

# Design of a High Speed Serializer, Timing Analysis and Optimization in TSMC 28nm Process Technology

Nupoor Sharma  
Department of ECE  
SJBIT, Bangalore  
Karnataka, India  
*nupoor591@gmail.com*

Dr Nataraj K R  
Professor & HOD,  
Department of ECE  
SJBIT, Bangalore  
Karnataka, India  
*krnataraj@sjbit.edu.in*

**Abstract**— The use of serializers and deserializers in SerDes devices is a compulsory requirement for chip to chip communication. They are useful in converting parallel to serial data and vice-versa. Multiple SerDes devices are housed in a single package.

In this paper, a high speed serializer targeted for speeds as high as 20Gbps is proposed and implemented. This is designed primarily for SerDes devices for chip to chip communication. The serializer is designed to facilitate high speed transfer data rates. This design employs differential logic implementation for the circuit, so as to owe high speed operation when compared to single ended implementation. The custom circuit design simulations are compared against standard library files generated by LIBERATE tool. Also Timing fixes were done using Synthesis flow by writing the RTL code for custom top module design and feeding it to DC and IC Compilers.

**Keywords**— LIBERATE; Differential logic; Synthesis, RTL; DC and IC Compiler.

\*\*\*\*\*

## I. INTRODUCTION

Owing to the multimedia applications, there is a rise in the demand for bandwidth of the transmission. This demand has paved way for the development of high speed and low cost serial link technology.

For getting data on and off of chips or boards and even boxes, a high-speed serial link is the ultimate choice. For Ex: With speeds from 1 to 12 Gbps and payloads from 0.8 to 10Gb, it accounts for a lot of data transfer. Thus, with fewer pins, no massive simultaneous switching output (SSO) problems, lower cost, and lower EMI, high-speed serial communication is the best option. Multi gigabit transceivers (MGTs) are by far the best option when we need to transfer lots of data at faster speeds.

As device geometry in integrated circuit (IC) grew smaller and smaller, and maximum toggle rate of flipflop (Fmax) increased, the requirement for I/O bandwidth exploded. In fact, some developments even allowed for I/O frequency being faster than Fmax.

The building blocks of serializer include multiplexors, latches, flipflops and xor gates which connect data output to drivers. As a result of these, it is difficult to fix timing violations in the path.

High-speed and Low power design have become two crucial pillars of modern VLSI circuits. The overall system performance is always determined by timing elements such as flip-flops and latches, and thus their improvement is one of the most critical tasks to enhance the system performance. The most important concerns in design of timing path elements such as D flip-flops (DFFs) and latches are power consumption and timing constraints.

Single ended design of serializer in earlier IBM transmitter architecture can support upto 10Gbps data rate. Differential implementation of the serializer overcomes this drawback by having fast speed, and higher data rates.

This implementation is improvised version of older, currently existing single ended implementation in IBM transmitter architecture. The circuit simulations were compared against .library files generated by STA tool, LIBERATE characterization tool from CADENCE VIRTUOSO. Also digital logic implementation of the top module design was done using Synthesis RTL to GDS flow. The results show the comparison in terms of area, power when comparing custom design to standard cells from Synthesis flow.

The rest of the paper is organized as follows. Section II briefly reviews the hardware characteristics of Serializer. The motivation of this work is also presented in this section. Section III then discusses existing topology for serializer and its basic functionality. The details of the proposed design are presented in Section IV. Experimental results are described in Section V.

## II. BACKGROUND

### A. Motivation

The SerDes devices use serializer and deserialiser to transfer data for chip to chip communications. Hence in order to achieve high speed and reliable data transfer for chip to chip, boards or packages, the serializer should be capable of sending data at high rate, by pumping the lower data rate input to higher speed data output. The usually available implementations of serializers include single ended design. So

here, an improved version is designed involving differential logic implementation to promote high speeds.

### B. Serializer

Serial-to-parallel and parallel-to-serial conversions have been an integral part of I/O design from the beginning. And, so has the idea of recovering a clock, or “locking a clock to an incoming stream.” But why has the SERDES suddenly become so important?

A Serializer takes  $n$  bits of parallel data changing at rate  $y$  and transforms them into a serial stream at a rate of  $n$  times  $y$ . On the contrary, a Deserializer takes serial stream at a rate of  $n$  times  $y$  and changes it into parallel data of width  $n$  changing at rate  $y$ .

