A STUDY TO IMPLEMENT SUPERSPEEDPLUS USB 3.1 PHYSICAL LAYER

[1]RAKESH ROSHAN MANDAL, [2]PHILEMON DANIEL

[1]M.TECH SCHOLAR, [2]ASSISTANT PROFFESER,
[1,2]Electronics &Communication EngineeringDepartment,
[1,2]NationalInstituteofTechnologyHamirpur(H.P.) -177005,India
E-mail:1rakeshroshan.mandal@gmail.com,2phildani7@nith.ac.in

Abstract— To work on a higher bandwidth, USB 3.1is the next evolutionarystep.AtpresentUSB3.1 is having the fastest data transmitting and receiving speed. The physical layer defines the signal in technology for the Super Speed bus for end-to-end communication between a host and device. Due to high significance of physical layer in the data transmission, USB 3.1 PHY has been implemented to enhance the overall performance. In this paper, the brief description of the overall functionality for implementing Physical Layer of USB3.1hasbeendiscussed.This paper also describes that, how USB3.1offersabetter performance overUSB3.0.

Keywords—Universal Serial Bus (USB); Physical Layer(PHY); Generation(Gen); Electromagnetic Interference(EMI);Running Disparity (RD);Register Transfer Logic (RTL).

I. INTRODUCTION

Initially, the USB was designed to provide a user-friendly plug-and-play way to connect external peripherals with a computer. But according to the present time scenario, USB has a plenty of different applications beyond just being as a connecting peripherals to computers. Earlier, USBs were providing two speeds (12Mbpsand1.5Mbps) for the use of peripherals. But as the technology grown, computers became more powerful and process larger amounts of data. As a result, users needed to get more and more data into and out of computers. This requirement led to the design of the USB2.0 in 2000 to provide a transfer rate up to 480Mbps while retaining backward compatibility. Further in 2006, USB3.0 was designed to provide users with the ability to have transfer rate up to 5Gbps while retaining backward compatibility with USB2.0.

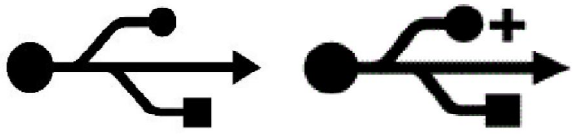

Figure 1(a).USB1.0symbolFigure 1(b).USB 2.0 symbol

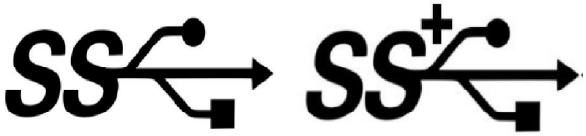

Figure 1(c).USB3.0symbol **Figure 1(d).USB 3.1 symbol**

To fulfill the continuous requirement of more bandwidth for larger & faster, storage & transfer of higher resolution video, and broader use of USB as an external expansion, USB3.1 was designed in 2010. USB3.1 extends the performance range of USB up to 10Gbps by doubling the Super Speed USB clock rate to 10Gbps and by enhancing more efficient data encoding.

II. OBJECTIVE

In this paper, our main focus will be to do the implementation and functionality verification of the physical layer of USB3.1 using Verilog HDL language in Xilinx tool for the target device XC7Z100-1-FFG900. The Verilog HDL coding for implementation of USB3.1 PHY has to be done for reducing there quired number of clock and percentage hardware over head of USB 3.1 PHY over USB3.0 PHY. The design has to be validated in Xilinx ISIM simulator and synthesized using XilinxISE14.7Designsuite.

III. ENHANCEDSUPERSPEEDSYSTEM

The Enhanced Super Speed system of USB3.1 is a layered communication system that provides two different speed range of5 Gbps (USB3.0) and 10Gbps (USB3.1).

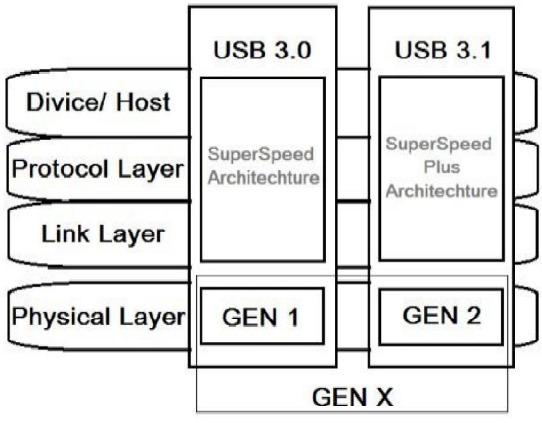

Figure2.USB3.1Enhanced Super Speed System

3.1. Device/Host Layer

In between Device and Host, one can actassource and the other as sink of information for exchange or vice versa.

