TOUIT, INC.

Touit Standard PCI Express Core

Datasheet
March, 2008

1. Features

- Complies with PCI Express Base Specification V1.1
- Complies with the PCI Express Architecture (PIPE) Specification V1.00
- Layered architecture with Configuration Space, Transaction Layer, Data Link Layer and Physical Layer
- Implements receive and transmit buffers, Flow Control (FC) and Transaction Ordering
- Implements the Message Signaled Interrupt, Power Management, Device Serial Number and PCI Express capability registers
- Optionally implements Virtual Channels and Advanced Error Reporting
- Supports up to 4 Virtual Channels
- Supports configurable maximum packet sizes in the range of 128B to 2KB
- Full Duplex communication
- Handles packet error detection and reporting
- Implements End-to-End Cyclic Redundancy Checking (ECRC)
- Up to 32 out standing requests with timeout control

2. Description

The Touit PCI Express Standard Core provides a PCI Express Endpoint solution with access to the PCI Express packet protocol data. This allows the user to work with the PCI Express packet formation details. The Standard core will handle all Configuration Space requests and PCI Express packet checking rules. The core handles the Flow Control (FC) checking on both the receive and transmit lanes. It will allow up to 32 outstanding read request packets to be issued and implements a completion timer for each request. If the completion timer expires the core can resend the request and alert the user to the timeout. Advanced Error Reporting, Power Management and Message Signaled Interrupts are PCI Express functions also included in the core. With the Standard core handling much of the low level PCI Express overhead, it allows the user to focus on the actual read and write data packets.

The Standard core is compliant to the PIPE interface for connection to PIPE compliant SERDES blocks. The core uses internal FPGA memory blocks for the Retry Buffer and Receive and Transmit buffers.

3. The Core

Configuration Space

The configuration space contains the register map for all the PCI Express configuration and status registers. It has an interface that provides read and write access to these registers for configuration read and write requests.
A complete set of configuration and status registers are provided. These include:

- PCI Type 0 header for Endpoint mode
- Power Management Capability
- Message Signaled Interrupt Capability
- PCI Express Capability
- Virtual Channels Capability
- Advanced Error Reporting Capability

Any read accesses to addresses not implemented returns zeros and any writes are treated as No Ops.

The registers are set via access signals provided by the core.

The Transaction Layer handles any accesses to the configuration space through configuration requests and will generate the appropriate completions as required.

Transaction Layer

The Transaction Layer is located between the Application Layer and the Data Link Layer. The Transaction Layer's primary responsibility is the assembly and disassembly of Transaction Layer Packets (TLPs). TLPs are used to communicate transactions, such as read and write, as well as certain types of events. Every TLP requiring a response packet is implemented as a split transaction between the receive and transmit sides of the Transaction Layer.

The transmit side of the Transaction Layer receives data from the Application Layer and forms TLPs, encapsulating the data, which are sent on to the Data Link Layer. Data is sent in either a write request TLP or in a completion TLP. The transmit side also sends read request TLPs when the Application Layer requests data. The Transaction Layer gives each read request TLP a unique identifier that enables completion TLPs to be directed to the originator of the request TLP. Each read request has a timer that monitors the read data returned via completion packets on the receive side of the Transaction Layer. The timer will reset when any of the requested data is received and will disable when the entire read transaction is completed. If the timer expires before the transaction completes, this is called a Completion Timeout. If enabled, on a Completion Timeout the read request will be resent by the Transaction Layer.

The receive side of the Transaction Layer receives TLPs from the Data Link Layer, analyzes the TLP and sends the data on to the Application Layer. The receive side also handles configuration space requests by issuing a read or write transaction to the configuration registers, and then schedules a completion packet to be sent with the transmit side of the Transaction Layer. If an error is detected in the TLP, the packet will be discarded and an error message will be scheduled for transmission to the root complex or host system.

The Transaction Layer is also responsible for managing credit-based Flow Control (FC) for TLPs. The amount of available space in the receive buffers determines the number of FC credits available. The Transaction Layer will hold-off any transaction waiting to be sent if there are not enough FC credits available on the other end of the link. Optional End-to-End Cyclic Redundancy Checking (ECRC) is done and generated in the Transaction Layer as well as Power Management events.