Conceptually, the input to the transmit stage of a serializer is an  $n$ -bit datapath which is serialized to a one-bit serial data signal for application to the Feed Forward Equalizers and Driver stages. Generally the value of  $n$  is a multiple of 8 or 10, and may be programmable on some implementations. Values of  $n$  which are multiples of 8 are useful for sending unencoded and/or scrambled data bytes; values of  $n$  which are multiples of 10 are useful for protocols which use 8B/10B coding.

A simplified schematic of the 2:1 Serializer which performs the serialization of even and odd data streams is shown in Fig.1 below. This simple tree structure has been successfully used at speeds over 40Gbps in CMOS technologies and over 132Gbps in SiGe technologies.

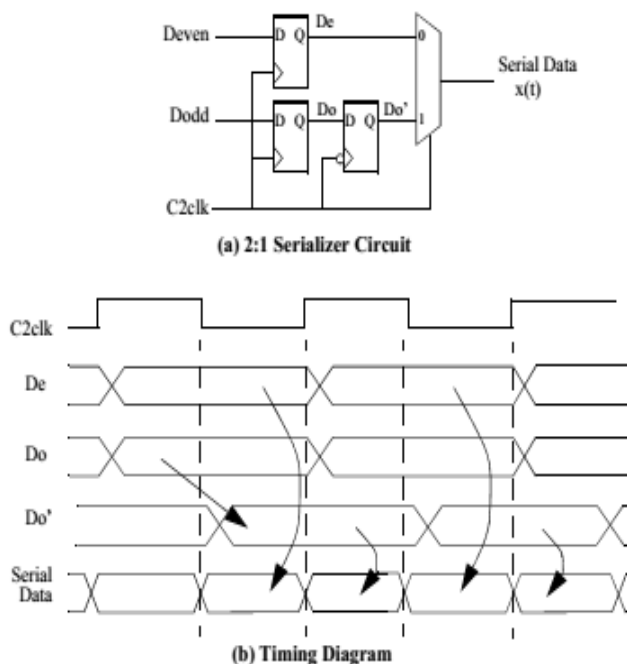


Fig. 1: Detailed 2:1 Serializer operation

The key design constraint that enables using this architecture for high baud rates is as follows:

the  $T_{sq}$  propagation delay of the multiplexor (from the select input to the multiplexor output) must be less than the  $T_{cq}$  propagation delay of the latch (from the clock input to the latch output). This is illustrated in Fig.1.b.

The two bits of parallel data, DEVEN and DODD, are assumed to be time-aligned into the serializer and are synchronized to the half-rate C2 clock. The first two latches capture the parallel DEVEN and DODD signals, creating De and Do outputs on the rising edge of the C2clk signal. The Do signal is generated by resampling the Do signal on the falling edge of the C2 clock. These two signals are skewed by 1 UI and provide the inputs to the 2:1 MUX. The C2 clock controls the select input of this MUX such that when the De input is selected when the clock is low, and Do is selected when the clock is high. If  $T_{sq} < T_{cq}$ , then the multiplexor always selects a stable input signal, resulting in clean, glitch-free operation. This pingpong action is illustrated in Fig. 1.b.

For example, the input to a 8:2 serializer stage consists of eight time-aligned parallel bits, and the output consists of two streams, i.e., Deven and Dodd half-rate serial streams feeding the Feed Forward Equalizer shift register. Implementation wise, the 8:2 serializer is built using a cascade of four 2:1 serializers feeding two 2:1 stages.

### III. RELATED WORK

The design shown below is a transmitter designed for low power, 10Gbps serial link transmitter of IBM architecture.

