PCI Express PIPE Overview

MindShare, Inc.

Dave Dzatko

March 2004

Introduction

PIPE, which stands for the **P**hysical **I**nterface for **P**CI **E**xpress Specification developed by Intel, has the stated intent of providing a standard interface between the internal logic of a PCI Express design and the analog and high-speed circuitry required to implement the serial link. This purpose of this functional separation is to allow ASIC and integrated circuit designers to focus on the PCI Express device core, Transaction, Data Link and logical Physical Layers, while relying on the PIPE-compliant physical design (PHY) for the electrical interface of the design. Other vendors can then provide macrocells to handle this high-speed analog interface layer, resulting in reduced time and risk for the overall design cycle.

The PIPE spec defines standard functionality that a PIPE-compliant PHY needs to implement, as well as a standard parallel interface between the PHY and the internal logic referred to in the spec as the Media Access Layer (MAC). The MAC in turn connects to the PCI Express Data Link Layer logic. The PIPE spec builds on the PCI Express base spec, so it should be noted that a working knowledge of that document is essential for a good understanding of the PIPE spec. This paper is based on the 1.0 version of the PIPE spec, and provides a brief introduction only. For more information on PIPE, refer to Intel's web site[1]. For more information on PCI Express, refer to Mindshare's web site[2] or the PCI Special Interest Group site[3].

PIPE Architecture

The PIPE architecture block diagram is shown in Figure 1. As mentioned earlier, the Data Link Layer (DLL) logic interfaces with the MAC, although the spec doesn't define this interface or any others except the one between the PHY and the MAC. All the other interconnects are understood to be implementation specific. The interface between the MAC and the PCS (Physical Coding Sub-layer) is a dual simplex, parallel bus called the PHY/MAC interface. The PCS and PMA (Physical Media Attachment Layer) interface is not defined by the spec. The functionality of the PCS and PMA are described conceptually in the PIPE and PCI Express specifications but, again, no specific implementations are implied. The PMA contains the high-speed analog and digital circuitry that connects to the PCI Express link via differential transmitters and receivers. The PIPE spec notes that there is overlap between it and the PCI Express spec, and states that in case of any conflict between them, the PCI Express spec takes precedence.

[1] http://www.pciexpressdevnet.org/apps/org/workgroup/devnetgf/pipe/

[2] http://www.mindshare.com/knowledge/pciexpress_overview.html

[3] http://www.pcisig.com/home

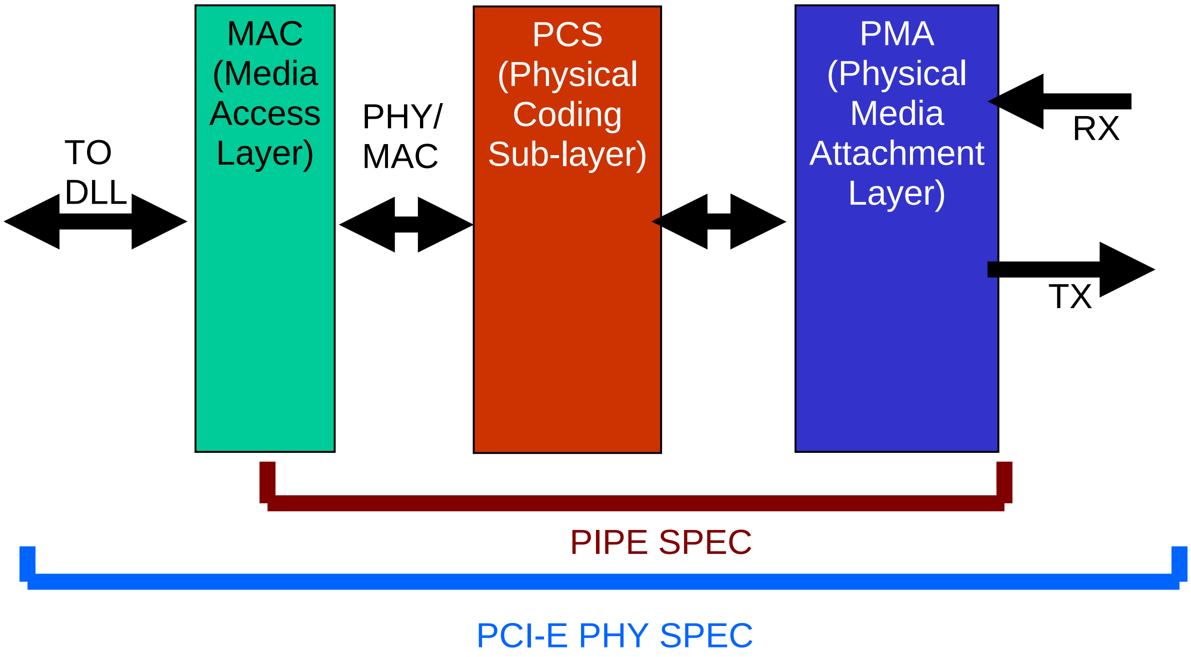

Figure 1. *PIPE Architecture*

MAC Architecture

The MAC contains many of the PCI Express logical Physical Layer circuits (such as the Link Training and Status State Machine (LTSSM), data scrambling, 8b/10b encoding, and byte striping), and functions as the bridge between the DLL and the PHY/MAC interface.

