

Design and Verification of High Speed IO Transmitter in 14nm Technology

K Nagaraju¹ and K Rajagopal²

¹M.Tech Student, CVR College of Engineering/ECE Department, Hyderabad, India.

Email: nraj055@gmail.com

²Asst. Professor, CVR College of Engineering/ECE Department, Hyderabad, India.

Email:@ rajgopalsushma@gmail.com

Abstract: For many years, parallel I/O schemes ruled the chip to chip, board to board or backplane communication. Parallel I/O had to experience performance issues like crosstalk, signal integrity and increased skew after passing certain I/O clock-frequency rates [1]. Also parallel I/O methods increased the complexity of the hardware (high pin count, more wires) and made bandwidth sharing inevitable. In many new communication protocols serial data transmission has become very common due to low pin count (reduced cost). Serial I/O methods can also transmit at much higher clock rates per bit transmitted, thus outweighing the parallel transmission method. High speed serialization with large bandwidth plays a major role in transmitter of high speed interfacing circuits such as PCIe, USB, and SATA. Various encoding schemes are used based on the protocols. Efficient equalizers with interface units are required to avoid ISI (Inter Symbol Interference) and to drive the back-panel line. This also makes the design complex, which in turn makes the design verification and validation more challenging.

Index Terms: PCIe, Serial I/O, USB, SATA

I. INTRODUCTION

High-speed serial I/O standards require only a half or less pin and wire count for the transmissions unlike the parallel transmission schemes. Surprisingly serial transmission methods, which are being widely used today in many new communication protocols, are able to transfer data at rates higher than 8 Gb/s. In high-speed serial transmission, clock and data are combined in a single stream thus reducing the problem of bit-to-bit skew. High-speed serial I/O standards span in more than one communication scenario like, Fibre channel, InfiBand and Gigabit & 10-Gigabit Ethernet etc. Dramatic increases in processing power, fueled by a combination of integrated circuit scaling and shifts in computer architectures from single-core to a possible many-core systems. This has rapidly scaled on-chip aggregate bandwidths into the Tb/s range [1], necessitating a corresponding increase in the amount of data communicated between chips not to limit overall system performance [2]. Due to the limited I/o pin constraints, high-speed serial link technology is employed for this inter-chip communication.

A. PCI Express overview

High-speed point-to-point electrical link systems employ specialized I/O circuitry that perform incident wave signaling over carefully designed controlled-impedance channels in order to achieve high data rates. General block view of High speed IO (Figure 1) divided into three sections namely 'Transmitter', 'Clocking' and 'Receiver' sections.

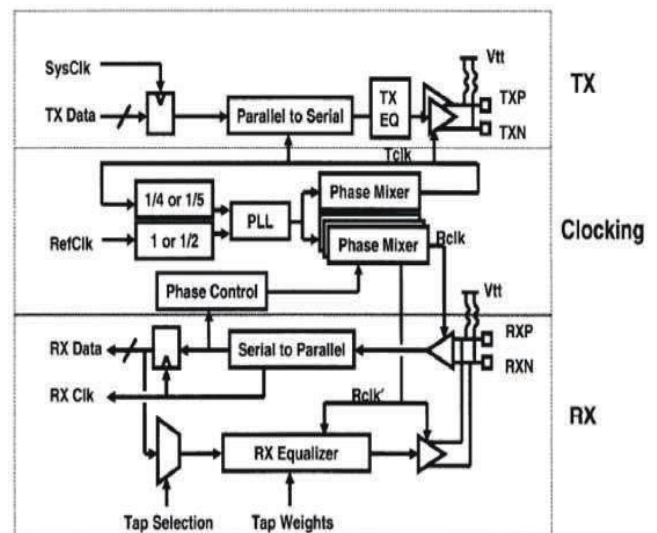


Figure 1. Block diagram of High Speed IO

Transmitter section contains encoding mechanism based on the protocol implementation, parallel to serial conversion (serializer), transmitter equalization circuitry and differential driver circuitry.

Clocking section is to control both transmitter and receiver contains phase lock loop, mixers, clock dividers and clock generator. This provides necessary clock required for transmitter and receiver sections.

Receiver section contains clock to data recovery, receiver equalizer circuit, decision feedback, error detection and correction, serial to parallel conversion (deserializer) and decoding scheme.

B. Electrical link system

Figure 2 shows the major components of a typical high-speed electrical link system. Due to the limited number of high-speed I/O pins in chip packages and printed circuit board (PCB) wiring constraints, a high-bandwidth transmitter serializes parallel input data for transmission. Differential low-swing signaling is commonly employed for common-mode noise rejection and reduced crosstalk due to the inherent signal current return path [7]. At the receiver, the incoming signal is sampled, regenerated to CMOS values and deserialized. The high-frequency clocks which synchronize the data transfer onto the channel are generated by a frequency synthesis phase-locked loop (PLL) at the transmitter, while at the receiver the sampling clocks are aligned to the incoming data stream by a timing recovery system.