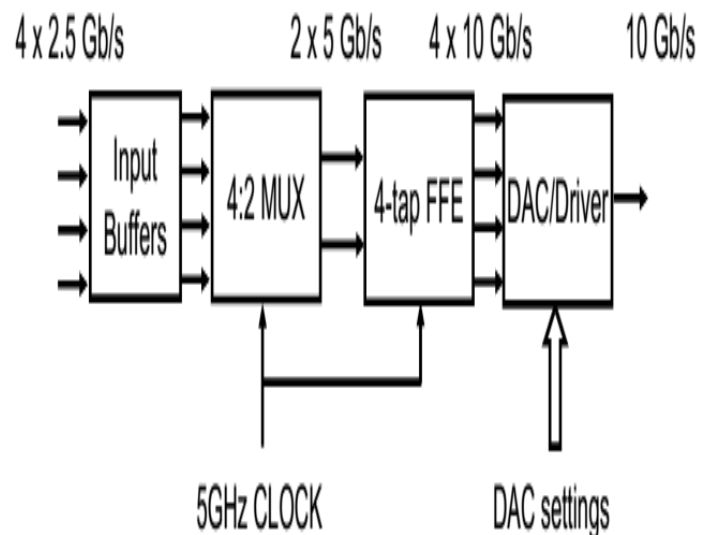


Fig. 2: Transmitter Top level block diagram

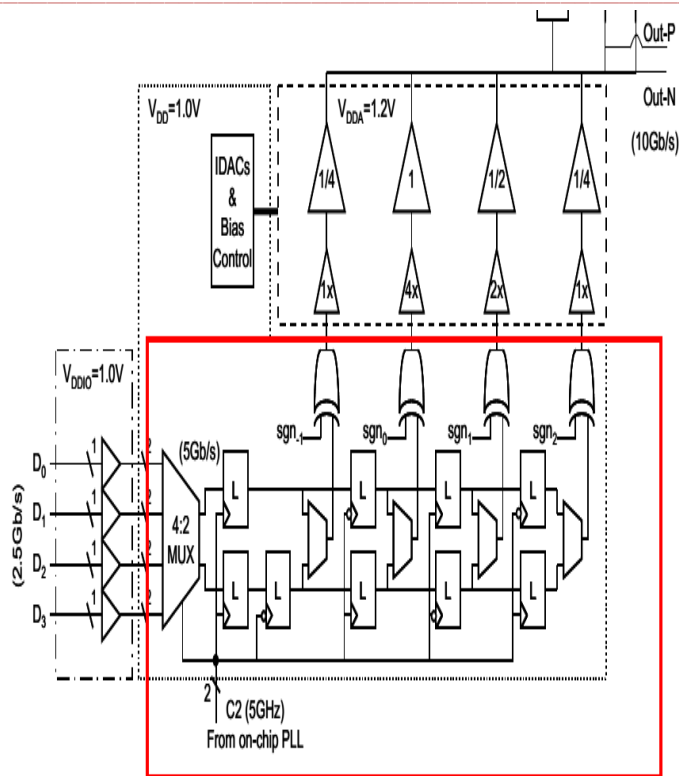


Fig. 3: Highlighted section depicting serializer

#### IV. THE PROPOSED DESIGN AND IMPLEMENTATION

In this section, the system architecture of serializer and the flows to generate liberty files for the sub building blocks, and RTL to GDS flow are discussed.

##### A. System Architecture of high speed, 10Gbps serializer

As illustrated in Fig. 8, high speed serializer top level design with its sub modules is shown.

In this design, differential level design was implemented. Differential signaling was such a method. It has several advantages over single-ended signaling. For example, it is much less susceptible to noise. It helps to maintain a constant current flow into the driving IC. And rather than comparing a voltage to a set value or reference voltage, it compares two signals to each other. Thus, if the signal referenced as the positive node has a higher voltage than the one referenced negative, the signal is high, or one. If the negative referenced signal is more positive, the signal is low, or zero. The positive and negative pins are driven with exact complementary signals. This implementation promotes high speed data rates, when compared to single ended implementation.

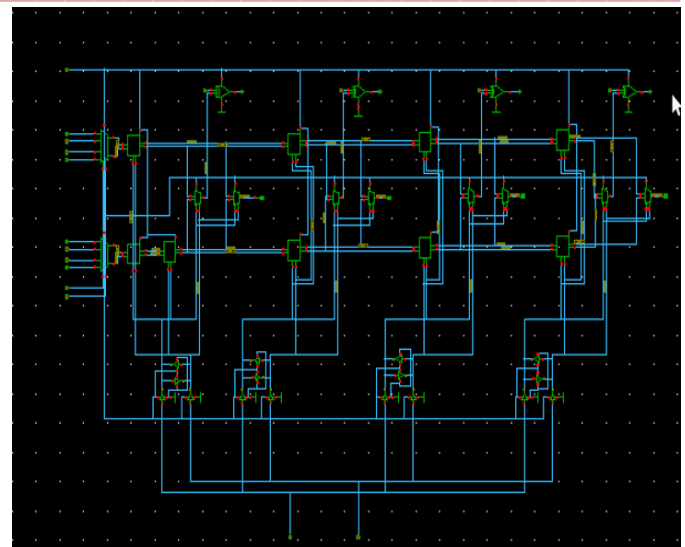


Fig. 4. System architecture of proposed design: high speed serializer.