Data Link Layer

The Data Link Layer (DLL) is located between the Transaction Layer and the Physical Layer in a PCI Express link. It provides packet tracking and error control between the two ends of a PCI Express link. The DLL also initializes the flow control information for the link.

Data integrity for a packet is ensured through the use of a CRC appended to the end of each packet. Packet tracking is handled by adding and tracking sequence numbers on each packet. The DLL on each end of the link will exchange acknowledgment packets, called Data Link Layer Message (DLLM) ACK and NAK packets, to signal successful or unsuccessful receipt of data. The DLL implements a Retry Buffer to store transmitted packets for re-transmission if required. After the DLL receives an acknowledgment that the other end of the link has successfully received a packet, the DLL will remove that packet from the Retry Buffer.

If any errors are detected the DLL will alert the Transaction Layer.

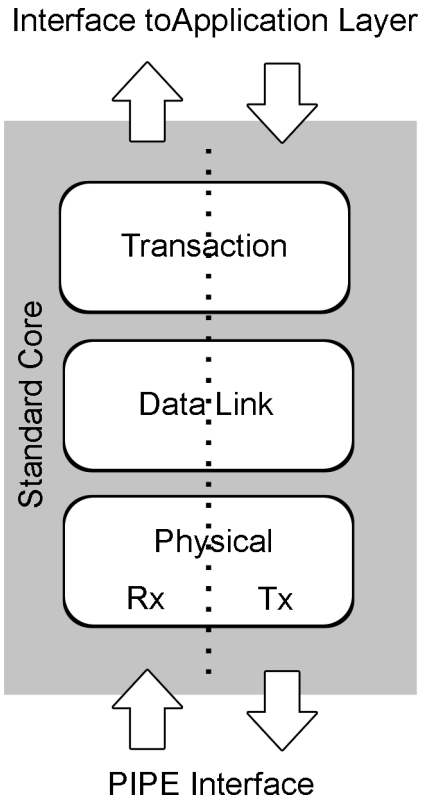
Physical Layer

The Physical Layer is located between the Data Link Layer and the SERDES or Phy. This layer is responsible for low level link management. The Physical Layer performs link initialization to establish lock and determine the width and speed of the link. After the link is initialized the Physical Layer will place specialized symbol characters before and after packets received from the Data Link Layer for transmission. These characters alert the Physical Layer on the receiving end of the link to the start and finish of a packet.

Lane to lane deskew and data scrambling/unscrambling are also implemented in the Physical Layer.

4. Core Block Diagram

Shown below is a simplified block diagram of the Standard PCI Express Core:



5. Signal Descriptions

PIPE Interface Signal Table

Signal Name	*I/O*	*Width*	*Description*
pcie_reset	I	1	PCI Express Bus reset signal.
pipe_reset	O	1	Resets the transmitter and receiver in the PHY.
pipe_tx_data	O	16	Parallel data output bus to the PHY.
pipe_tx_data_k	O	2	Data/Control for the symbols of transmit data. Bit 0 corresponds to the low-byte of pipe_tx_data, Bit 1 to the upper byte. A value of zero indicates a data byte, a value of 1indicates a control byte.
pipe_rx_data	I	16	Parallel data input bus from the PHY.
pclk	I	1	Parallel interface data clock. All data movement across the parallel interface is synchronized to this clock.

