, if not stopped already, and return to its initial state. When this bit is zero, the PLL programmer is ready to accept control inputs.

PLL Read Enable: When this bit is set to a one and the PLL programmer is enabled, the programmer will perform a read of the PLL device internal registers. The 40 bytes of data obtained will be written into the PLL read FIFO as ten long-words. When this bit is zero and the PLL programmer is enabled, the programmer will write data into the PLL device or simply check for a response to the selected ID value depending on the PLL Check ID control bit.

PLL Check ID: When this bit is set to a one and the PLL programmer is enabled, the programmer will begin a write operation, but will stop after the device ID has been sent. If the ID was acknowledged successfully, the done status will be set and the error status will be cleared. If the ID was not acknowledged successfully, the done status will be cleared and the error status will be set. When this bit is zero and the PLL programmer is enabled, the PLL programmer will perform a write or read operation depending on the PLL Read Enable control bit.

PLL Use Alternate ID: When this bit is set to a one, the device ID sent will be the alternate ID: 0x6A. When this bit is zero, the normal ID: 0x69 will be sent to the PLL device.

PHLNK_BASE_USER_INFO

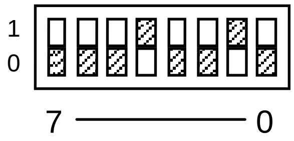
[0x004] Base User Info ï (read only)

User Info Port

Data Bit	Description
31-24	FPGA Design Revision
23-16	FPGA Design ID
15-8	PCI Core Revision
7-0	User ID Switch

FIGURE 5 PCIE-HOTLINK BASE STATUS PORT

User ID Switch: The user switch is read through this port. The bits are read as the lowest byte. Access the read-only port as a long word and mask off the undefined bits. The dip-switch positions are defined in the silkscreen. For example the switch figure below indicates a 0x12.



PCI Core Revision: This is the revision that was entered when the core was created, and is the value that is reported to the operating system. Currently this value is 0x00.

FPGA Design ID: This value distinguishes this design from other HOTLink designs using the same PCI vendor/device ID. For this design the value is 0x02.

FPGA Design Revision: The value of the fourth byte of this port is the revision number of the Xilinx design (currently 0x01 ï rev A).

PHLNK_BASE_INT_STATUS

[0x008] Base Interrupt Status ï (read only)

Base Interrupt Status Register

Data Bit	**Description**
31-27	Spare
26-24	Last Channel Implemented
23-19	Spare
18	PLL Error
17	PLL Done
16	PLL Ready
15	Spare
14	PLL Read FIFO Data Valid
13	PLL Read FIFO Full
12	PLL Read FIFO Empty
11	Spare
10	PLL Write FIFO Data Valid
9	PLL Write FIFO Full
8	PLL Write FIFO Empty
7-6	Spare
5-0	Channel Interrupt Active

FIGURE 6 PCIE-HOTLINK BASE STATUS PORT

Channel Interrupt Active: These six bits are used to report which channel's interrupts are active. When a one is read, it indicates that the corresponding channel has requested an interrupt; when a zero is read, that channelôs interrupt is not active.

PLL Write/Read FIFO Empty: When a one is read, it indicates that the corresponding FIFO contains no data; when a zero is read, there is at least one word in the FIFO. Although the FIFO is empty, there may still be one valid data word in the pipeline. The FIFO data valid bit indicates whether this is the case.

PLL Write/Read FIFO Full: When a one is read, it indicates that the corresponding FIFO is full; when a zero is read, there is room for at least one word in the FIFO.

PLL Write/Read FIFO Data Valid: When a one is read, there is valid data available; when a zero is read, there is no valid data available.

PLL Ready: When a one is read, the PLL programmer is idle and ready to accept a new command; when a zero is read, the programmer is actively sending data or reading data to/from the PLL device.

PLL Done: When a one is read, the programmer has successfully completed an input or output request; when a zero is read, this is not the case. This bit is latched and must be cleared by writing the PLL done bit back to this register.

PLL Error: When a one is read, an error occurred while processing a read or write request; when a zero is read, no error occurred. This bit is latched and must be cleared by writing the PLL error bit back to this register.

Last Channel Implemented: This three-bit binary field encodes the number of the last channel that is implemented on the board. From one to six channels can be populated and enabled on the PCIe-HOTLink board. The number of the last channel must be between zero and five.

