Configurable Design and Simulation of Synchronous Retry Buffer for PCI-Express 3.0 Data Link Layer

Tejas A. Bavarva[1]
[1]M.E. student
[1]VLSI & Embedded System Design Department
[1]Gujarat 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, exclusive and synchronous retry buffer used in PCI-Express data link layer transmitter with control logic that manages write and read operations, generates status flags and provides optional handshake signals for interfacing with user logic. It is often used to control the flow of packets between transmitter and receiver. It also represents various flags like fifo_full, fifo_empty, fifo_ae, fifo_af, fifo_hf and err. In this way, data integrity between transmitter and receiver is maintained. RTL coding of retry buffer has been written in Verilog language and it is simulated and verified in Modelsim PE student edition tool. At the end of the design, Linting and Synthesis have been done with the help of Xilinx Vivado tool and analyzed timing, power and utility report.

Keywords: Elastic buffer, FIFO, Synthesis, PCI-Express

I. INTRODUCTION

A synchronous retry buffer or FIFO refers to a FIFO design where packets are written sequentially into a memory array using a clock signal and packets are sequentially read out from the memory array using the same clock signal. The numbers of rows of the array is called the DEPTH of the FIFO. The bit length of each row (the number of columns of the array) is called the WIDTH of the FIFO.

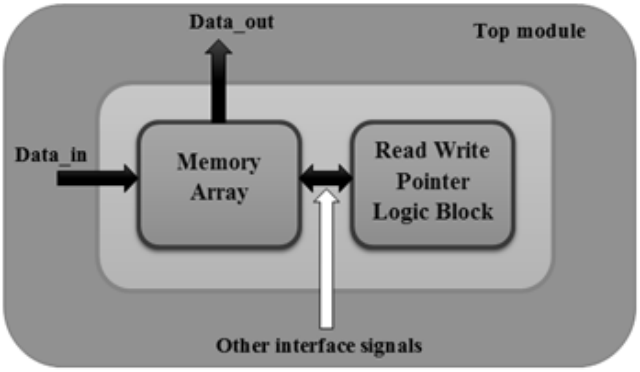

Fig. 1: Proposed architecture of Retry Buffer

A synchronous FIFO has a single clock port for both data-write and data-read operations. Data packets are applied at the module's input port "Data_in" is written into the next available empty location on a rising clock edge when the write request signal is asserted. The explanation of various flags is as follows:

1) fifo_full: When all buffer locations are full and if there is any write request signal is received, then this flag will be asserted.
2) fifo_empty: When all buffer locations are empty and if there is any read request signal is received, then this flag will be asserted.
3) fifo_af: When ¾ of the buffer locations are full, this flag will be asserted. This flag will be asserted till fifo_full = 1.
4) fifo_ae: This flag will be asserted till ¼ of the buffer locations will full and it will be disabled after those many locations will be written into memory array.
5) fifo_hf: This flag will be enabled when half of the memory locations in retry buffer will be written.
6) error: This flag will be enabled if there is no space in buffer and write request signal will be received or if the buffer is empty and read request signal will be received.

A synchronous buffer is needed when transmitter and receiver operates at different speed. Suppose transmitter is supplying data at rate which receiver cannot handle or vice versa. So to bridge the gap between transmitter and receiver or we can say to match the throughputs between transmitter and receiver, Synchronous Buffer is act as an Elastic buffer.

II. DETAILED IMPLEMENTATION OF DESIGN

This section covers the whole implementation of retry buffer of PCI-Express 3.0. Now we will see the detailed pin diagram of top module, memory array and read write pointer logic which are as follows:

A. Top module

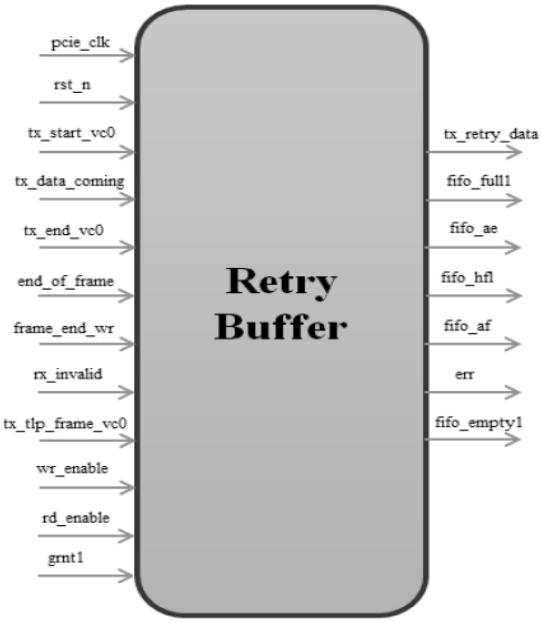

Fig. 2: Pin diagram of top module

Pin diagram of top module of retry buffer is shown in Fig.2. It shows the main input signals and output signals required to trigger the memory array and read write pointer logic block. There is only one clock to synchronize both write and read operations. By giving active low asynchronous rest

signal, buffer can be reset when needed. One extra signal named "grnt2" is used for taking the grant from transmit arbiter which is one of the main part of PCI-Express 3.0 data link layer transmitter.