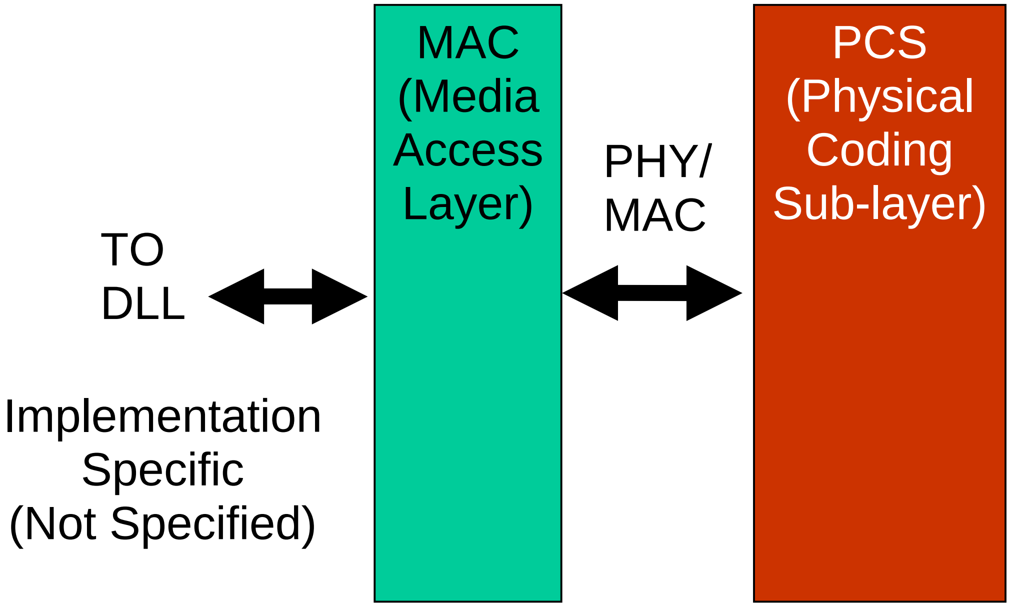

Figure 2. *MAC Architecture*

.

PHY/MAC Interface

The PHY/MAC interface is the major normative area of the PIPE spec, as shown in Figure 3. The pins that make up the interface and their functionality are described in the spec, and timing diagrams are provided to show the synchronous timing relationships, but no detailed timing parameters for the signals are given. Instead, the spec provides a description of implementation-specific timings that a vendor of a PIPE-compliant PHY macrocell or discrete chip must specify. Much of the functionality of the PHY/MAC Interface is described in the spec using timing diagrams, which are not repeated here.

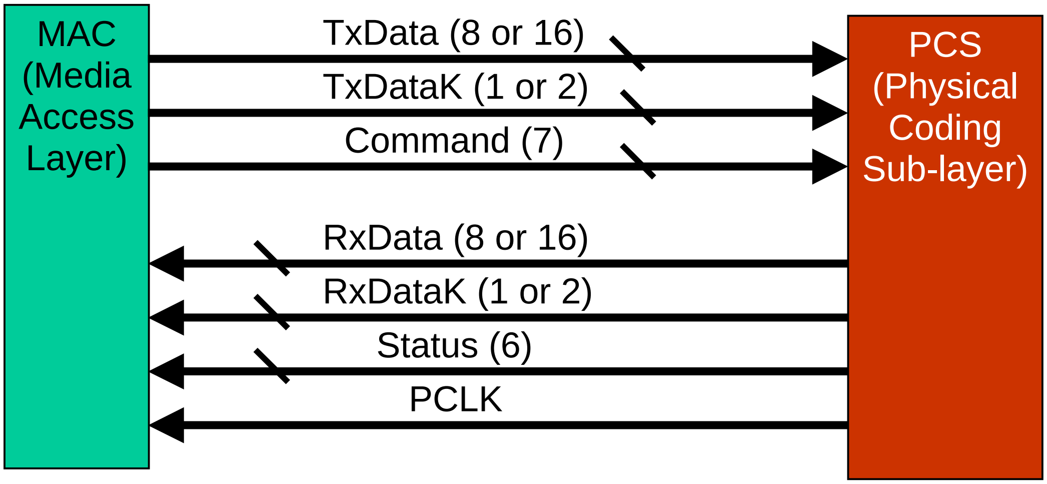

Figure 3. *PHY/MAC Interface*

The PHY/MAC Interface is a parallel interface for transferring data to be transmitted on the PCI Express bus. The width of this parallel interface for bytes of data is shown as either be 8 or 16 bits in each direction. In addition to the data path, the PHY/MAC Interface includes one Control (K) lines in each direction for each byte to indicate whether the corresponding byte is a data or a control character. If the interface is 16 bits wide, there will be two Control lines, but only one is needed if the interface is 8 bits wide.