PHLNK_BASE_PLL_FIFO

[0x00C] PLL Output FIFO Write/ PLL Input FIFO Read

PLL Output/Input FIFO Ports

Data Bit	**Description**
31-0	FIFO data word

FIGURE 7 PCIE-HOTLINK BASE PLL FIFO PORT

Writes to this port load PLL programming data into the PLL TX FIFO. This data is used to configure the PLL device. Reads from this port return data from the PLL RX FIFO. This data is the PLL deviceôs internal register data that was read by the PLL programmer. Both FIFOs are 32 words deep and 32 bits wide.

PHLNK_BASE_CNT_CNTRL

[0x010] Counter Control (read/write)

Trigger Counter Control Register

Data Bit	**Description**
31-5	Spare
4	Start Clear Enable
3	Start Immediately
2	Continuous Enable
1	End Count Clear Enable
0	Trigger Counter Enable

FIGURE 8 PCIE-HOTLINK COUNTER CONTROL REGISTER

Trigger Counter Enable: When this bit is set to a one, the counter will begin counting. If the Start Clear Enable bit is a one, the counter will start counting from zero, otherwise the Start Count will be loaded. When this bit is zero, the counter is disabled.

End Count Clear Enable: When this bit is set to a one, the counter will be set to zero when the End Count is reached. When this bit is zero, the Start Count will be loaded into the counter when the End Count is reached.

Continuous Enable: When this bit is set to a one, the counter will continue to count and generate triggers after a trigger has been generated. When this bit is zero, the counter will cease counting after a trigger has been generated.

Start Immediately: When this bit is set to a one and the Trigger Counter Enable is zero, a single trigger will be sent at once. When this bit is zero, triggers will only be generated by the Trigger Counter mechanism.

Start Clear Enable: When this bit is set to a one, the counter will be cleared to zero when the counter is enabled. When this bit is zero, the Start Count will be loaded into the counter when it is enabled.

PHLNK_BASE_START/TRIGGER/END_COUNT

[0x014, 0x018, 0x01C] Trigger Counter Counts (read/write)

Counter Control Count Registers

Data Bit	**Description**
31-20	Spare
19-0	Count

FIGURE 9 PCIE-HOTLINK BASE COUNT REGISTERS

These three 20-bit registers hold the Start Count, Trigger Count and End Count respectively. These counts are used to specify the timing behavior of the trigger counter circuit.

PHLNK_CHAN_CNTRL_0-5

[0x020, 0x04C, 0x078, 0x0A4, 0x0D0, 0x0FC] Channel Control (read/write)

Channel Control Register

Data Bit	**Description**
31	Group Start Enable
30	Suppress Stop Sequence
29	RX Frame Done Interrupt Enable
28	TX Frame Done Interrupt Enable
27	Receiver Start Mode
26	Transmitter Select
25	I/O Multiplexor Enable
24	Transmitter Send Frame
23	Send Frame Auto-Clear Enable
22-21	Receiver Storage Mode
20	Receiver Reframe Enable
19	Receiver BIT Enable
18	Receiver A Input Select
17	Transmitter Load Enable
16	Transmitter Output Enable
15	Transmitter BIT Enable
14	Receiver Enable
13	Transmitter Enable
12	Transmit Mode Select
11	Read DMA Arbitration Priority Enable
10	Write DMA Arbitration Priority Enable
9	Read DMA Interrupt Enable
8	Write DMA Interrupt Enable
7	RX FIFO Overflow Interrupt Enable
6	RX FIFO Almost Full Interrupt Enable
5	TX FIFO Almost Empty Interrupt Enable
4	Force Interrupt
3	Master Interrupt Enable
2	FIFO Data Loop-Back Test Enable
1	Receive FIFO Reset
0	Transmit FIFO Reset

FIGURE 10 PCIE-HOTLINK CHANNEL CONTROL REGISTER

All bits are active high and are reset on system power-up or reset. If an external control or status bit is active low, it is inverted from the value in this register.

Transmit / Receive FIFO Reset: When one or both of these bits are set to a one, the corresponding data FIFO and control and status circuitry will be reset. When these bits are zero, normal FIFO operation is enabled. FIFO resets are synchronous, referenced to the PCI clock.

FIFO Data Loop-Back Test Enable: When this bit is set to a one, any data written to the transmit FIFO will be immediately transferred to the receive FIFO. This allows for fully testing the data FIFOs without sending data through the HOTLink interface. When this bit is zero, normal FIFO operation is enabled.