PIN	SIZE	Description
pcie_clk	1 bit	System clock of 100 MHz
rst_n	1 bit	Active high reset signal
tx_start_vc0	1 bit	Active high start of frame signal
tx_data_coming	1 bit	Indicates that the data is passing through the bus
tx_end_vc0	1 bit	Active high end of frame signal
end_of_frame	1 bit	Indicates the end of frame signal (FSM output)
frame_end_wr	1 bit	Indicates the end of frame and ready to write into data_reg variable
rx_invalid	1 bit	Indicates the invalid signal, occur only when rx_retry_notification[12]=1
wr_enable	1 bit	Write enable signal to write the data in buffer
rd_enable	1 bit	Read enable signal to retry data from buffer
tx_tlp_frame_vc0	32 bit	Input packet coming from the TLP MUX
grnt1	1 bit	Grant from arbiter to Retry Buffer
tx_retry_data	59 bit	Retry data for retransmission purpose from Buffer
fifo_full1	1 bit	Indicates whether fifo is full or not
fifo_empty1	1 bit	Indicates whether fifo is empty or not
fifo_ae	1 bit	Indicates whether fifo is almost empty or not
fifo_hfl	1 bit	Indicates whether fifo is half full or not
fifo_af	1 bit	Indicates whether fifo is almost full or not
err	1 bit	Indicate error signal when fifo is full and wr_enable=1 or fifo is empty and rd_enable=1

Table. 1: Pin description of top module

B. Memory array

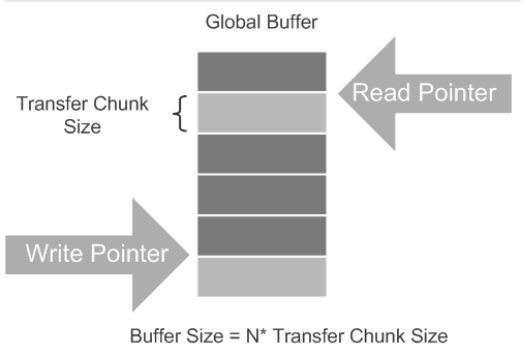

Fig. 3: Memory array operation [8]

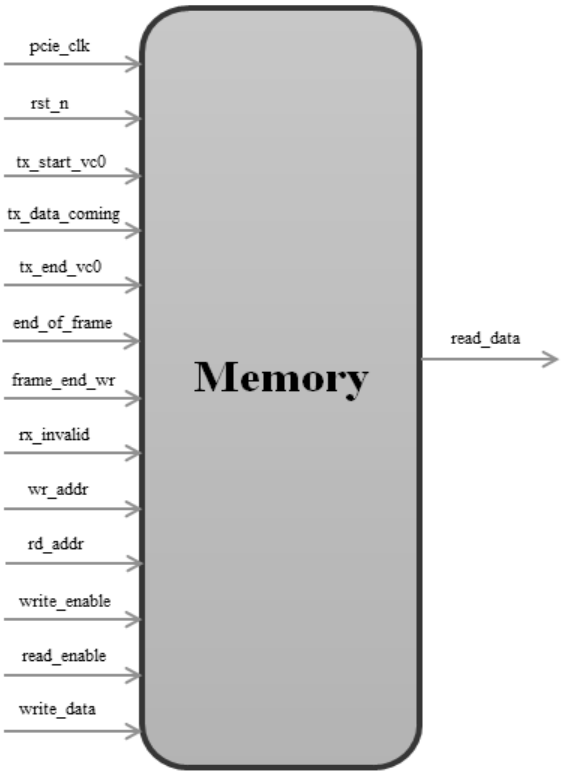

Fig. 4: Pin diagram of memory array

PIN	SIZE	Description
pcie_clk	1 bit	System clock of 100 MHz
rst_n	1 bit	Active Low asynchronous reset
tx_start_vc0	1 bit	Active high start of frame signal
rx_invalid	1 bit	Indicates the invalid signal, occur only when rx_retry_notification[12]=1
tx_data_coming	1 bit	Indicates that the data is passing through the bus
tx_end_vc0	1 bit	Active high end of frame signal
end_of_frame	1 bit	Indicates the end of frame signal (FSM output)
frame_end_wr	1 bit	Indicates the end of frame and ready to write into data_reg variable
wr_addr	12 bit	Indicates the write address of memory
rd_addr	12 bit	Indicates the read address of memory
write_enable	1 bit	Indicates the write enable signal to write the data into buffer
read_enable	1 bit	Indicates the read enable signal to read data from buffer
write_data	32 bit	Indicates the data coming from TLP MUX0
read_data	59 bit	Indicates the data coming from buffer