Seven signals from the MAC to the PCS make up the command bus that allows the MAC to tell the PHY to begin performing a variety of tasks, including:

- Begin Receiver Detection
- Enter the External Loop-back state
- Enter the Electrical Idle state
- Set the current running disparity to positive or negative
- Invert the polarity of the received data

- Reset the PHY
- Power down the PHY into various Power Management (PM) states

The PHY/MAC Interface also includes a 6-bit status bus for communication of PHY status from the PCS to the MAC. The status bus has codes and signals that indicate conditions including:

- The PHY has obtained symbol lock
- The received data is valid
- The PHY has completed various PM state transitions
- The PHY has detected that a receiver is attached
- The PHY has detected an Electrical Idle state on the Link
- There are 8 additional status and error codes defined

The PHY includes a clock output called PCLK which is used to synchronize data transfers across the parallel PHY/MAC Interface. If the parallel interface is 16 bits wide, then PCLK runs at 125 MHz, but if the interface is 8 bits wide, then the PCLK runs at 250 MHz so as to maintain the data rate with respect to the serial link.

PCS Architecture

The Physical Coding Sub-layer (PCS), although part of the Logical Sub-block of the PHY, is included within a PIPE-compliant discrete device or macrocell. The PCS supports the PHY side of the standard PHY/MAC Interface as well as an implementation-specific interface between it and the Physical Media Attachment Layer (PMA). No specifications are provided in the PIPE document for this interface.

The clock reference input (CLK) is used by the PHY to generate the internal bit rate clocks for transmitting and receiving PCI Express data. Specifications for this implementation-specific clock, which will be used internally to generate the bit-rate clock for the PHY transmitter and receiver as well as the PCLK for the PHY/MAC interface, must be provided by PIPE-compliant PHY vendors. Spread spectrum modulation that matches the system reference clock modulation is permitted for this signal.

The 8B/10B logic required by PCI Express resides in the PCS. The input to the 8B/10B encoder is the data or control character from the PHY/MAC Interface, the value of the current running disparity, and the TxDataK signal that is needed because the encoding for a data character is different from that of a control character.

The PCS also contains the elastic buffer used to compensate for the slight variations in frequency between the transmitter’s clock and the clock used by the receiver to process the incoming data.

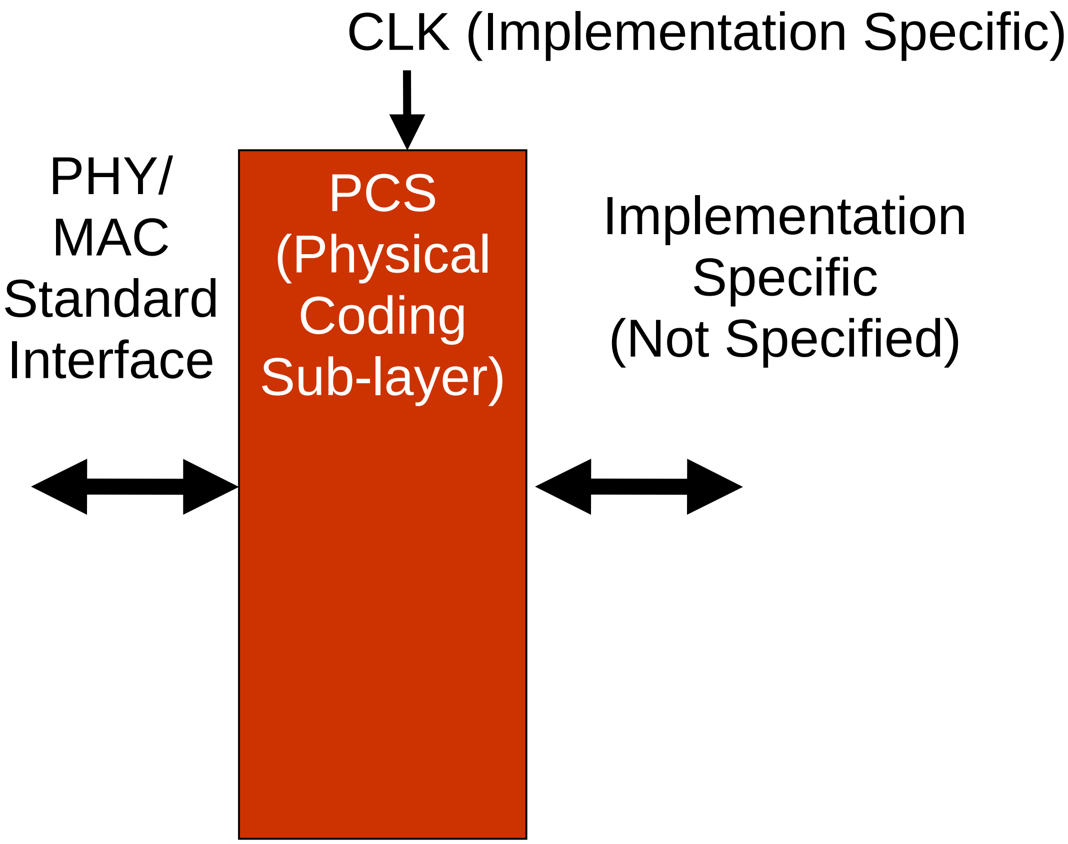
CLK (Implementation Specific)
PHY/ MAC Standard Interface
PCS (Physical Coding Sub-layer)
Implementation Specific (Not Specified)

Figure 4. *PCS Architecture*

After reset, the first state of the link initialization and training process is the receiver detect, during which the transmitter detects whether a receiver is attached to it at the other end of the link. The PCS contains the logic that controls the receiver detection and reports the detect status to the MAC via the PHY/MAC Interface.

