IEEE SPONSORED 3rd INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEMS (ICECS 2016)

Efficient Implementation of High Speed PCI Express MAC Transmitter with PIPE Interface

Ravikumar Javali, Jim Terence Colaco
Atria Logic Pvt. Ltd.
Bangalore, India
Email: ravijavali@atrialogic.com
jim@atrialogic.com

***Abstract*— PCI Express (PCIe) has come a long way since its inception and today has become the foundation for inter component communication. PCIe is the protocol with the highest speed available in the industry today. Since dealing with high speed protocols is very complex, many aspects need to be taken care of, mainly the architecture implementation. This paper details an efficient method of implementation of Physical Interface for PCI Express (PIPE) Transmitter Interface. The implementation shows how the packets are formed in PHY layer for Link training and Initialization, Power Management and Packetization of data. The proposed design of the PCIe MAC (Media Access Control) runs at only a five clock cycle latency in order to form a packet, scramble it and place it on the bus. It consumes only 534 LUTs which is only 1% of the available slice LUTs on Zynq ZC706 Board.**

***Keywords*—PIPE, PCIe, PCI Express, MAC, PHY, High Speed, Serial Communication, SERDES, 8b10b encoding, PMA, PCS, HSSI, LTSSM, FPGA implementation.**

I. Introduction

Current trends in computing dictate the need for very high speed data transfer due to the increasing size of data needed to be processed per unit time by novel applications like multi-core graphics processors and solid state drives(SSDs). Such applications that require high data rates and bandwidths require a high speed protocol capable of handling the transfers. The PCI Express (PCIe) protocol is an ideal fit for this technological niche as an interconnect mechanism. Compared to the existing Serial AT Attachment (SATA) interconnects PCIe provides an improvement of up to 33.33% and a transfer rate of up to 8GB/s.

To tackle the aforementioned issues during design of high speed interconnects many good design practices need to be followed. For example pipelining, parallel execution, partitioning of logic, minimization of gate count and latency and modular design practice. Having said this, our proposed design incorporates such design practices providing an efficient means of implementation that follows the PCIe Specification [1]. The implementation comprises of a Media Access Control (MAC) Transmitter Engine which also follows the Intel Physical layer Interface for PCI Express (PIPE) Specification [3].

We have come up with an efficient architecture for PCIe MAC Transmitter with PIPE interface for PCIe version 2.1 which runs at an overall latency of five clock cycles, uses up 534 slice LUTs, and can send packets at the rate of 62.5MHz or 125MHz based on data interface width.

II. Survey

A. *What does Intel's PIPE Spec say?*

The PCI Special Interest Group (PCI-SIG) publishes and maintains the base specification for PCIe. The PCIe base specification revision 2.1 [1] provides an overview of the functionality the MAC-PHY implementation is required to provide while giving the designer free will in coming up with a design for the same. However the PIPE 3.1 specification [3] which is published by Intel specifies certain PIPE signals which are followed as a standard in the industry and have been incorporated in our design [2].

PIPE signals:

i. Tx_data : 8 bit or 16 bit or 32 bit width
ii. Tx_D/K : Indicates whether packet is data / control
iii. Command signals:
   - TxdetectRx : Begin receiver detection
   - Pipewidth : Indicates data width
   - Piperate : Indicates link speed
   - Txdatavalid : Indicates valid data is present
   - Pwrdown: Indicates various power management states
   - TxCompliance : For compliance pattern test
   - Resetphy : Physical layer reset signal
iv. Rx_data : 8 bit or 16 bit or 32 bit width
v. Rx_D/K : Indicates whether received packet is data/control character
vi. Status: 5 bit status signal from PHY to MAC
   - PHY obtained symbol lock
   - PHY completed various PM states
   - Received data is valid
   - PHY detected receiver
   - PHY detected Electrical Idle on link

978-1-4673-7832-1/16/$31.00©2016 IEEE

The Serializer/De-serializer (SERDES) is controlled through the signals that are sent to it by the Transmitter Interface, which include some Power Management signals to control the state of operation of the SERDES and the MAC.

B. *Related work*

Earlier work done on the PCIe MAC includes proprietary architectures for implementation of the MAC, since neither PCI-SIG nor Intel's PIPE specification prescribes a particular method for implementation. However when we look at PCIe base specification 2.1 [1] we can see the MAC and PHY layer interface exists and the MAC is responsible for generating all the necessary link training packets, power management (PM) signals and packetizing the data being sent. We have considered, for this implementation, the base specification provided by PCI-SIG alone and have not considered any other implementations in our survey.