Table. 2: Pin description of memory array

C. Read write pointer logic block

This module explains how read and write pointer will increment and decrement when retry notification will be received. This block will work on different conditions than normal FIFO. This logic block will generate different flags like fifo_full, fifo_empty, fifo_af, fifo_ae, fifo_hf and err. As tx_start_vc0 signal will be asserted, this block will start to store the data frame by concatenating three frames which are made up of next sequence number as start frame, payload data and CRC of that data as end frame. When CRC of that particular data will be received, end signal will be asserted to indicate the frame completion. As it will be received, buffer address and pointer will be incremented by one and it points to the location next to top location. It shows the error signal when there is no space in buffer to store the data frames and it receive the write request signal or when the buffer is empty and it receive the read request signal.

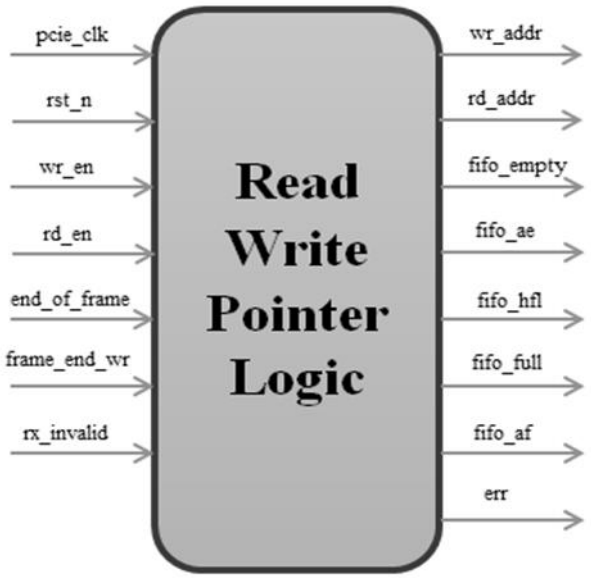

Fig. 5: Pin diagram of read write pointer logic block

PIN	SIZE	Description
pcie_clk	1 bit	System clock of 100 MHz
rst_n	1 bit	Active Low asynchronous reset
wr_en	1 bit	Write enable signal to write the data in buffer
rd_en	1 bit	Read enable signal to retry data from buffer
end_of_frame	1 bit	Indicates the end of frame signal (FSM output)
frame_end_wr	1 bit	Indicates the end of frame and ready to write into data_reg variable
rx_invalid	1 bit	Indicates the invalid signal, occur only when rx_retry_notification[12]=1
wr_addr	12 bit	Indicates the write address of memory
rd_addr	12 bit	Indicates the read address of memory
fifo_full	1 bit	Indicates whether fifo is full or not
fifo_empty	1 bit	Indicates whether fifo is empty or not
fifo_ae	1 bit	Indicates whether fifo is almost empty or not
fifo_hfl	1 bit	Indicates whether fifo is half full or not
fifo_af	1 bit	Indicates whether fifo is almost full or not
err	1 bit	Indicate error signal when fifo is full and wr_enable=1 or fifo is empty and rd_enable=1