The PCS also contains a Phase Lock Loop (PLL) to generate the internal, high speed clocks used for the PHY based on the CLK input.

PMA Architecture

The Physical Media Attachment (PMA) Layer implements the high-speed analog and digital circuitry for PCI Express signaling, including the differential drivers and receivers for each lane of a link. Although the data is serially transmitted over the link, the connection from the PCS to the PMA is a ten bit wide, implementation-specific parallel interface. The high-speed serialization and de-serialization logic (SERDES), needed to create the serial data stream of PCI Express, resides in the PMA.

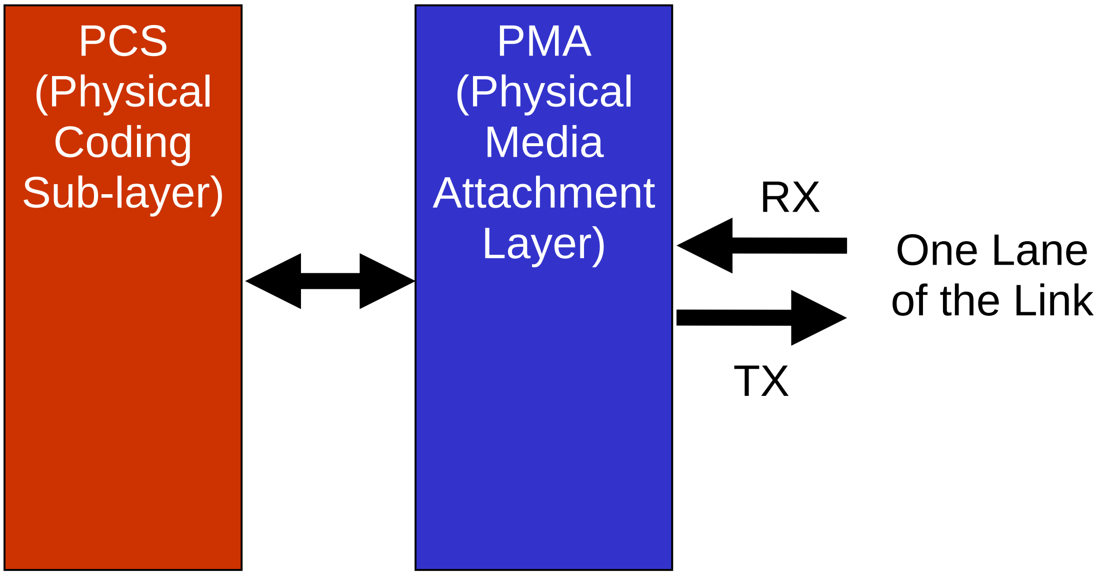
PCS (Physical Coding Sub-layer)
PMA (Physical Media Attachment Layer)
RX
TX
One Lane of the Link

Figure 5. *PMA Architecture*

PHY Functionality and Features

A PIPE-compliant PHY discreet or macrocell, as shown in Figure 6, is designed to handle all the low-level PCI Express protocol and high-speed PCI Express signaling. To review, the required functionality provide by the PIPE-compliant PHY includes:

- The SERDES or serialization/de-serialization circuitry
- The 8B/10B Encoder and Decoder
- Analog buffering for the received data and the data to be transmitted
- Elastic buffering for clock tolerance compensation
- Receiver detection circuitry
- Disparity control for use in transmitting the PCI Express compliance pattern
- Beacon transmission and reception circuitry for implementing remote wakeup capability

The PIPE spec does not define the specific internal architecture or design of the PCS and PMA functional blocks. These functional blocks simply represent high-level functionality that must exist in a PIPE-compliant PHY. The PIPE specification does describe, in general, the architecture and behavior of the PCS and PMA.