Master Interrupt Enable: When this bit is set to a one all enabled interrupts for the 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.

TX FIFO Almost Empty Interrupt Enable: When this bit is set to a one, an interrupt will be generated when the transmit FIFO level is equal or less than the value specified in the PHLNK_CHAN_TX_AMT register, provided the channel master interrupt enable is asserted. When this bit is zero, an interrupt will not be generated, but the status can still be read from the channel status register.

RX FIFO Almost Full Interrupt Enable: When this bit is set to a one, an interrupt will be generated when the receive FIFO level is equal or greater to the value specified in the PHLNK_CHAN_RX_AFL register, provided the channel master interrupt enable is asserted. When this bit is zero, an interrupt will not be generated, but the status can still be read from the channel status register.

RX FIFO Overflow Interrupt Enable: When this bit is set to a one, an interrupt will be generated when an attempt is made to write to a full receive FIFO, provided the channel master interrupt enable is asserted. When a zero is written to this bit, an interrupt will not be generated when an overflow condition occurs, but the latched status can still be read from the channel status register.

Write / Read DMA Interrupt Enable: These two bits, when set to one, enable the DMA arbiter to use the TX almost empty and/or RX almost full status to give priority to a channel that is approaching the limits of its FIFOs. The levels written to the TX almost empty and RX almost full registers are used to determine these status values. When these bits are zero normal round-robin arbitration is used to determine access to the PCI bus for DMA transfers.

Transmit Mode Select: This bit, when set to one and the Group Start Enable is zero, enables transmitter data streaming mode operation. In this mode the transmit state-machine, if enabled, will read the transmit FIFO and send only data bytes until the transmit byte-count is reached. Each data-frame starts with the programmable start sequence and ends with the stop sequence When this bit is zero, the transmit state-machine, when enabled, simulates the target system operation by sending 128 lines of 128 16-bit pixels followed by 96 NULL bytes each as described in the Theory or Operation section above. See the Group Start description for more information.

Transmitter Enable: This bit, when set to one, enables the HOTLink transmitter state-machine. The behavior of the transmitter depends on the state of the Test Mode bit described above, whether there is data in the FIFO and whether the Send Frame control bit is set. When this bit is zero, the transmitter state-machine is disabled.

Receiver Enable: This bit, when set to one, enables the HOTLink receiver state-machine. The receiver will start storing data when either the specified start sequence is detected, or immediately if the receiver start mode is set to a one. The format of the stored data depends on the receive storage mode field described below. When this bit is zero, the receiver state-machine is disabled.

Transmitter BIT Enable: This bit, when set to one, enables the HOTLink transmitter Built-In-Test mode. When this bit is zero, Built-In-Test mode is disabled.

Transmitter Output Enable: This bit, when set to one, enables the external output of the HOTLink transmitter. When this bit is zero, only the internal output, which is always on, is enabled.

Transmitter Load Enable: When the Built-In-Test mode is enabled, setting this bit to a one starts the test sequence. This bit is zero in normal operation.

Receiver A Input Select: This bit, when set to one, selects input A of the HOTLink receiver. This input is driven by the transformer-coupled external input. When this bit is zero, input B is selected. This input is driven by the internal signal coming from the channelòs HOTLink transmitter output C.

Receiver BIT Enable: This bit, when set to one, enables the HOTLink receiver Built-In-Test mode. When this bit is zero, Built-In-Test mode is disabled.

Receiver Reframe Enable: This bit, when set to one, asserts the HOTLink RF (reframe) control line. This causes the HOTLink receiver chip to search for a comma character to align the serial bit-stream to a byte boundary. When this bit is zero, the RF control signal to the HOTLink receiver chip will not be asserted.

Receive Storage Mode: This two-bit field controls which received characters are stored in the receive FIFO and how the FIFO word is formatted. When this field is set to "01", only data characters will be stored using eight bits per character and each FIFO word will contain four data-bytes; when this field is set to "00", ten bits will be used to represent a character, all control and data characters will be stored and each FIFO word will contain three characters in bits 0-9, 10-19 and 20-29. Bits 30 and 31 are unused and set to zeros. When this field is set to "10", all data characters will be stored, but a control character will only be stored if it differs from the previous character received. Again, ten bits are used to represent a character and each FIFO word will contain three characters.