Table. 3: Pin Description of read write pointer logic block

III. SIMULATION RESULTS

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

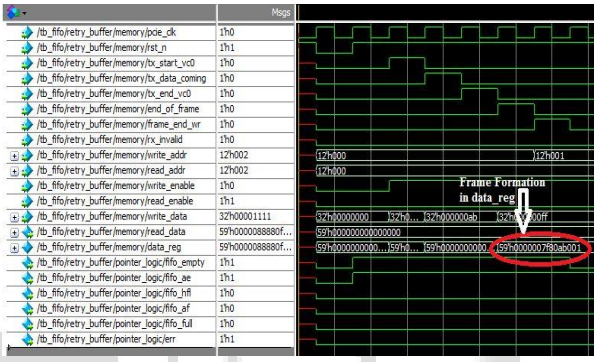

Fig. 6: Waveform of buffer, when writing first data into memory

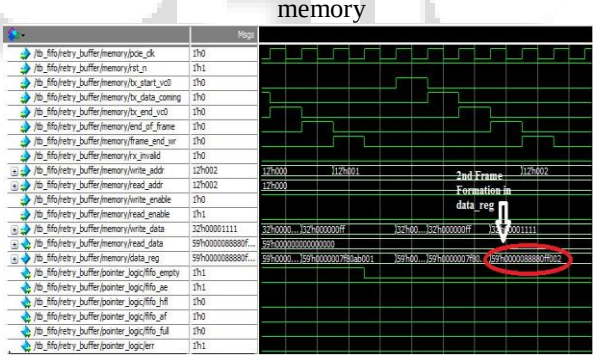

Fig. 7: Waveform of buffer, when writing second data into memory

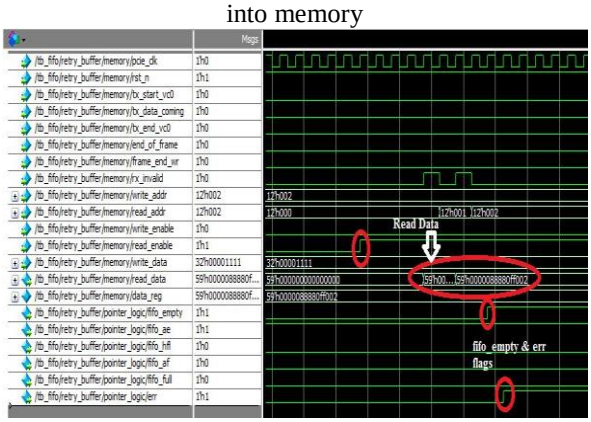

Fig. 8: Waveform of buffer, when reading the data depending upon the retry notification received

Design of Synchronous Retry Buffer for PCI-Express 3.0 Data Link Layer Transmitter has been prepared in Verilog language and its unit testing (simulation) has been done in *Modelsim PE Student edition* software. After designing the Synchronous Retry Buffer, 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.

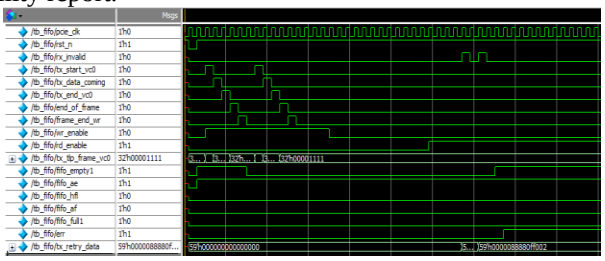

Fig. 9: Waveform of top module of retry buffer

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.

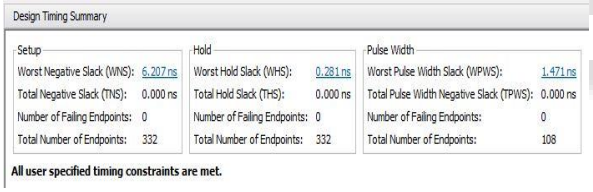

Fig. 10: Design Timing Summary

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

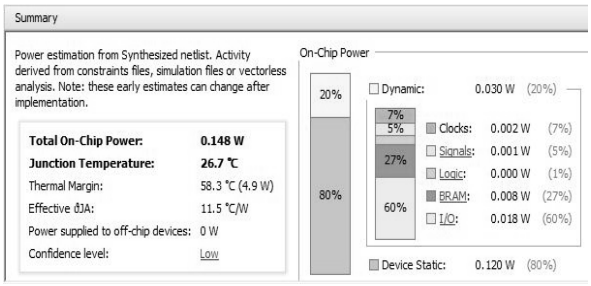

Fig. 11: Power Summary

Here static power is more than dynamic power and total power consumption is 0.148 W. There should be less power as much as possible with junction temperature.

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	134	53200	1
Slice Registers	93	106400	1
Memory	7	140	5
IO	106	127	83
Clocking	1	32	3

Fig. 12: 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, it states that this is the first paper on buffer for PCI-Express 3.0 Data Link Layer Transmitter. This buffer contains three configurable parameters which are DATA_WIDTH, DEPTH and ADDR_WIDTH, can be changed on demand as version of PCI-Express changes. Simulated results which are shown above operated at 100 MHz clock speed. These results are very much accurate according to paperwork. Synthesis has been completed with design optimization and analyzed timing, power and utility report.

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] "Design, Verification And Syntehsis Of Synchronous Fifo" 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] Eugin Hyun and Kwang-Su Seong, The Design of PCI Express for Future Communication Platform, VLSI Lab., Dept. of Electronics Engineering, 2004
[4] PCI Express 1.1 Root Complex Lite x1, x4 IP Core User's Guide, LATTICE SEMICONDUCTOR, February 2012
[5] Xilinx User Guide "LogiCORE™ PCI EXPRESS® ENDPOINT BLOCK PLUS v1.2", UG341 February 15, 2007
[6] Xilinx User Guide "LogiCORE IP ENDPOINT BLOCK PLUS v1.15 for PCI Express", UG341 July 22, 2011
[7] Don Anderson and Tom Shanley, "PCI System Architecture", MINDSHARE INC., 1999
[8] "Reference Design for Streaming Data from Multiple High-Speed Digitizer Modules", April 19, 2010, retrieved from [online] http://www.ni.com/white-paper/9412/en/

