



Enabling success from the center of technology™

2007 Xfest

Virtex-5 PCI Express Endpoint Block

- **Present Xilinx PCI Express solutions**
- **Understand the Xilinx Virtex-5 PCI Express Endpoint Block**
- **Use Xilinx Core Generator to design with the Virtex-5 PCI Express Endpoint Block**

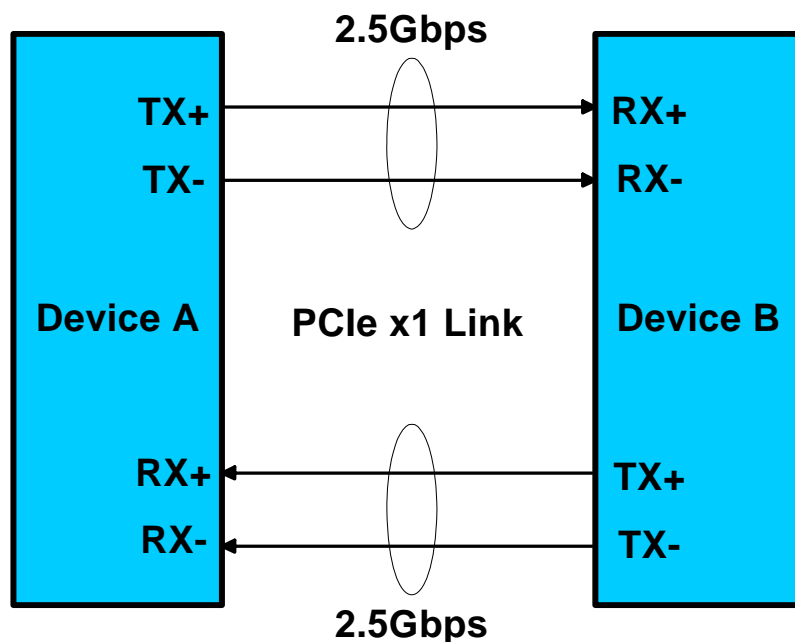
- **Introduction to PCI Express**
- **Xilinx Virtex-5 PCI Express Endpoint**
- **Designing with the Virtex-5 PCI Express Endpoint Block**
- **Virtex-5 PCI Express Demo**

- **Introduction to PCI Express**
- Xilinx Virtex-5 PCI Express Endpoint
- Designing with the Virtex-5 PCI Express Endpoint Block
- Virtex-5 PCI Express Demo

- **The PCI Express (PCIe) standard is a next-generation evolution of the older PCI and PCI-X parallel bus standards**
 - It is a high-performance, general-purpose interconnect architecture, designed for a wide range of computing and communications platforms
 - It is a packet-based, point-to-point serial interface that is backward compatible with PCI and PCI-X configurations, device drivers, and application software

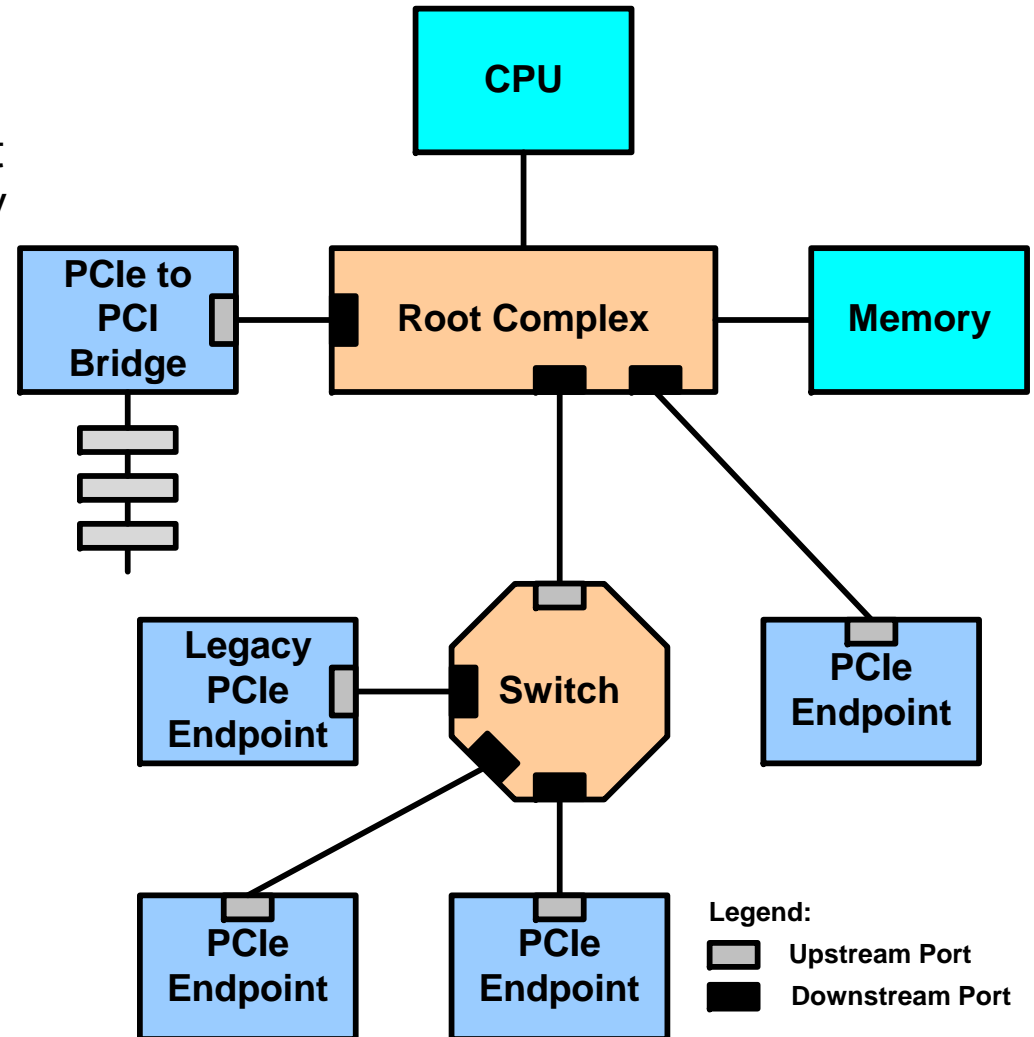
PCI Express Bandwidth		
Link	Raw Bandwidth Per Direction	Effective Bandwidth Per Direction
x1	2.5Gbps	2Gbps
x2	5Gbps	4Gbps
x4	10Gbps	8Gbps
x8	20Gbps	16Gbps
x12	30Gbps	24Gbps
x16	40Gbps	32Gbps
x32	80Gbps	64Gbps

- Point-to-point serial interface
- One transmit and one receive differential pair per lane
- Each lane transfers data at 2.5Gbps in both directions



Link	# of Lanes
x1	1
x2	2
x4	4
x8	8
x12	12
x16	16
x32	32

- **Root Complex**
 - A Root Complex (RC) denotes the root of an I/O hierarchy that connects the CPU and memory subsystem to the I/O
 - Comparable to the PCI North Bridge
- **Switch**
 - Logical assembly of multiple virtual PCIe-PCIe bridge devices
 - Comparable to the PCI South Bridge
- **Endpoint**
 - Previously known as the Peripheral



- **PCI Express configuration space consists of 4096 bytes**
 - First 256 bytes match the PCI configuration space

- **Some configuration data is simply read by the Root Complex**
 - Vendor and Device ID
 - Memory blocks and size requirements
 - Certain capabilities of the subsystem such as the device type and power management

- **Some configuration data is assigned by the Root Complex**
 - Base Address Register (BAR)
 - Specific address for each requested memory block
 - Six maximum BARs per subsystem
 - Transaction ID consisting of Bus and Device Number
 - Consistent with the physical slot



- **The communication between two PCIe devices is referred to as a transaction**
 - Transactions are packet-based

- **Each PCIe device can be a Requester and/or a Completer**
 - Requester initiates a transaction
 - Completer responds to a request
 - Both the Root Complex and the Endpoint device can function as a Completer as well as a Requester

- **The PCI Express protocol supports four types of transactions**
 - Memory (read and write)
 - I/O (read and write)
 - Configuration (read and write)
 - Message (communication information outside of the Memory, I/O, and Configuration spaces such as interrupt signaling, error signaling, etc.)

