DESIGN OF SERIALIZER AND DESERIALIZER OPERATING IN 65 nm CMOS TECHNOLOGY FOR HIGH-SPEED SERIAL LINK (HSSL) APPLICATIONS

By

Zexian Li

Senior Thesis in Electrical Engineering

University of Illinois at Urbana-Champaign

Advisor: Professor José E. Schutt-Ainé

December 2015

Abstract

In order to explore the issue for the increasing need of high speed serial link to transit massive data, this thesis addresses the Serializer and Deserializer components design within the link. The serializer at the transmitter-side converts the data form from parallel into a single serial data stream for transmission while the Deserializer will convert the data back to its original parallel form at the receiver end; this allows for higher speed data transmission, less number of interconnects, power consumption, area and crosstalk compared to the traditional parallel ways.

For the Serializer's basic unit – the 2 to 1 serializer, the four building blocks are: two negative edge triggered D flip-flops, a positive latch and a 2-to-1 Mux. The basic unit will be used to build the 4 to 1 serializer and 8 to 1 serializer. Circuit design processes, testbench constructions and simulations for the 2 to 1 serializer, 4 to 1 serializer and 8 to 1 serializer will be presented. Experimental results show the feasibility of the design discussed. Also, a shift register based 8-bit Deserializer is designed to Deserializer the data back to its original from. The design discusses above could operate in 65 nm CMOS technology at desired frequencies 3.2 GHz in Cadence Virtuoso to work with the entire high-speed serial link with a supply voltage of 1.2 V.

Subject Keywords: SerDes; Serializer; Deserializer; Cadence; High-Speed Serial Link (HSSL); SoC

Acknowledgments

I would like to express my deepest thanks to my professor, José E. Schutt-Ainé, who unconditionally supported me and instructed me through the research project. I also want to express my sincere gratitude to Dr. Xinying Wang, who patiently answered all my trivial circuit questions along the way and was always willing to take time to discuss the project with me. In addition, I appreciate Dr. Da Wei for the through guidance and support when I was in the bottleneck of the project. I also want to thank Rushabh Mehta who patiently gave me tutorials and advice on using Cadence Virtuoso. Without their help, the project could not have gone so far.

Contents

1. Introduction

1.1 Motivation

With the advancement of technology, there has been more and more need for the gigabit rate link for storage application, data communication, computer networks and etc. To meet with the high processing multi-gigabit speeds and system performance, it becomes necessary to have prompt and efficient high-speed inter-connects. Traditional parallel link has been used in circuits for a long time, which let the data be sent over multiple channels simultaneously. But as the data rates gets higher and higher shown in the figure below, the parallel links can't keep up with the system performance due to cross-coupling noise and data skew especially in long distance transmission.

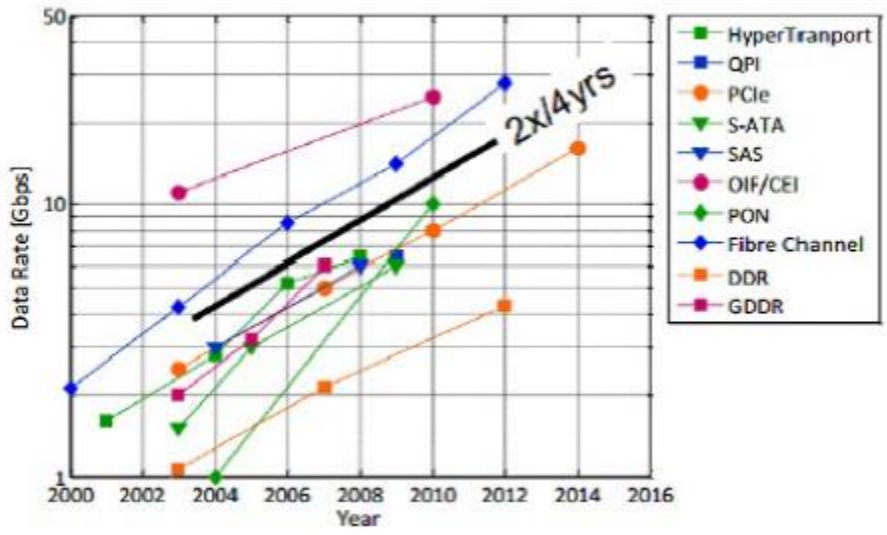

Figure 1.1: I/O Link Signaling Data Rate Trends [1]

While serial communication has lots of advantages over the parallel links. In serial communication, the data is sent over a single channel sequentially. Because of that, there is no crosstalk noise, Furthermore, due to more interconnects, wires and ports are employed in parallel links, much more area, dynamic power and leakage power will be required. Therefore, there arises a need for SERDES. As the name suggests, is a serial transceiver that converts parallel data to serial data on the Transmitter side and serial data back to parallel data on the receiver side. A Serializer and Deserializer are to be designed to

work with the SERDES in the research group to achieve the goal of high-speed serial data transmission with signal integrity.

1.2 Purpose

The objective of this thesis is to present an introduction to the theoretical concepts behind the design of a Serializer and Deserializer for a high-speed Serial link. This thesis entails the design architectures and building blocks of 8-to-1 Serializer and 1-to-8 Deserializer in 65 nm CMOS technology operating at 3.2 GHz with a supply voltage of 1.2 V.

1.3 Outline

This thesis is organized as follows. Chapter 2 entails the fundamental and basic concepts of SerDes. Chapter 3 describes of the architecture of the SerDes and SerDes components. Chapter 4 presents the topology and design of a Serializer and Deserializer, including choices of implementation, design, and schematic and simulations on Cadence Virtuoso. Chapter 5 concludes the Serializer/Deserializer design.

2. High Speed Serial Link Overview

2.1 Parallel vs. Serial Links

This section discusses parallel communication and serial communication's advantages and disadvantages after comparison. A study conducted by R. Dobkin [2], shows that serial links outperform parallel links especially when long-range communication is needed.

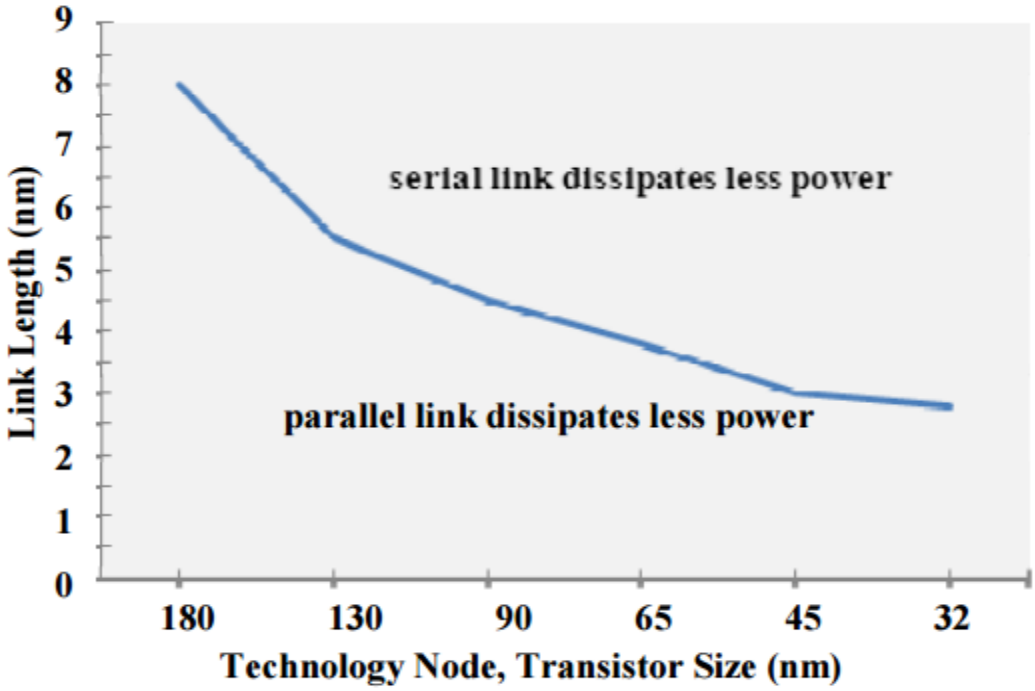

Figure 2.1: Minimum Link Length above which Less Dynamic Power is Dissipated by the Serial Link than the Parallel Link

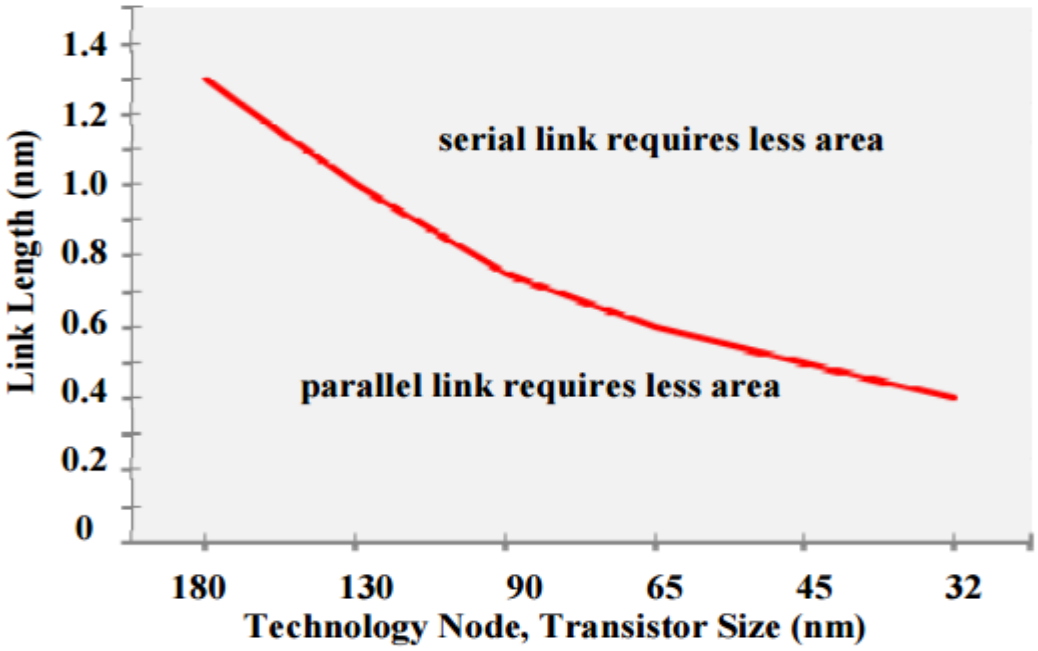

Figure 2.2: Minimal Link Length Above Which the Active Area and Leakage of the Serial Link Are Smaller than in Parallel Link

The above two figures show that for any particular technology node there is a limiting value of the link length above which, it is better to implement the link as serial rather than parallel because it is more advantageous in terms of power and area. [3]

2.1 Data Transmission

Data transmission is the physical transfer of data over a point-to-point communication channel. Nowadays there are two common methods of data transmission: parallel communication and serial communication. Historically, parallel communication had been widely used. However, as the technology advanced and the data rate scaled into multi-gigabit ranges, serial communication has become the preferred method of data transmission over the parallel one. [2]

2.2 Parallel Communication

Parallel communication is a method of transmitting multiple lines of data simultaneously. In many cases, the data from the transmitting parallel ports will be sent to the corresponding receiving ports without any handling of data. Therefore, for an n-bit parallel communication, 2n ports and n wires are needed as the figure below shows.