The design uses clock buffers as a part of clock tree to transmit the clock to all latches. The data inputs are d0, d1, d2, d3 and their complements which are fed to two 4:2 differential multiplexors. The outputs from these devices are fed into differential latch. In the lower section of the design, an extra latch enabled by the complementary clock signal is added to facilitate half cycle delay. Henceforth, the design employs half cycle delay in its outputs. The positive outputs from the latches are fed into 2:1 mux which then feeds the output to an xor gate. The clocks fed to the initial input section multiplexors is 2.5Ghz, and the clock fed to latches is 5Ghz. Thus, obtaining data rates at 10Gbps at the output of xor gates.

The clocks supplied for multiplexors and latches used for this design are complementary in nature.

##### B. The Proposed Differential latch design

The basic differential latch design employs two enable signals clk and its complement clkb which are fed to stacked inverters. The inputs are complementary too, inp and inn supplied to the stacked inverter gates. A feedback inverter loop is connected to help in restoring action of latched data.

For designing a 4:2 differential mux, two 2:1 muxes are linked by combining sel and selb inputs, supplying them simultaneously for two mux sections. So the top section takes d0, d1 and the lower mux section takes d0\_b and d1\_b inputs and selects accordingly

ULVT devices are used for the design to allow high speed as these devices allow fast operation but consume more power. Thus a tradeoff is made between power and speed.

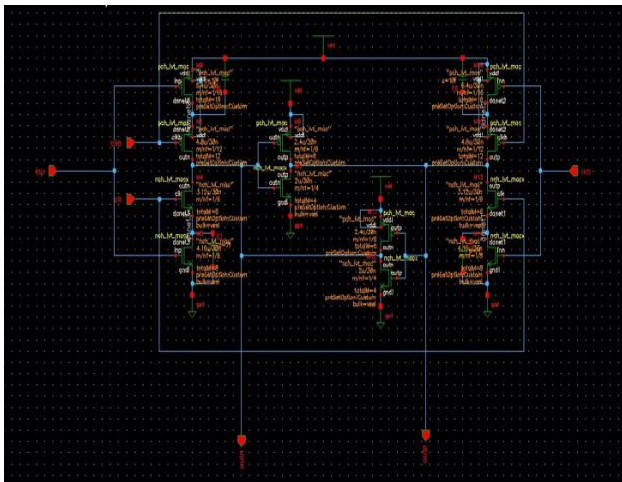


Fig. 5. Differential latch topology

### C. The proposed 4:2 differential multiplexor design

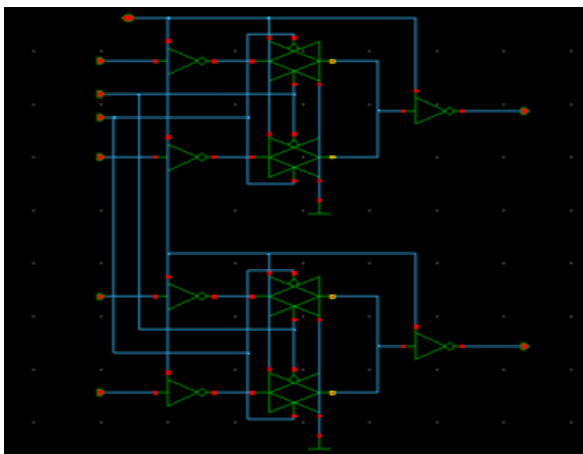


Fig.6. Differential 4:2 mux design

### D. The proposed 2:1 multiplexor design

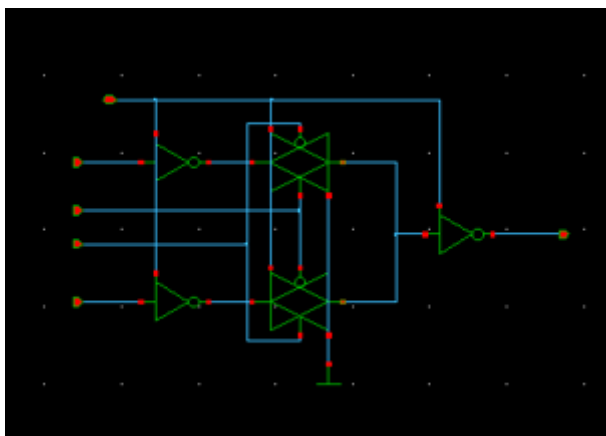


Fig.7. Proposed 2:1 Mux design

Let us now dive into the concept itself. The working goes as follows. The proposed design takes four data inputs and

their complements generated using pseudo random sequence generator block, written using VerilogA code.