- **Transactions are divided into three categories**
 - Posted transactions
 - Non-posted transactions
 - Completion transactions

- **Memory writes and message transactions are posted transactions**
 - The requester sends a packet, but the receiver does not return a completion

- **Non-posted transactions (memory reads, I/O reads and writes, and configuration reads and writes) require a response and are implemented as split transactions**
 - Completion packets can be directed to the correct originator because each packet has a unique identifier

- **DO-DI-PCIE-PIPE**
 - PCI Express PIPE Interface Protocol Layer LogiCORE
 - Spartan-3/3E, x1 lane
 - Support for Spartan-3 and Spartan-3E

- **DO-DI-PCIEXP**
 - PCI Express Endpoint LogiCORE
 - Virtex-IIPro and Virtex-4 support
 - Spartan-3/3E PIPE support also included in this version



- **Xilinx offers two different PCI Express Endpoint LogiCORE™ IP solutions to help you configure the built-in low power PCI Express Endpoint Block in Virtex™-5 LXT/SXT FPGA devices**
 - *PCI Express Endpoint Block LogiCORE* - Connects the built-in PCIe block with the GTP transceivers and block RAMs. It also creates and connects the clock and reset logic blocks to the PCIe block
 - *PCI Express Endpoint Block Plus LogiCORE* - In addition to the above integration of GTP, block RAM, and clock and reset logic blocks, also enables seamless migration for current Xilinx PCI Express soft IP users
 - Value added features such as Xilinx LocalLink User Interface to connect to user logic and the Memory BAR filtering and Non-memory TLP ID checking and filtering are included

- **Endpoint applications**

- Provides a soft IP (RTL level) that lies on top of V5LXT's embedded PCIe endpoint block (EZ DMA)
- Provides an easy-to-use PCI-like interface effectively converting PCIe TX/RX signals into Master-Slave signals with multi-channel scatter-gather DMA capabilities
- Backward compatible with PLDA PCI/PCI-X/PCIe core user interface

- **RootPort/Switch applications**

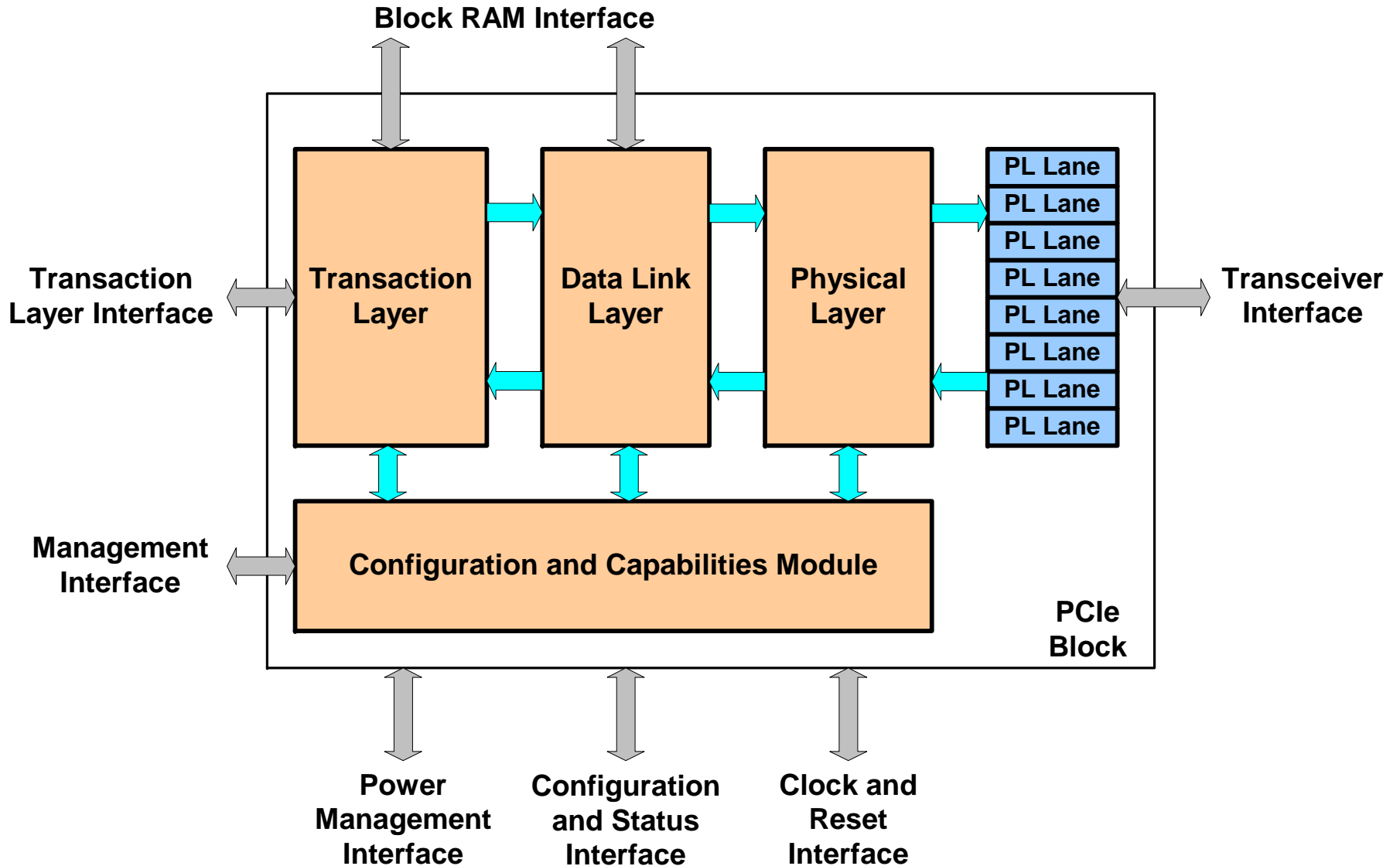
- EZ DMA block for V5LXT
- XpressRICH x1 x4 RP/SW support
- XpressLITE x8 RP support (available end of Q2 2007)

- **What transaction types are supported by the PCI Express protocol?**
 - Memory (read and write)
 - I/O (read and write)
 - Configuration (read and write)
 - Message (communication information outside of the Memory, I/O, and Configuration spaces such as interrupt signaling, error signaling, etc.)

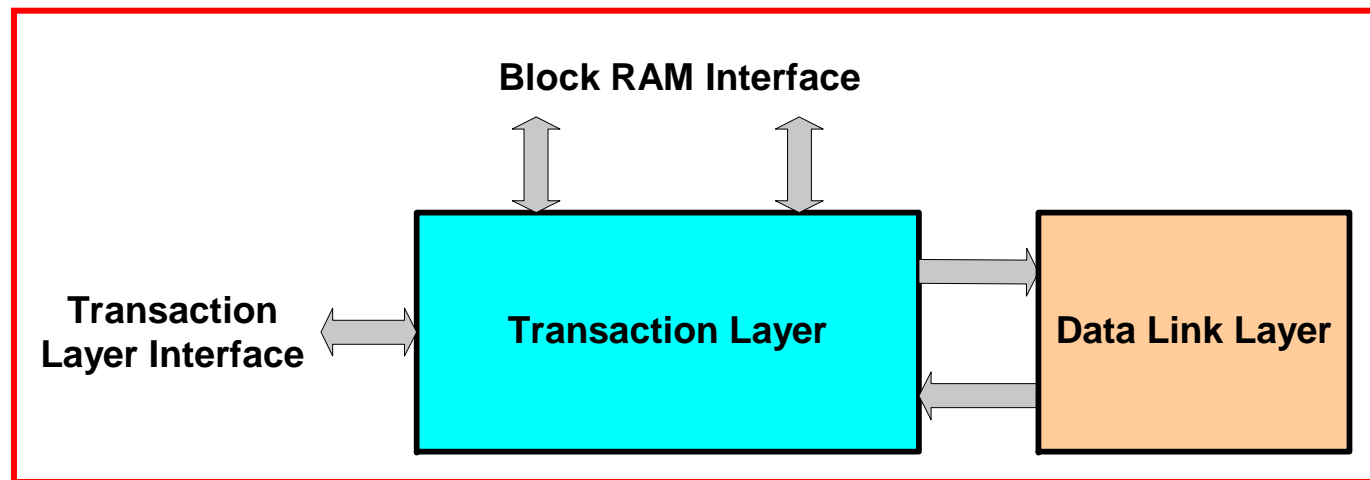
- **What transaction categories are supported for PCI Express transactions?**
 - Posted transactions
 - Non-posted transactions
 - Completion transactions

- Introduction to PCI Express
- **Xilinx Virtex-5 PCI Express Endpoint**
- Designing with the Virtex-5 PCI Express Endpoint Block
- PCI Express Demo

- **The PCIe Endpoint block contains the functionality defined in the specifications maintained by the PCI-SIG**
 - Available in all Virtex-5 LXT/SXT devices
 - Compliant with PCI Express base specification, revision 1.1
 - PCI Express Endpoint block or Legacy PCI Express Endpoint block
 - x8, x4, x2, or x1 lane width
 - RocketIO™ GTP transceivers can implement a fully compliant PHY
 - Block RAMs used for buffering
 - Fully buffered Transmit and Receive
 - Management interface to access PCIe Configuration Space and internal configuration
 - Full range of maximum payload size (128 to 4096 bytes) supported
 - Up to two Virtual Channels (VCs)
 - Up to 6 x 32 bit or 3 x 64 bit BARs (or a combination of 32 bit and 64 bit)
 - BARs configurable for memory or I/O



- **The Transaction Layer (TL) module is the upper layer in the architecture**
 - This module takes Transaction Layer Packets (TLPs) presented by user logic at the Transaction Layer interface and schedules them for transmission over the link
 - The Transaction Layer module also advises the user application when TLPs are received
 - TLPs can both make requests and complete requests from another device