In paper [4] the authors have designed the MAC-PHY interface based on PCI express for WLAN. The interface between MAC and PHY in the said paper reduces the work of the CPU by doing it on the Field Programmable Gate Array (FPGA) for a Wireless Local Area Network (WLAN) application.

In paper [5] the authors have designed a verification solution to ensure that the design under test follows the specification set by PCI-SIG in [1]. This paper does not concentrate on efficiency and gate count as it is a verification solution.

III. Proposed design architecture

Since neither the PCI-SIG nor Intel's PIPE specification [3] specifies the method of implementation, as they have left it up to the designer, we have come up with an efficient way of implementation which includes the PHY packet generation modules sending packets to an arbiter which decides the module for which the bus grant should be given. The data then flows through a scrambler which scrambles the data packets.

The scrambled data is then passed to a Physical Coding Sub-Layer (PCS) unit. The PCS has an 8b10b encoding and decoder unit which is followed by a Physical Media Attachment (PMA) which contains the SERDES and other analog circuits.

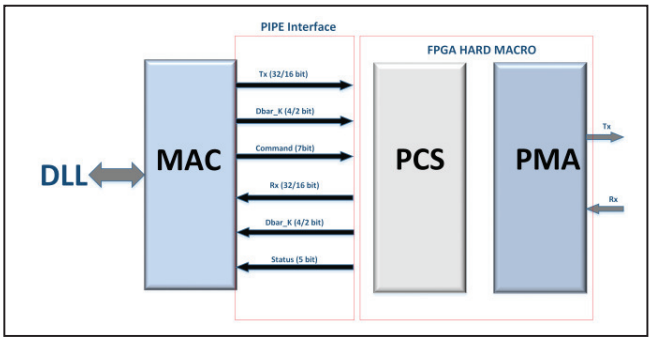

Fig 1: PCIe PHY layer proposed architecture

Fig [1]; depicts the PIPE signals specified by Intel. The implementation is designer specific. The efficiency depends on how well the designer architects it. We have come up with architecture for PIPE Interface, which is configurable where the end user can choose the interface width as required by his application.

The interface width for our design is configurable for 16 bit and 32 bit operation. The configurable data width provides the advantage of varying the parallel clock (pclk) for its operation. If the user selects a higher data width, a lower pclk frequency will be required. Since the design mainly targets an FPGA development board, minimal frequency of operation is ideal. If the frequency of operation is too high, the required timing closure might not be met. This might lead to the design not working according to requirements. So, it is preferable that the design operates at a lower frequency and a higher data width.

Interface Width	Pclk frequency	Link speed
8 bit	250 MHz	2.5 GT/s
16 bit	125 MHz	2.5 GT/s
32 bit	62.5 MHz	2.5 GT/s

Table 1: Pclk frequency variation with data width interface variation at a link speed of 2.5 GT/s (Gen 1)

Interface Width	Pclk frequency	Link speed
8 bit	500 MHz	5 GT/s
16 bit	250 MHz	5 GT/s
32 bit	125 MHz	5 GT/s

Table 2: Pclk frequency variation with data width interface variation at a link speed of 5 GT/s (Gen 2)

Table [1] and Table [2] show the variation of pclk frequency with change in data width interface at a link speed of 2.5 GT/s and 5 GT/s. Selection of a higher data width interface substantially reduces the operating clock frequency.

IV. Implementation

The proposed architecture is implemented in two sections. Namely

- MAC Layer Implementation
- Interface between MAC and PCS i.e. PIPE

The MAC Layer implementation is depicted in fig [2]. The MAC Layer is responsible for following operations

- Formation of Training sequence 1 (TS1) & Training Sequence 2 (TS2) packets
- Formation of skip ordered set (SOS).
- Formation of electrical idle ordered sets (EIOS).
- Formation of Logical idles.

- Formation of fast training sequence ordered sets (FTSOS).
- Handling power management.

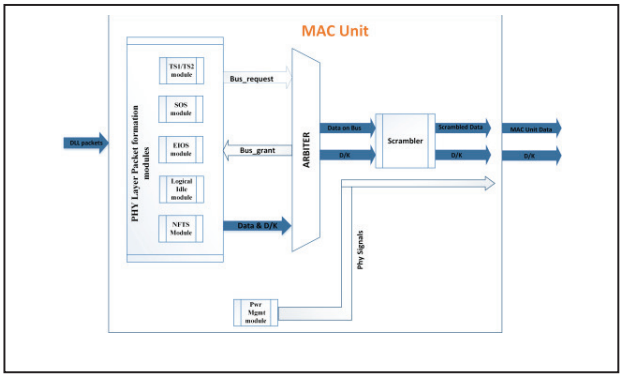

Fig 2: MAC layer Implementation

TS1/TS2 formation module: This module generates training sequences that are required for training the link before setting up the link for sending the actual data (i.e TLP or DLLP). TS1/TS2 packets are 16 bytes wide. The MAC module is capable of generating TS1/TS2 based on LTSSM states. After receiving and sending a certain number of training sequences in each state the LTSSM progresses to the next state [1].

0	COM	K 28.5
1	Link #	0 -255 = D0.0 – D31.7, PAD = K23.7
2	Lane #	0 -31 = D0.0 – D17.1, PAD = K23.7
3	# FTS	No. of FTS required for L0s recovery
4	Rate ID	Indicates speed
5	Train Cntrl	
6 – 9	TS ID or EQU Info	EQU Info when changing speed
10 – 16	TS ID	TS1 Identifier = D10.2 TS2 Identifier = D5.2

Fig 3: Training sequences TS1/TS2 formats

As shown in Fig [3] the TS1/TS2 packets will be formed with negotiated lane and link number, rate ID (speed) and also the TS1 or TS2 ID based on particular state of LTSSM. Once the link is trained the LTSSM moves to L0 active state.

SOS formulation module: SOS is a packet containing the COM symbol followed by three SKP symbols. SOSs are sent anywhere in the interval of 1180 – 1538 symbols [1]. SOS prevents the overflow or underflow of the Elastic buffer present in the receiver PCS of each device.

EIOS formation Module: Electrical idle ordered set is a packet consisting of the COM symbol followed by three IDLE symbols. EIOSs are generated and sent by the Transmitter engine before it wants to move to low power states like L0s, L1 or L2. When the Transmitter engine sends EIOS, the configuration registers and contents of variables are stored to be loaded back when the device is restored back to active state L0.

FTSOS formation module: Fast training sequence is a packet consisting of a COM symbol followed by three FTS symbols. A certain number of FTS ordered sets are sent by the Transmitter engine of the device which is in low power state if it wants to come back to active state L0. The number of FTSOSs needed to be sent is decided by the N_FTS field present in TS1/TS2 [1].

Logical Idle module: Logical idle symbols are just zeros sent as data. After LTSSM training is done before moving to active state the device needs to send and receive a certain number of logical idles.

Power Management module: Power Management module generates the power down signal based on the phystatus signal given by the SERDES.

Arbiter Module: The main block of the MAC is the arbiter. The arbiter module is capable of handling the requests generated by the different packet formation modules. Depending on the LTSSM state change one or the other packet formation modules will make a bus request to arbiter and arbiter grants the bus to one of those modules which has made the request. The decision to grant the bus to packet formation module is done by the internal artificial intelligence logic incorporated into the arbiter module.

In order to achieve the low latency we have designed our MAC unit in such a way that it follows pipelining all the time. The bus request from packet formation module, the bus arbitration, bus grant and sending of packets is done in two initial clock cycles. Followed by which we get data for every clock cycle.

If the bus is engaged by one module and if another module makes the request for the bus to the arbiter, it will be held until the bus becomes free. A “BUSY” signal will be generated by the module which is engaged in sending packets on the bus, as soon as the packets are sent the module will de-assert the BUSY signal so immediately in the next clock cycle the bus will be engaged by the other module without stalling. There is no glitch or stalled cycle in the designed MAC architecture.

Scrambler module: The Scrambler is intended to prevent repetitive patterns in data streams [1]. These repetitive patterns create a pure tone i.e. a noise of high intensity around a particular frequency which is much more than the noise caused by EMI. For this reason PCI-SIG has included a scrambler in the specification which is mandatory.

The Scrambler scrambles the data and logical idles, it bypasses all the training sequences, ordered sets and compliance patterns. The scrambler can be disabled for compliance patterns externally by an implementation specific signal.

V. Results

The proposed architecture for PCIe Tx PIPE interface is targeted for Xilinx Zynq ZC706 development board and the

results (both simulation and synthesis) for the same are included in the following section.

A. Simulation Results:

The designed PCIe PIPE interface at transmitter side is simulated on Xilinx ISE design suite version 14.6 by choosing the ZC706 as the target board. The simulation results were captured using Questasim and are shown in the following section.

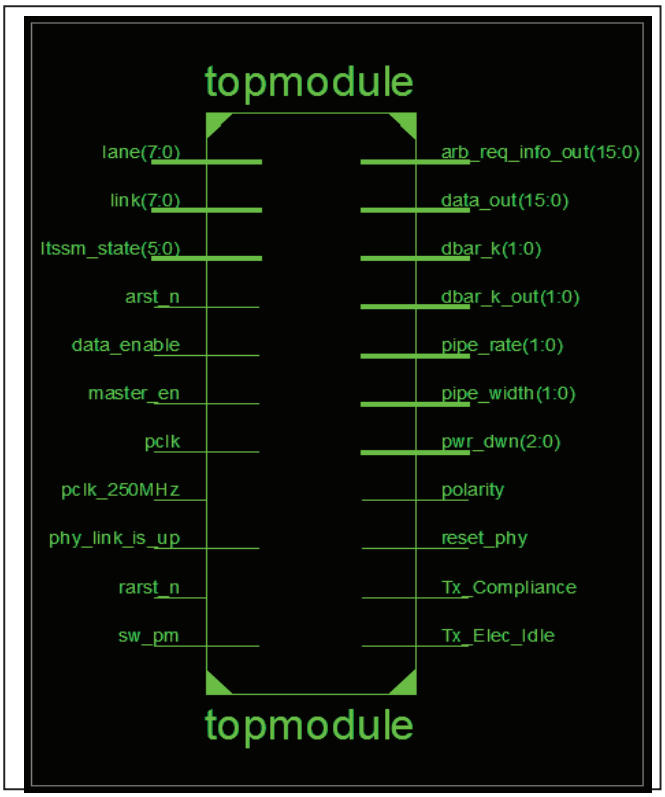

Fig 4: PIPE Interface implementation

The top module shown in Fig [4] is top module of entire MAC unit. This top module gives out a few PIPE interface signals as well as other scrambler data information. Top module gets its input from the Data Link Layer (DLL) and LTSSM.

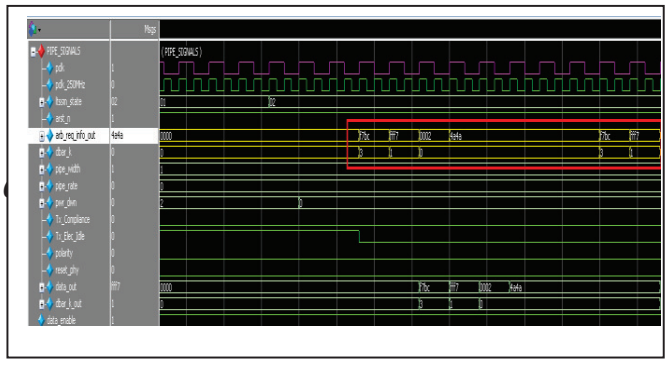

Fig 5: Training Sequence (TS1) being sent from Transmitter

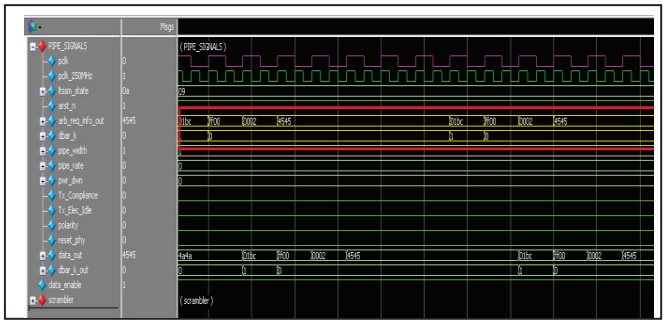

Fig 6: Training Sequence (TS2) being sent from Transmitter

Training sequences (TS1 & TS2) that are sent during the link training are shown in Fig [5] & Fig [6]. TS1/TS2 packet is 16 bit wide and it starts with COM character (BC).

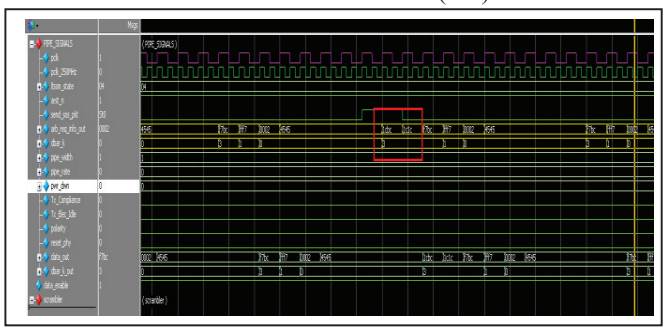

Fig 7: SOS sent for every 1538 symbols

The skip ordered sets are sent for once in every 1180 – 1538 symbols. SOS is represented as {BC1C1C1C} in hexadecimal notation.