Send Frame Auto-Clear Enable: When this bit is set to a one, the Transmitter send frame control bit will be automatically cleared when the frame completes. When this bit is zero, the send frame bit must be explicitly cleared.

Transmitter Send Frame: When this bit is set to a one, the transmitter will begin to send data-frames if the FIFO contains data. When this bit is zero, no frames will be sent.

I/O Multiplexor Enable: When this bit is set to a one, the two-to-one multiplexor is enabled. This multiplexor selects between the transmitter output and the receiver input. When this bit is zero, the I/O multiplexor is not enabled. This bit has no effect on channels 4 and 5.

Transmitter Select: When this bit is set to a one and the I/O multiplexor is enabled, the transmitter output is selected. When this bit is zero and the I/O multiplexor is enabled, the receiver input is selected. This bit has no effect on channels 4 and 5.

Receiver Start Mode: When this bit is set to a one, the receiver state-machine will immediately begin receiving and storing input data to the receive FIFO without waiting for a matching start-sequence. When this bit is zero, the receiver state-machine waits until received data matches the designated start-sequence before acquiring data.

TX/RX Frame Done Interrupt Enable: When one of these bits is set to a one, an interrupt will be generated when a transmit/receive frame completes. When this bit is zero, an interrupt will not be generated when a frame completes, but the status can still be read from the channel status register.

Suppress Stop Sequence: When this bit is set to a one, the stop sequence will not be sent when the end of a frame is reached. When this bit is zero, the normal stop sequence will be sent when the end of a frame is reached.

Group Start Enable: When this bit is set to a one, the transmitter will wait for a trigger from the base trigger control counter to initiate a transmission. This allows selected channels to transmit concurrently. If the Transmit Mode Select bit is also one, the transmitter will wait for a trigger for all successive frames as well. If that bit is zero, only the first frame is synchronized and local controls determine when successive frames are sent. When this bit is zero the Group Start feature will be disabled for this channel.

PHLNK_CHAN_STATUS_0-5

[0x024, 0x050, 0x07C, 0x0A8, 0x0D4, 0x100] Channel Status Read/Latch Clear Write

Channel Status Register

Data Bit	**Description**
31	Channel Interrupt Active
30	User Interrupt Active
29-28	Transmit I/O Data Valid
27	Transmit FIFO Data Valid
26	Aux FIFO Full
25	Aux FIFO Almost Full
24	Aux FIFO Empty
23	Receiver I/O FIFO Full
22	Receiver Running
21	Receive Frame Done (latched)
20	Transmit Frame Done (latched)
19	Transmit Data Read (latched)
18	Receive Data Ready (latched)
17	Read DMA Ready (Idle)
16	Write DMA Ready (Idle)
15	Read DMA Error (latched)
14	Write DMA Error (latched)
13	Read DMA Complete (latched)
12	Write DMA Complete (latched)
11	Receive Symbol Error (latched)
10	Receive FIFO Overflow (latched)
9	Receive FIFO Almost Full (latched)
8	Transmit FIFO Almost Empty (latched)
7	Receive Data Valid
6	Receive FIFO Full
5	Receive FIFO Almost Full
4	Receive FIFO Empty
3	Transmit Data Valid
2	Transmit FIFO Full
1	Transmit FIFO Almost Empty
0	Transmit FIFO Empty

FIGURE 11 PCIE-HOTLINK CHANNEL STATUS REGISTER

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

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

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

Transmit Data Valid: When a one is read, there is at least one valid transmit data word left. This bit can be set even if the transmit FIFO is empty, because there is a one-word pipeline after the FIFO output to feed the transmit I/O or FIFO bypass path. When this bit is a zero, it indicates that there is no more valid transmit data.

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

Receive FIFO Almost Full: When a one is read, the number of data-words in the receive data FIFO is greater or equal to the value written to the PHLNK_CHAN_RX_AFL register; when a zero is read, the level is less than that value.

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

Receive Data Valid: When a one is read, there is at least one valid receive data word left. This bit can be set even if the receive FIFO status reports empty, because there is a four-word pipeline after the FIFO output to facilitate a PCI read DMA. When this bit is a zero, it indicates that there is no more valid receive data.