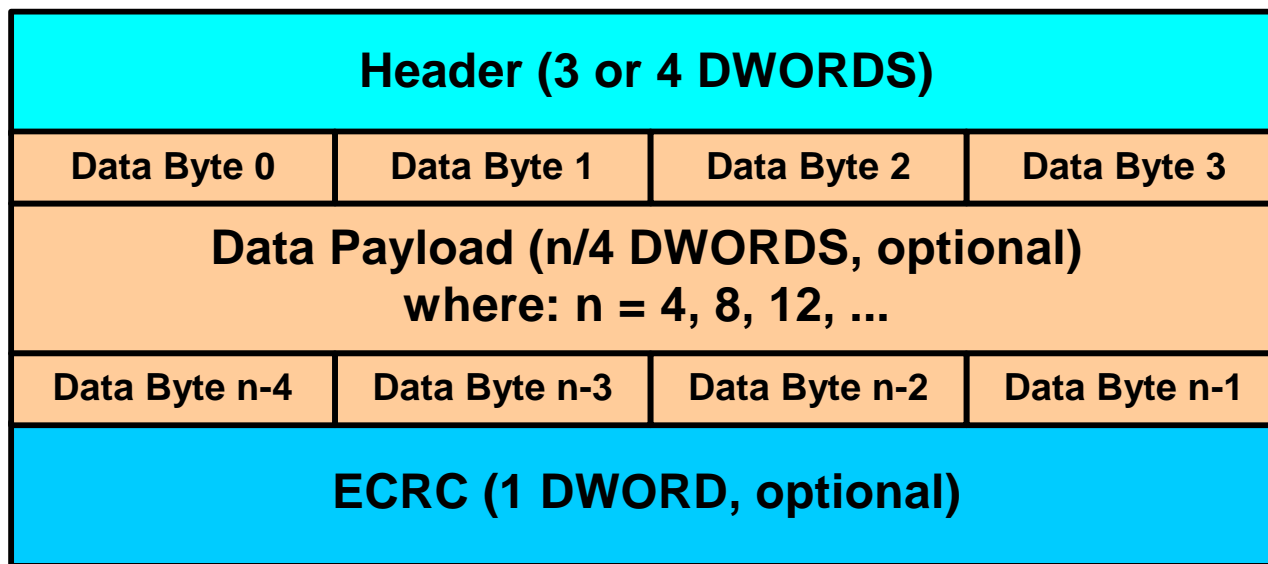
- **A Transaction Layer Packet (TLP) is composed of a header, data payload (for most packets), and optional end-to-end CRC (ECRC)**
 - Packets must be formed by the user in accordance with the PCI Express specification
 - Packets must be decoded by the user properly

	Criteria	Size	Notes
Header	32-bit Machine	3 DWORDS	1 DWORD for address
	64-bit Machine	4 DWORDS	2 DWORD for address
Payload	Read Request	0	No Payload
	Write or Read Completion	Max Payload Size (MPS) or less	4096 bytes in Virtex-5 PCIe Block
ECRC (Digest)	Included	1 DWORD	Passed by TL to User
	Trimmed	0	Still exists, but trimmed by TL

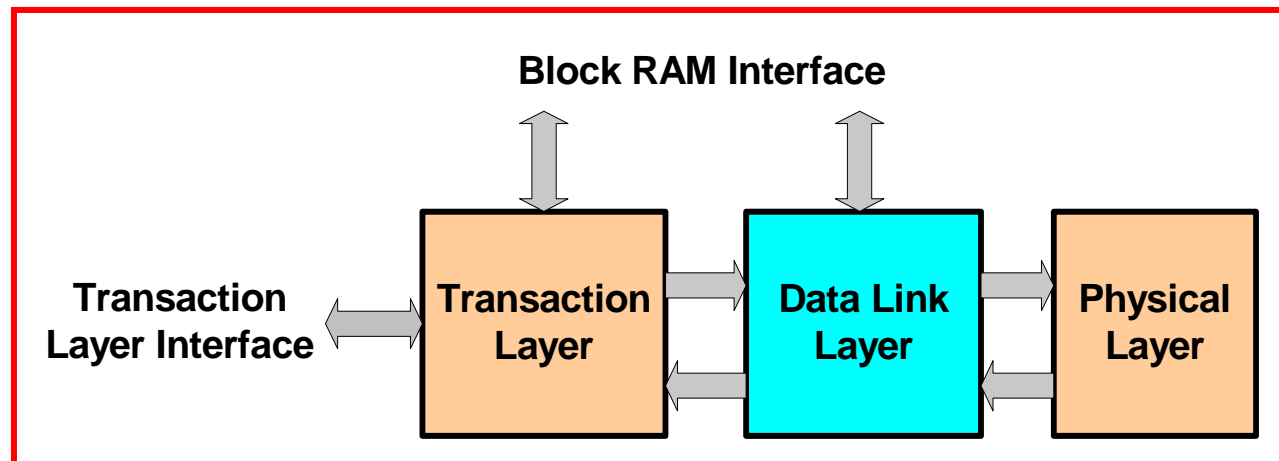
- **The TLP Header is a set of fields that appear at the front of a Packet that contain the information required to determine the characteristics and purpose of the Packet**

- **Depending on the type of a packet, the header for that packet will include some of the following types of fields**
 - Format of the packet (header size and presence of payload)
 - Type of the packet (memory, I/O, configuration, etc.)
 - Length for any associated data
 - Transaction Descriptor, including
 - Transaction ID
 - Traffic Class
 - Completion status

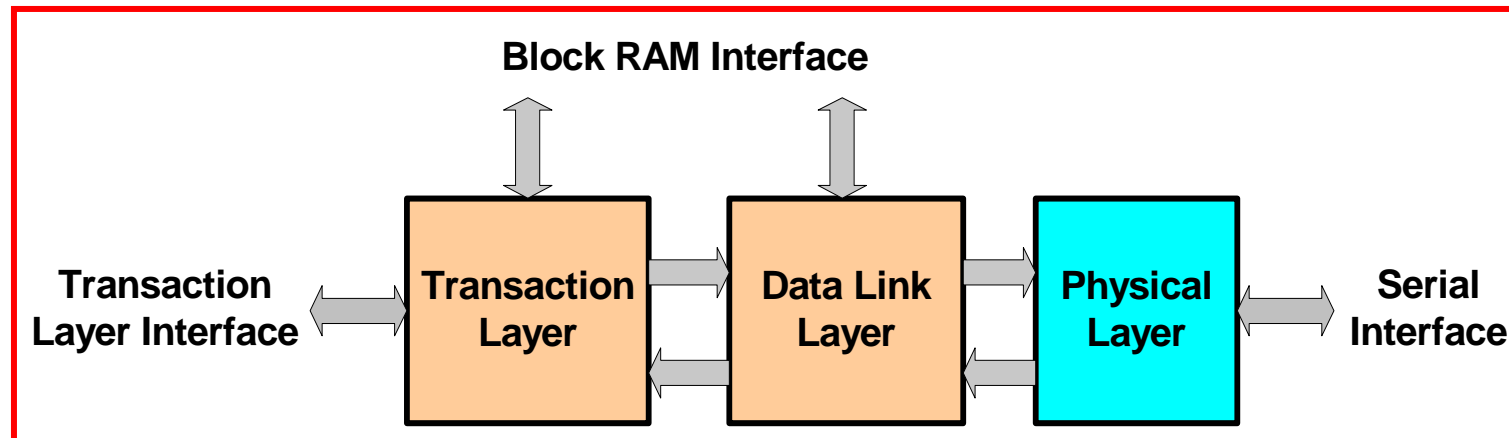
- **Payload is only included for Write Requests and Read Completions**
 - A Read Request TLP would consist of the 3 DWORD Header (for a 32-bit machine) and the optional ECRC
 - A Read Completion TLP returns the payload to the user
 - A Write Request TLP forwards the user payload



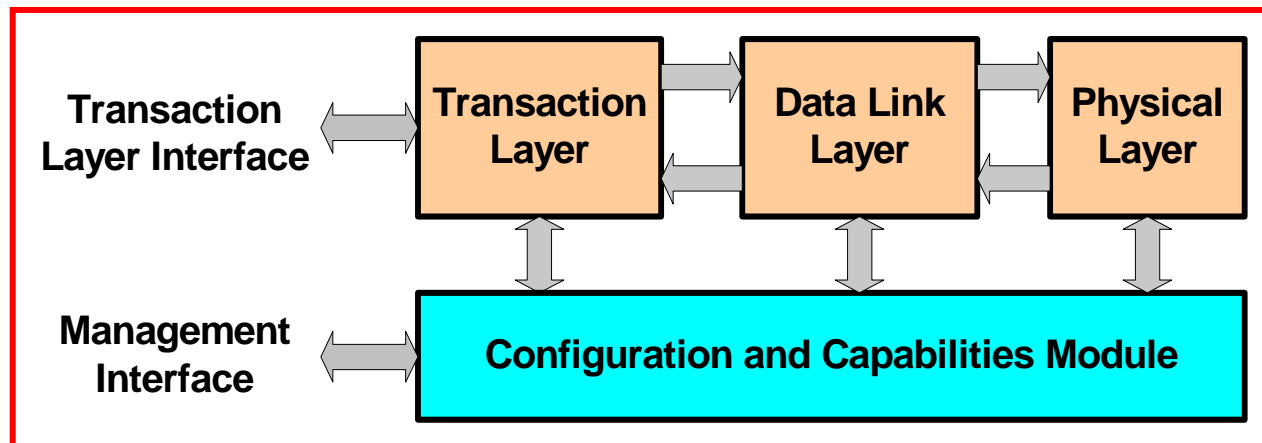
- **The Data Link Layer (DLL) Module resides between the Transaction Layer and the Physical Layer modules**
 - Its primary responsibilities are link management and data integrity, including error detection and correction
 - The transmission portion of the DLL accepts TLPs from the Transaction Layer and generates the appropriate TLP sequence number and Link CRC (LCRC), then passes the packet to the Physical Layer
 - It also places a copy of the packet in a retry buffer, making it available if the packet needs to be resent
 - Nullified packets are automatically purged from the retry buffer