The four data inputs of 2.5Gbps data rates are fed into multiplexors which are then accordingly sent via latches to the final mux section which feeds the output to drivers through xor gates. The data is latched at 5Ghz clock and thus the data inputs are converted to 5Gbps data. The final mux converts the data to 10Gbps outputs.

The design involves four sections of 2:1 serializers which thus convert four data inputs to four data outputs available at the rear end of the xor gates. Thus the outputs o1, o2, o3 and o4 are delayed by half clock cycle.

Circuit simulations of the setup and hold time of latch are carried out by keeping data constant and then shifting clock, and vice-versa. These values are verified by generating liberty files by using LIBERATE Characterization tool . Also the mux delays are measured using STA process, LIBERATE tool and by custom circuit simulations.

Also, a RTL code for the top module design is written from the scratch which is then fed to Synthesis DC compiler. It generates a set of timing reports. It is this stage where we perform the setup time violation fixes. This data is then fed to IC compiler tool, which generates APR( Auto placement and route) design and layout for our RTL code input. Hold times fixes are done in this stage, by inserting buffers in data path and very rarely into clock paths. The layout is done by picking cells from standard library. Thus, a comparison is done between standard cells and custom designed cells in performance, area.

## V. PERFORMANCE EVALUATION

In order to investigate the effectiveness of the proposed design, the complete top module layout was done and simulations were carried out. It was seen that this implementation can easily work upto 20Gbps data rates, owing to the differential logic implementation. Rise and Fall times of 15ps was set for the clock signals for faithful design.

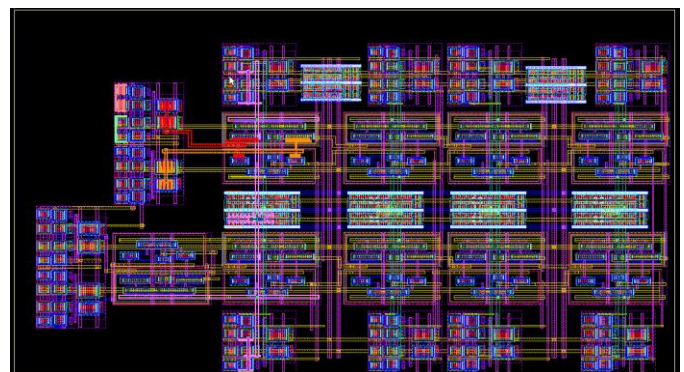
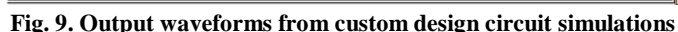


Fig. 8. Top module design layout





The setup and hold times of latch, under worst case corner, Slow corner [SSS, voltage: 0.925V and temperature: 110C] are tabulated as follows for comparison.

<b>Latch</b>	<b>Circuit Simulations</b>	<b>LIBERATE analysis</b>
<b>Setup time</b>	17.96ps	22ps
<b>Hold time</b>	-9ps	-15ps

**Table. 1. Latch data comparison**

Mux	Circuit Simulations	LIBERATE analysis
-----	---------------------	-------------------

<b>Duty Cycle</b>	48.99%	50%
<b>Cell delay</b>	21ps	29ps

**Table. 2. Mux data comparison**

This paper introduces an effective high speed serializer design using differential logic implementation in circuit design. This design can support upto 20Gbps data rates in top module layout, which is a very high speed targeted. It also shows the comparison between circuit simulations and liberty files generated by STA tool. RTL Code and Synthesis flow was also run on the design, from digital end perspective.