3.2. Protocol Layer

This provides application data information required to build a communication relationship, also known as pipe between a host and a device endpoint.The host determines when application data has to be transferred.

3.3. Link Layer

The link layer describes the logical portion of a port and the communications between two ports or link partners. This also initialize and manage the physical layer connection.

IV. PHYSICAL LAYER OF USB 3.1

This describes the physical connection between a downstream facing port on a host and the upstream facing port on a device. As shown in the functional block diagram of the physical layer in Figure 3.1, the PHY is the intermediate layer between the higher layers of the host and the external devices. The PHY generally consist of both transmitter block and receiver block. The transmitter block of PHY, transmits the signals from higher layers of host to the receiver of external device. The receiver block of PHY, receives the signals from the external device and further transmits them in to the higher layers of host. With transmitted data signals, different control signals are also provided for transmission in the PHY. Along with the data and control signals, many commands are also transmitted for the operation of PHY.

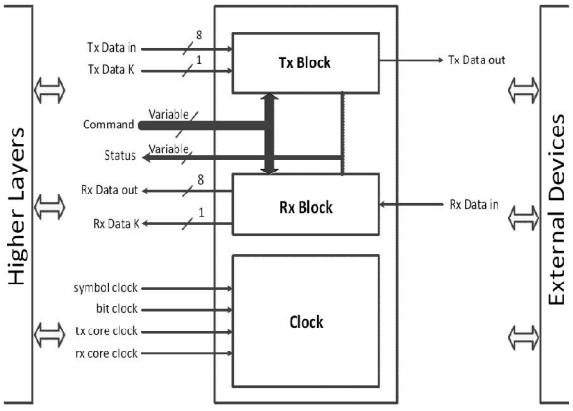

Figure 3. PHY Functional Block Diagram

For the implementation of USB 3.1 PHY, all the individual blocks of the transmitter and receiver of both GEN 1 and GEN 2, described below have to be designed using the Verilog HDL language. Every individual block has to be synthesized and the functionally verified using Xilinx ISE 14.7 Design suite.

4.1 GEN 1 Physical Layer

The Gen 1 PHYcan transfer the data at the rate of 5 Gbps. As shown in Figure 4 and Figure 5, the D/K bit indicates that whether the received information is data or control bit. In both the Gen, scrambler only scrambles the data bit, not the control bit. After receiving 8-bit data and D/K bit from the link layer, the Gen 1 PHY transmitter scrambles the data to reduce EMI emissions. After scrambling, the scrambled 8-bit data is encoded into 10-bit symbols for transmission over the physical connection. Then the encoded data are sent at a rate that includes spread spectrum to further lower the EMI emissions.

At the receiver end, the bit stream is received and assembled into 10-bit symbols. Then the assembled data stream is decoded and descrambled, producing 8-bit data which are further sent for further processing to the link layer.