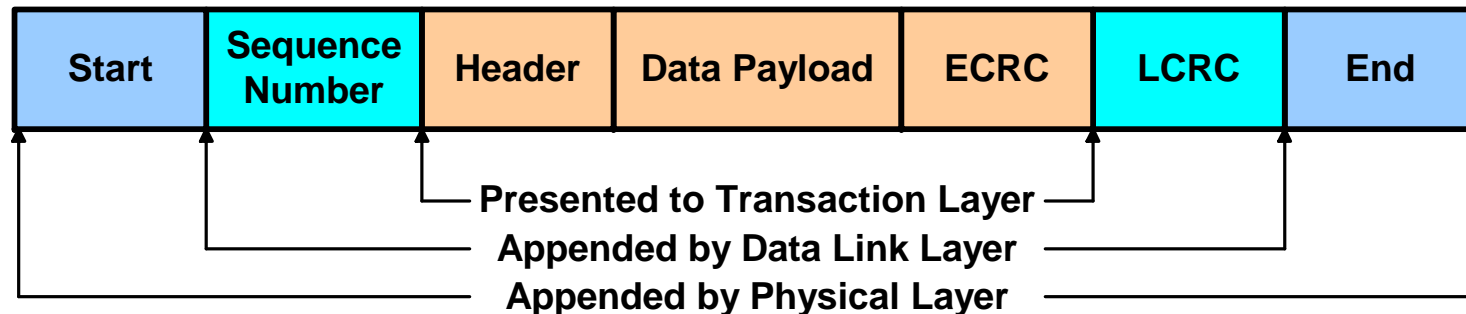
- **The Physical Layer (PL) module carries out the following functions defined for the PL of a PCIe device**
 - Packet framing and de-framing (Start of Frame and End of Frame)
 - Byte striping and un-striping; that is, distributing Tx packets over the associated PL lanes and reassembling Rx packets received over the different PL lanes
 - Link initialization and training
- **The PL is also responsible for power state control and generating the scramble and de-scramble codes that are applied to the data**



- **The Configuration and Capabilities module provides the repository for the different registers within the Configuration Space**
 - It implements the legacy PCI configuration header defined in both the PCI Express Base Specification and the earlier PCI bus specifications
 - It also implements the extended configuration space supported by PCI Express systems that contain
 - Power management
 - Message signaled interrupts
 - Error reporting



- A TLP is composed of a header, data payload (for most packets), and optional end-to-end CRC (ECRC)
 - The optional ECRC is not supported by the Virtex-5 PCIe block
- The transmission portion of the DLL accepts TLPs from the Transaction Layer and generates the appropriate TLP sequence number and Link CRC (LCRC)
- The Physical Layer appends the Start and End to the packet



- **How many layers are included in the PCI Express data path?**
 - Transaction Layer
 - Data Link Layer
 - Physical Layer

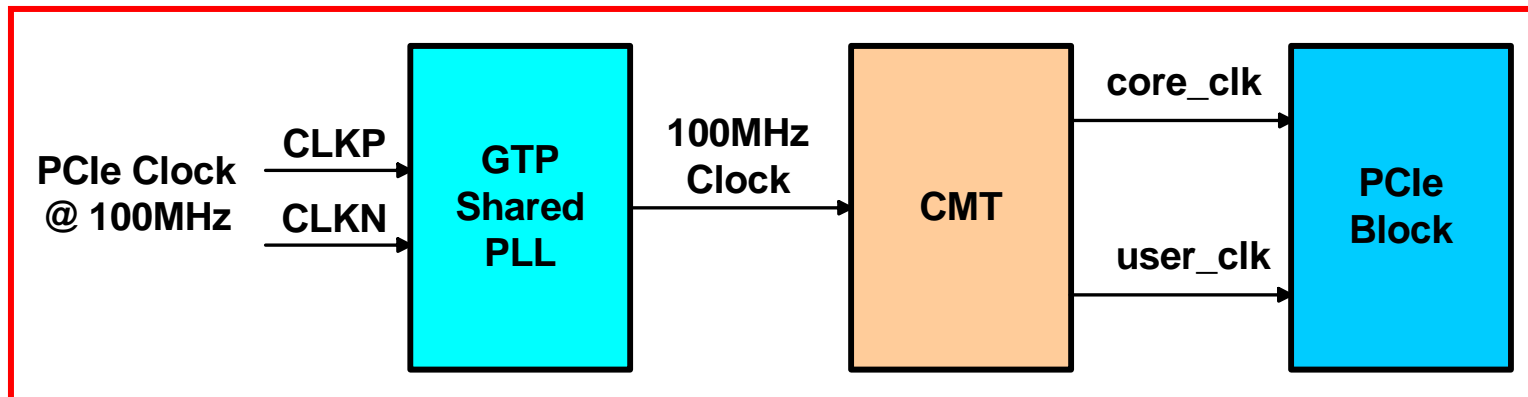
- **Which PCI Express transactions include data payload in the packet?**
 - Write Request
 - Read Completion

- **The Virtex-5 PCIe Endpoint block has two synchronous clock domains, the `core_clk` and the `user_clk` clock domain**
 - The `user_clk` clocks the following interfaces
 - The Management interface
 - The Transaction Layer interface
 - The write port of the Tx buffer
 - The read port of the Rx buffer
 - User logic in the fabric connected to the above interfaces
 - The `core_clk` clocks the following interfaces
 - The rest of the PCIe block
 - The read port of the Tx buffer
 - The write port of the Rx buffer
 - The Retry buffer
 - The Transceiver interface
 - Portions of the GTP transceiver (TXUSRCLK2, RXUSRCLK2)

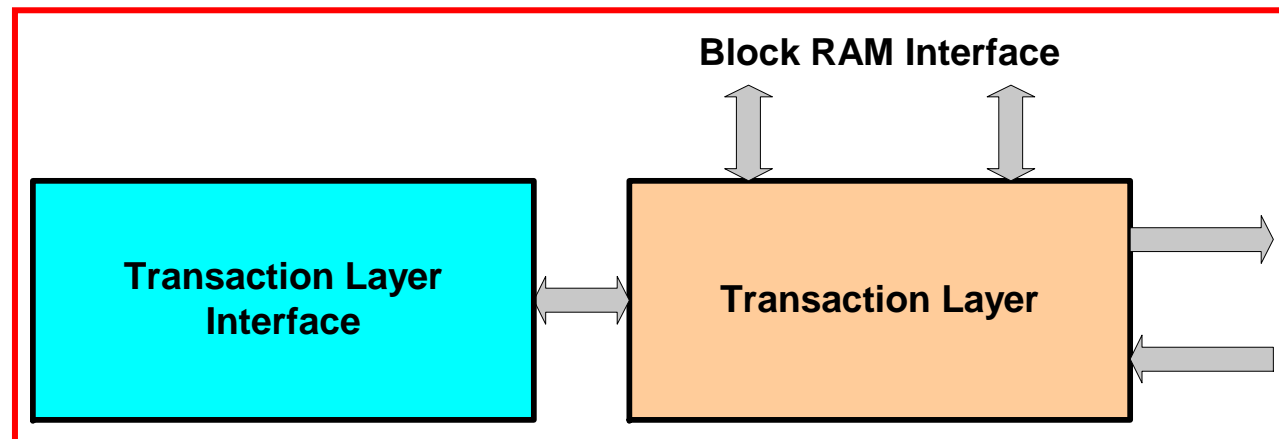
- The `core_clk` always runs at 250 MHz
- The `user_clk` must also run at 250 MHz for x8 configurations to maintain full bandwidth
 - The `user_clk` can be run at lower frequencies for x1, x2, or x4, while still maintaining full bandwidth, lowering power, and simplifying timing closure

Configured Lane Width	User Interface Data Width	core_clk Frequency (MHz)	user_clk Frequency (MHz)	Bandwidth (each direction)
x1	64	250	62.5, 125 or 250	2Gbps
x2	64	250	62.5, 125 or 250	4Gbps
x4	64	250	125 or 250	8Gbps
x8	64	250	250	16Gbps

- **The core_clk and user_clk are obtained by using a Clock Management Tile (CMT)**
 - The reference clock input is connected to the GTP transceiver via the CLKP and CLKN differential reference clock pins
 - The reference clock is forwarded from the GTP transceiver to the CMT via the REFCLKOUT output of the GTP Shared PLL
 - The CMT PLL is used to derive the 250 MHz core_clk as well as the user_clk clock input from the 100MHz reference clock