Signal Name	*I/O*	*Width*	*Description*
pipe_rx_data_k	I	2	Data/Control bit for the symbols of receive data. Bit 0 corresponds to the low-byte of pipe_rx_data, Bit 1 to the upper byte. A value of zero indicates a data byte; a value of 1 indicates a control byte.
pipe_tx_detect_loopback	O	1	Used to tell the PHY to begin a receiver detection operation or to begin loopback.
pipe_tx_elecidle	O	1	Forces PHY Tx output to electrical idle.
pipe_tx_compliance	O	1	Sets the running disparity to negative. Used when transmitting the compliance pattern.
pipe_rx_status	I	3	Encodes receiver status and error codes for the received data stream and receiver detection.
pipe_rx_polarity	O	1	Tells PHY to do a polarity inversion on the received data.
pipe_rx_elecidle	I	1	Indicates receiver detection of an electrical idle.
pipe_rx_valid	I	1	Indicates symbol lock and receipt of valid data.
pipe_powerdown	O	2	Power up or down the transceiver.
pipe_status	I	1	Used to communicate completion of several PHY functions including power management state transitions, and receiver detection.

Application Layer Interface Signal Table

Signal Name	*I/O*	*Width*	*Description*
vc_x_np_ready	O	1	Signal receipt of valid read request packet ready for readout.
vc_x_np_pop	I	1	Pops the current read request off the read request fifo and loads the next read request packet.
vc_x_np_req_id	O	16	Read requestor ID.
vc_x_np_high_addr	O	1	Signals 64 bit (high) or 32 bit (low) address value.
vc_x_np_addr	O	64	Address for the read request.
vc_x_np_length	O	10	Length, in DWORDS, of the read requst.
vc_x_np_tag	O	5	Tag associated with the read request.
vc_x_p_ready	O	1	Signal receipt of valid write request packet ready for readout.

Signal Name	*I/O*	*Width*	*Description*
vc_x_p_pop	I	1	Pops the current write request off the write request fifo and loads the next write request packet.
vc_x_p_high_addr	O	1	Signals 64 bit (high) or 32 bit (low) address value.
vc_x_p_addr	O	64	Address for the read request.
vc_x_p_length	O	10	Length, in DWORDS, of the write requst.
vc_x_p_valid	O	1	Indicates valid data is available on the write request data port.
vc_x_p_last	O	1	Idicates the data on the write request data port is the last data for the write packet.
vc_x_p_enable	I	1	Signals successful read of the write request data on the data port.
vc_x_p_data	O	32	Write request data port.
vc_x_cpl_ready	O	1	Signal receipt of valid completion packet ready for readout.
vc_x_cpl_pop	I	1	Pops the current completion packet off the completion fifo and loads the next completion packet.
vc_x_cpl_done	O	1	Idicates that current completion packet is the final completion packet finishing the associated read request.
vc_x_cpl_status	O	2	Completion packet status.
vc_x_cpl_tag	O	5	Tag from the associated read request.
vc_x_cpl_local_id	O	32	ID value for current completion packet.
vc_x_cpl_length	O	10	Length, in DWORDS, of the completion.
vc_x_cpl_valid	O	1	Indicates valid data is available on the completion data port.
vc_x_cpl_last	O	1	Idicates the data on the completion data port is the last data for the completion packet.
vc_x_cpl_enable	I	1	Signals successful read of the completion data on the data port.
vc_x_cpl_data	O	32	Completion data port.
ext_local_id	I	32	Local ID value for reqd request. Will be output with any received completion packet for the read request.
ext_type	I	3	Idicates the type of TLP to be sent. Valid values are: 000: Read Request

Signal Name	*I/O*	*Width*	*Description*
			001: Write Request 111: Completion
ext_tc	I	3	Traffic class for TLP.
ext_length	I	10	Length field, in DWORDS, for TLP. Can not be more than the value indicated by max_payload_size.
ext_high_address	I	1	Signals 64 bit (high) or 32 bit (low) address value.
ext_address	I	64	Address for TLP. For read and write request packets, upper 32 bits are set to zero for 32 bit request. For Completions, the upper 32 bits are set to zero and the bottom 4 bytes are as follows: byte 3 and 2: requester ID byte 1: tag byte 0: lower address
ext_first_be	I	4	Byte enables for first DWORD.
ext_last_be	I	4	Byte enables for last DWORD,
ext_status	I	3	Status field for completion TLP.
ext_ready	I	1	Signals TLP ready to be sent.
ext_done	O	1	Signals successful transmission of TLP.
ext_last	I	1	Idicates the data on the data port is the last data for the packet.
ext_enable	O	1	Signals successful read of the data on the data port.
ext_data	I	16	Transmit TLP data port.
interrupt	I	1	Sends an interrupt to the root complex.
retransmit_enable	I	1	Enables retransmission of read request packets on a completion timeout.
completion_timeout	O	1	Idicates a completion timeout occurred.