Transmit FIFO Almost Empty (latched): When a one is read, it indicates that the transmit FIFO data count has become less than or equal to the value in the PHLNK_CHAN_TX_AMT register. A zero indicates that the FIFO has not become almost empty. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Receive FIFO Almost Full (latched): When a one is read, it indicates that the receive FIFO data count has become greater than or equal to the value in the PHLNK_CHAN_RX_AFL register. A zero indicates that the FIFO has not become almost full. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Receive FIFO Overflow (latched): When a one is read, it indicates that an attempt has been made to write data to a full receive data FIFO. A zero indicates that no overflow condition has occurred. This bit is latched and can be cleared by writing back to the Status register with a one in this bit position.

Receive Symbol Error: This is a latched version of the RVS (Received Violation Symbol) signal from the channelôs HOTLink receiver. This bit is intended to be used for Built-In-Test operation as it indicates an error during the test sequence. This bit is latched and must be cleared by writing the same bit back to the channel status port.

Write/Read DMA Complete: When a one is read, it indicates that a corresponding DMA descriptor list has completed. These bits are latched and must be cleared by writing the same bit back to this channel status port. A zero indicates that a corresponding DMA descriptor list has not completed since the bit was last cleared.

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

Write/Read DMA ready (Idle): These two bits report the DMA state-machine status. If a one is read, the corresponding DMA state-machine is idle and available to start a transfer. If a zero is read, the corresponding DMA state-machine is already processing a data transfer.

Receive Data Ready: This is a latched version of the ready signal from the channelôs HOTLink receiver. This bit is intended to be used for Built-In-Test operation as the ready signal will pulse once per test loop, so polling this bit will indicate the completion of the receive test sequence. This bit is latched and must be cleared by writing the same bit back to the channel status port.

Transmit Data Read: This is a latched version of the data read signal from the channelôs HOTLink transmitter. Similarly to the ready bit above, this bit is intended to be used for Built-In-Test operation as the data read signal will pulse once per test loop, so polling this bit will indicate the completion of the transmit test sequence. This bit is latched and must be cleared by writing the same bit back to the channel status port.

Transmit Frame Done: When a one is read, it indicates that the transmitter has completed a requested transfer. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that the transmitter has either not completed a requested transfer or no transfer was initiated.

Receive Frame Done: When a one is read, it indicates that the receiver has detected a valid stop sequence while in the run state. If a zero byte stop sequence is requested, the receiver uses the byte-count to determine when the frame is done. This bit is latched and must be cleared by writing the same bit back to the channel status port. A zero indicates that the receiver has either not completed a requested transfer or a transfer was not initiated.

Receiver Running: When a one is read, it indicates that the receiver is actively processing an I/O frame. A zero indicates that the receiver is either idle, waiting for the start sequence or the receiver start mode is set to a one (causes the receiver to start storing characters immediately without waiting for a start sequence).

Receiver I/O FIFO Full: When a one is read, it indicates that the 10-bit wide receiver sample FIFO is full. A zero indicates that this FIFO is not full.

Transmit FIFO Data Valid: When a one is read, it indicates that there is valid data at the output of the transmit data FIFO. A zero indicates that there is not valid data at the output of the transmit data FIFO.

Transmit I/O Data Valid: This two-bit field indicates the number of transmit data-words that are in process in the transmit data state-machine. This field can have a value of zero, one, or two data-words.

User Interrupt Active: When a one is read, it indicates that an enabled user interrupt condition (other than the DMA interrupts) is active for the channel. A zero indicates that no enabled interrupt condition is active.

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

PHLNK_CHAN_FIFO_0-5

[0x028, 0x054, 0x080, 0x0AC, 0x0D8, 0x104] TX FIFO Write/RX FIFO Read

RX and TX FIFO Ports

Data Bit	Description
31-0	FIFO data word

FIGURE 12 PCIE-HOTLINK CHANNEL RX/TX FIFO PORT

These ports are used to make single-word accesses into the channel transmit FIFO and out of the channel receive FIFO.

PHLNK_CHAN_WR_DMA_PNTR_0-5

[0x02C, 0x058, 0x084, 0x0B0, 0x0DC, 0x108] Input DMA Control (write only)

Input DMA Pointer Address Port

Data Bit	Description
31-0	First Chaining Descriptor Physical Address

FIGURE 13 PCIE-HOTLINK 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 three 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 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 is set in one of the next pointer values. Writing a zero to this port will abort a write DMA in progress.