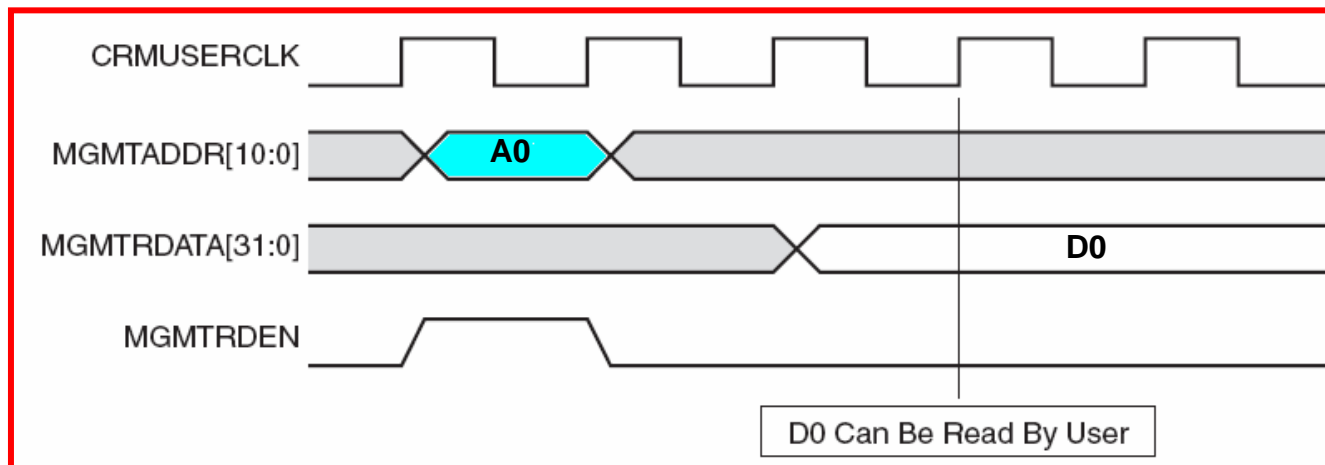


- **Packets are presented to and received from the PCIe block through the Transaction Layer interface**
 - On this interface, a *beat* is a clock cycle where both the source and destination are ready
 - The Transaction Layer interface resides in the **user_clk** clock domain
 - The Transaction Layer Interface framing signals indicate the
 - Start and End of frame
 - Start and End of payload
 - Destination ready
 - Source ready

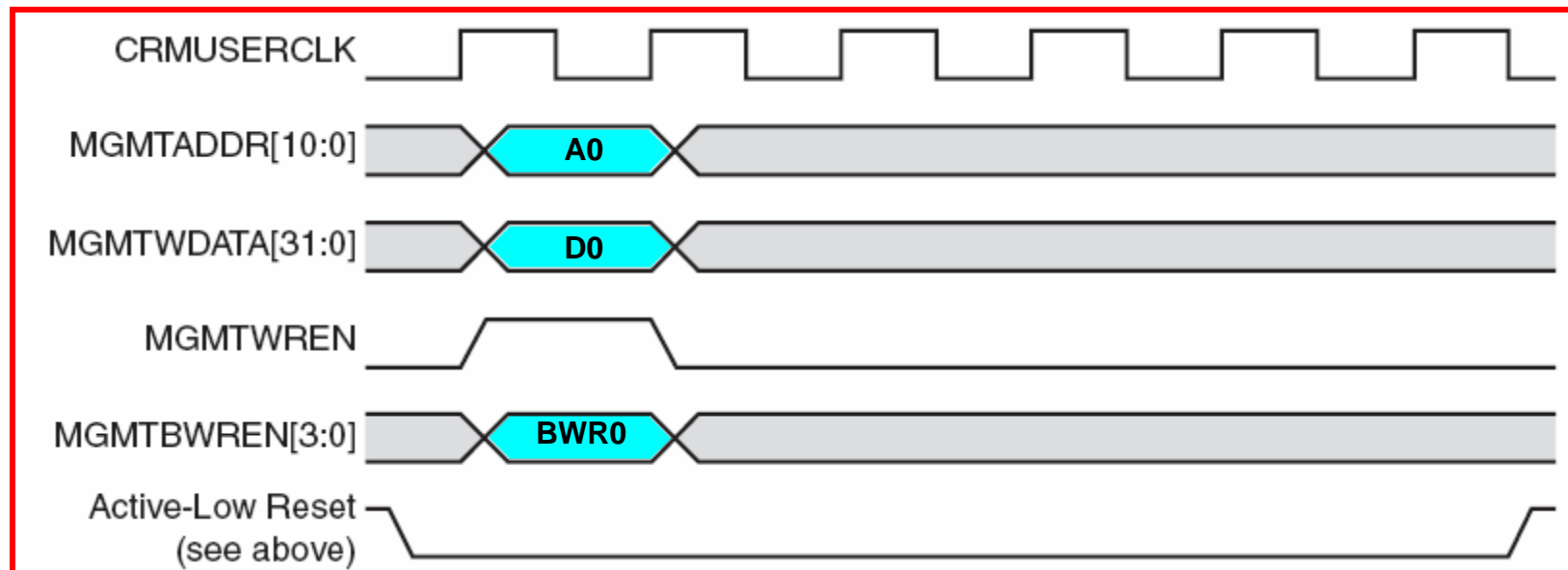


- **The Management interface is used to access various registers and signals in the PCIe block**
 - The PCIe Configuration Space
 - Various control and status registers
 - The Management Interface also contains output signals for statistics and monitoring
- **The Management interface has separate 32-bit data read and write buses**

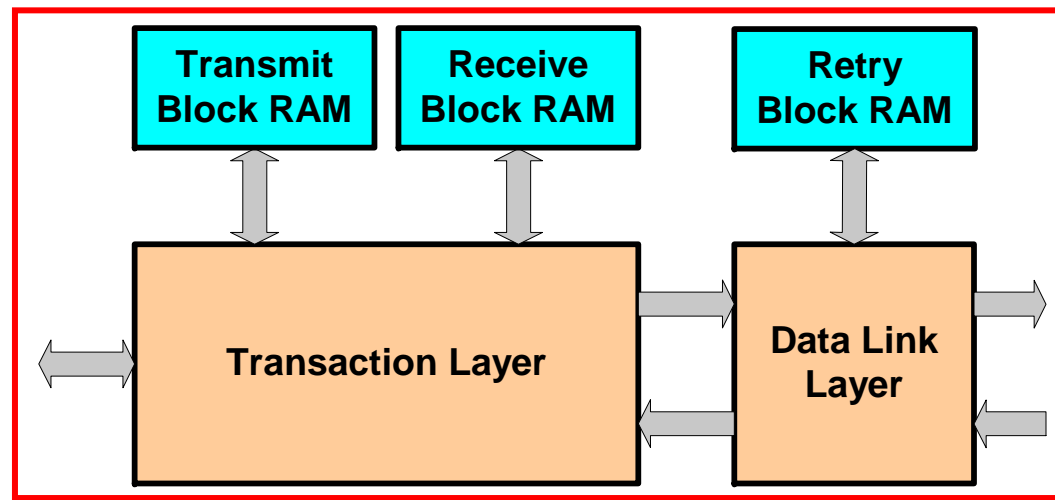
Management Interface Read Timing



- **To use the Management interface to override an attribute (for example, DEVICEID)**
 - The PCIe block must be held in reset during and for at least four cycles (user_clk) after performing a Management write to the attribute address
 - This must be done to allow the new value to propagate within the PCIe block
 - For writes, byte-write enables determine which byte of the 32-bit data is written



- **The PCIe block buffers are implemented using block RAMs. Each buffer has separate read and write interfaces. The sizes of the buffers can vary based on the application's needs**
 - *Transmit (Tx) buffer* - It is divided into separate areas for the different VCs, and each area is further divided into separate FIFOs for posted, non-posted, and completion transactions
 - *Receive (Rx) buffer* - It is divided into separate areas for the different VCs, and each area is further divided into separate FIFOs for posted, non-posted, and completion transactions
 - *Retry buffer* - Holds a copy of each TLP that is currently in the process of being transmitted until the information has been received correctly



- **The amount of block RAM needed can vary greatly, depending on the user requirements**
 - More block RAM is needed for the Tx and Rx buffers when there is a larger maximum payload size or more VCs
 - The amount of block RAM needed for the Retry buffer can increase with multilane designs because the bandwidth is greater
 - The total amount of RAM on each port of the Retry buffer must always be a power of two
 - The TX, RX, and Retry buffers are instantiated and configured in the CORE Generator wrappers, based on selections made in the CORE Generator GUI
 - The Block RAM data paths are 64 bits wide

- **The Virtex-5 PCIe block supports three types of resets, as defined by the PCI Express Base Specification**
 - *Cold reset* - A fundamental reset that occurs following the application of power
 - *Warm reset* - A fundamental reset that is triggered by hardware without the removal and reapplication of power
 - *Hot reset* - An in-band mechanism for propagating reset across a PCIe link

- **Transceiver Interface**
 - Connections between the Transceiver Interface and the GTP transceivers are included in the CORE Generator wrappers
 - Software automatically connects PCIe hard block to GTP transceivers and configures GTP transceiver settings for specific PCIe requirements and lane widths

- **Power Management Interface**
 - This interface includes ports related to Power Management. Most ports in this interface are tied off by the CORE Generator wrapper

- **Configuration and Status Interface**
 - This interface includes control and status, error, backend interface configuration, and interrupt ports

- The following table shows the Virtex-5 PCI Express Endpoint Block Power estimates using the Xilinx Power Estimator (XPE) tool
 - GTP transceiver power dissipation is less than 100mW

Configured Lane Width	user_clk Frequency (MHz)	Power (mW)
x1	250	409
	125	340
	62.5	306
x4	250	420
	125	351
	62.5	317
x8	250	435
	125	366
	62.5	332