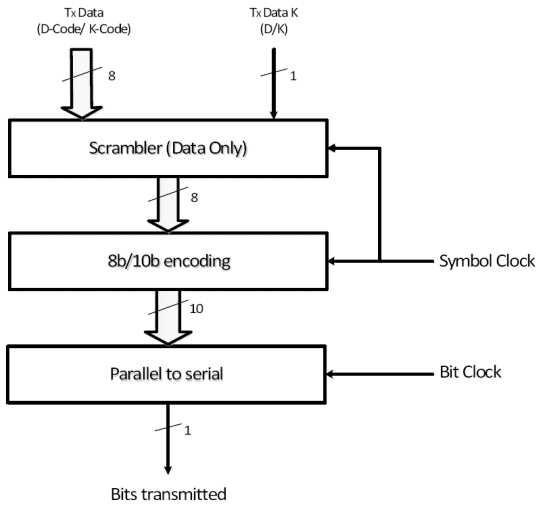

Figure 4. GEN 1 Transmitter block diagram

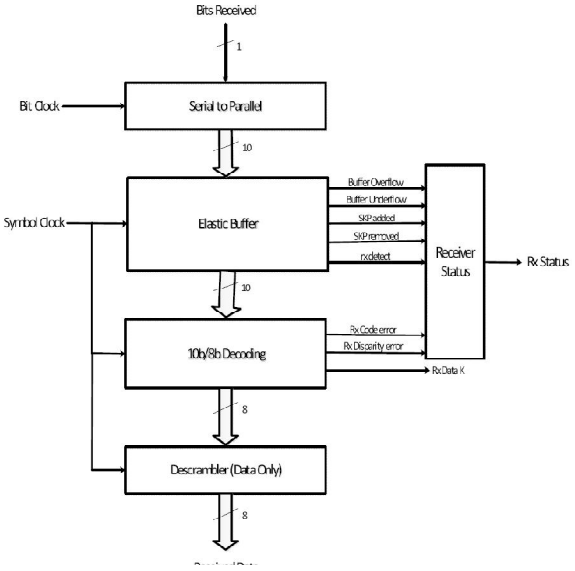

Figure 5. GEN 1 Receiver block diagram

4.2 GEN 2 Physical Layer

The Gen 2 PHYcan transfer the data at the rate of 10 Gbps. As shown in Figure 6 and Figure 7, the Block Start bit acts as the enable signal of Gen 2. There is a 4 bit block header which identify the 16 different 8-bit data. After receiving 8-bit data, 4-bit block header and 1-bit block start from the link layer, the Gen 2 PHY transmitter scrambles the data to reduce EMI emissions. After scrambling, the scrambled 128-bit data is encoded into 132-bit symbols for transmission over the physical connection. Then the encoded data are sent at a rate that includes spread spectrum to further lower the EMI emissions. At the receiver end, the bit stream is received and assembled into 132-bit symbols. Then the assembled data stream is decoded and descrambled, producing 8-bit data and 4-bit block header which are further sent for further processing to the link layer.

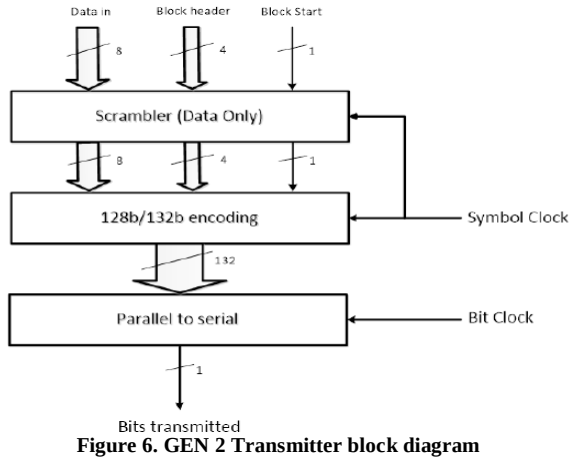

Figure 6. GEN 2 Transmitter block diagram

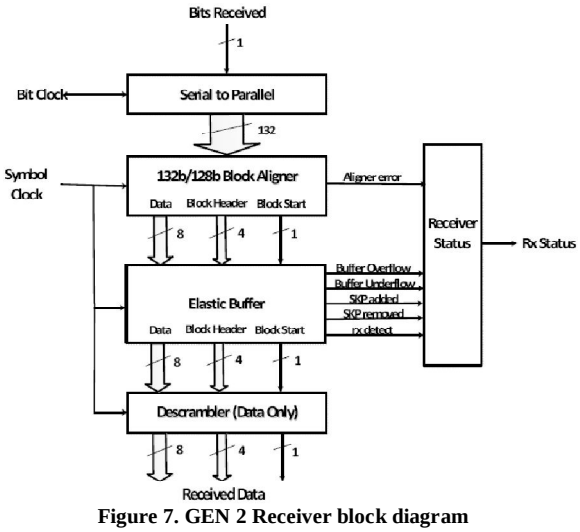

Figure 7. GEN 2 Receiver block diagram

V. FUNCTIONS OF DIFFERENT BLOCKS OF PHYSICAL LAYER

5.1 Encoder/Decoder

Encoding is used in serial data transmission standards to ensure sufficient data transitions, which is essential for the receiving of data.

5.1.1. GEN 1 Encoder/Decoder

In high-speed serial data transfer, mostly8b/10b coding scheme is used. The Gen 1 transmitter encoder converts the 8-bit parallel datainput to 10-bit parallel output. This 10-bit parallel output data is then converted to a serial data stream using a high-speed Serializer (parallel to serial converter). The serial data stream will be then transmitted to the receiver side through a transmission media. Then on the receiver side the serial data stream is again converted to 10-bit parallel output data using a high-speed Deserializer (serial to parallel converter).