PHLNK_CHAN_TX_FIFO_COUNT_0-5

[0x02C, 0x058, 0x084, 0x0B0, 0x0DC, 0x108] TX FIFO Word Count (read only)

TX FIFO Data Count

Data Bit	Description
31-14	Spare
13-0	TX data words stored

FIGURE 14 PCIE-HOTLINK CHANNEL TX FIFO DATA COUNT PORT

These read-only register ports report the number of 32-bit data words in the corresponding transmit FIFO. There is a 16-word auxiliary FIFO and three additional latches in the transmit data path that may contain data if enabled, which allows this value to be a maximum of 0x2013.

PHLNK_CHAN_RD_DMA_PNTR_0-5

[0x030, 0x05C, 0x088, 0x0B4, 0x0E0, 0x10C] Output DMA Control (write only)

Output DMA Pointer Address Port

Data Bit	**Description**
31-0	First Chaining Descriptor Physical Address

FIGURE 15 PCIE-HOTLINK 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 three 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 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 is set in one of the next pointer values. Writing a zero to this port will abort a read DMA in progress.

PHLNK_CHAN_RX_FIFO_COUNT_0-5

[0x030, 0x05C, 0x088, 0x0B4, 0x0E0, 0x10C] RX FIFO Word Count (read only)

RX FIFO Data Count

Data Bit	**Description**
31-15	Spare
14-0	RX data words stored

FIGURE 16 PCIE-HOTLINK CHANNEL RX FIFO DATA COUNT PORT

These read-only register ports report the number of 32-bit data words in the corresponding receive FIFO. There are four additional latches in the read DMA pipeline that may contain data, which allows this value to be a maximum of 0x4004.

PHLNK_CHAN_TX_AMT_0-5

[0x034, 0x060, 0x08C, 0x0B8, 0x0E4, 0x110] TX FIFO Almost Empty Level (read/write)

TX FIFO Almost Empty Level Register

Data Bit	Description
31-14	Spare
13-0	TX FIFO almost empty level

FIGURE 17 PCIE-HOTLINK CHANNEL TX FIFO AMT LEVEL REGISTER

These read/write ports access the transmitter almost-empty level registers for the channel. When the number of data words in the transmit data FIFO is equal or less than this value, the almost empty status bit is set and an interrupt may be generated if it is enabled.

PHLNK_CHAN_RX_AFL_0-5

[0x038, 0x064, 0x090, 0x0BC, 0x0E8, 0x114] RX FIFO Almost Full Level (read/write)

RX FIFO Almost Full Level Register

Data Bit	Description
31-15	Spare
14-0	RX FIFO almost full level

FIGURE 18 PCIE-HOTLINK CHANNEL RX AFL LEVEL REGISTER

These read/write ports access the receiver almost-full level registers for the channel. When the number of data words in the receive data FIFO is equal or greater than this value, the almost full status bit is set and an interrupt may be generated if it is enabled.

PHLNK_CHAN_START_SEQ_0-5

[0x03C, 0x068, 0x094, 0x0C0, 0x0EC, 0x118] Start Sequence (read/write)

Frame Start Sequence Register

Data Bit	**Description**
31-30	Sequence Character Count
29	Spare
28	Character 2 Control Character/Data Select
27-20	Character 2 Byte
19	Spare
18	Character 1 Control Character/Data Select
17-10	Character 1 Byte
9	Spare
8	Character 0 Control Character/Data Select
7-0	Character 0 Byte

FIGURE 19 PCIE-HOTLINK CHANNEL START SEQUENCE REGISTER

Start and stop sequences of up to three characters can be specified in this register and the following register. The number of characters to be used is coded in the Sequence Character Count two-bit field. The individual characters are entered in the appropriate character fields. A one in the Control Character/Data Select field indicates a control character, a zero indicates a data character. Character 0 is the first character sent or received, followed by character 1 and 2 as appropriate depending on the character count. If this register is not programmed or is set to all zeros, a two character sequence with K28.5 in character 0 and K28.4 in character 1 will be used.