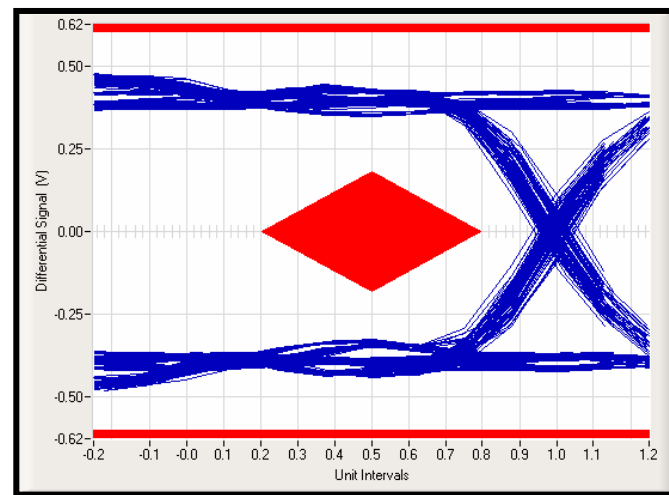
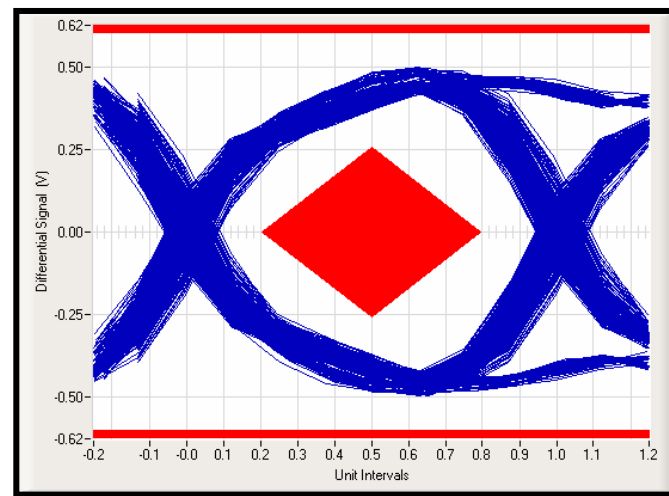
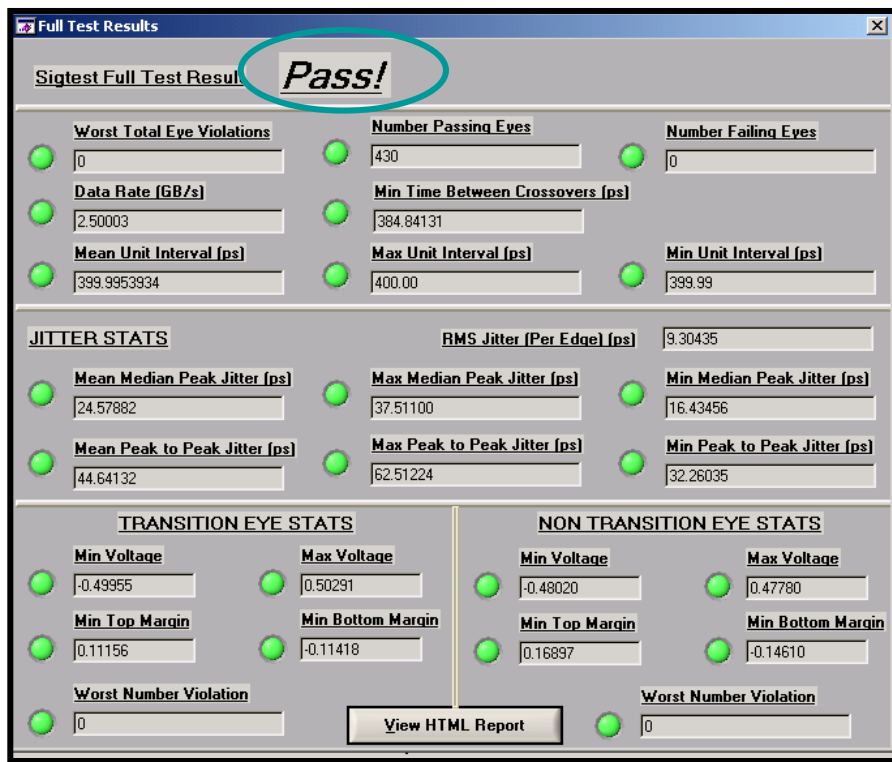
- **Latency of the Virtex-5 PCI Express Endpoint Block Power plus GTP**
 - Memory read initiated from a Root Complex

Configured Lane Width	user_clk Frequency (MHz)	Memory Read Latency (1 DW)
x8	250	380 ns
x4	125	400 ns
x1	125	570 ns

- **Virtex-5 PCIe Endpoint block has passed the following tests**
 - PCI-SIG compliance test
 - 3 SIG Gold suites (Electrical, Configuration and Protocol)
 - Interoperability at 100% (26 suites)
 - X1, X4 & X8 Endpoint configurations

- **Reference boards used for PCI-SIG compliance**
 - ML555 – X4 & X8 configurations
 - ML505 – X1 configuration
 - ML523 – X1 configuration





- **Typical design flow uses the CORE Generator**
 - The CORE Generator wizard configures the required blocks such as GTP, BRAM, Clock, and Reset

- **The CORE Generator wizard supports two user selectable wrappers**
 - *Light-weight native interface (Endpoint Block)* – Compact, enables advanced features such as multiple VCs and larger max payload size
 - *Full-featured (Endpoint Block Plus)* – User friendly interface, supports legacy design migration

- **SmartModel available for the following simulators**
 - Cadence NC Verilog
 - Mentor ModelSim
 - Synopsys VCS



Virtex-5 Integrated PCIe Endpoint Block Benefits

43

Enabling success from the center of technology™

- **Full featured, compliant PCIe solution**
- **Integrated in all Virtex-5 LXT/SXT devices**
- **Guaranteed functionality and performance**
 - Certified in PCI-SIG integrators list
- **Pre-engineered and verified block**
 - Saves design time and effort
 - Saves logic resources, power, and cost
- **Easy-to-use PCIe wizard configures the system**

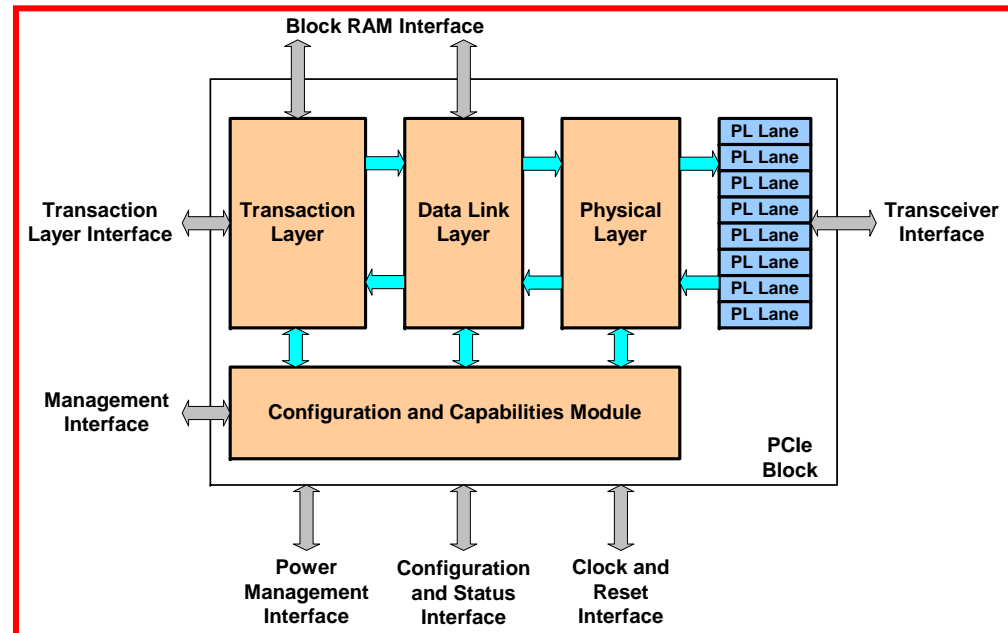
- **What clock domains are used in the Virtex-5 PCI Express Endpoint block and what are their frequencies?**
 - **core_clk** clock domain (250MHz)
 - **user_clk** clock domain (250, 125, or 62.5MHz)

- **How many different buffers are implemented in a Virtex-5 PCI Express Endpoint design with block RAM?**
 - Transmit Buffer
 - Receive Buffer
 - Retry Buffer