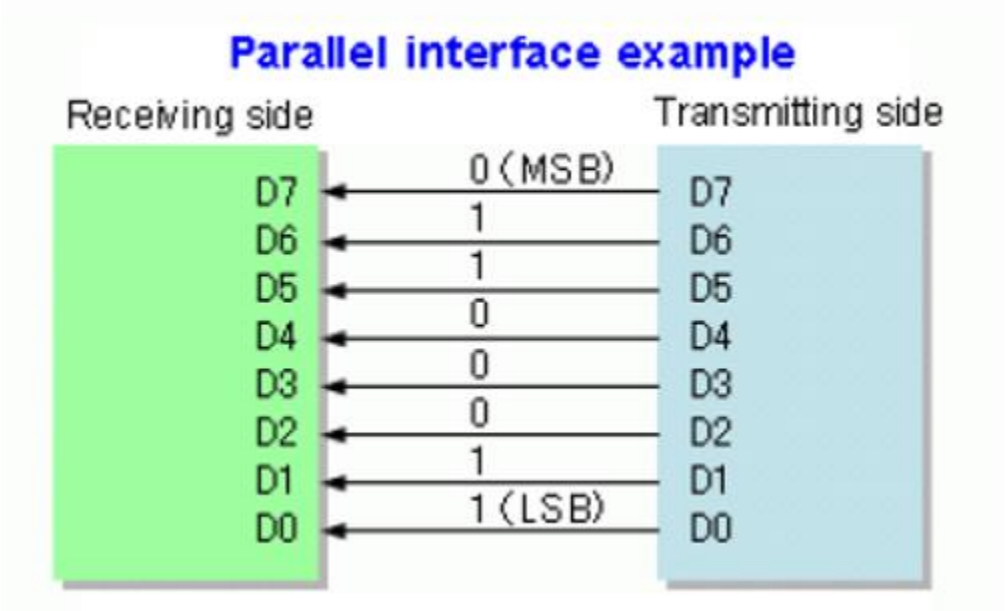

Figure 2.3: Parallel Communication Interface

Parallel communication is usually easier to implement, but occupies lots of area. But data skew is common in parallel communication which is caused by inter-symbol interference (ISI), noise and slight difference in capacitance and inductance of wires. Because of these factors, the arrival times for different data lines can vary significantly from the system's specification. Furthermore, data skew could destroy the integrity of the data. Crosstalk is when one channel creates an undesired effect in other channels due to the conductors being in proximity. Parallel communication also suffers from crosstalk because of the multiple data line as shown above.

2.3 Serial Communication

Serial Communication, on the contrast to parallel communication, only sends data one bit at a time over a single channel. Evidently, serial communication requires fewer transmitting and receiving ports and channels. Signals in serial communication are originally parallel data, and are converted into serial data before the transmitting port by Serializer. Therefore, serial communication requires extra circuit design.

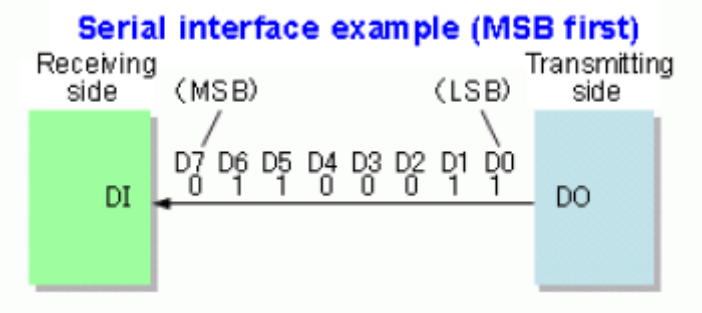

Figure 2.4: Serial Communication Interface

Serial communication is now more commonly used in both long distance and short distance (on-chip) communication because of its advantages over parallel communication. First of all, serial communication has lower cost than parallel communication. For long distance communication, the transmitting cost is significantly reduced because of the use of fewer cables. For on-chip communication, fewer pins and inter-connecting wires are required for serial communication because the number of pins is not directly related to the number of data signals, which can lead to smaller chips and lower cost. Moreover, serial communication does not have issues of data skew. Serial communication uses only one channel to transmit data, so the arrival of each signal is unique. Lastly, serial communication has less of an issue with crosstalk since it uses fewer interconnecting wires.

3. High-Speed Serial Link Overview

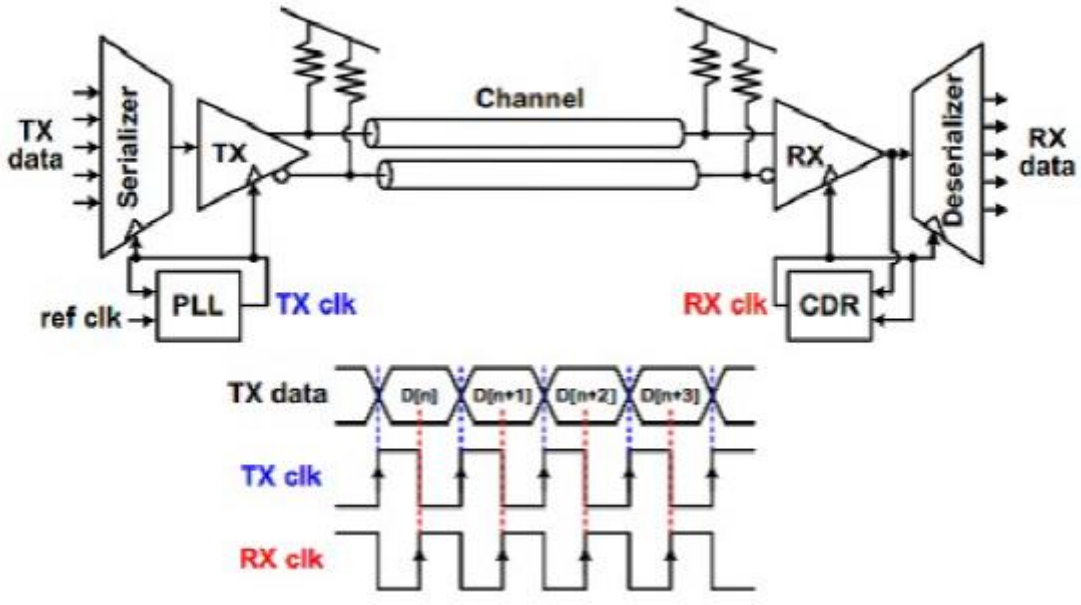

Figure 3.1: A typical High-Speed Link Block Diagram [6]

The SERDES system refers to the complete assembly of transmitter, channel and receiver that constitute the high-speed serial link. A typical block diagram of the SERDES is shown above. In this section, the details of the circuit blocks of SERDES will be discussed.

3.1 Phase-Locked Loop (PLL)

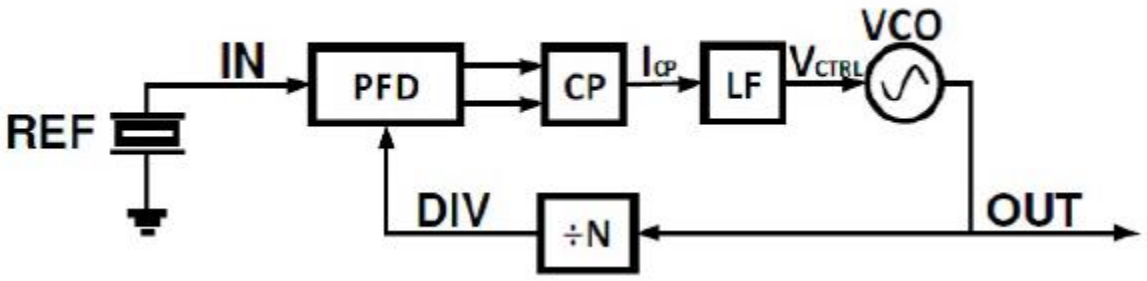

Figure 3.2: A typical Phase-Locked Loop Block Diagram

PLLs are widely used in communication systems and high speed systems. In the SERDES, the purpose of a PLL is to generate the clock signal for the Deserializer to recover the parallel data. Crystal Oscillator could not be applied here in SERDES as it could only produce reference clock with good integrity up to around 200 MHz Phase-locked loop is a negative-feedback system shown above to generate a local clock

at a setup frequency while referring to the input clock. Just as the figure above shows, the working principal of a typical PLL is: the phase-frequency detector (PFD) detects differences between the frequency and phase of a divided version of the generated divided output clock signal with the reference clock. And then the phase-frequency detector produces pulse width modulated (PWM) outputs which are used to drive a charge pump (CP). The charge pump then converts the digital output of the PFD to an analog signal by pumps or drains charge into/from the capacitor in the loop filter (LF). While the loop filter (LPF) is usually second order low pass filter, which filters out the high frequency components in the output of PD. The low pass filter filters out the noise and then sends the output to the Voltage Controlled Oscillator (VCO). At last, the VCO generates the output signal – clock waveform proportional to the control voltage from the output of the low pass filter. Overall, the PLL's purpose in SERDES is to produce clock signals with minimal timing noise, minimal jitter (in time domain) and minimal phase noise (in frequency domain).

3.2 Driver Amplifier

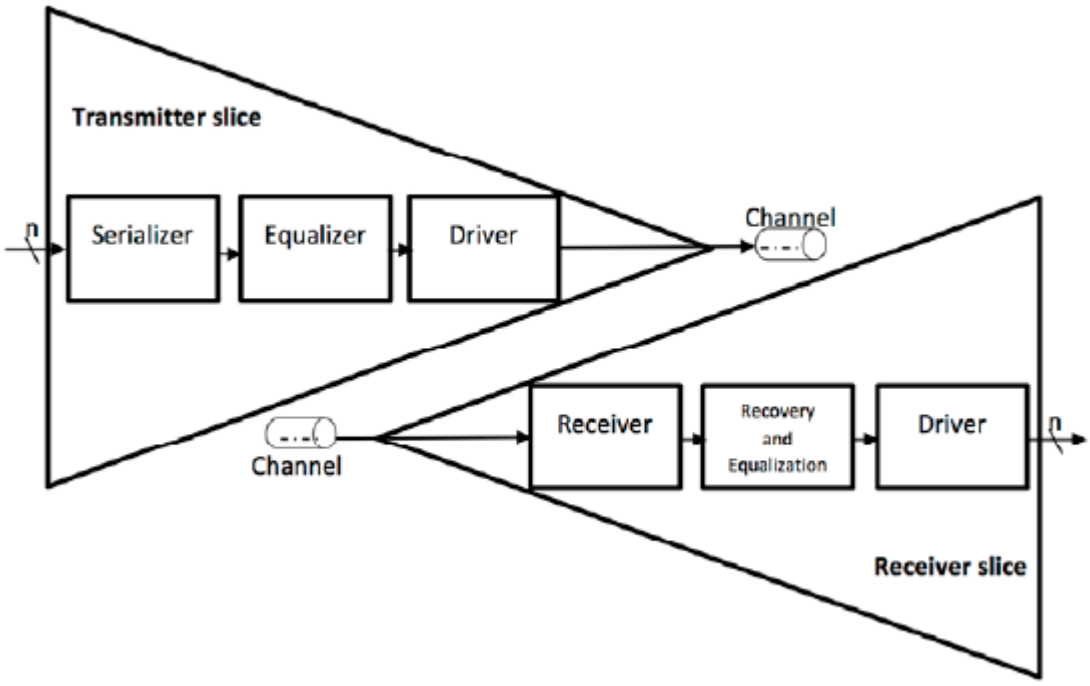

Figure 3.3: A High-Speed Link Block Diagram [1]

As the figure above shows, driver amplifiers are needed both in the transmitter and receiver sides. The Driver Amplifier is used to amplify the input serial bit-stream before it is sent through the channel to the receiver. Driver Amplifier is also needed because it could provide impedance terminations to terminate the channel input and output with its characteristic impedance 50 Ω.

3.3 Clock Data Recovery (CDR)

In SERDES, the clock is embedded in the transmitted data stream and the receiver is expected to extract the clock from the received data and use it to sample the data stream, which is what Clock Data Recovery circuit block is for.

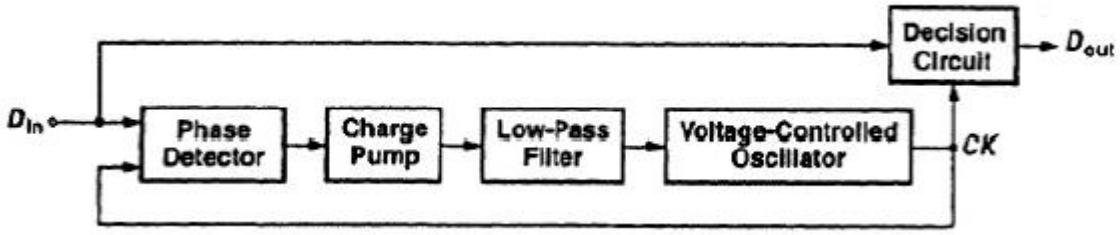

Figure 3.4: Typical Clock Data Recovery Block Diagram [4]

The figure above shows the typical block diagram of the Clock Data Recovery. The phase detector is to make sure the phase difference remained constant to allow optimal sampling, usually it will be the center of the eye in the eye diagram. Because the center of the eye is usually the most optimum point to take a data sample as it has the least jitter mostly. The rest of the blocks are similar with PLL as discussed above. The only difference is that there is no divider in PLL as the generated clock is divided by 1. After extracting the clock signal from the received the serial bit-stream, the CDR will pass it to a decision circuit to generate data which often require a power-hungry PLL circuit.

3.4 Equalizer

In SERDES, equalizer is to recover the signal after the channel so that the signal can be recognized in the next stage of the SERDES as channel behavior distorts the transmitted pulses, causing more and more bit

errors as the data rates increase. There are usually linear and non-linear equalizer implemented in SERDES. One of the circuit techniques used to compensate this behavior is equalization at the receiver. A commonly used type of equalization is to provide a response that directly compensates for the channel's frequency response. Since the channel attenuation is larger at higher frequencies, one possible equalization solution is to boost the higher frequency components which have been severely attenuated. Another solution is to suppress the lower frequency components without altering the higher frequency components.

3.5 Serializer/Deserializer

SerDes stands for Serializer/Deserializer. The Serializer takes parallel data and serializes the data into serial bit stream at the transmitter side. It is a complete digital circuit and precedes the TX driver circuit. At a fundamental level, a serializer is essentially a Multiplexer circuit whose driving clock for the serialization process is the TX_CLK signal generated by the TX PLL. On the other end of the serial link, receiver side, the Deserializer takes the serial data, recovers the clock from the data to original parallel form as before.

Figure 3.5: Typical SERDES Link Diagram

4. Design Architecture

4.1 Design of Serializer

The design of Serializer is presented in Figure 1 and Figure 2. Figure 1 below shows the basic block diagram of a 2 to 1 serializer. The serializer mainly consists of three parts: the negative edge D flip-flop, the D latch and the 2-to-1 Mux.

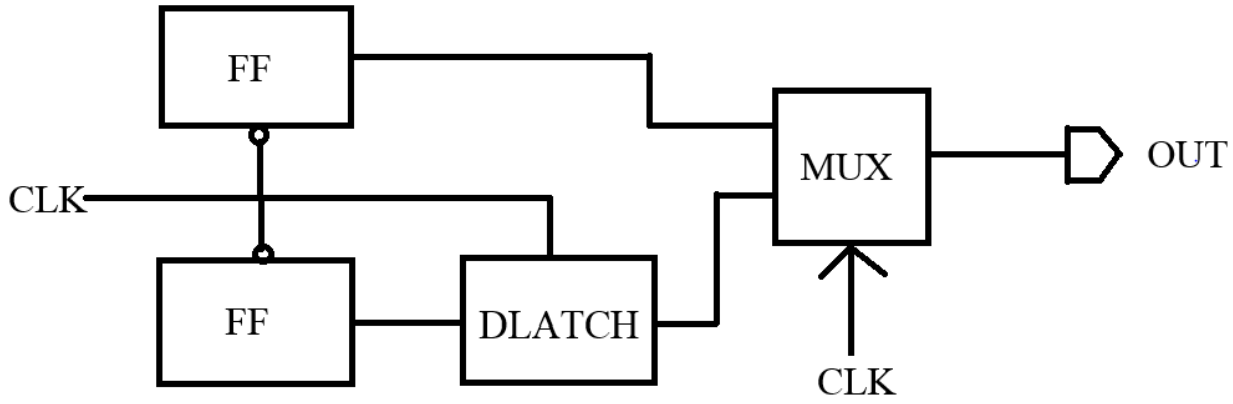

Figure 4.1: 2-to-1 Serializer Block Diagram

First, the 8-bit data stream is generated by an on-chip programmable data generator. A simplified diagram of the 8-to-1 Serializer will be shown in figure 2. The parallel input data is latched by eight flip-flops that are clocked at eighth rate because there are four 2-to-1 Serializer at the first level of the 8-to-1 Serializer. And each 2-to-1 Serializer has two flip-flops as the figure 1 above shows. Four latches, running on the opposite phase of the eighth-rate clock are used to delay four of the data bits by half a clock period next. Four 2-to-1 Serializers will convert the 8-bit wide bus operating at eighth-rate into a 4-bit wide bus operating at quarter rate. The process is repeated using two 2-to-1 Serializers, two latches, and the quarter-rate clock to create a 2-bit wide bus operating at half-rate. [7]

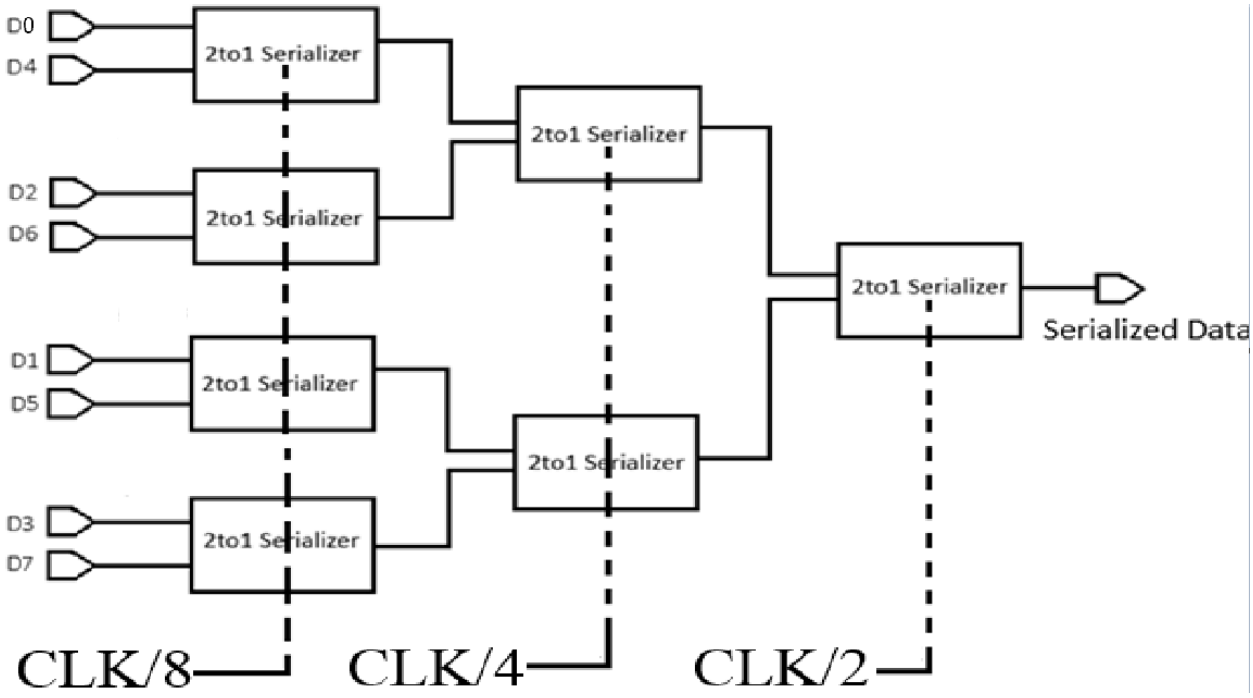

Figure 4.2: 8-to-1 Serializer Block Diagram in Binary Tree Form

Figure 2 shows that the 8 to 1 serializer employs seven 2to1 Serializers constructed as the figure shows above.

The schematic of 2-to-1 Serializer in Cadence Virtuoso is shown below,

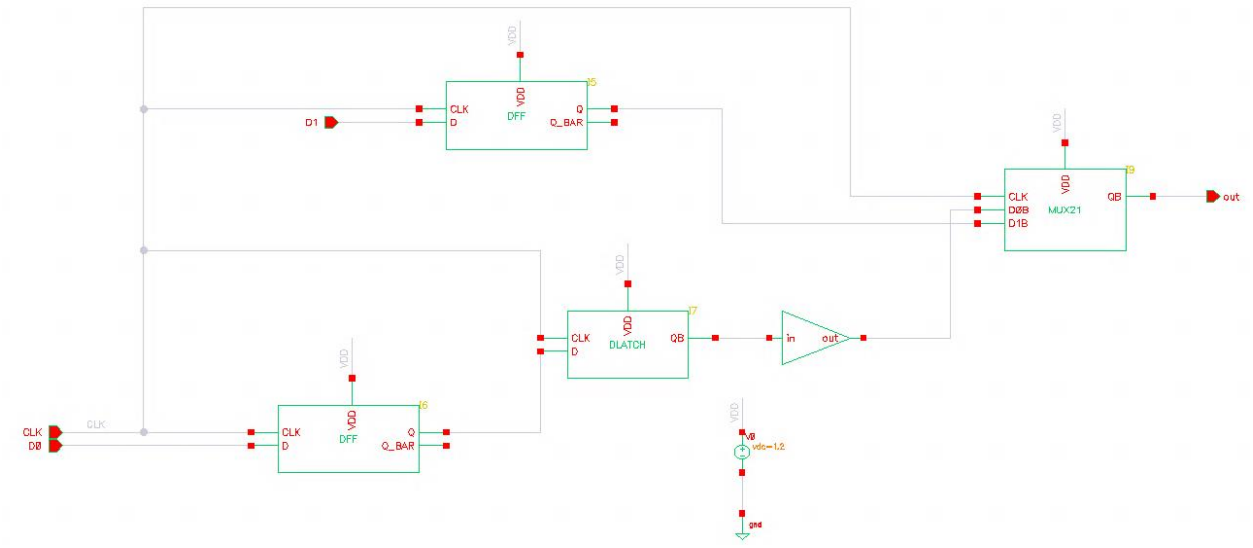

Figure 4.3: 2-to-1 Serializer Schematic in Virtuoso Cadence

The schematic of 4-to-1 Serializer in Cadence Virtuoso is shown below,

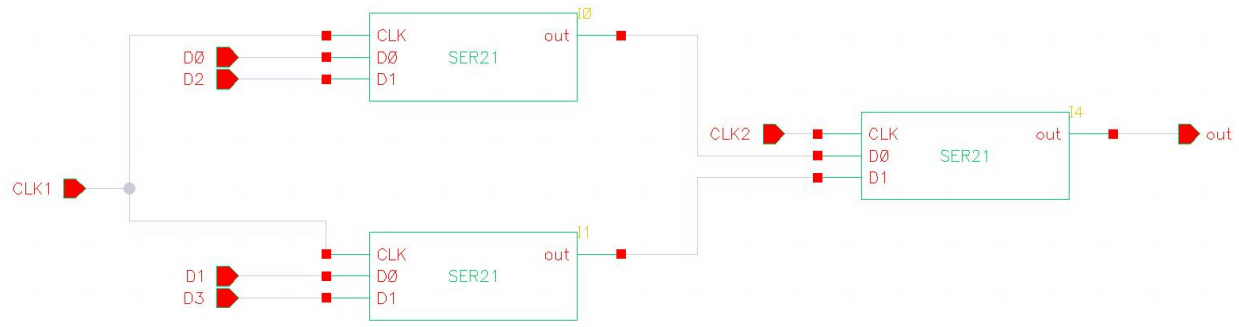

Figure 4.4: 4-to-1 Serializer Schematic in Binary Tree Form in Virtuoso Cadence

The schematic of 8-to-1 Serializer in Cadence Virtuoso is shown below,

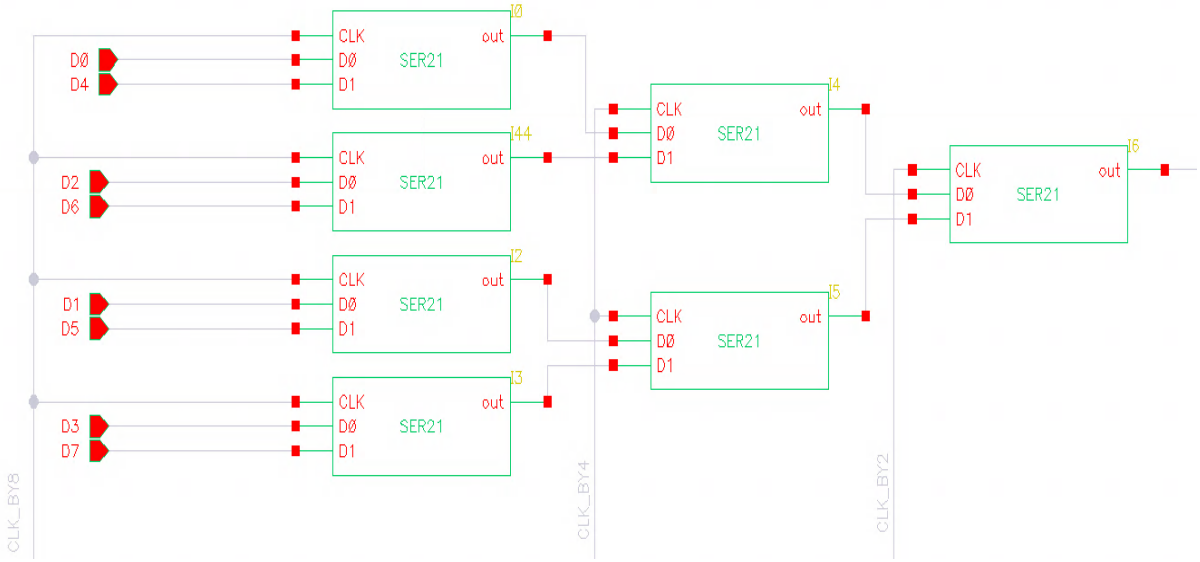

Figure 4.5: 8-to-1 Serializer Schematic in Binary Tree Form in Virtuoso Cadence

The last stage of 2-to-1 Serializer is directly connected to the 1-to-8 Deserializer which will be described later in this thesis.

4.1.1 Design of Negative Edge Triggered D Flip-flop

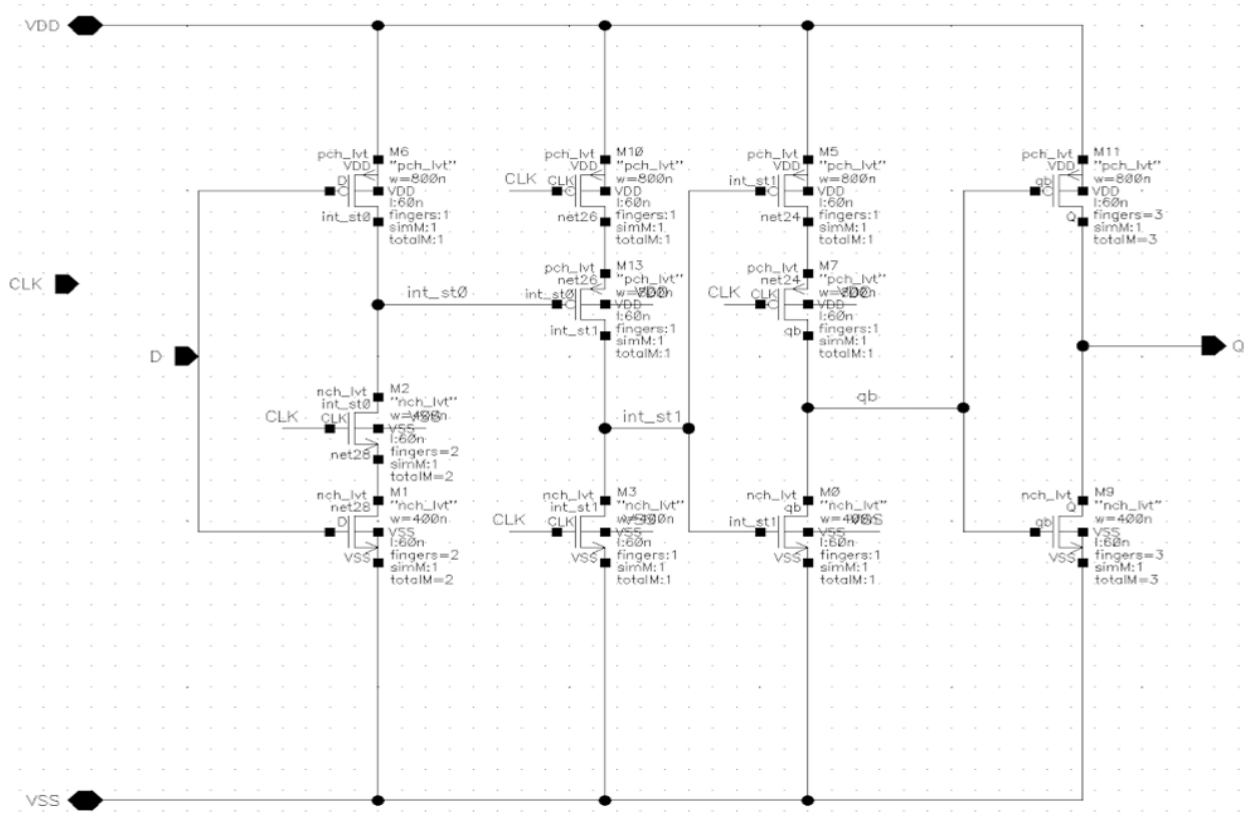

Figure 4.6: Negative Edge Triggered D Flip-flop Schematic in Virtuoso Cadence

For a negative edge triggered D flip-flop, as the name suggest, the falling edge of the clock is the triggering edge. After triggering, D will be propagated to the output. Without triggering, the output will stay the same without changing.

4.1.2 Design of Positive D Latch

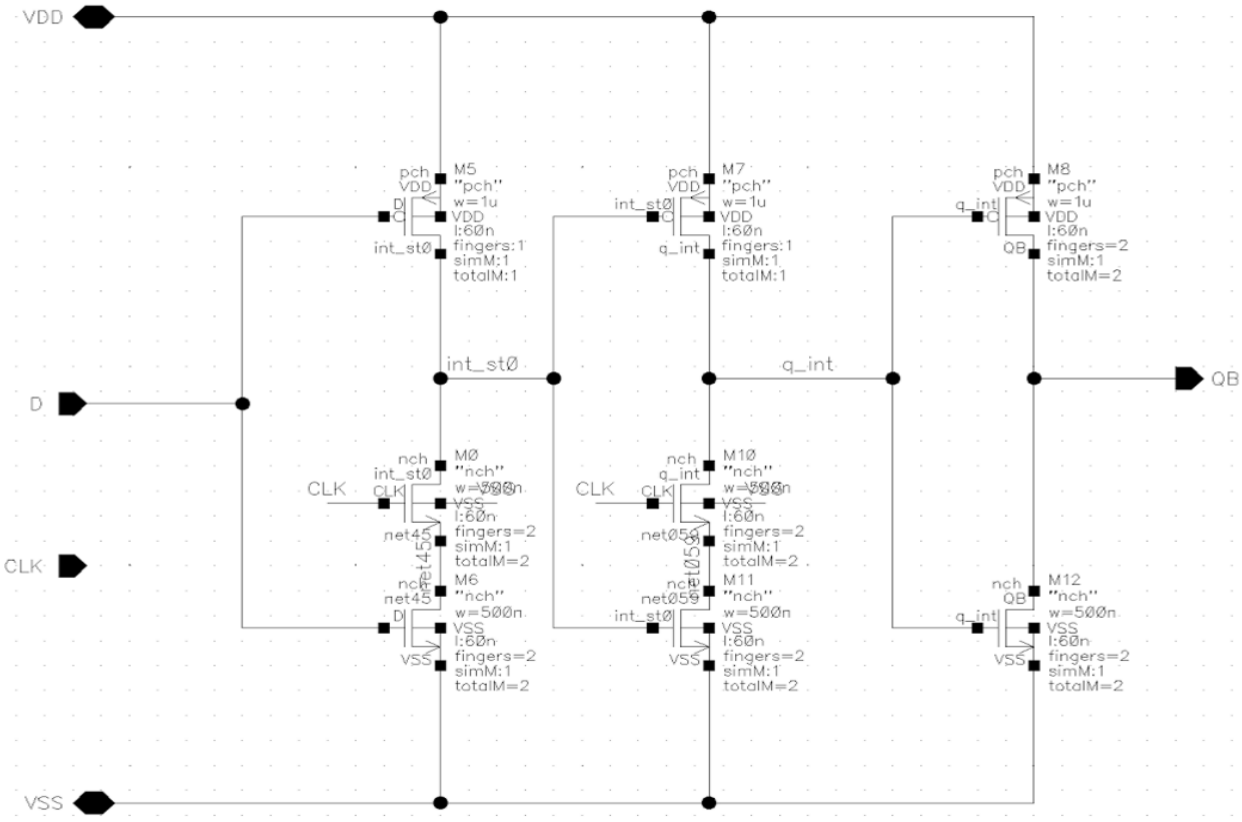

Figure 4.7: Positive latch (with inverter at last stage) Schematic in Virtuoso Cadence

For a positive latch, when CLK is high, the latch is in the transparent mode and corresponds to two cascaded inverters. So D will be simply propagated to q_int. On the other hand, when CLK is low, both inverters are disabled. So the latch is in hold mode and D will not be propagated. Because at this time, M0 and M10 are deactivated.

4.1.3 Design of 2-to-1 Mux

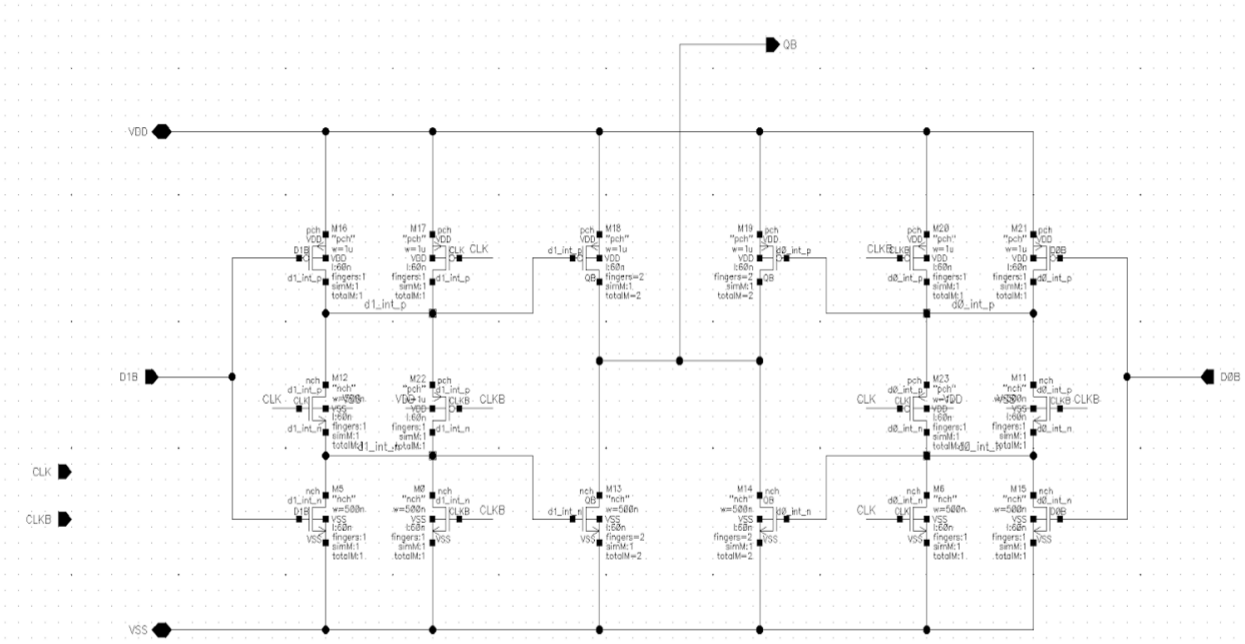

Figure 4.8: 2-to-1 Mux Schematic in Virtuoso Cadence

As the schematic shown, when CLK is high, M12, M17, M22 and M0 transistors are activated. Thus, D1B could be propagated to the output. While when CLK is low and CLKB is high, M11, M20, M23 and M6 transistors are to be activated. Hence, D0B will be propagated to the output. So CLK signal here act as the selecting bit, when CLK is low, D0B will be selected. While when CLK signal is high, D1B will be selected.

4.2 Deserializer

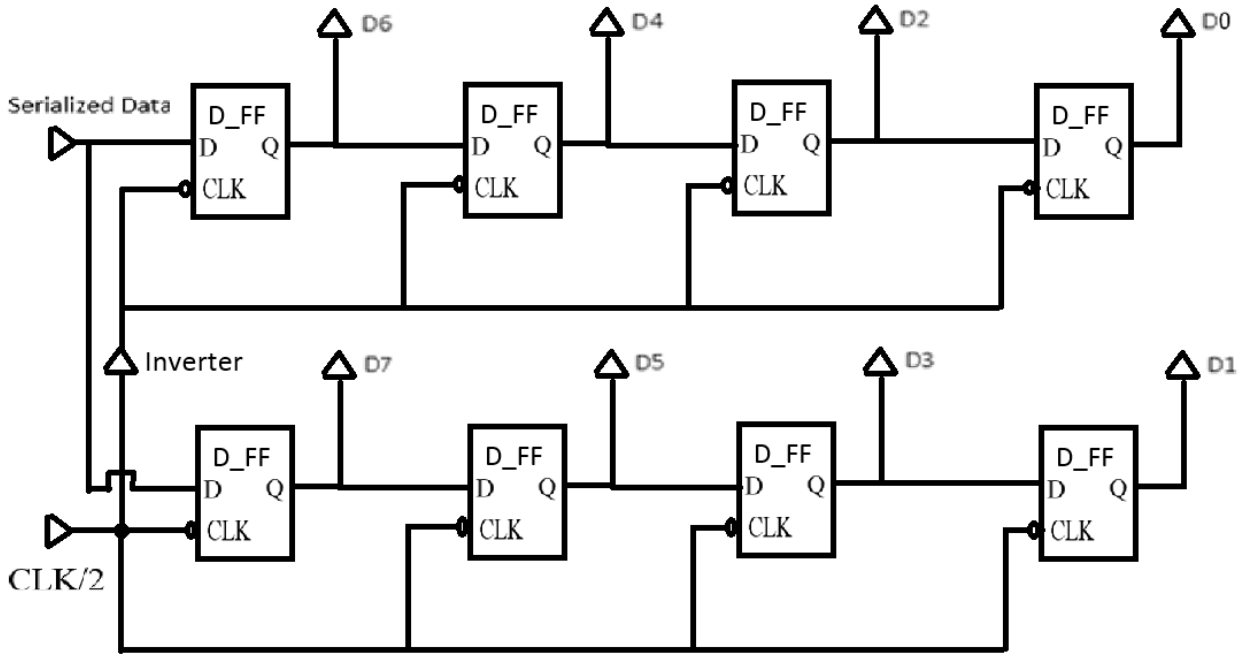

Figure 4.9: The 1-to-8 Deserializer Block Diagram

As the figure above shows, the Deserializer employs negative edge triggered flop-flop. These flip flops are implemented using clock overlap insensitive clocked CMOS registers as discussed above. The Deserializer samples input on both edges of clock which is basically right shift data whenever there is a rising or a falling edge clock. And only half rate clock 1.6 GHz is employed in this design. So when the CLK is at the rising edge, D7 will be selected. And then when CLK is at falling edge, D6 will be selected next. After all, D7 to D0 will all be selected in order.

The schematic of Deserializer in Cadence Virtuoso is shown below with the delay circuit which will be described later,

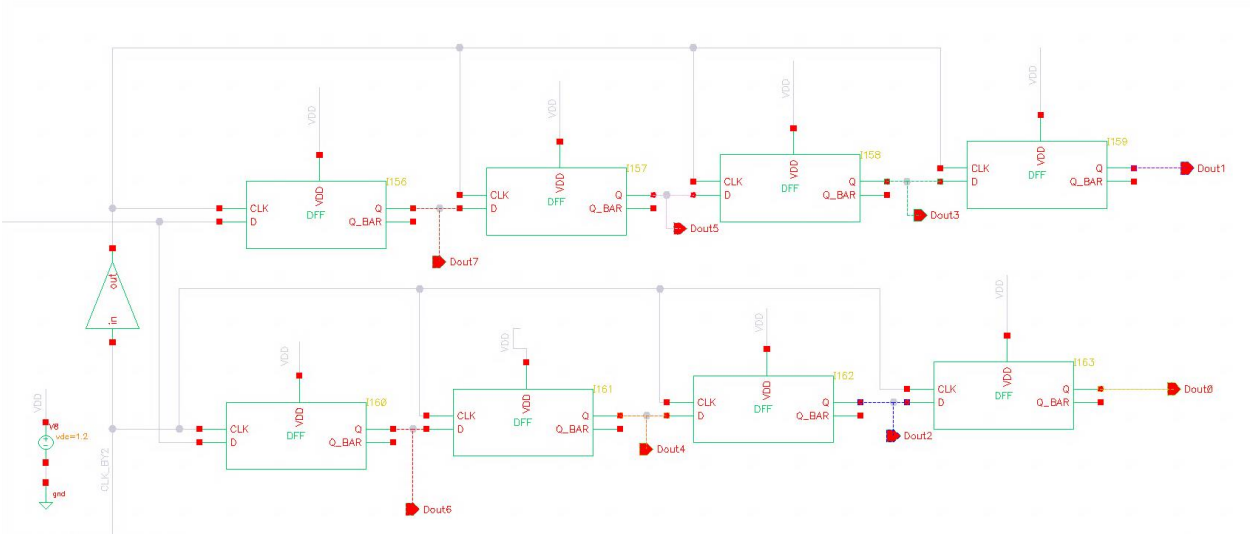

Figure 4.10: 1-to-8 Deserializer Schematic in Virtuoso Cadence

4.3 Clock Divide Circuit Block

In the SERDES design, three clock frequencies are required for the Serializer and Deserializer to work properly. Asynchronous divider as asynchronous divider could reduce high frequency clock loading. Furthermore, because each stage runs at lower frequency, the power consumption is lowered. While the disadvantage is the jitter accumulation as jitter accumulates with the clock-to-Q delays through the divider. While the disadvantage is it is slowed down by stacked PMOS, signals goes through three gates per cycle. And it requires full swing input clock signal. The divider circuit consists of 3-DFFs that are connected together shown in Figure below to realize a divide-by-8 operation.

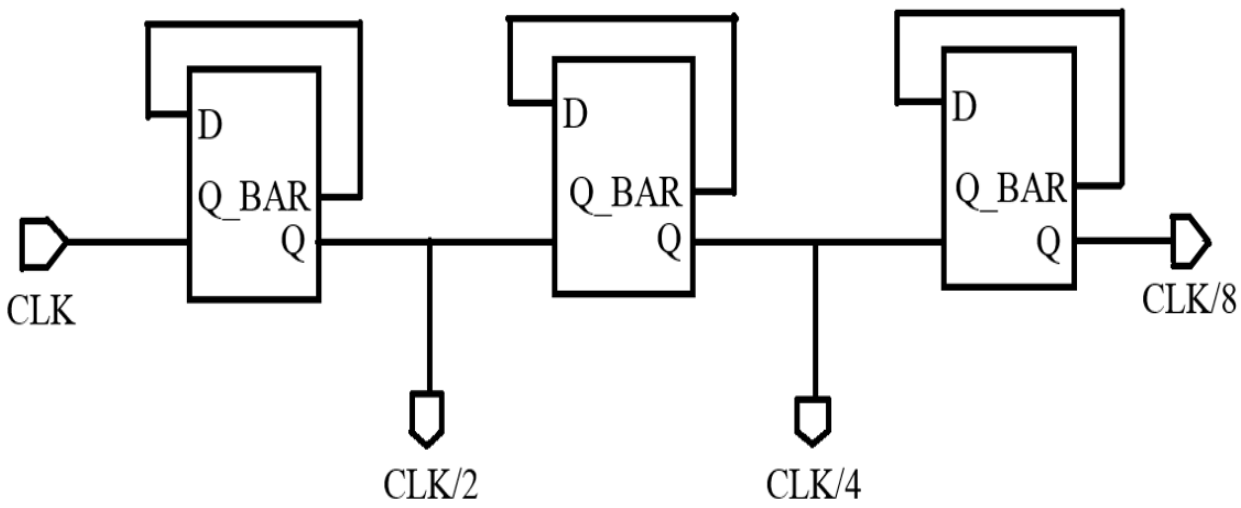

Figure 4.11: Divide-by-8 Clock Circuit Block Diagram

The schematic of the asynchronous divider in the SerDes design is as follows.

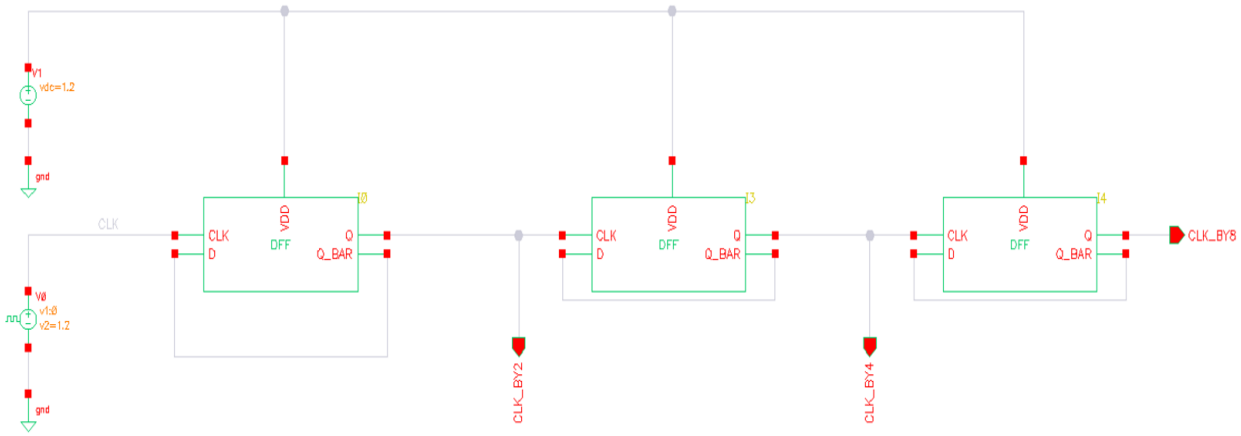

Figure 4.12: Divide-by-8 Clock Circuit Schematic in Virtuoso Cadence

Figure below shows the testing waveform of the asynchronous divider in the SerDes design. The input signal CLK is a 3.2 GHz clock. CLK_BY2 is the clock with frequency of 1.6 GHz, CLK_BY4 is the clock with the frequency of 0.8 GHz, and CLK_BY8 is the clock with frequency of 0.4 GHz. After all, half-rate, quarter-rate and eighth rate clocks are generated and distributed to the appropriate places with the

buffers which will be introduced in the later section.

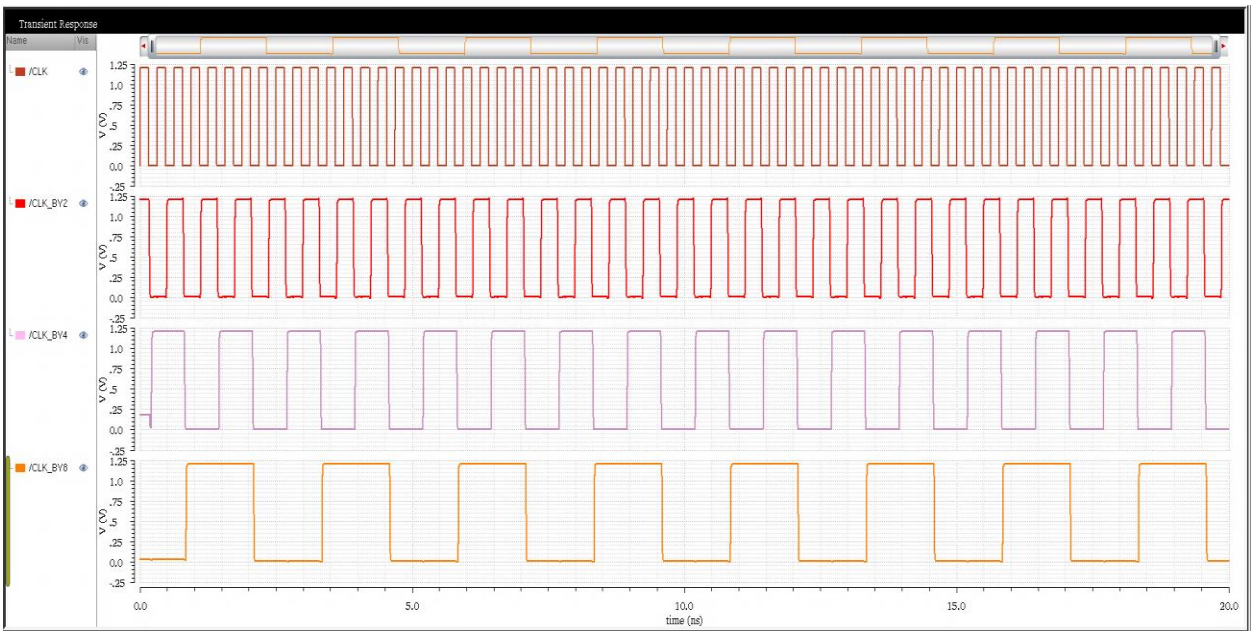

Figure 4.13: Divide-by-8 Clock Circuit Test Waveform Result in Virtuoso Cadence

4.4 Delay Circuit Block

Adding necessary delay circuits are crucial for letting sampling clock's rising and falling edge be at the same time when data streams are in transitions as discussed at Section 4.1: Design of Serializer. Furthermore, the 2-to-1 Mux in every 2-to-1 Serializer has a Mux which has three inputs, the sampling clock, D1 and D0. To avoid the race condition of three inputs, the sampling clock is delayed inside the 2-to-1 Serializer to make the sampling clock's rising and falling edges, the transitions of D1 and D0 not happen in the same time or too closely. In other words, the sampling clock's rising and falling edge should be in the center of the eye diagram formed by the data streams D0 and D1. In this thesis, inverters are used to make buffers shown below to make delays to let the Serializer/Deserializer function properly.

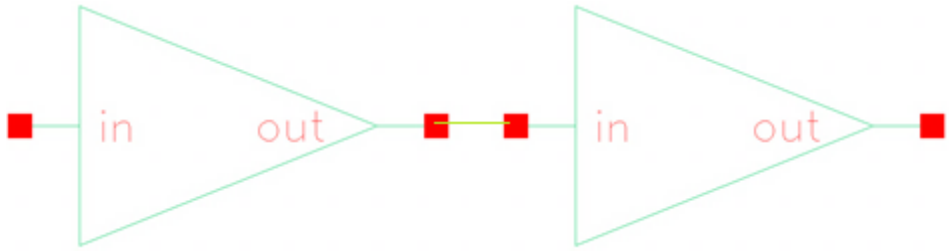

Figure 4.14: Buffer Symbol in Virtuoso Cadence

Figure above shows the inverter symbol used in test bench to let the clock edges could be synchronized with the data stream.

Delays are observed from the simulations of the eye diagrams formed by the D1, D0 data streams and the clock. To have the proper delay, the clock rising edge and falling edge should be at the center of both eyes of data streams. The figure below shows the delayed clock's rising edge and falling edge (red line) properly sit in the middle of the two transitions. With eye diagram simulating and adding proper delays, the 2-to-1 Serializer timing would be optimal for the proper function.

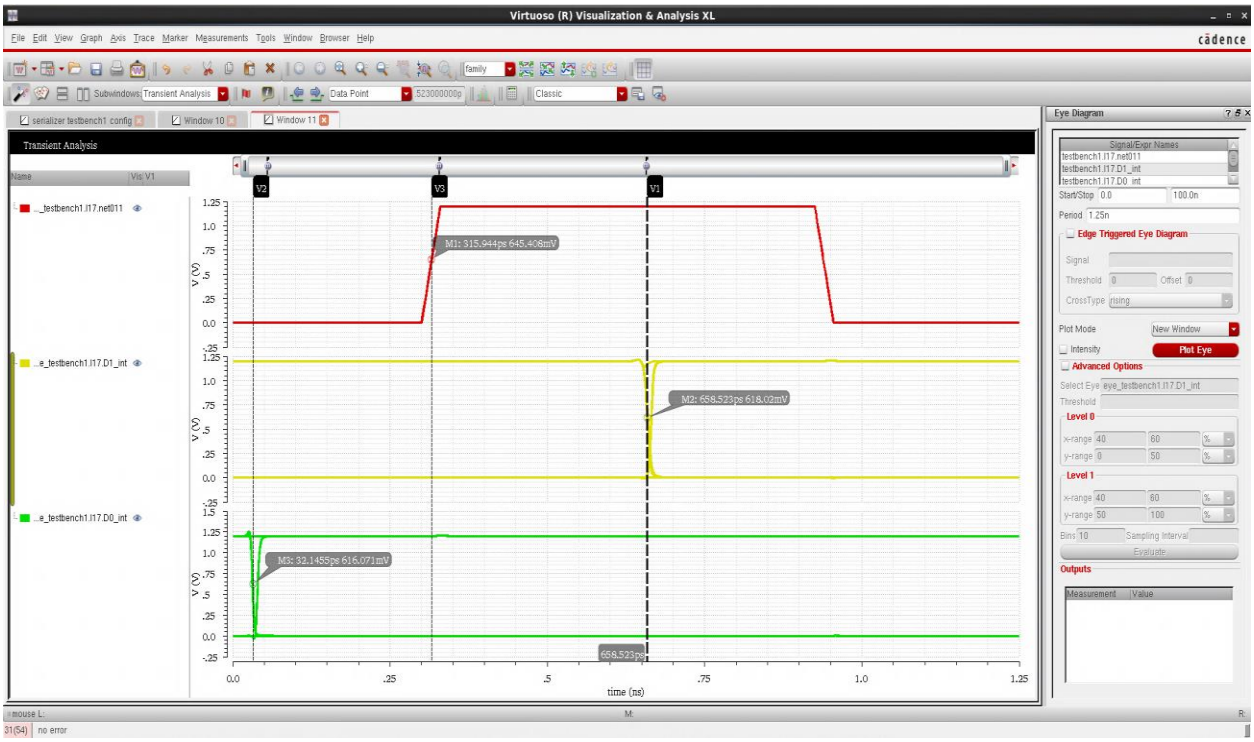

Figure 4.15: Eye Diagram of D1(Yellow Line), D0(Green Line) and Mux Clk (Red Line)'s Eye Diagram showing that 2-to-1 Serializer Timing is Optimal

After adding proper delay for the 2-to-1 Serializer, 4-to-2 and 8-to-4 Serializers should also be added the proper delay to let their outputs' clocks rising and falling edges be the same with the data stream's transitions time. The following figures will discuss this more in details.

Figure 4.16: Eye Diagram of the Outputs of 4-to-2 Serializer and the Sampling Clock of Last Stage 2-to-1 Serializer *(Red Line: Output [0] of 4-to-2 Serializer, Yellow Line: Output [1] of 4-to-2 Serializer, Green Line: Sampling Clock of the Last Stage 2-to-1 Serializer)*

As the above figure shown, to let the sampling clock and the two output of 4-to-2 Serializer arrive at the last stage 2-to-1 Serializer, 678 ps – 638 ps = 40 ps delay is needed here.

Figure 4.17: Eye Diagram of the Outputs of 8-to-4 Serializer and the Sampling Clock of Next Stage 4-to-2 Serializer *(Red Line: Output [0] of 8-to-4 Serializer, Yellow Line: Output [1] of 8-to-4 Serializer, Green Line: Output [2] of 8-to-4 Serializer, Light Blue: Output [3] of 8-to-4 Serializer, Sampling Clock of the Last Stage 2-to-1 Serializer)*

As the above figure shown, to let the sampling clock and the four output of 8-to-4 Serializer arrive at the next stage 4-to-2 Serializer, 52 ps – 17 ps = 35 ps delay is needed here.

And same rules apply for the Deserializer. The following figure shows that the Deserializer's driving clock's rising and falling edges are delayed to avoid the eight data streams transitions.

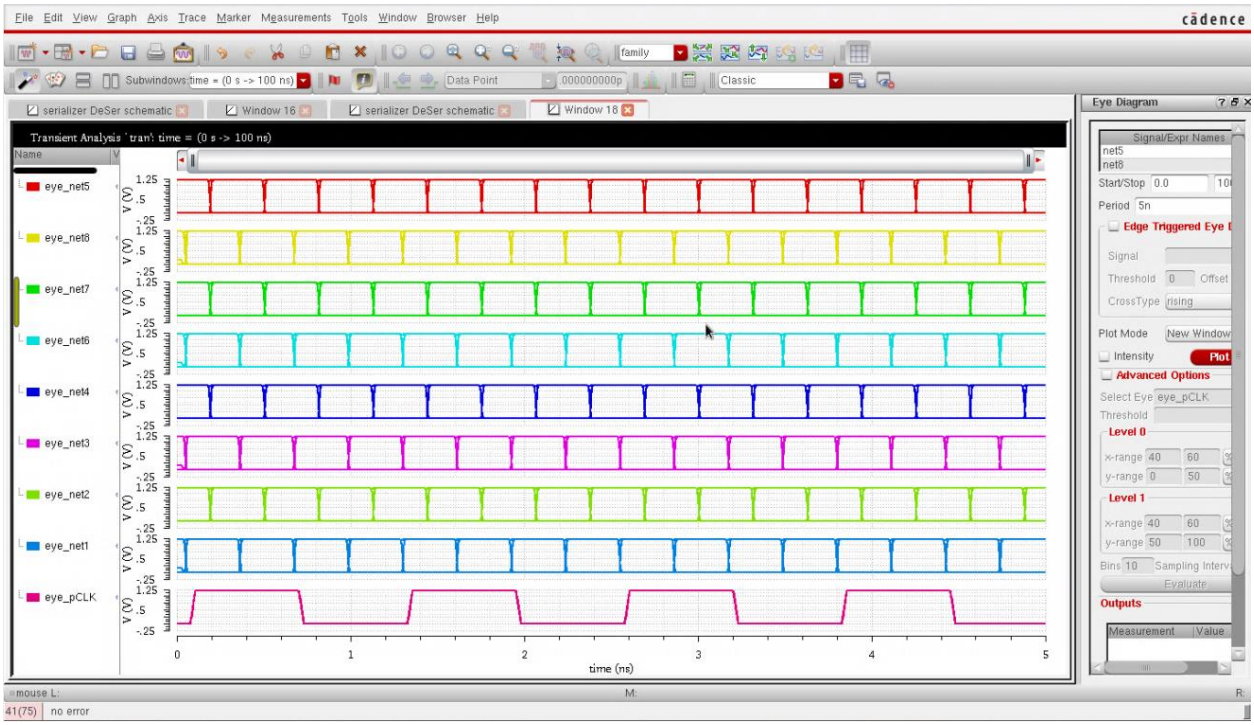

Figure 4.18: Eye diagram of Deserializer (Data streams are the top eight ones; the sampling clock is the last one)

5. Transistor-Level Simulations

With Deserializer's serial data bit line Connected directly to the Serializer's serial data output line, the simulation result in Virtuoso Cadence could be seen below,

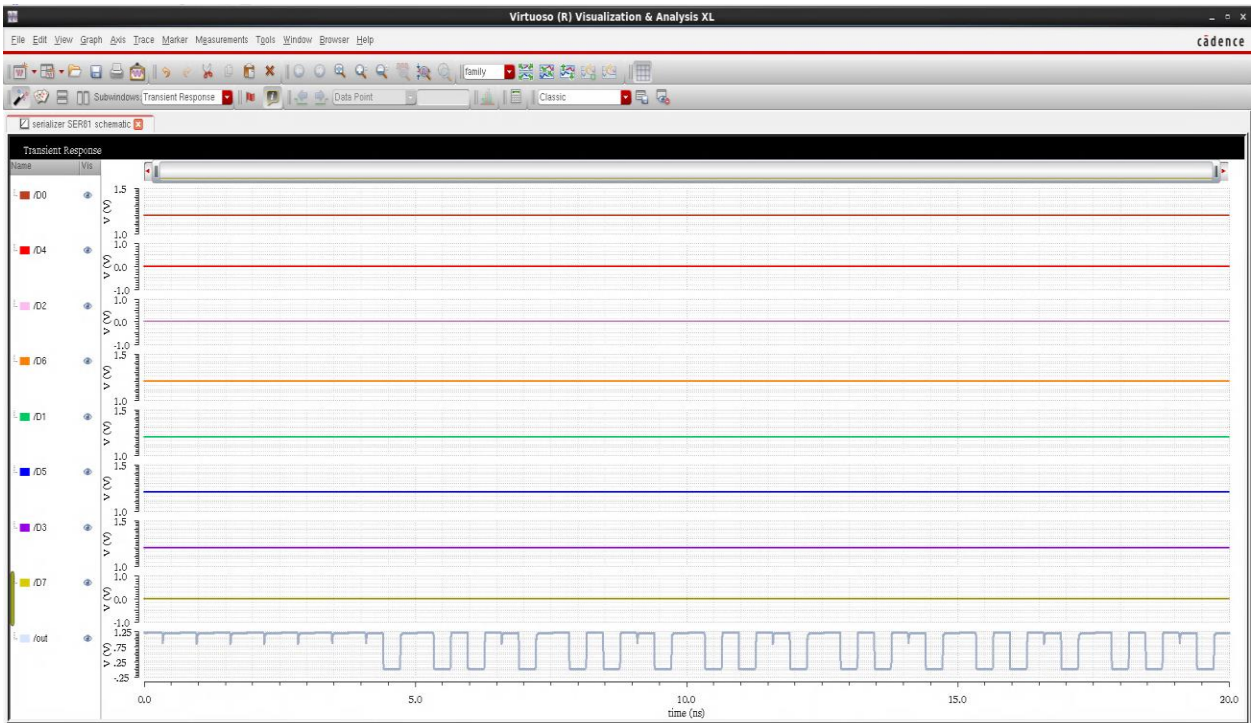

Figure 5.1: Simulation Result of the Proposed Serializer for Parallel Data Input “11010110”

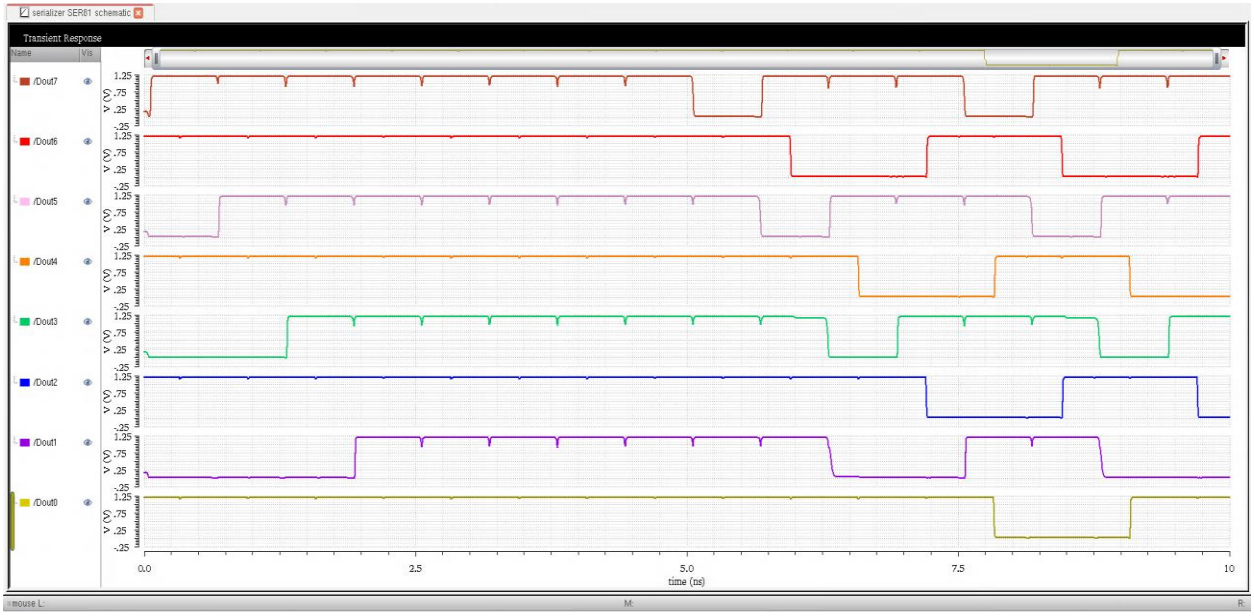

Figure 5.2: Simulation Result of the Proposed Deserializer for Serial Data Input “11010110”

6. Conclusion

The increasing trends in SoCs and SiPs technologies demand integration of large numbers of buses and metal tracks for interconnections. High-speed serial links outperform parallel links especially for long-range communication. So SERDES Transceiver is a promising solution which can also reduce the number of interconnects and offers significant benefits in context with less power consumption, routing congestion, area and crosstalk noises. This paper reports a design of Serializer and Deserializer architecture for basic functional operations of serialization and deserialization used in SERDES Transceiver. Deserializer architecture employs a design technique which samples input on both edges of clock. The main advantage of this technique which is input sampled with both edges of a clock, which results in power savings in the clock distribution network. This proposed Serializer and Deserializer architecture is designed using 65 nm CMOS technology and simulation is done using Cadence Spectre simulator with a supply voltage of 1.2 V.

References

[1] R. Ratan, "Design of a Phase Locked Loop Based Clocking Circuit for High Speed Serial Link Applications," 2014.

[2] R. Dobkin, A. Morgenshtein, A. Kolodny, R. Ginosar. (2007) Parallel vs. Serial On-Chip Communication, CCIT TR674, EE Pub No. 1631, EE Department, Technion

[3] Alser, M.H. and Assad, M.M. (2011) Design and Modeling of Low-Power Clockless Serial Link for Data Communication Systems. *National Postgraduate Conference (NPC), Kuala Lumpur, 19-20 September 2011, 1-5.* http://dx.doi.org/10.1109/NatPC.2011.6136441

[4] S. Ravikumar, "Circuit Architectures for High Speed CMOS Clock and Data Recovery Circuits," 2015

[5] Jaiswal, N. and Gamad, R. (2015) Design of a New Serializer and Deserializer Architecture for On-Chip SerDes Transceivers. Circuits and Systems, 6, 81-92. http://dx.doi.org/10.4236/cs.2015.63009

[6] S. Palermo, CMOS Nanoelectronics Analog and RF VLSI Circuits, Chapter 9. New York City, N.Y. : McGraw-Hill, 2011

[7] Philpott, R.A., Humble, J.S., Kertis, R.A., Fritz, K.E., Gilbert, B.K. and Daniel, E.S. (2008) A 20 Gb/s SerDes Transmitter with Adjustable Source Impedance and 4-Tap Feed-Forward Equalization in 65nm Bulk CMOS. *IEEE Custom Integrated Circuits Conference (CICC)*, San Jose, 21-24 September 2008, 623-626