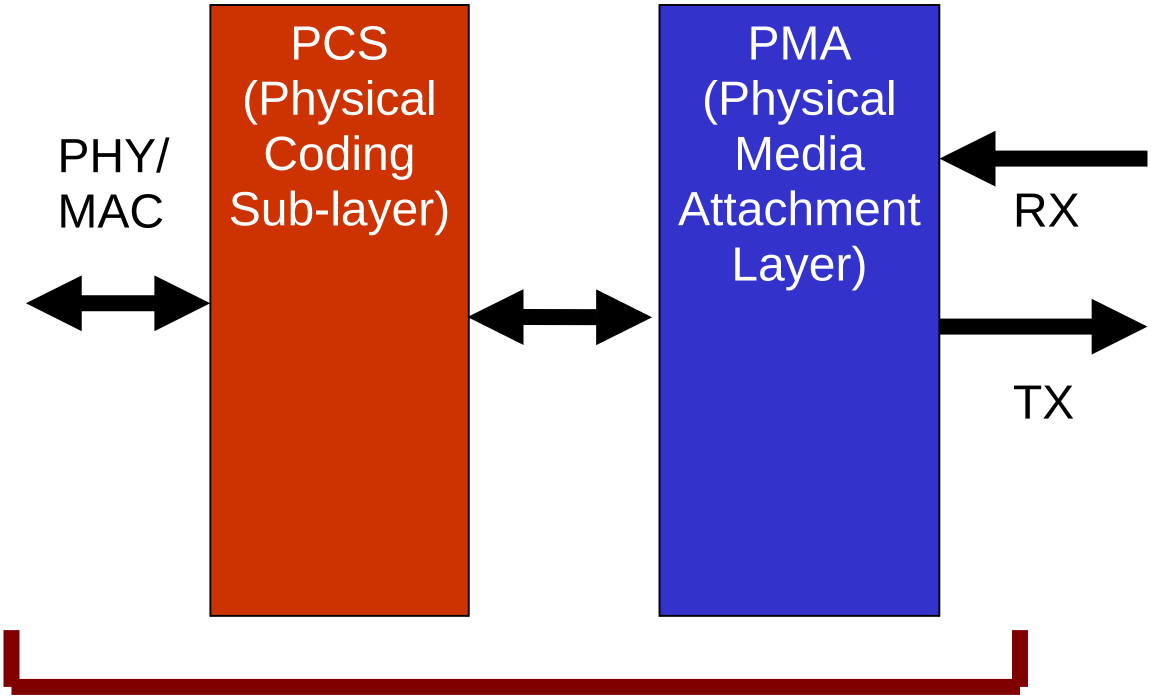

Figure 6. *PIPE Functionality*

PIPE Signal Organization

A PIPE-compliant PHY will have signal groups as shown in Figure 7. There are four basic signal groups:

1. Clocks to and from the PLL
2. Transmitter Data and Control
3. Receiver Data and Control
4. Command and Status

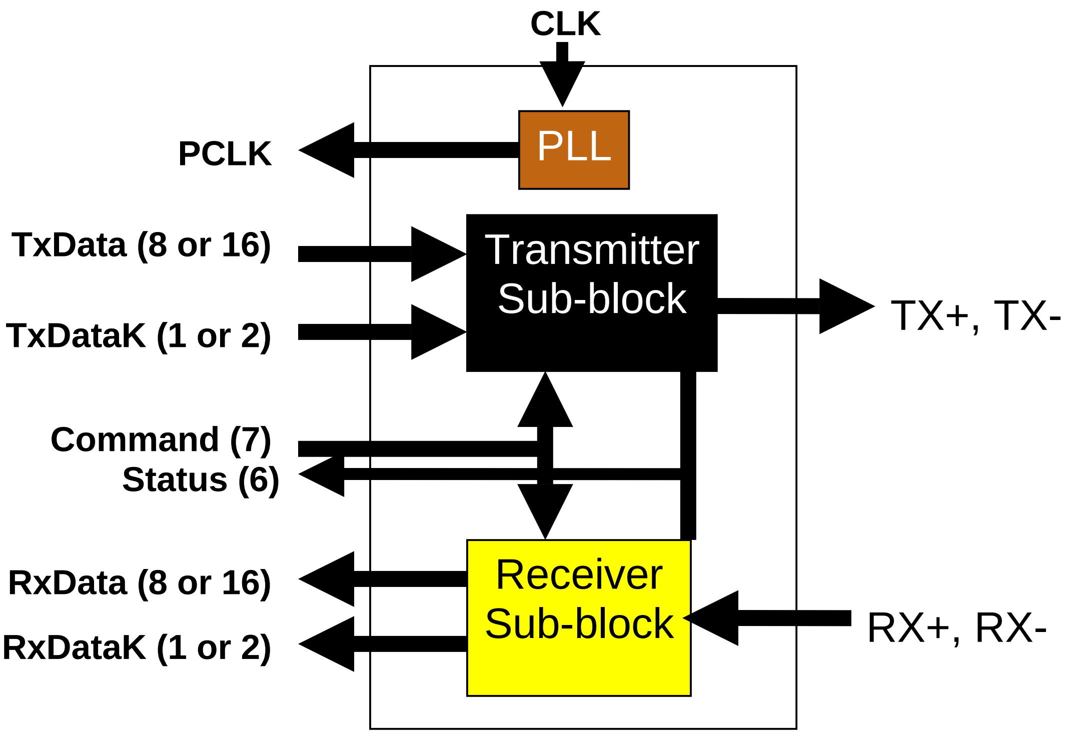

Figure 7. *PIPE Signal Organization*

PIPE PLL

The PLL generates the PCLK used in synchronizing the parallel PHY/MAC Interface based on the CLK input. The PCLK frequency is 125 MHz for 16-bit implementations and 250 MHz for 8-bit implementations. The PLL will produce a 250 MHz clock used as an input to the 8B/10B encoder and decoder and the elastic buffer. The PLL will produce a 2.5 GHz clock that is used as an input to the SERDES and the clock recovery circuitry.