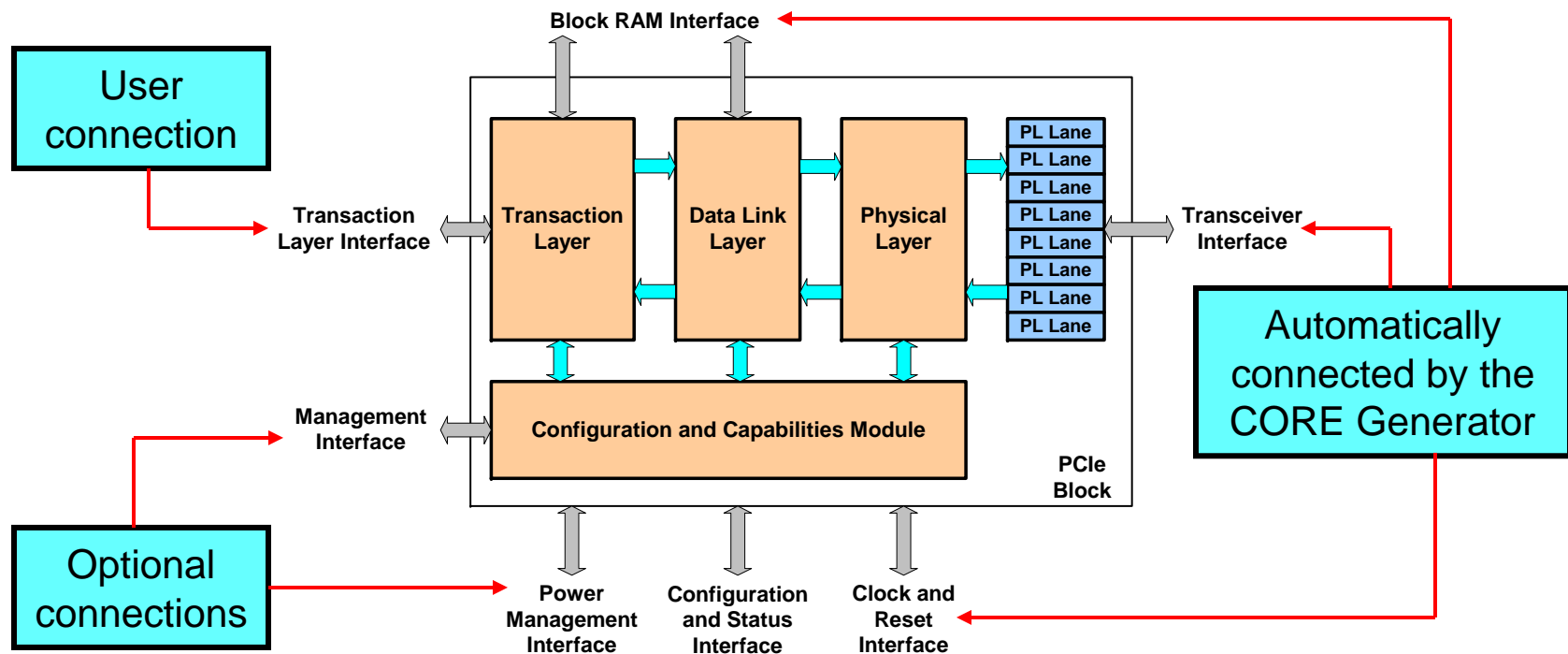
- Introduction to PCI Express
- Xilinx Virtex-5 PCI Express Endpoint
- **Designing with the Virtex-5 PCI Express Endpoint Block**
- Virtex-5 PCI Express Demo

- The Virtex-5 PCIe block is configurable by using a combination of attributes and port tie-offs, as part of the standard FPGA configuration using the CORE Generator wizard

- There are several interfaces to the PCIe block
 - Block RAM
 - Transceiver
 - Clock and reset
 - Configuration/Status
 - Power management
 - Management
 - Transaction layer



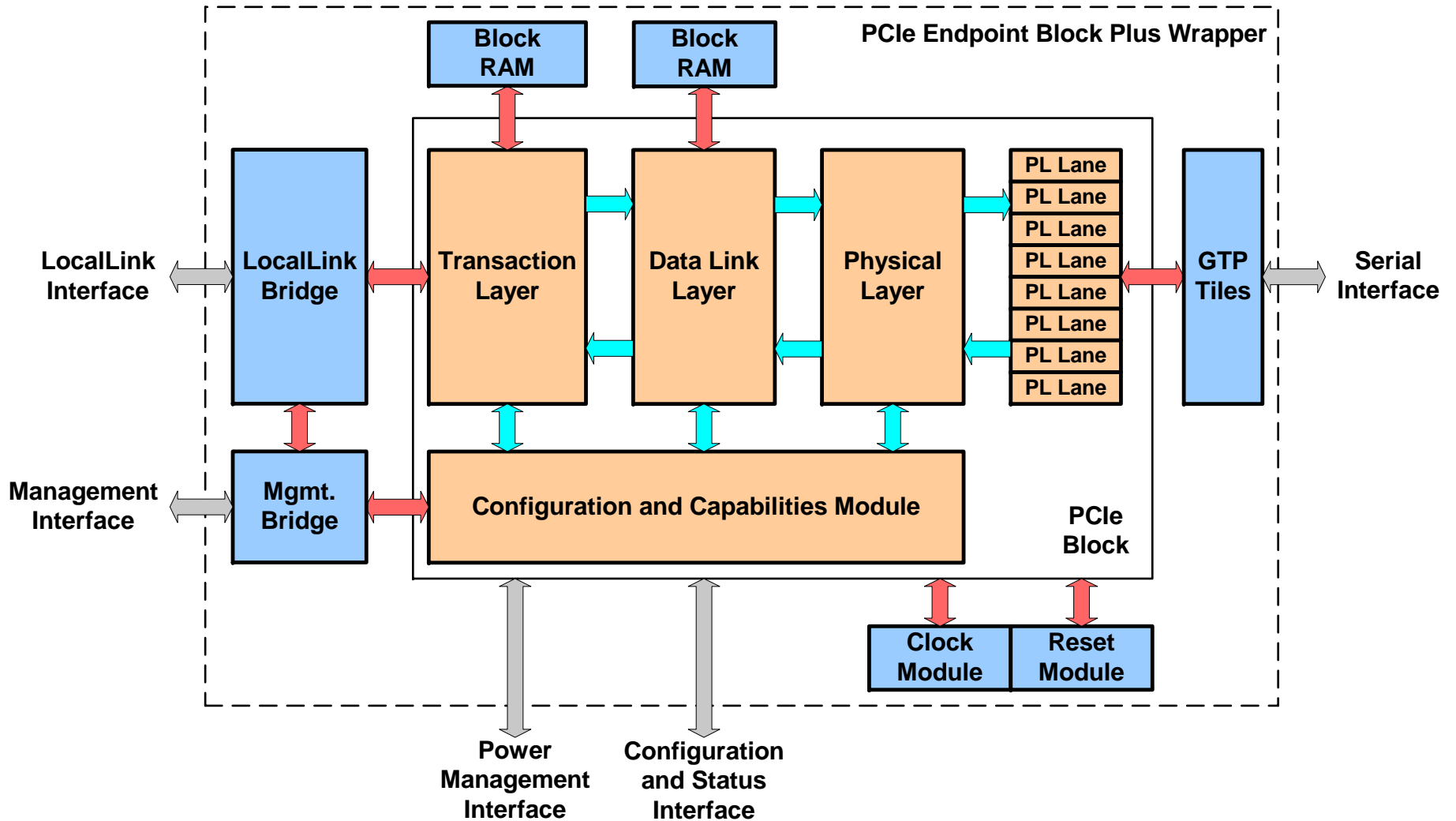
- **The Transceiver, Memory, Clock and the Reset interfaces are automatically connected in the CORE Generator wrappers**
 - These interfaces are not visible outside of the wrappers
 - User application must be implemented in the FPGA fabric and interfaced to the PCIe block using the Transaction Layer Interface
 - The rest of the interfaces such as Power, Configuration, and Management are optional