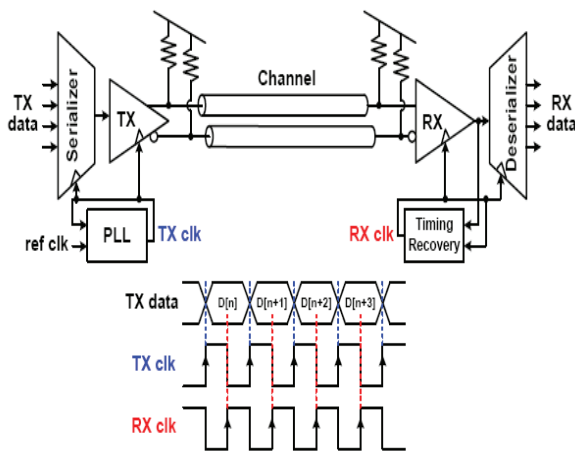


Figure 2. High speed electrical link system

C. Transmitter system

The transmitter must generate an accurate voltage swing on the channel while also maintaining proper output impedance in order to attenuate any channel-induced reflections. Either current or voltage-mode drivers, shown in Figure 3, are suitable output stages. Current-mode drivers typically steer current close to 20mA between the differential channel lines in order to launch a bipolar voltage swing on the order of $\pm 500\text{mV}$. Driver output impedance is maintained through termination which is in parallel with the high-impedance current switch. While current-mode drivers are most commonly implemented [8], the power associated with the required output voltage for proper transistor output impedance and the “wasted” current in the parallel termination led designers to consider voltage-mode drivers. These drivers use a regulated output stage to supply a fixed output swing on the channel through a series termination which is feedback controlled [9]. While the feedback impedance control is not as simple as parallel termination, the voltage-mode drivers have the potential to supply an equal receiver voltage swing at a quarter [7] of the common 20mA cost of current-mode drivers.

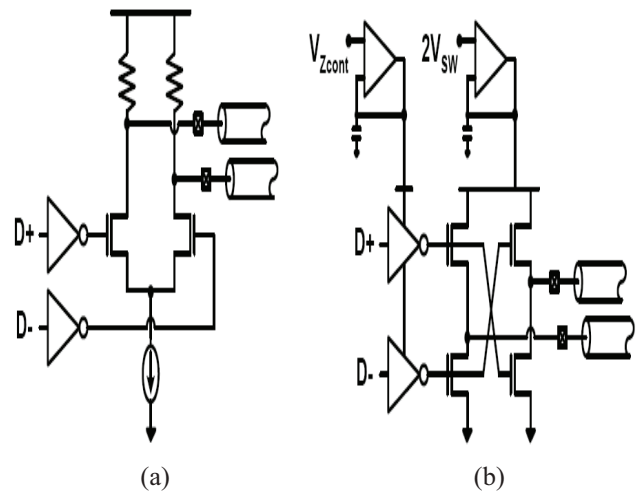


Figure 3. Transmitter output stages:

(a) current-mode driver, (b) voltage-mode driver.

D. Receiver system

Figure 4 shows a high-speed receiver which compares the incoming data to a threshold and amplifies the signal to a CMOS value. This highlights a major advantage of binary differential signaling, where this threshold is inherent, whereas single-ended signaling requires careful threshold generation to account for variations in signal amplitude, loss and noise [11]. The bulk of the signal amplification is often performed with a positive feedback latch [12,13]. These latches are more power-efficient versus cascaded linear amplification stages since they don’t dissipate DC current. While regenerative latches are the most power-efficient input amplifiers, link designers have used a small number of linear pre-amplification stages to implement equalization filters that offset channel loss faced by high data rate signals [14,15].

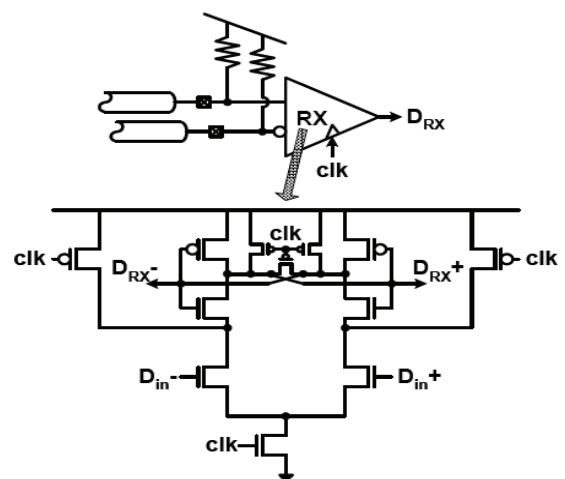


Figure 4. Receiver input stage with regenerative latch

One issue with these latches is that they require time to reset or “pre-charge”. Thus to achieve high data rates, often multiple latches are placed in parallel at the input and

activated with multiple clock phases spaced a bit period apart in a time-division-demultiplexing manner (Figure 5). This technique is also applicable at the transmitter, where the maximum serialized data rate is set by the clocks switching the multiplexer. The use of multiple clock phases offset in time by a bit period can overcome the intrinsic gate-speed which limits the maximum clock rate that can be efficiently distributed to a period of 6-8 fanout-of-four (FO4) clock buffer delays [16,17], shown in [18]

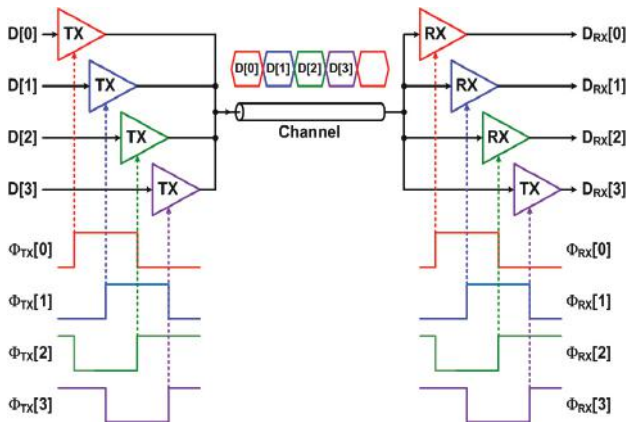


Figure 5. Time Division-Multiplexing link

II. PCIe TRANSMITTER BLOCK OVERVIEW

Figure 6 shows the block diagram of PCIe transmitter, depends on the input data PCLK will varies as stated in above section. Depending on the PCIe version encoding scheme varies as shown in table 1. In order to transfer data over a high speed serial interface, data is encoded prior to transmission and decoded upon reception. The encoding process ensures that sufficient clock information is present in the serial data stream to allow the receiver to synchronize to the embedded clock information and successfully recover the data at the required error rate. In addition, the 8b/10b or 128b/130b encoding improves the line characteristics, enabling long transmission distance and more effective error-detection at the receiver.

TABLE I
ENCODING SCHEMES IN PCIe

PCIe mode	Encoding scheme
Gen1	8b/10b
Gen2	8b/10b
Gen3	128b/130b
Gen4	128b/130b

Using the 8b/10b encoding algorithm adds 20% overhead to each character so the effective data-rate of PCIe Gen 1 becomes $(1-0.20) \times 2.5 = 2$ Gbps and for Gen 2 becomes $0.8 \times 5 = 4$ Gbps similarly by using 128b/130b encoding algorithm in Gen 3 and Gen 4 adds 1.538% overhead so the effective data of Gen 3 is $0.985 \times 8 = 7.88$ Gbps, for Gen 4 is $0.985 \times 16 = 15.76$ Gbps as shown in table 2.

TABLE II
ENCODING SCHEMES IN PCIe

PCIe modes	Operating frequency	Data rate	Bandwidth
Gen 1	2.5 GHz	2.5 Gbps	2 Gbps
Gen 2	5 GHz	5 Gbps	4 Gbps
Gen 3	8 GHz	8 Gbps	~8 Gbps
Gen 4	16 GHz	16 Gbps	~16 Gbps

Parallel to serial conversion is done by using multiple stages in order to perform high speed operations without any data error in the serialized output, because using traditional FSM, built with multiple flops in the path will increase the overall path delay and may lead to race conditions with high speed input clock, leading to setup and hold violations.

In order to receive and recover the data coming from noise channel at receiver end, output at the transmitter end has to be differential so that channel noise can be compensated. Hence differential drivers are required to convert single ended data to differential data.

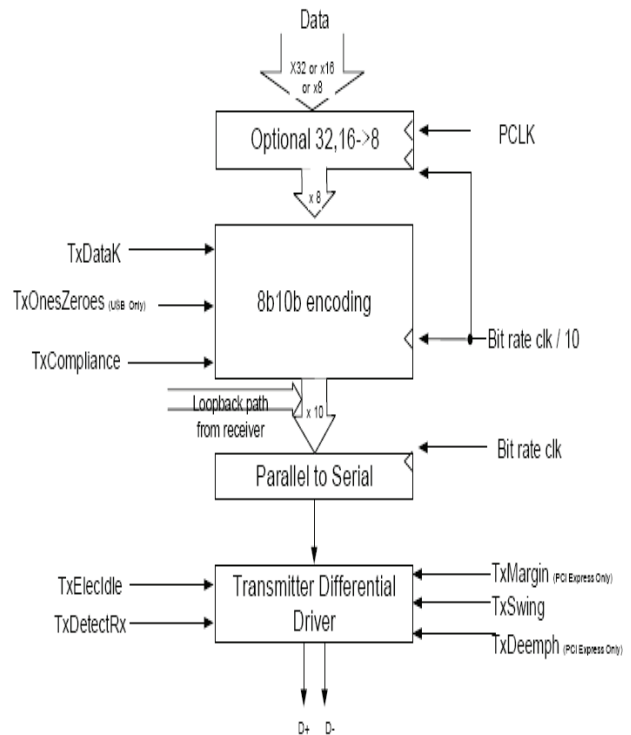


Figure 6. Transmitter Block diagram

III. CADENCE CONFORMAL

A systematic method that uses mathematical proof to check the design's properties is formally known as Logic Equivalence Checking (LEC). Verification starts writing systemverilog/verilog coding to mimic the behavior of

circuitry and compares the schematic with systemverilog code. Some of the advantages of Formal verification are:

- Enables "White box" verification
This provides full controllability and visibility on internal structure
- No test vectors required
 - No input vector creation necessary
 - No test vectors for logical function
 - Testbench setup is not required
 - Easy debugging
- Exhaustive verification
- Find bugs earlier

This will help in finding bugs as early as possible and also in the progress of work, target a class of bugs typically found with simulation late in the design cycle

- Speed-
Formal is magnitudes faster than simulation

It verifies the logical equivalence of RTL, gate or transistor level netlists to each other. But does not guarantee that the initial design meet the design specification. It also ignores timing information and performs only boolean equivalence checks.

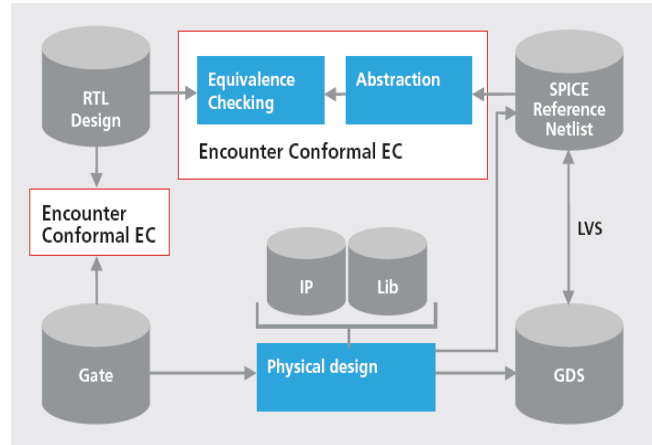
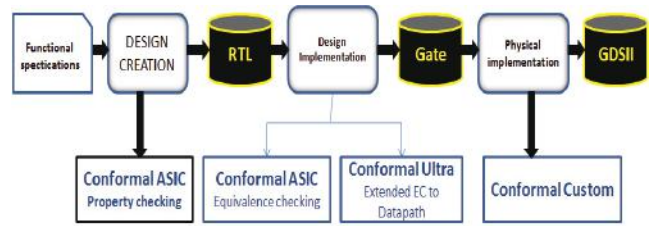


Figure 8. Cadence Conformal in design flow

Figure 7 shows the LEC Flow in the design cycle, LEC can be performed between RTL to RTL or RTL to pre-layout netlist or RTL to post-layout netlist or netlist to netlist. Figure 8 shows cadence conformal flow in ASIC design flow.

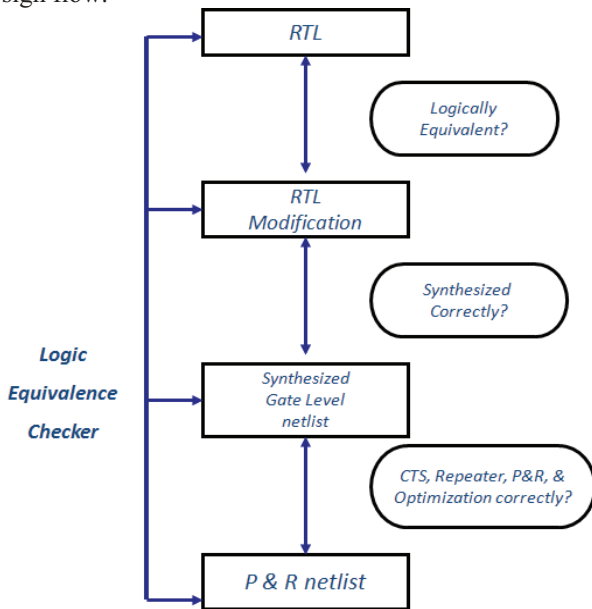


Figure 7. LEC in design flow

Figure 9 shows the LEC tool flow, it contains two modes namely, 'setup' and 'LEC', all the reading of systemverilog/verilog code, netlist is done in setup mode and mapping the instance to match the functionality, generation, comparison and debugging is done in LEC mode. This process repeats until schematic meets the functional/logical equivalence

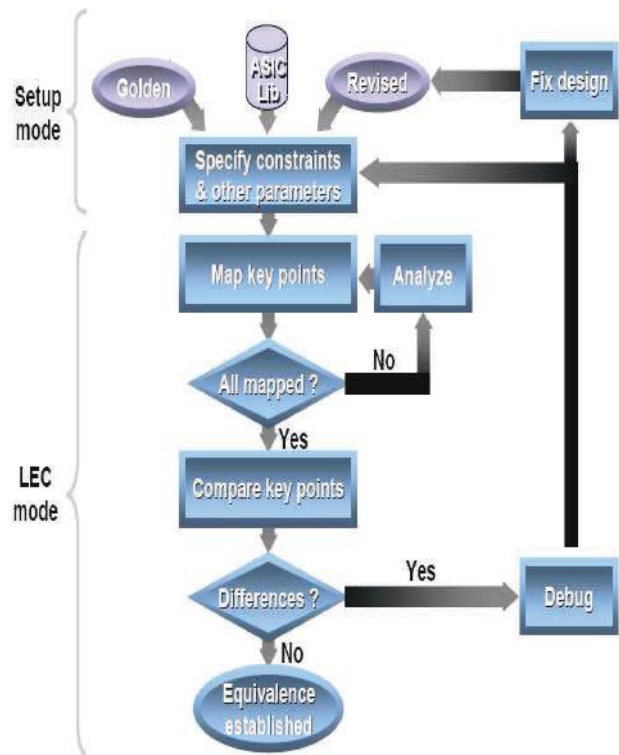


Figure 9. LEC Flow

IV. PROPOSED DESIGN AND VERIFICATION

A. Proposed verses existing

Existing High speed IO transmitters are suffering with ISI due to high frequency attenuation, behaves like low pass filter along the path and usage of CML logic leads to large number of transistor count in the design with low delay. Proposed method uses CMOS logic with custom D-Flipflop, latches, mux etc.. which provides high speed operation with full logic swing at high frequencies. This potentially eliminate requirement of equalizer with more number of taps at the transmitter. Existing design uses flatten design of serializer which increases possibility of risk during parallel to serial conversion. Proposed design divides the parallel to serial conversion into stages as 8bit user-interface (8UI) and 2bit user-interface (2UI) operating at different clock frequencies. This leads to clock domain crossing problem. This is eliminated by using feed-forward synchronization mechanism as shown in figure 10. Existing CML driver requires large tail currents. This is eliminated in proposed design by using digital driver with pull-up resistance and pull-down resistance same as line characterization resistance.

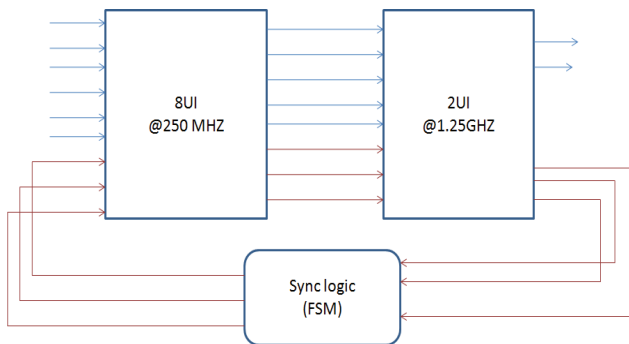


Figure 10. Proposed method of serIALIZATION

Figure 11 shows the block view of design. Parallel to serial conversion plays a major role in the Transmitter. Focusing on ‘parallel to serial conversion’ provides good insight on real need of equalizer at the transmitter, whether it is necessary or not. Encoding scheme just provides the data encryption at the transmitter.

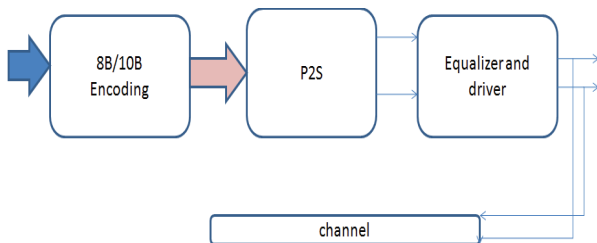


Figure 11. Block view of design

B. Design plan

Design contains 8b/10b encoding, parallel to serial conversion (P2S) block, followed by driver circuit connected to channel as shown in figure 20. 8b/10b

encoding consist of disparity blocks to add two parity bits for every byte of input data from controller. P2S converts parallel inputs to differential serial outputs by converting 40bit input data to 10bit data using 8UI, 10bit output of 8UI connected to 2UI separates 10 bit data into 2bit even and odd stream. Finally this 2bits are converted to 1bit using MUX and D-Flipflop at output.

C. Verification plan

Verification is the act of testing a design against its specifications. This makes sure that the design is right. Verification is a quality control process. This is a repetitive process to make sure the design is right against its specification. Validation is the act of test system against its operational goals. This makes sure that it is a right design. Validation is a quality assurance process.

The proposed verification plan starts with writing systemverilog/verilog code to mimic the behavior of design. This code is simulated using VCS simulator to speed up the process once the required behavior is attained these RTL codes are used in formal verification using cadence conformal , which compares the schematic with RTL code for logic equivalence. Because of no test-vectors requirement, it is easy way of verification. It provides exhaustive verification and leads to finding bugs as early as possible in the design.

Figure 12 shows the proposed method of verification plan. Behavioral Models (BMODS) imitates the behavior of design using verilog/systemverilog and compares with the actually schematic level implementation using cadence conformal for Formal verification which verifies without simulation. Finally to make sure the design is working properly, use mixed signal validation by synopsys VCS-XA which rely on the stimulus. Hence the final simulation results makes sure of the assurance of quality in the design.

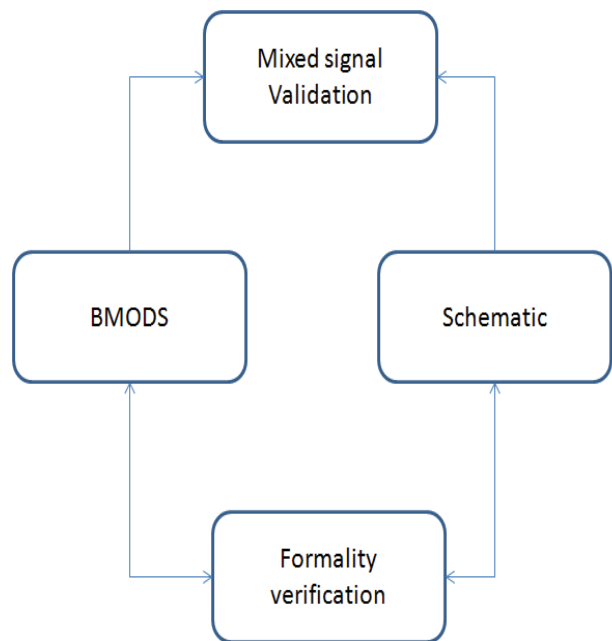


Figure 12. Proposed verification plan

V. DESIGN IMPLEMENTATION

A. Schematic overview of Transmitter

Final schematic overview is as show in figure 13. And its symbol view in figure 14.

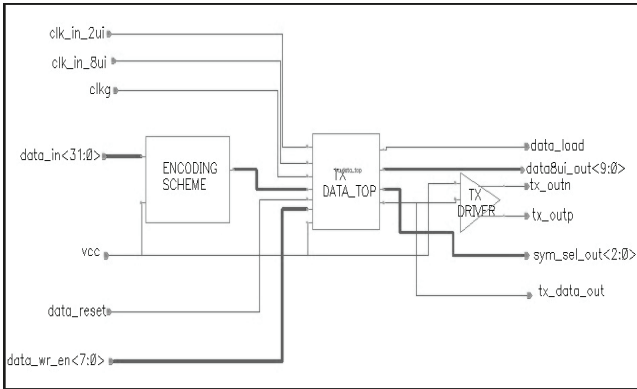


Figure 13. Transmitter schematic overview

Parallel to serial conversion is done in series of step by step conversion and also provides different level of user interface (UI) signals like 10bit interface output from data_8ui, 2bit interface output from data_2ui and 1bit interface output from TX_DATA_TOP. such that implemented design is compatible to both single and multi bit interfacing devices. Hence require multiple clock input signal in the design such as cllk_in_8ui for data_8ui operation, clk_in_2ui for data_2ui operation and clk_g for final serialization, these clock frequency changes according to the PCIe mode as shown in table 3. data_wr_en input is used for output selection of data output for data_8ui and also for TX_DATA_TOP. data_in is the 32 bit input from PCIe controller to the transmitter for transmission, in order to clear the data. When power is just switched ON and for force reset of transmitter, data_reset is provided. Due to multi hierarchical design structure, it is complex to debug the failure occurred in the design, hence few signals such as data_load, sym_sel_out, data8ui_out, tx_data_out, are brought out to verify and debug the functionality of design, 'tx_outp' and 'tx_outn' are dual outputs that are to be transmitted through the channel.

TABLE III.
OPERATING CLOCKS INFORMATION IN DESIGN

PCIe mode	clk_in_8ui	clk_in_2ui	Clkg
Gen1	250MHz	1.25GHz	2.5GHz
Gen2	500MHz	2.5GHz	5GHz
Gen3	800MHz	4GHz	8GHz
Gen4	1.6GHz	8GHz	16GHz

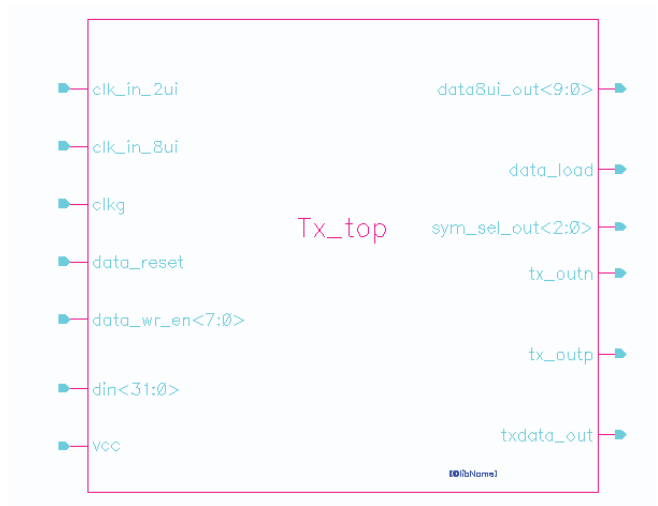


Figure 14. Tx_Top symbol view

VI. RESULTS

A. Verification report

Design is Formal verified using Cadence Conformal. Conformal as capability of inspecting clock domain crossing. Figure 15. shows the hierarchical view of data_top in the design. Figure 16 shows the Verification Report of data_top.

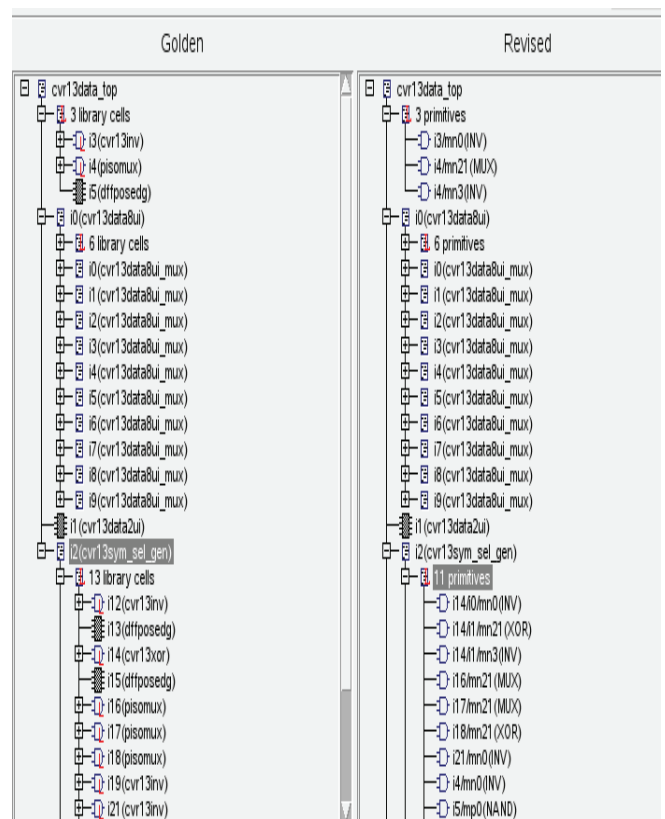


Figure 15. Hierarchical view of data_top

Verification Report		
Category		Count
1. Non-standard modeling options used:		0
2. Incomplete verification:		2
User added black box:	yes	
Black box mapped with different module name:	yes	
3. User modification to design:		0
4. Conformal extended checks recommended:		0
5. Design ambiguity:		0
6. Compare Results:		PASS

Figure 16. Verification Report of data_top

As the design is Formal verified, verification report in figure 16, ensures that implemented design is bug free without cross domain crossing violations and behavior of design is as expected. So this can be signed-off. Since we have not applied any stimulus some functionality may differ. To make sure the correct functionality, mixed signal validation is done and specter simulation is done for small blocks in hierarchy.

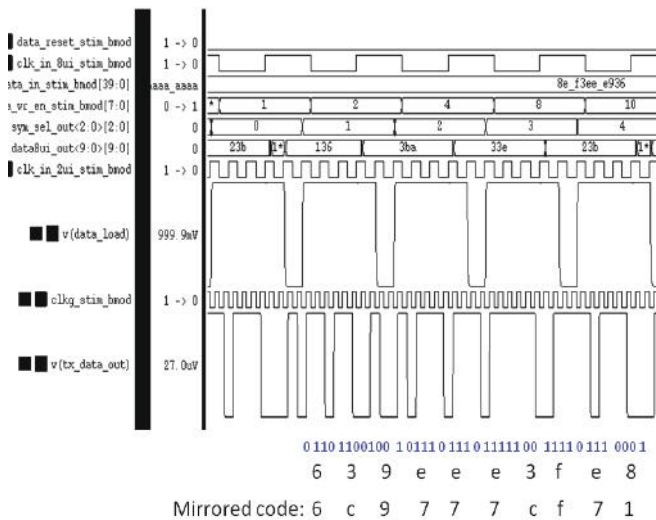


Figure 17. Mixed signal validation results of data_top

Figure 17 shows validation results of data_top with tx_driver for differential signaling. As the input 40bits parallel data feed to data_8ui, which slices 40 bit data into four slices of 10 bit each. Slices are selected using data_wr_en and sym_sel_out to provide output 10bit data parallel as 8UI interfacing at 250MHz in case of Gen 1, 500MHz in Gen 2, 800MHz in Gen 3 and 1.6GHz in Gen 4. data_2ui is separated, even and odd positioned bits of 10bits parallel data into 2 bit serial data at 1.25GHz in Gen 1, 2.5GHz in Gen 2, 4GHz in Gen 3 and 8GHz in case of Gen 4. Finally even and odd stream of 2bit data merged into one bit data tx_data_out using 2x1 mux with select line as the clock (clk) as 2.5GHz in Gen 1, 5GHz in Gen 2, 8 GHz in Gen 3 and 16GHz in Gen 4. The output serial data

tx_data_out in all generations is same as input 40bits parallel data.

VII. CONCLUSIONS

The designed high speed IO Transmitter provides high speed operation with less delay (1.81 ns) which is approximately two time faster than design implemented in 45nm technology (delay 4.45ns) and four times faster than design implemented in 90nm technology (7.23ns), with no data loss at the transmitter end. Equalizer is not required at the transmitter because no ISI occurred in the design. There were no Clock domain Crossing violations found which was fully verified using Cadence Conformal tool and validated using Mixed signal validation tool. But care should be taken to avoid ISI at receiver by placing equalizer with sufficient number of taps. Also, because of low pass effect of channel, signal gets degraded at the receiver's end.

The designed circuits are implemented in CMOS logic that can be replaced with CML logic to minimize power. Care should be taken on tail currents driving the logic through current mirror hence variations in the tail current in turn leads to glitches in the circuits. High capacitive loads formed at the current mirror may hold sufficient charge which may effect the operation.

For reducing ISI an efficient equalizer needs to be designed which may be linear or non-linear. Equalizer should be high pass because of low pass effect of routing channels so that it makes the overall response flat. Equalizer should provide sufficient gain at high frequencies. Linear equalizer such as Continuous Time Linear Equalizer (CTLE) or non-linear equalizers such as FIR filters with large or sufficient number of taps will provide the expected response at the driver output of the transmitter.

REFERENCES

- [1] S. R. Vangal *et al.*, "An 80-Tile Sub-100W TeraFLOPS Processor in 65-nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 1, Jan. 2008.
- [2] B. Landman and R. L. Russo, "On a Pin vs. Block Relationship for Partitioning of Logic Graphs," *IEEE Transactions on Computers*, vol. C-20, no. 12, Dec. 1971.
- [3] R. Payne *et al.*, "A 6.25-Gb/s Binary Transceiver in 0.13- μ m CMOS for Serial Data Transmission Across High Loss Legacy Backplane Channels," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, Dec. 2005.
- [4] J. F. Bulzacchelli *et al.*, "A 10-Gb/s 5-Tap DFE/4-Tap FFE Transceiver in 90-nm CMOS Technology," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 12, Dec. 2006
- [5] B. S. Leibowitz *et al.*, "A 7.5Gb/s 10-Tap DFE Receiver with First Tap Partial Response, Spectrally Gated Adaptation, and 2nd-Order Data-Filtered CDR," *IEEE International Solid-State Circuits Conference*, Feb. 2007.
- [6] Semiconductor Industry Association (SIA), International Technology Roadmap for Semiconductors 2008 Update, 2008.
- [7] W. Dally and J. Poulton, *Digital Systems Engineering*, Cambridge University Press, 1998.

- [8] K. Lee *et al.*, "A CMOS serial link for fully duplexed data communication," *IEEE Journal of Solid-State Circuits*, vol. 30, no. 4, Apr. 1995
- [9] K.-L. J. Wong *et al.*, "A 27-mW 3.6-Gb/s I/O Transceiver," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 4, Apr. 2004,
- [10] C. Menolfi *et al.*, "A 16Gb/s Source-Series Terminated Transmitter in 65nm CMOS SOI," *IEEE International Solid-State Circuits Conference*, Feb. 2007.
- [11] S. Sidiropoulos and M. Horowitz, "A Semidigital Dual Delay-Locked Loop," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 11, Nov. 1997
- [12] J. Montanaro *et al.*, "A 160MHz, 32b, 0.5W CMOS RISC Microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 11, Nov. 1996.
- [13] A. Yukawa *et al.*, "A CMOS 8-bit high speed A/D converter IC," *IEEE European Solid-State Circuits Conference*, Sep. 1988.
- [14] B. Casper *et al.*, "A 20Gb/s Forwarded Clock Transceiver in 90nm CMOS," *IEEE International Solid-State Circuits Conference*, Feb. 2006.
- [15] J. Poulton *et al.*, "A 14mW 6.25Gb/s Transceiver in 90nm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, Dec. 2007.
- [16] C.-K. Yang and M. Horowitz, "A 0.8- μ m CMOS 2.5Gb/s Oversampling Receiver and Transmitter for Serial Links," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 12, Dec. 1996.
- [17] J. Kim and M. Horowitz, "Adaptive-Supply Serial Links with Sub-1V Operation and Per-Pin Clock Recovery," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, Nov. 2002, pp. 1403-1413.
- [18] M. Horowitz, C.-K. Yang, and S. Sidiropoulos, "High-Speed Electrical Signaling: Overview and Limitations," *IEEE Micro*, vol. 18, no. 1, Jan.-Feb. 1998.