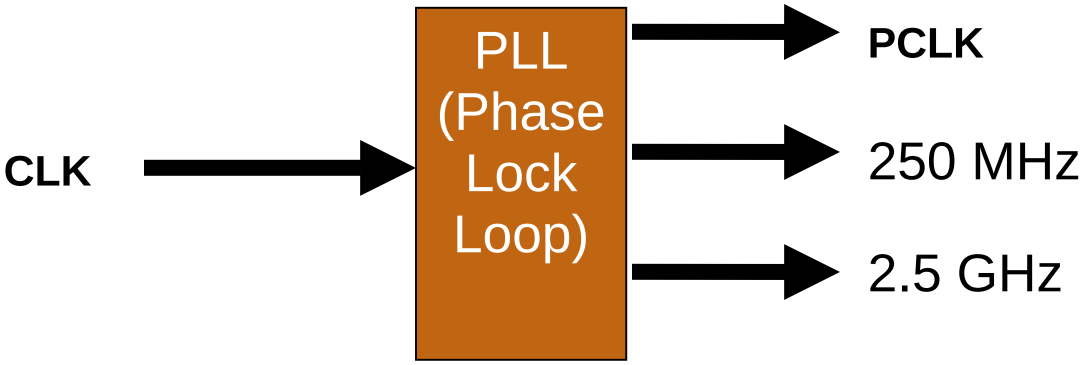

Figure 8. *PIPE PLL*

Transmitter Sub-block

The Transmitter Sub-block of the PHY has 4 distinct sections. Data to be transmitted comes from the PHY/MAC Interface at the rate of either 8 or 16 bits per rising edge of the PCLK into an optional 16-to-8 bit converter. This converter would, of course, be unnecessary if the PHY/MAC interface is already 8 bits wide, but it would ensure delivery of 8 bits per tick of a 250 MHz clock to the 8B/10B Encoder if a 16-bit interface is used.

The 8B/10B Encoder converts the 8-bit character into a 10-bit symbol at 250 MHz, using the TxDataK input to determine whether it is encoding a control or data character. The TxCompliance input signal is used to force the current running disparity at the encoder to be negative, permitting forced error conditions for compliance testing.

The Loop-back path is shown to define the flow of data when the PHY is placed into the loop-back mode. In that case, the data to be looped back from the receiver bypasses most of the transmitter logic and goes directly to the parallel-to-serial converter.

The parallel to serial converter creates a serial bit stream running at 2.5 GHz using a clock derived from the internal PLL and the differential driver sends the bit stream out in accordance with the PCI Express specification. The TxElecIdle signal informs the driver that it should enter the Electrical Idle state. If this is not set, the driver should be sending something – either data during normal operation, or else the beacon signal if operating in lower power states. If it is set, the transmitter should switch to electrical idle as quickly as possible, though it is required to send at least three bytes of the electrical ordered set first.

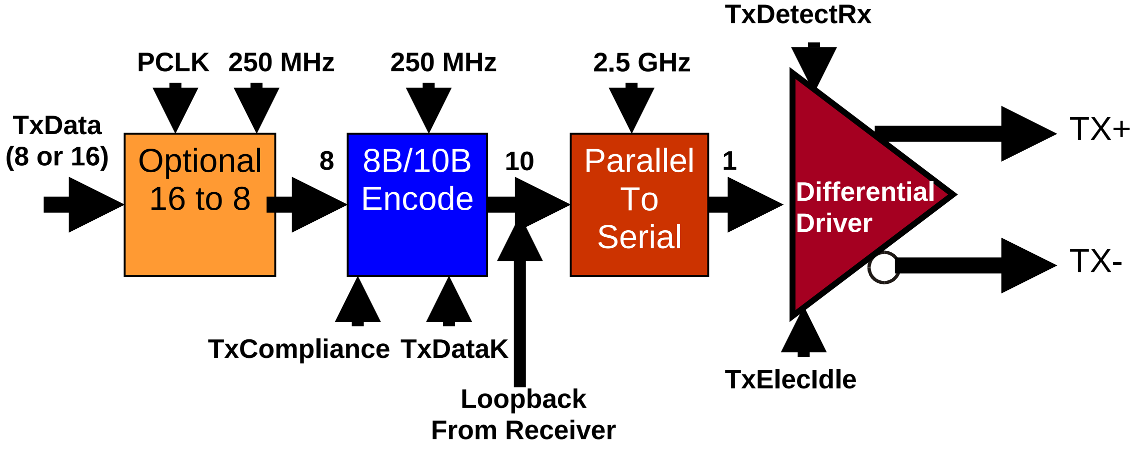

Figure 9. *Transmitter Sub-block*

Receiver Sub-block

The Receiver Sub-block has nine distinct sections. The received differential data is converted to a serial bit stream running at 2.5 GHz by the receiver, which can also detect the Electrical Idle state on the Link and report that to the MAC via the RxElecIdle status signal. Based on transitions in the incoming data stream, the Clock Recovery Circuit recovers the frequency of the transmitted clock using the 2.5 GHz clock input provided by the PLL. The recovered clock is used to latch data into the Data Recovery Circuit, from which it is fed into the Serial to Parallel converter.

The Serial to Parallel converter supplies a 10-bit symbol to both the Elastic Buffer and the K28.5 (Comma character) Detect circuitry. The Elastic Buffer is used for clock tolerance compensation and this is explained in much more detail in a Mindshare whitepaper[4]. The Comma Detect circuitry is used to establish symbol lock on the 10-bit symbols by detecting the special Comma control character that begins every PCI Express ordered set. This detect circuitry indicates to the MAC that received symbols are valid with the RxValid status signal. The output of the Elastic Buffer is fed into the 8B/10B Decoder and the Rx Status circuitry, and is also routed back to the transmitter for use when the PHY is placed into the External Loop-back mode.

The Receiver Status circuitry creates the RxStatus signals based on the characteristics of the received data. If the data has an 8B/10B decode error or experiences an Elastic Buffer overflow or underflow condition, it will report these conditions to the MAC with the RxStatus signals. The 8B/10B Decoder converts the 10-bit symbol back into an 8-bit character and determines the state of the RxDataK signal based on whether the decoded character is a data or control character. If the PHY/MAC interface is 16 bits wide, the optional 8-to-16-bit converter transfers two bytes at the 125 MHz PCLK frequency. If the interface is 8 bits wide, the optional converter is not needed and the received data byte is sent at 250 MHz.

[4] http://www.mindshare.com/classroom/pciexpress/elastic_buffer_implem.pdf

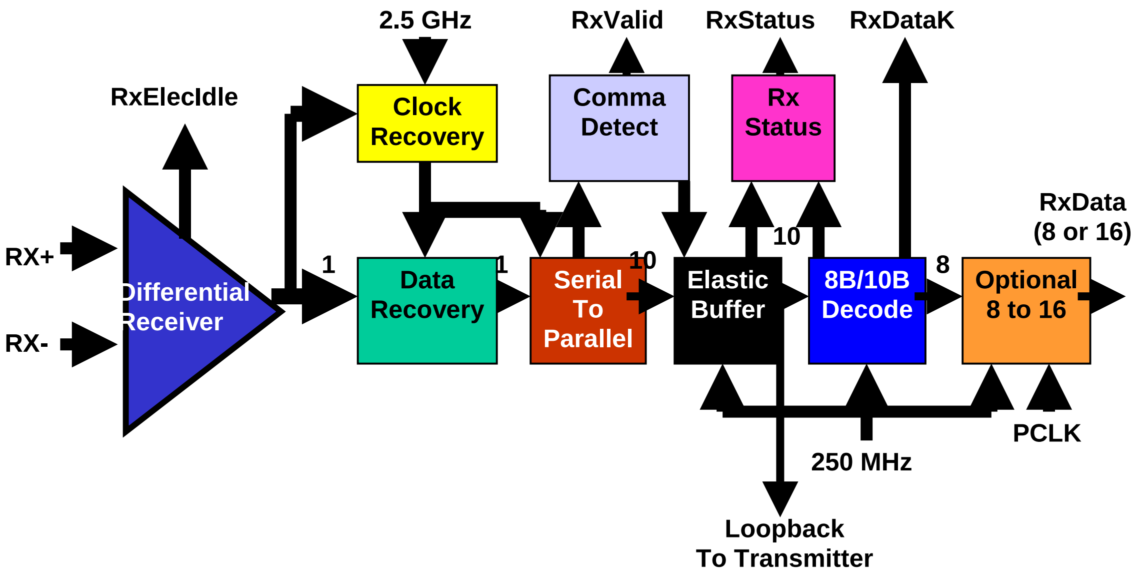

Figure 10. *Receiver Sub-block*

PIPE Signal Descriptions

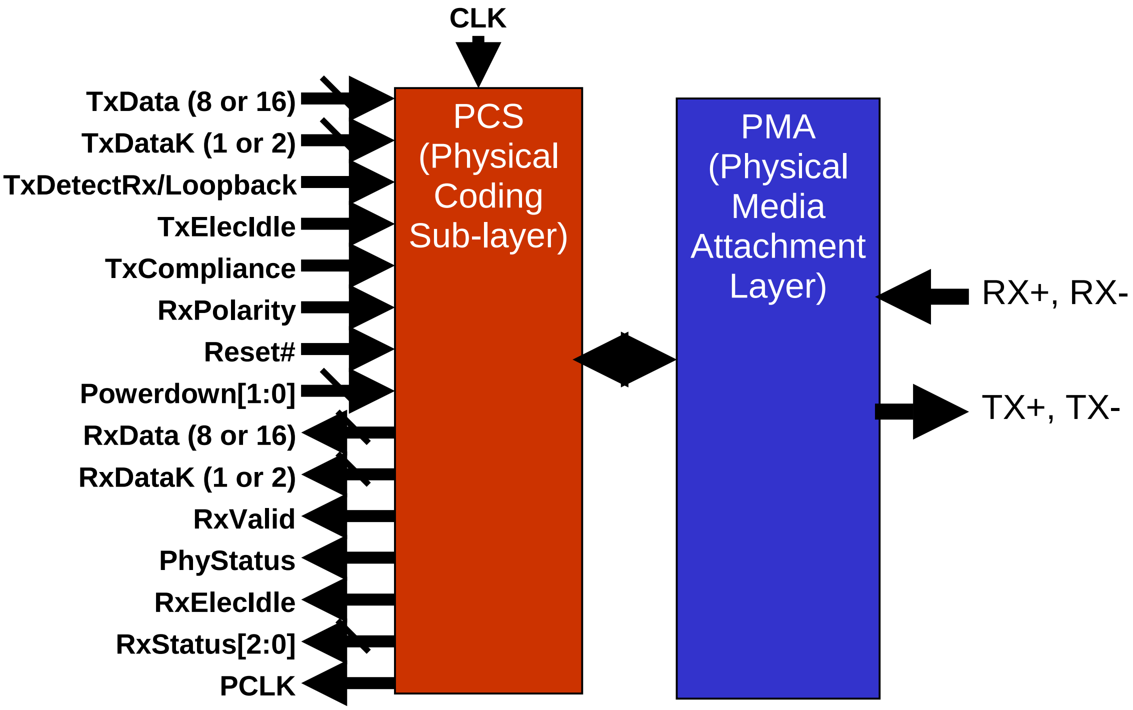

Figure 11. *PHY/MAC Interface Signals; Refer to Table 1*

Table 1. *PHY/MAC Interface Signals*

Signal Name	Description
CLK	This input to the PCS is used to generate the high speed (2.5 GHz) transmit and receive clocks and PCLK from the PLL. Specifications for this signal are implementation specific, and must be supplied by the PHY implementer. This signal may use spread spectrum modulation.
TxData (8 or 16)	These signals make up the parallel data bus from the MAC to the PCS. If implemented as a 16-bit interface then two character of data are transferred per clock. The first character is transmitted on bits [7:0] of the parallel bus, and the second character, if there is a 16-bit interface, is on bits [15:8] of the parallel bus.
TxDataK (1 or 2)	These signal(s) indicate the type of characters on the TxData bus. Bit 0 is for the lower byte on bits [7:0] of the parallel bus and Bit 1 is for the upper byte on bits [15:8] of the parallel bus. A zero indicates a data character; while a one indicates a control character.
TxDetectRx/ Loopback	This signal indicates to the PHY to begin the receiver detect process or to begin external loopback as described in the PCI Express Specification. Like many PIPE signals, the meaning of this signal depends on which power state the PHY is in. When in the P1 state, this signal informs the PHY to do a receiver detect, but when in the P0 state, this signal informs the device to go into the loopback mode as quickly as possible.
TxElecIdle	When this signal is asserted (active high), it tells the PHY to put the Tx outputs into the electrical idle state. This is true for all power states. When this signal is deasserted, valid data from the TxData and TxDataK signals should be transmitted in the P0 state. If the PHY is in the P2 state, a beacon should be transmitted, if the TxElecIdle signal is deasserted. In the P0s and P1 states, the TxElecIdle signal must be asserted. The use of this signal is also affected by the PHY power state, since there are some states in which the transmitter must be electrically idle. See section 6.14 of the PIPE spec for more detail.
TxCompliance	When asserted, this ordinarily forces he current running disparity (CRD) to negative. As the name implies, this is useful in conjunction with the transmission of the compliance pattern to generate test data. It is also used in a multi-lane implementation to turn off any lanes that are not going to be used, as in a x4 link that must operate as a x1. When both TxCompliance and TxElecIdle are asserted, that informs the affected lane to “turn off” and conserve as much power as possible.
RxPolarity	When asserted active high, this signal tells the PHY to do a polarity inversion on the received data.
Reset#	When asserted, this active low input to the PHY puts the transmitter and receiver in the default state
Powerdown[1:0]	These inputs put the transceiver into one of four power states: 1. P0 – normal operational mode 2. P0s – PCLK remains on, but Rx conserves power; entered when Rx detects electrical idle. Corresponds to link state L0s. 3. P1 – PCLK remains on, both Rx and Tx are in electrical idle. Corresponds to link state L1. 4. P2 – PCLK is off, PHY should minimize power consumption now, since it must operate within the limits of Vaux. Corresponds to link state L2. See section 6.3 of the PIPE spec for more details on PHY power management.
RxData (8 or 16)	These signals make up the parallel data bus from the PCS to the MAC. If implemented as a 16-bit interface, two symbols of data are transferred per clock. The first symbol received on the PCI Express data lines is on bits [7:0] of the parallel bus. The second symbol received on the PCI Express data lines is on bits [15:8] of the parallel bus.

RxDataK (1 or 2)	These signal(s) are the Data#/Control indicator(s) for the received symbols on the RxData bus. Bit 0 is for the lower byte on bits [7:0] of the parallel bus and Bit 1 is for the upper byte on bits [15:8] of the parallel bus. A zero indicates a byte of data; a one indicates a byte of control.
RxValid	This signal qualifies the data on the RxData and RxDataK signals. When this signal is asserted, the data on the receiver data bus is valid and the PHY has achieved symbol lock.
PhyStatus	Indicates completion of PHY power state transitions and receiver detection.
RxElecIdle	Indicates that the PCI Express receiver pins have detected an Electrical Idle state on the Link.
RxStatus[2:0]	Delivers receiver status and error codes for the received data and receiver detect status from the PHY to the MAC, noting such things as SKP symbol added or removed, disparity error, elastic buffer overflow or underflow, 8b/10b decode error, etc.
PCLK	This clock synchronizes the parallel interface between the PHY and the MAC. The frequency is 125 MHz for 16-bit data and 250 MHz for 8-bit data. Spread spectrum modulation is allowed on this clock.
TX+, TX-	These are the PCI Express differential outputs.
RX+, RX-	These are the PCI Express differential inputs.