PHLNK_CHAN_STOP_SEQ_0-5

[0x040, 0x06C, 0x098, 0x0C4, 0x0F0, 0x11C] Stop Sequence (read/write)

Frame Stop Sequence Register

Data Bit	Description
31-30	Sequence Character Count
29	Spare
28	Character 2 Control Character/Data Select
27-20	Character 2 Byte
19	Spare
18	Character 1 Control Character/Data Select
17-10	Character 1 Byte
9	Spare
8	Character 0 Control Character/Data Select
7-0	Character 0 Byte

FIGURE 20 PCIE-HOTLINK CHANNEL STOP SEQUENCE REGISTER

If this register is not programmed or is set to all zeros, a two character sequence with K28.4 in character 0 and K28.5 in character 1 will be used.

PHLNK_CHAN_BYTE_COUNT_0-5

[0x044, 0x070, 0x09C, 0x0C8, 0x0F4, 0x120] Frame Byte-Count (read/write)

Frame Byte Count Register

Data Bit	Description
31-24	Spare
23-0	I/O Frame Byte-Count

FIGURE 21 PCIE-HOTLINK CHANNEL BYTE COUNT REGISTER

This register specifies the number of data-bytes to be sent between the start and stop sequence. If the value written is zero, a default value of 0x8000 bytes will be used.

PHLNK_CHAN_FRM_SPCR_0-5

[0x048, 0x074, 0x0A0, 0x0C8, 0x0F8, 0x124] Inter-Frame Byte-Count (read/write)

Inter-Frame Byte Count Register

Data Bit	Description
31-24	Spare
23-0	Frame Spacer Byte-Count

FIGURE 22 PCIE-HOTLINK CHANNEL FRAME SPACER COUNT REGISTER

This register specifies the number of Inter-Frame (K28.5) bytes that will be sent between frames. If the value written is zero, a default value of 0x03F will be used.

Applications Guide

Interfacing

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

ESD

Proper ESD handling procedures must be followed when handling the PCIe-HOTLink. The card is shipped in an anti-static, shielded bag. The card should remain in the bag until ready for use. When installing the card the installer must be properly grounded and the hardware should be on an anti-static workstation.

Start-up

Make sure that the "system" can see your hardware before trying to access it. Many BIOS will display the PCI devices found at boot up on a "splash screen" with the VendorID and CardId and an interrupt level. Look quickly, if the information is not available from the BIOS then a third party PCI device cataloging tool will be helpful. We use PCIView.

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.

We provide the components. You provide the system. Only careful planning and practice can achieve safety and reliability. Inputs can be damaged by static discharge, or by applying voltage outside of the device rated voltages.

Construction and Reliability

PCIe Modules while commercial in nature can be conceived and engineered for rugged industrial environments. The PCIe-HOTLink is constructed out of 0.062-inch thick FR4 material.

Surface mount components are used. Most devices are high pin count compared to mass of the device. For high vibration environments inductors and other higher mass per joint components can be glued down.

Conformal Coating is an option. For condensing environments conformal coating is required.

ROHS processing is an option. Standard lead solder is used unless ñ-ROHSò is added to the part number.

Thermal Considerations

The PCIe-HOTLink 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. 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 Street, Suite C
Santa Cruz, CA 95060
831-457-8891
831-457-4793 fax
support@dyneng.com

Specifications

Host Interface:	PCIe
Serial Interfaces:	One to Six HOTLink inputs and HOTLink outputs
TX Bit-rates generated:	160 - 320 MHz for the HOTLink I/O, Frequencies can be varied by reprogramming the PLL
Software Interface:	Control Registers, FIFOs, 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:	Each channel has an interrupt for TX done, RX done, TX almost empty, RX almost full and RX FIFO overflow. 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:	All Options are Software Programmable
Interface Options:	Channel 0-3 HOTLink receive or transmit lines are available on J1. Channel 4 and 5 receive and transmit lines are available on J2.
Dimensions:	Standard Single PCIe board.
Construction:	FR4 Multi-Layer Printed Circuit, Surface-Mount Components
Power:	Max. **TBD** mA @ 5V

Order Information

PCIe-HOTLink
http://www.dyneng.com/pciehotlink.html
Standard version with one 16K x 32-bit FIFO, and one 8K x 32-bit FIFOs per channel.

PCIe-HOTLink-Eng-1
Engineering Kit for the PCIe-HOTLink board-level schematics (PDF)

PCIe-HOTLink-Eng-2
Board-level schematics (PDF), Linux Driver and sample application

PCIe-HOTLink-Eng-3
Board-level schematics (PDF), Windows Driver and sample application

All information provided is Copyright Dynamic Engineering