Configuration Space Signal Table

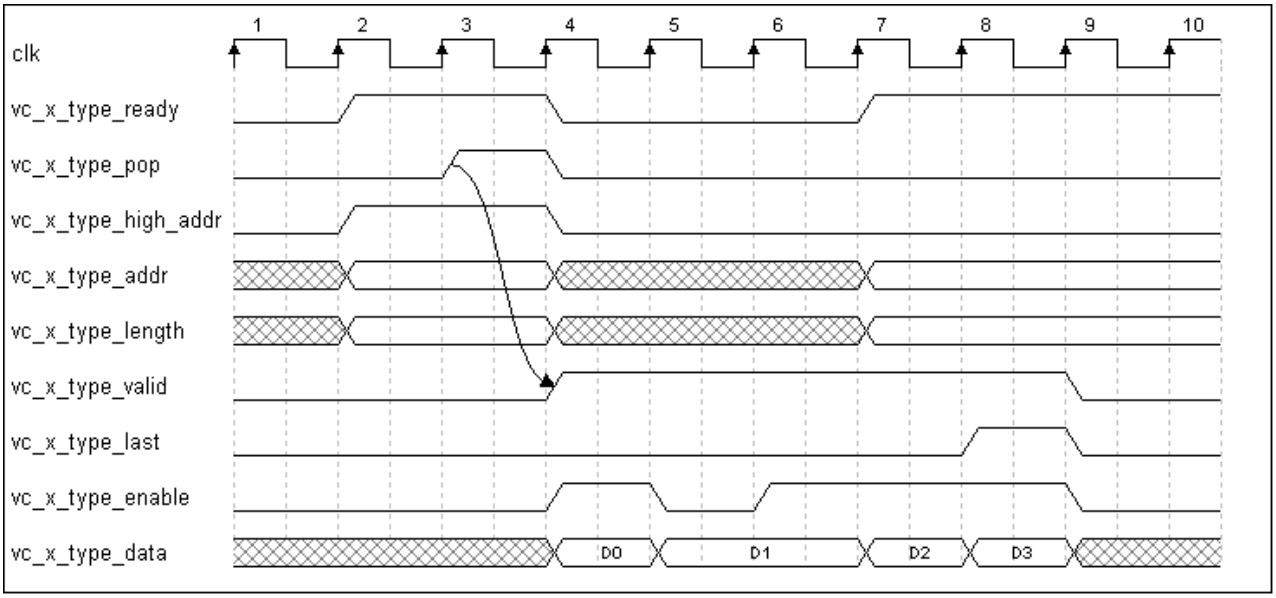
These values should not be changed after system initialization. It is recommended that these values be harcoded in the HDL.

Signal Name	*I/O*	*Width*	*Description*
max_payload_size	O	10	Maximum allowed data payload for a TLP.
bar_x_enable	I	1	Enables the Base Address Register (BAR) in

Signal Name	*I/O*	*Width*	*Description*
			configuration space. NOTE: only BARs 0, 2, and 4 are available for use as memory space. BAR 1, 3, and 5 are reserved for use when BAR 0, 2, or 4 are set to 64 bit address mode. BAR x+1 is the upper 32 bit address value for BAR x. Where 'x' is 0, 2, or 4.
bar_x_64bit	I	1	Sets BAR to 64 bit address mode.
bar_x_size	I	25	Sets the size of the address region in bytes.
bar_x	O	25	Address assigned to BAR.
bar_x+1_size	I	32	Sets the size of the upper 32 bit address region.
bar_x+1	O	32	Address assigned to BAR.

6. Timing Diagrams

Virtual Channel NP, P, or CPL Interface to Application Layer:



External TLP Transmit Interface to Application Layer: