

Configurable Design and Simulation of PCI-Express 3.0 Data Link Layer Transmitter

Tejas A. Bavarva¹

¹M.E

¹VLSI & Embedded System Design GTU PG School

¹Gujarat Technological University, Ahmedabad, Gujarat, India

Abstract--- PCI-Express is a high performance, general purpose I/O interconnect communication protocol. This paper presents the detailed implementation of configurable Data Link Layer Transmitter of PCI-Express 3.0. The proposed architecture presents the transmitter module which contains the sophisticated features of PCI-Express 3.0. It explains how TLPs are processed in data link layer transmitter by appending the next sequence number as start frame and 32-bit LCRC as end frame. Protocols of PCI-Express 3.0 like link acknowledgement time out replay mechanism, flow initialization protocol and error correction protocol have been implemented and verified. The architecture of retry buffer is also presented with all the experimental results and it also explains how the retransmission is happening in the transmitter module. At the end of the design, Linting and Synthesis have been done and analyzed timing, power and utility report.

Keywords: PCI, TLP, DLLP, Arbiter, LCRC, ECRC

I. INTRODUCTION

PCI-Express 3.0 is a high performance, general purpose I/O interconnects defined for a wide variety of future computing and communication platforms [1]. PCI-Express 3.0 takes advantage of recent advances in point-to-point interconnects, switch based technology and packetized protocol to deliver new level of performance and features [1]. Power management, Quality of Services (QoS), Hot-plug/Hot-swap support, Data integrity and error handling are among some of the advanced features supported by PCI-Express 3.0 [1].

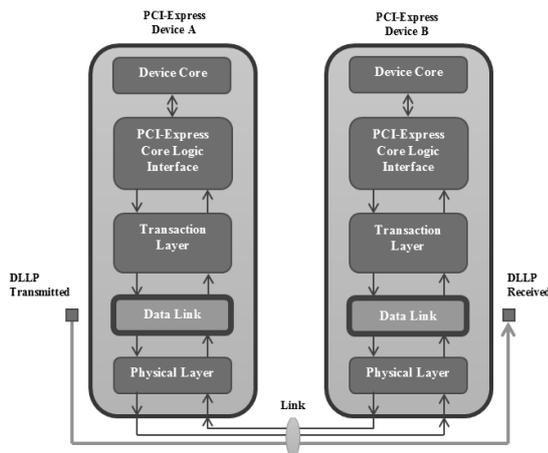


Fig. 1: Layering diagram of PCI-Express data link layer [1]

The fundamental goal of this layering definition is to facilitate the reader's understanding of the specification [1]. Note that this layering does not imply a particular PCI-

Express implementation [1]. PCI-Express 3.0 uses packets to communicate information between the components [1]. Packets are formed in transaction and data link layers to carry the information from the transmitting component to the receiving component [1]. As the transmitted packets flow through the other layers, they are extended with additional information necessary to handle packets at those layers [1]. At the receiving side, the reverse process occurs and packets get transformed from their physical layer representation to the data link layer representation and finally (for Transaction Layer Packets) to the form that can be processed by the transaction layer of the receiving device [1]. Fig.1 shows the conceptual flow of transaction level information through the layers.

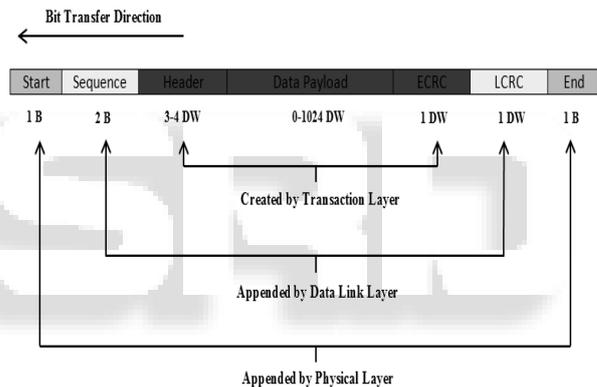


Fig. 2: Packet formations through the layers

The data link layer is an intermediate stage between transaction layer and physical layer. Link acknowledgement time out replay mechanism, flow initialization protocol, error correction protocol and power management are some of the basic responsibilities of the data link layer. As shown in Fig.2, packet is assembled in transaction layer by appending header (3-4 DW) as start frame and ECRC (1 DW) as end frame. Again when the packet assembled in transaction layer passes through the data link layer, packet sequence (2 B) will be appended as start frame and LCRC (1 DW) will be appended as end frame. Again physical layer adds start frame and end frame of 1 Byte to data link layer packet. In this way, packet formations through the layer will be completed and finally it will pass through the link which is established between transmitter and receiver. Data link layer also produce and consume the packets that are used for link acknowledgement and time out replay mechanism [1]. To differentiate these packets from those used by transaction layer, the term Data Link Layer Packet (DLLP) will be used when referring to packets that are produced and consumed at the data link layer [1].

II. PROPOSED ARCHITECTURE AND DESIGN

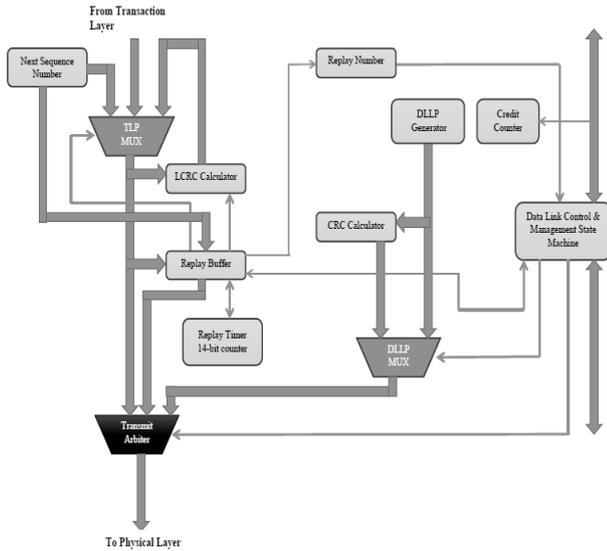


Fig. 3: Proposed architecture

The transmitter receives TLPs from Transaction Layer and it sends them to the Physical Layer with a sequence number that identifies each TLP and CRC. The transmitter is responsible for the retransmission of TLPs if necessary and it also sends the DLLPs that are generated in the data link layer. Now we will discuss all the individual blocks in detail as follows:

A. Data Link Control & Management State Machine

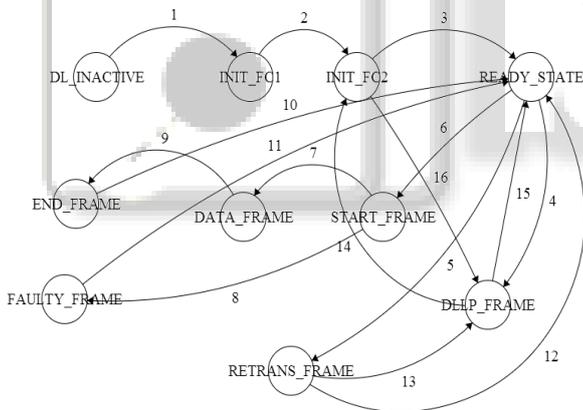


Fig. 4: Control FSM

CONDITION NO.	CONDITION
1	phy_up = 1
2	fi1 = 1
3	tx_req_vc0 & (~fifo_full1) & gmt0
4	gmt2 = 1
5	gmt1 = 1
6	tx_start_vc0 & gmt0
7	tx_data_coming & gmt0
8	(&tx_retrans_no) & gmt0
9	tx_end_vc0 & gmt0
10	end_of_frame & gmt0
11	tx_faulty_frame = 1
12	tx_req_vc0 & (~fifo_full1) & gmt0
13	gmt2 = 1
14	gmt0 = 1
15	tx_req_vc0 & (~fifo_full1) & gmt0
16	gmt2 = 1

Table. 1: Conditions for FSM

Data Link Control & Management State Machine is the heart of PCI-Express Data Link Layer. The data link layer tracks the state of the link. It communicates link status with the transaction layer and physical layer. One variable named as phy_up is taken for the physical layer whether it is ready to transmit the packet towards the receiver or not. All other transitions are made using various conditions as shown in Table.1. As the flags like start frame, data frame and end frame will be enabled, particular transitions will be done and we can see the status of data link layer.

B. Next Sequence Number

This module generates the 12-bit sequence number that will be added at the beginning of the next TLP to transmit. As the flag for data coming will be enabled, next sequence number will be incremented by ne. This counter sends the sequence number before a TLP is dispatched by the transaction layer.

C. LCRC (Link CRC) Calculator

This module calculates 32-bit CRC for particular TLP received from transaction layer and this value is added to the tail of TLP as end frame. This module calculates the various lengths of data coming from transaction layer and gives the output in one clock cycle. This module has a parallelization of a diagram that originally shows a serial realization [2]. This module is kept only to verify whether the data sent to receiver is same or not.

D. TLP MUX

This module permits the flow of next sequence number, data from transaction layer and LCRC on the basis of selection line. First, next sequence number will pass through the bus, then data coming from transaction layer will pass through it and finally LCRC will pass on the bus and complete the whole frame structure respectively. TLP MUX output is of 32-bit. Data received from transaction layer has variable size of 128 byte to 4096 bytes (15-bit).

E. Retry Buffer

This is the most important and sophisticated module in PCI-Express 3.0 data link layer. It stores the TLPs coming from transaction layer and also sent to the physical layer when it will receive the negative acknowledgement from the receiver side. As the start flag will be enabled, next sequence number will pass through TLP MUX and stores into the buffer at location 12'h0000. Sequentially, when data and LCRC will come to the retry buffer, it will be appended to the particular sequence number and write pointer will be incremented by one. Memory manager which is a part of the retry buffer will detect the retry notification which is of 13-bit (1-12 bit = sequence number and 13th bit shows the validation or invalidation of TLP received). If 13th bit is 0, then it is the valid frame and if 13th bit is 1, then it is invalid frame. If it receive positive acknowledgement, then buffer will delete the TLP of particular location. If it will receive negative acknowledgement, then read pointer will read the location of that TLP. Retry buffer parameter are ADDR_WIDTH = 12, DEPTH = 4096, DATA_WIDTH = 59.

F. Replay Timer

This timer starts when the start flag will be enabled. It counts the amount of time for particular TLP transmission or retransmission. Time will be predefined for particular length of data. If retry notification will not be received in that predefined time, then timer will be expired and it generates the time out flag. If it receives the retry notification within that predefined time, then timer will be stopped at that moment. Replay timer counts are given below in Table.2 according to its payload size and link width.

Max_Payload_Size (bytes)	Link Operating Width						
	x1	x2	x4	x8	x12	x16	x32
128	999	672	507	489	462	432	387
256	1536	939	642	609	558	504	423
512	1965	1155	750	546	615	546	444
1024	3501	1923	1134	738	870	738	540
2048	6573	3459	1902	1122	1383	1122	732
4096	12717	6531	3438	1890	2406	1890	1116

Table. 2: Replay Timer limit according to Payload size and Link width [1]

G. Replay Number

This module counts how many number of retransmission occurred in the transmitter design. Here maximum limit is 4 (2-bit). Retransmission occurs only when if replay timer expires or if negative notification will be received.

H. CRC Calculator

This module calculates the CRC for DLLP. Again do not confuse between TLP and DLLP. DLLP is sent to check whether link is operational or not. DLLP packets are sent every 34 microseconds according to the PCI-Express 3.0 manual. This module is same as LCRC calculator. CRC output is of 32-bit.

I. DLLP MUX

This module permits the flow of DLLP and CRC of DLLP packets on the basis of selection line which is manually operated. This flow should occur at every 34 microseconds. Transmit arbiter is responsible to that particular flow of DLLP packets.

J. Credit Counter

This module counts the number of credits (12-bit) available for header and data in retry buffer. When it detects tx_start_vc0, credits for header and data will be decremented by one. When rx_valid flag is detected, credits for header and data will be incremented by one.

K. Transmit Arbiter

This is a very sophisticated and complex module. This module gives the final output of the data link layer transmitter. There are three possible grants for three different requests. There are some priority values for all requests. Depends on those priority values, particular grant will be given to the particular module. First request line is from TLP MUX, second request line is from retry buffer and third request line is from DLLP MUX. This arbiter is such a smart module that selects the particular module depends on

time and priority values of their requests. The grant for DLLP MUX must be given from the transmit arbiter every 34 microseconds to retraining the link.

III. SIMULATION RESULTS

Design of PCI-Express 3.0 Data Link Layer has been prepared in Verilog language and its unit testing (simulation) has been done in *Modelsim PE Student edition* software. After designing the PCI-Express 3.0 Data Link Layer Transmitter, Linting has been done over it using *HDL companion* tool to check the modularity of the design code. After Linting, Synthesis has been completed using *Xilinx Vivado* software and verified timing, power and utility report.

Simulation results of some of the important modules are given below.

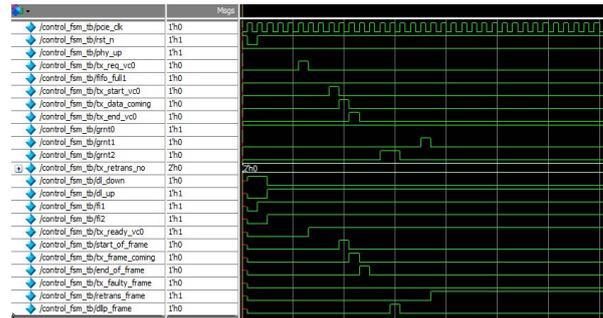


Fig. 5: Waveform of Control FSM

Above waveform of Control FSM shows the various states transitions depends on various flags like phy_up, tx_req_vc0, grnt0, grnt1, grnt2, etc. If grnt0 = 1, normal transmission should occur from TLP MUX. If grnt1 = 1, retransmission should occur from retry buffer. If grnt2 = 1, transmission of DLLP packets should occur.

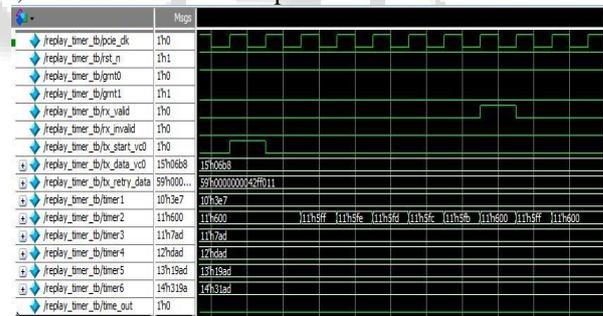


Fig. 6: Waveform of Replay Timer when timer doesn't expire

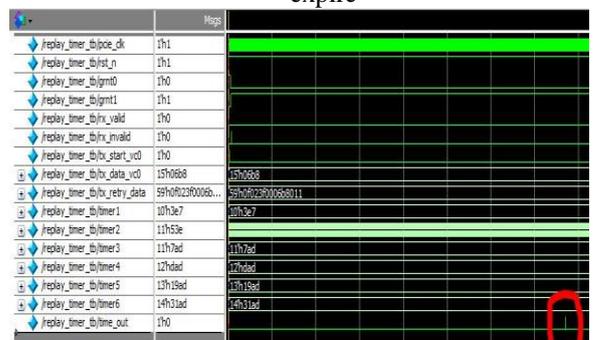


Fig. 7: Waveform of Replay Timer when timer expires

Replay Timer starts when tx_start_vc0 = 1, If retry notification gives positive acknowledgement, timer will be reset. If retry notification gives negative acknowledgement, timer will continue to count and when it reaches its

maximum value, again timer will be reset and tx_time_out flag will be enabled.

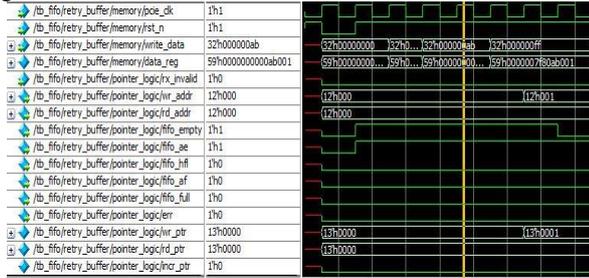


Fig. 8: Waveform of Retry Buffer when writing the data

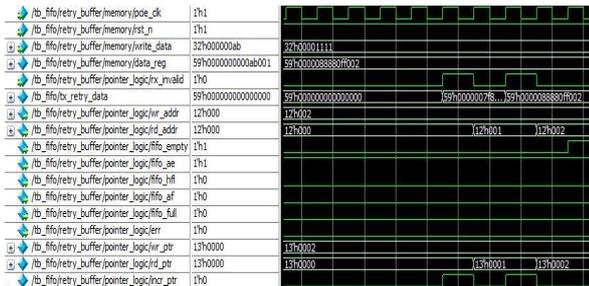


Fig. 9: Waveform of Retry Buffer when reading the data

Retry Buffer stores the packet formed in data link layer and depending on the retry notification received from receiver, retransmission occur. For each packet, retry buffer indicates the flags like fifo_full, fifo_empty, fifo_oe, fifo_af, fifo_hf and err. For each different length of packet, timer is different. When timer expires and tx_time_out signal is enabled, retransmission occurs.

In Transmit Arbiter, one condition is there that when request from DLLP MUX is received, other transmission must be suspended till the transmission of the DLLP packets. It is necessary to send the DLLP packets periodically to retrain the link between Tx and Rx. Transmit Arbiter checks the condition between 33 microseconds to 34 microseconds, whether the counter output is reset to 0 or not. After that, arbiter gives the high priority to either TLP MUX or Retry Buffer.

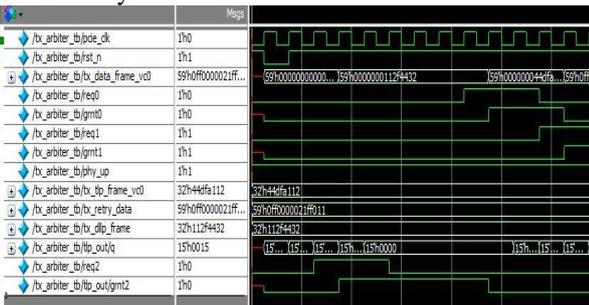


Fig. 10: Waveform of Transmit Arbiter

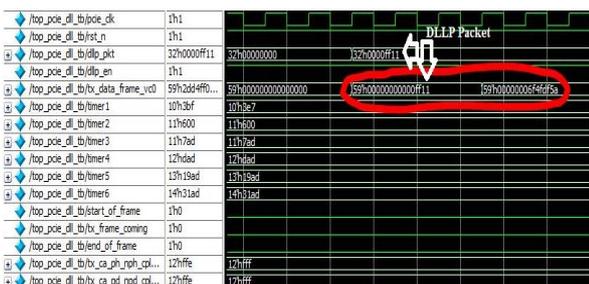


Fig. 11 Waveform of Top module, DLLP packet transmission

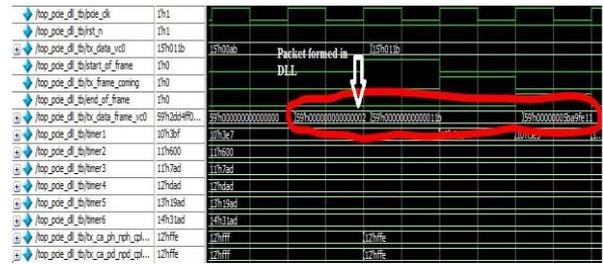


Fig. 12: Waveform of Top module, normal transmission

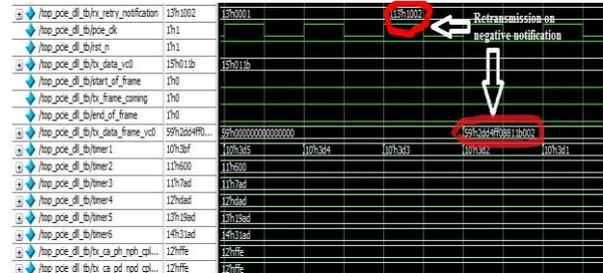


Fig. 13: Waveform of Top module, retransmission when negative notification received

There are three test cases are taken in snapshots of top module when DLLP packets transmit, normal transmission occurs and retransmission occurs when negative notification received. So in this way, we can take so many test cases to verify the design of PCI-Express 3.0 Data Link Layer Transmitter.

IV. SYNTHESIS RESULTS

Logic synthesis is a process of converting a high level description of design into an optimized gate level description of the design into an optimized gate level presentation, given a standard cell library and certain design constraints. In synthesis, timing, power and utility analysis have been done and their reports are shown below.

A. Timing Report

In timing analysis, constraints like create_clock, set_input_delay and set_output_delay have been added and completed the analysis.

Design Timing Summary		
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.598 ns	Worst Hold Slack (WHS): 0.262 ns	Worst Pulse Width Slack (MPPWS): 5.471 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 719	Total Number of Endpoints: 719	Total Number of Endpoints: 204

All user specified timing constraints are met.

Fig. 14: Design Timing Summary

Timing analysis is tested on the basis of slack and slack must be positive so that setup and hold violation don't occur in design.

B. Power Report

Here static power is more than dynamic power and total power consumption is 0.154 W. There should be less power as much as possible with junction temperature.

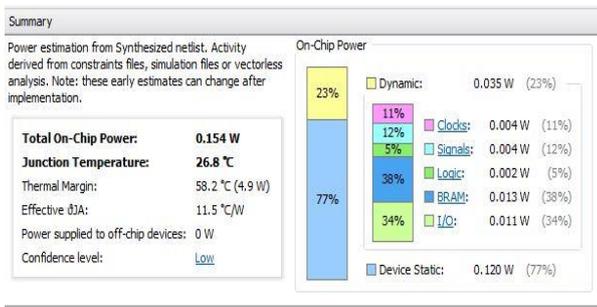


Fig. 15: Power Summary

C. Utilization Report

Utilization report gives the information about which resources have used how much area in the design.

Resource	Utilization	Available	Utilization %
Slice LUTs	623	53200	1
Slice Registers	379	106400	1
Memory	7	140	5
IO	229	127	180
Clocking	4	32	13

Fig. 16: Utilization Summary

From utilization report, conclusion can be drawn that utilization % must be less than 100 % that is the range. Beyond that range, either hardware must be changed or utilization % must be within 100 %.

V. CONCLUSION

From this paper, the various capabilities and protocols of PCI-Express 3.0 Data Link Layer Transmitter have been demonstrated. This presented design of PCI-Express 3.0 Data Link Layer Transmitter has backward compatibility with existing PCI-Express versions. By changing the parameters (constants) in the current design, it can be used in upcoming version i.e. PCI-Express 4.0 to improve the data transfer rate.

From the simulated results, it can be stated that the response of replay timer and transmit arbiter is very fast compared to earlier version of PCI-Express. All the modules have been optimized by the optimization technique.

ACKNOWLEDGEMENT

The results in this paper are from the work which was conducted as a Master Thesis by me in the field of VLSI & EMBEDDED system design, in cooperation with Centre for Development Advanced Computing (C-DAC), Pune. I take the opportunity to heartily thank our project guide respected Mr. Santosh S. Jagtap for his valuable guidance and touch of inspiration and motivation throughout the project. I am very grateful for those who supported and helped me during conducting this work.

REFERENCES

- [1] PCI - SIG, "PCI Express® Base Specification Revision 3.0", Retrieved 10th November, 2010
- [2] M. Aguilar, A. Veloz, M. Guzman, "Proposal Of Implementation Of Data Link Layer Of PCI-Express", First International Conference On Electrical And Electronics Engineering, 2004

- [3] PCI Express 1.1 Root Complex Lite X1, X4 IP Core User's Guide, Lattice Semiconductor, February 2012
- [4] Peter Bohm, "Incremental And Verified Modeling Of The PCI Express Protocol", IEEE Transaction On Computer-Aided Design Of Integrated Circuits And Systems, Vol. 29, NO. 10, October 2010
- [5] Eugin Hyun And Kwang-Su Seong, The Design Of PCI Express For Future Communication Platform, VLSI Lab., Dept. Of Electronics Engineering, 2004
- [6] "PCI Express – An Overview Of The PCI Express Standard", August 13, 2009, Retrieved From [Online] [Www.Ni.Com/White-Paper/3767/En/](http://www.ni.com/White-Paper/3767/En/)
- [7] Xilinx User Guide "Logicore™ PCI Express® Endpoint Block Plus V1.2", UG341 February 15, 2007
- [8] Xilinx User Guide "Logicore IP Endpoint Block Plus V1.15 For PCI Express", UG341 July 22, 2011
- [9] Don Anderson And Tom Shanley, "PCI System Architecture", Mindshare INC., 1999
- [10] Sandra Mohan And Anish Joseph, A Dynamic Priority Based Arbitration Algorithm, IJITEE, ISSN: 2278-3075, Volume-3, Issue-1, June 2013