- [1] Alexander Rylyakov and Sergey Rylvov, IBM T.J. Watson Research Center, "A Low Power 10 Gb/s Serial Link Transmitter in 90-nm CMOS", CSIC Symposium, November 2005.
- [2] John W. Poulton, Fellow, IEEE, William J. Dally, Fellow, IEEE, XiChen, Member, IEEE, John G. Eyles, "A 0.54 pJ/b 20 Gb/s Ground-Referenced Single-Ended Short-Reach Serial Link in 28 nm CMOS for Advanced Packaging Applications " , IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 48, NO. 12, DECEMBER 2013.
- [3] Pavan Kumar Hanumolu, Bryan Casper\*, Randy Mooney\*, Gu-Yeon Wei\*\* and Un-Ku Moon, "JITTER IN HIGH-SPEED SERIAL AND PARALLEL LINKS", IEEE JOURNAL, 2004.
- [4] Chih-Hsien Lin , Chung-Hong Wang and Shyh-Jye Jou , "5Gbps Serial Link Transmitter with Pre-emphasis " ,
- [5] Chenjie Gu, INTEL Corp, "Challenges in Post silicon validation of High Speed I/O Links," IEEE Journal, 2004.
- [6] Emre Salman, Ali Dasdan, Eby Friedman, " Exploiting Setup and hold time interdependences in Static Timing Analysis", IEEE Transactions on Computer aided design of Integrated circuits and systems, Vol .26, June 2007.
- [7] A Significant Technology Advancement in High-Speed Link Modeling and Simulation, ALTERA Corporation.
- [8] High speed I/O design, Text guide, INTEL corporation 2001.
- [9] Jitter in High speed Serial & Parallel Links, ISCAS, IEEE 2004.
- [10] LIBERATE Characterization Suite manual from CADENCE VIRTUOSO.
- [11] PRIMETIME manual from SYNOPSYS.
- [12] Vladimir Stojanovic and Mark Horowitz, Stanford University, "Modeling and Analysis of High-Speed Links", Research supported by MARCO Interconnect Focus Center and Rambus Inc.
- [13] Ganesh Balamurugan, Bryan Casper, James E. Jaussi, Mozghan Mansuri, Frank O'Mahony, and Joseph Kennedy, Member, IEEE, "Modeling and Analysis of High-Speed I/O Links", IEEE 2009.
- [14] Circuits for the Design of a Serial Communication System Utilizing SiGe HBT Technology, By Thomas W. Krawczyk Jr.
- [15] Mike Peng Li, Masashi Shimanouchi, and Hsinho Wu, Altera Corporation, "Advancements in High-Speed Link Modeling and Simulation" (An Invited Paper for CICC 2013), IEEE 2013.
- [16] Julien Saadé, Frédéric Pétrot, André Picco, Joel Huloux, Abdelaziz Goulahsen, "A System-Level Overview and Comparison of Three High-Speed Serial Links: USB 3.0, PCI Express 2.0 and LLI 1.0", IEEE 2013.
- [17] Vladimir Stojanovic and Vojin G. Oklobdzija, Fellow, IEEE, "Comparative Analysis of Master-Slave Latches and Flip-Flops for

- 
- High-Performance and Low-Power Systems”, IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 34, NO. 4, APRIL 1999.
- [18] Fatemeh Aezinia, Ali Afzali-Kusha, and Caro Lucas, Nanoelectronics Center of Excellence, “Optimizing High Speed Flip-Flop Using Genetic Algorithm”, IEEE 2006.
- [19] “High Speed Serial I/O made simple” A designer’s guide, By Abhijit Athavale, Carl Christensen.
- [20] Liberty User Guides and Reference Manual Suite Version 2013.03.
- [21] Frank O’Mahony, Intel, “Energy-Efficient I/O Design for Next-Generation Systems”, IEEE International Solid State Circuits Conference, 2014.
- [22] Emre Salman, Ali Dasdan, Feroze Taraporevala, Kayhan Küçükçakar, and EbyG.Friedman, Members, IEEE, “Exploiting Setup–Hold-Time Interdependence in Static Timing Analysis”, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, VOL. 26, NO. 6, JUNE 2007.
- [23] Hongli Gao Fei Qiao Dingli Wei Huazhong Yang , Division of Circuits and Systems, Dept. of Electronic Engineering, Tsinghua University, “A Novel Low-Power and High-Speed Master-Slave D Flip-Flop “, IEEE 2006.
- [24] Y. Berg, “NOVEL HIGH SPEED DIFFERENTIAL CMOS FLIP-FLOP FOR ULTRA LOW-VOLTAGE APPLICATIONS”, IEEE 2011.
- [25] “Digital Integrated Circuits”, By Jan. M .Rabaey.
- [26] “Design of High-Speed Serial-Links in CMOS”, Won Namgoong, SRC Research Review, September 2003.
- [27] “High Speed Serial Links, Design Trends and Challenges”, Vladimir Stojanović, Integrated Systems Group, Massachusetts Institute of Technology.
- [28] SystemVerilog 3.1a, Language Reference Manual, Accellera’s Extensions to Verilog.
- [29] Vivado Design Suite, User Guide, Design Analysis and Closure Techniques, XILINX, January 2014.
- [30] VerilogHDL, A guide to Digital Design and Synthesis, By Samir Palnitkar.
- [31] Synopsys, Timing Constraints and Optimization, User Guide, December 2011.
- [32] The Quartus II TimeQuest Timing Analyzer, ALTERA Corporation.