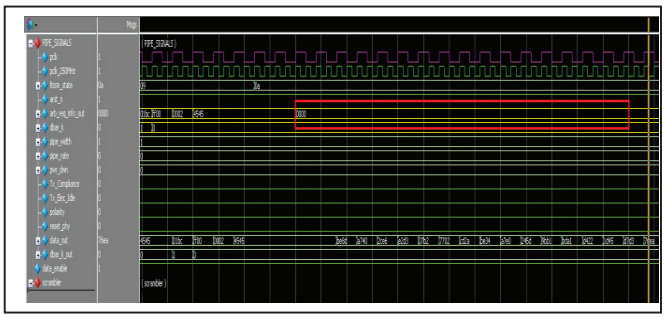

Fig 8: Logical Idles sent after training sequences

Logical idle is just a data with all zero symbols. Logical idles are sent after the link training before LTSSM moves to active state L0.

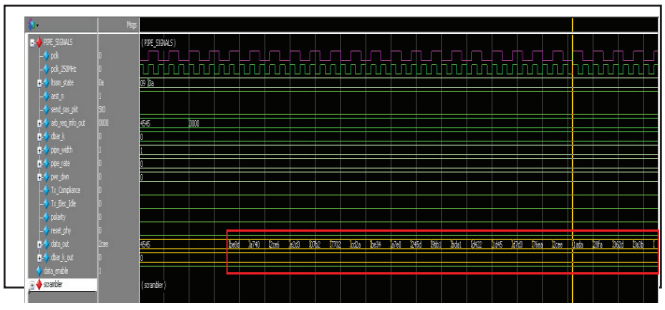

Fig 9: Scrambled logical idles

The scrambled logical idles is shown in Fig [9]. The scrambler scrambles all the data and logical idles.

B. Synthesis Results:

The RTL code for the MAC unit is synthesized for Xilinx Zynq ZC706 board with package XC7Z045. The synthesis results are included in the following table.

Sl. No.	Parameter	Unit
1.	Latency	5 clock cycles
2.	Number of slice LUT	543
3.	No. of slice registers	384
4.	No. of bonded IOB	75

Table 3: Synthesis results

The module takes only 5 clock cycles to form the packet, send it to the arbiter, scramble it and give out the scrambled data. The design takes only 543 slice LUTs and 75 bonded IOBs on the FPGA.

Pipelining of stages is done to achieve higher throughput, as a result of which, after an initial latency of 5 clock cycles we get the data for every clock cycle. There is no stalling or in other words there is no dead cycle while sending the packets.

VI. Conclusion

The PCIe protocol is the fastest bus interface for communication available in the industry. The complexity involved in the physical layer design for power management and LTSSM is solved with the modular design approach, where a complex architecture is broken into modular units and all the units are integrated into one top module. The simplified design architecture makes it possible to achieve lower latency and lower gate count. The proposed design is able to send the packets at the rate of 62.5 MHz and 125 MHz with the data interface width of 32 bit for version 1 (2.5GT/s) and version 2 (5GT/s) respectively before the SERDES.

The proposed design takes only 543 slice LUTs and 75 bonded IOBs. This is very minimal when compared to available LUTs and IOBs on the FPGA board. Hence this is an efficient design of high speed PCIe PIPE Tx interface.

VII. Future Scope

The PHY layer of PCI Express is incomplete without the LTSSM and the MAC Receiver Engine. The designed Transmitter module can be connected to the LTSSM, which in turn is connected to the Receiver Engine, and it can take the LTSSM state as input and train the link accordingly. The MAC Unit then can be integrated to the SERDES package which consists of the PCS and PMA which gives out serial data at the rate of 2.5 GT/s or 5 GT/s.

Acknowledgment

The authors would like to thank Atria Logic Pvt. Ltd. for their continuous support and help in carrying out this project.

References

[1] *PCI Express Base Specification 2.1*, PCI-SIG, November 10, 2010.

[2] Mike Jackson and Ravi Budruk, *PCI Express Technology*, e-book, ver. 1.02, Mindshare Inc.

[3] *PHY Interface for PCIE, SATA and USB 3.0 Architectures*, Intel, ver. 4.0, 2011.

[4] Wang Guixin, Wang Hao and Kang Guixia, *Design and FPGA Implementation of MAC-PHY Interface Based on PCI Express for Next-Generation WLANs,* International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2012, ISSN 2161-9646.

[5] Sagar Kumar K S, Venkategowda N, *Sampling and Reconstruction of Ordered Sets in PCIe 3.0*, International Journal of Innovative Research in Science, Engineering and Technology, vol. 4, Issue June 2015, ISSN 2319-8753.

[6] Chandana K N , Karunavathi R K, *Link Initialization and Training in MAC Layer of PCIe 3.0*, International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, pg 2717-2719, ISSN 0975-9646