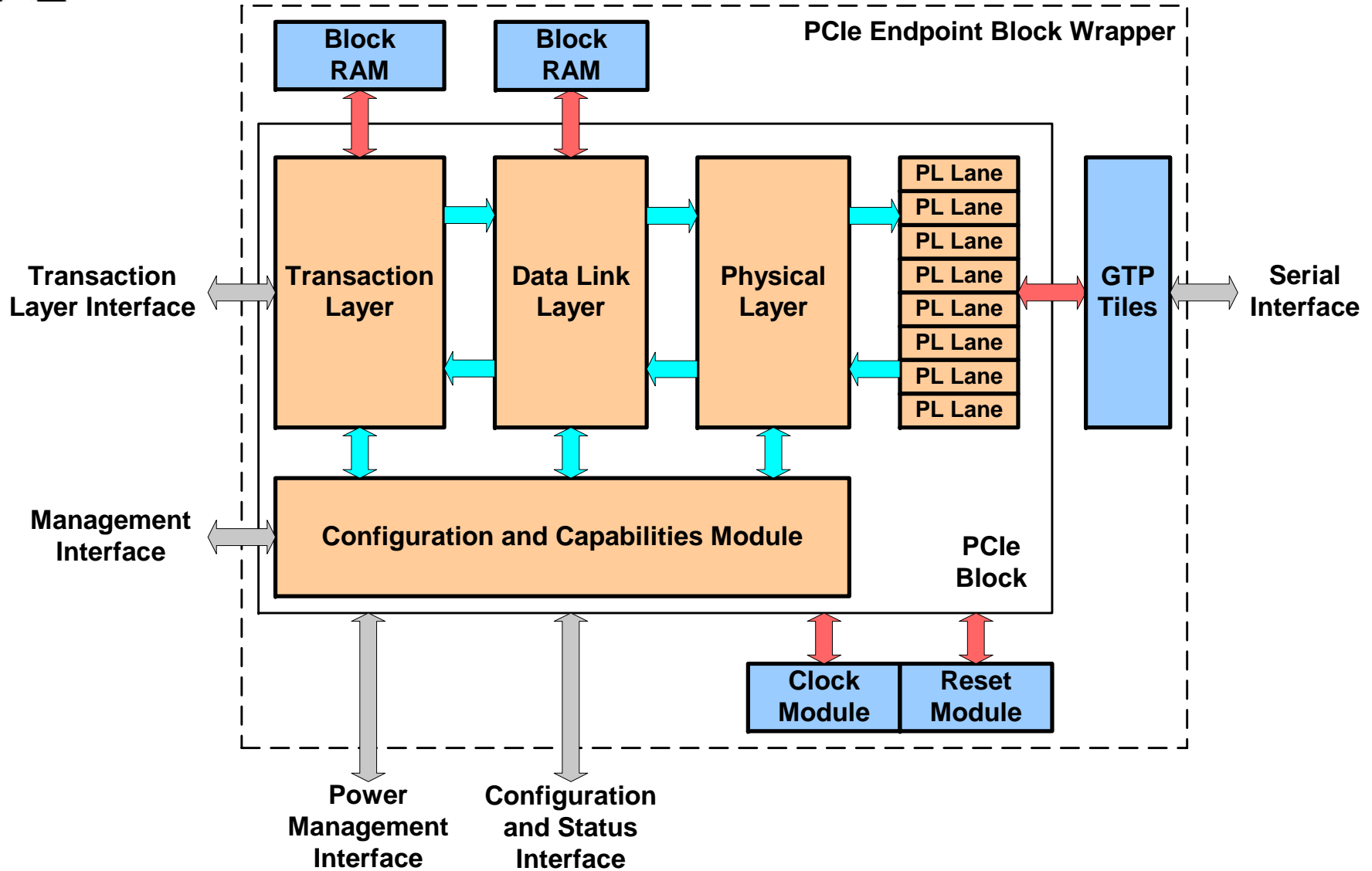
- **The PCI Express wizard is included in the ISE CORE Generator**
- **The PCIe Wizard enables user access to all features of the PCI Express Endpoint block, including the advanced features**
- **The PCIe block is highly complex and customizable. The PCIe Wizard is provided to customize and generate a PCIe subsystem using a simple set of menu options**
 - The PCIe subsystem contains the PCIe block, GTP tiles, block RAMs, clock module, and reset module, which are all automatically configured and connected
 - The options available in the wizard determine the correct attribute settings and tie off any unneeded ports
 - Clicking on the desired options in the wizard generates a completely customized wrapper

- **The PCI Express wizard performs the following functions**
 - Sets the attributes of the PCIe block
 - Ties off unneeded ports of the PCIe block
 - Customizes, instantiates, and connects the desired number of GTP tiles to the PCIe block (collectively referred to as the GTP wrapper)
 - Customizes, instantiates, and connects the desired number of block RAMs to the PCIe block (collectively referred to as the block RAM wrapper)
 - Customizes, instantiates, and connects a clock module to provide a complete clocking solution for the PCIe block, GTP tile, block RAMs, and user application
 - Customizes, instantiates, and connects a reset module to provide the necessary PCIe block resets
 - All of the above are contained in a single entity, the PCIe wrapper or PCIe subsystem

- **Xilinx provides two Virtex-5 PCI Express Endpoint LogiCORE wrapper solutions**
 - **Endpoint Block Plus** for faster-time-to-market
 - Optimized configuration for Endpoint applications
 - x1, x4 and x8 lane width
 - Block RAM usage optimized for 512 Bytes Max Payload Size
 - Enables 1 Virtual Channel (VC) operation
 - Transaction Layer interface via Xilinx LocalLink interface
 - BAR checking, reporting, filtering of Memory Transactions
 - Error Reporting and Management
 - **Endpoint Block** for maximum design flexibility
 - User customization of PCIe built-in block features
 - Configurable Max Payload Size from 128 to 4096 bytes
 - 1 or 2 Virtual Channels
 - x1, x2, x4 and x8 lane width
 - Customizable user interface for power users



Enabling success from the center of technology™



- **Parameterized Core Net-list**
- **Programmed Input Output (PIO) example design**
- **Customer Simulation Demonstration Test bench**
 - Verilog HDL simulation flow supported for PIO (VHDL planned)
 - Includes complete Downstream PCIe port models (1 and 4 Lane)
 - May be used to verify complex customer Endpoint designs
- **Customer Implementation Demonstration**
 - Example UCF delivered targeting ML555 Rev C board
 - Complete implementation scripts delivered for PIO design

- **Easy to use solution**
 - Hides most PCIe Endpoint Block complexity/details

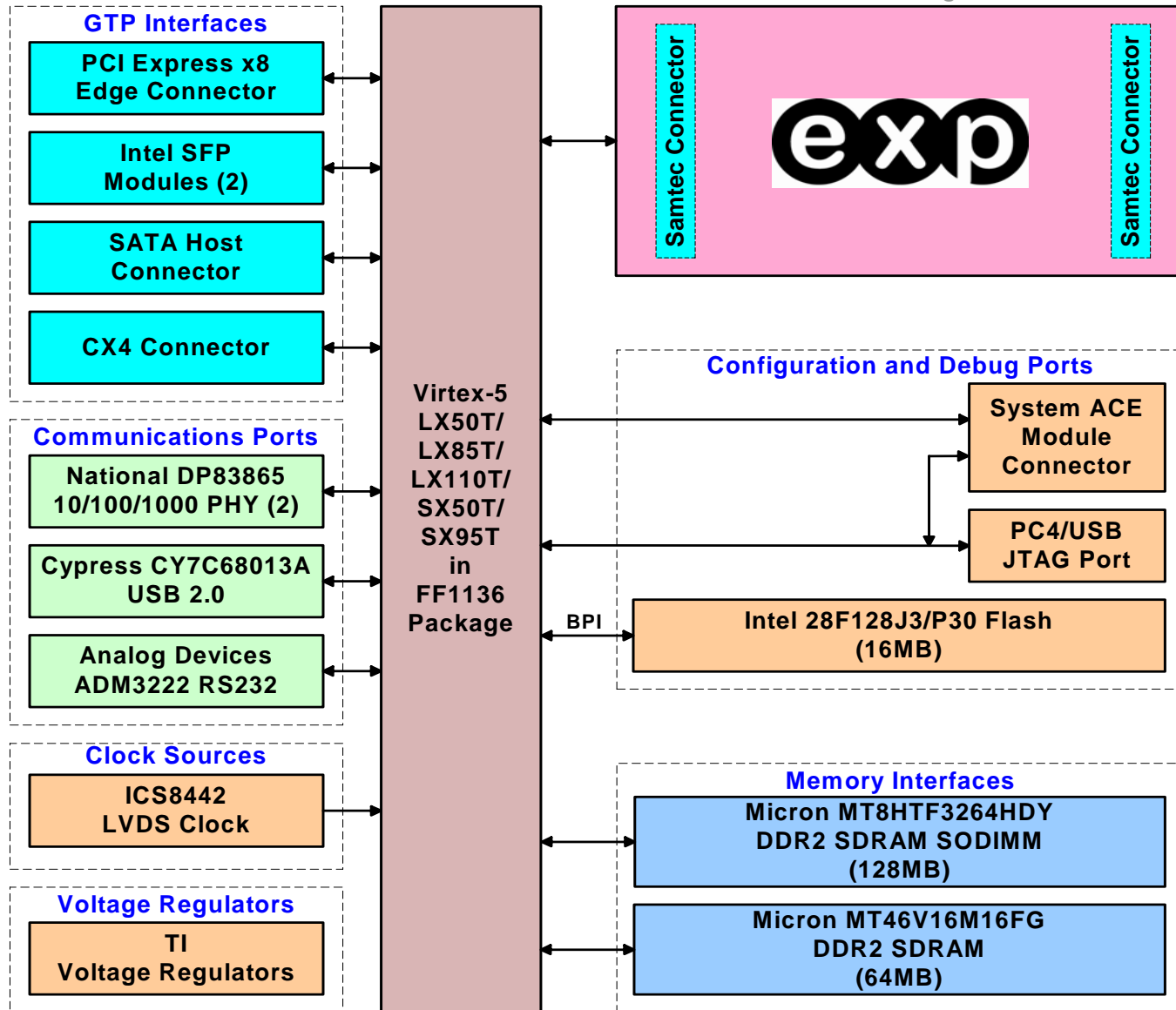
- **User does not have to be a PCIe expert**
 - Knowledge level required is similar to that of using Xilinx PCIe soft core

- **Provides many PCI Express required features**
 - Significant reduction in customer engineering/verification effort

- **The Endpoint Block Plus core should also be used when**
 - Converting designs based on Xilinx Virtex-4/VII-PRO PCIe soft cores
 - Endpoint application needing Xilinx LocalLink backend interface

- Introduction to PCI Express
- Xilinx Virtex-5 PCI Express Endpoint
- Designing with the Virtex-5 PCI Express Wizard
- **Virtex-5 PCI Express Demo**

Enabling success from the center of technology™



- **Using the CORE Generator PCIe wizard to create a PIO reference design for the Avnet V5LXT/SXT PCIe development board**
 - Implementing the design using the ISE tool in batch mode