The decoder will then reconvert the 10-bit data againto the original 8-bit data. During the execution of the 8b/10b coding scheme, the serial data stream is DC-balanced and has a maximum run-length without transitions of 5.

5.1.1.1. DC Balance and Run Length

When the number of 0s and 1s are equal for a given length of serial data stream, the data stream is DC-balanced. It is important to maintain a DC-balance as it avoids a charge being built up in the media. The maximum numbers of contiguous 0s or 1s in a serial data stream is known as run-length.

5.1.1.2. 8B/10B Code Mapping

The 8b/10b mapping is how the encoder converts 8-bit code groups into 10-bit codes. The group of code include 256 data characters and 12 control characters namedas Dx.y and Kx.y respectively.

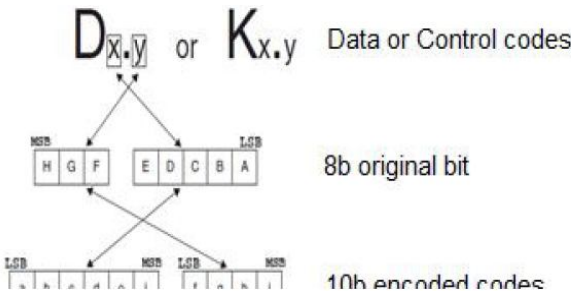

Figure 8. The 8b/10b Coding Scheme System

As shown in Figure 8, according to the coding scheme the original 8-bit data is broken into two blocks, 3 most significant bits (y) and 5 least significant bits (x). They are named as H, G, F and E, D, C, B, A starting from the most significant bit to the least significant bit. The 3-bit block is encoded into least significantblock of 4 bits named j, h, g, f. The 5-bit block is encoded into most significant block of 6 bits named i, e, d, c, b, a.

5.1.1.3. Disparity

In order to maintain a DC-balance in a data stream, the disparity concept is used. The disparity of a block is calculated by subtracting the number of 0s from the number of 1s. The value of a block with zero disparity is called disparity neutral. If both the 4-bit and 6-bit blocks are disparity neutral, then the

combination of 10-bit encoded data will also be disparity neutral which is a perfect DC-balanced code. But a perfect DC-balanced code is not possible to implement completely, as only 6 out of the 16 possible values of the 4-bit block are disparity neutral, they are insufficient for encoding the 8 values of the 3-bit block. Similarly, only 20 out of the 64 possible values of the 6-bit block are disparity neutral and they are insufficient for encoding the 32 values of the 5-bit block. Therefore, the values with a disparity of +2 and -2 are also used in the 8b/10b coding scheme to cover all the possible 3-bit and 5-bit values. Because both 4-bit and 6-bit blocks have an even number of bits, it is not possible to get a disparity of +1 or -1. The 8b/10b coding scheme was designed to combine the values of the 4-bit and 6-bit blocks in such a way that at the extreme condition, disparity value of the 10-bit code group will be at most +2 or -2.

5.1.1.4. Running Disparity

Atsomecases,10-bit encoded data values with +2(or-2) disparity maybe transmitted through the serial data stream. In that case, the DC-balance of the data stream become in valid. For maintaining a DC-balance data stream, each code group will be converted to one of the two possible values of RD-(with disparity of-2or0) and RD+(with disparity of+2or0).The encoder will consider either RD+orRD-value based on the calculation of current Running Disparity to maintain DC- balance.

Foreach3-bitor5-bitdata, the corresponding RD+ and RD- value is calculated and stored in the encoding table. Initially, the transmitter considers the current running disparity as RD-value from the encoding table. When an 8-bit data is encoded, the encoder will use the RD-value for encoding. After checking, if the 10-bit data been encoded is disparity neutral, then the same value of RD-will be used for encoding. Otherwise, the current running disparity will be changed to the RD+value which will be used for encoding. Similarly, if the current Running Disparity starts with RD+, the opposite function takes place.

5.2 Scrambling

Scrambling takes place by generating a pseudo-random data pattern which is XORed with the outgoing data stream bit. The algorithm used for scrambling data is expressed as a polynomial that is implemented as a linear feedback shift register (LFSR) which is different and unique for Gen 1 and Gen 2. The repetitive patterns are eliminated from the bit stream by the use of Scrambling. These repetitive patterns in the bit stream results in large amounts of energy concentrated in discrete frequencies which lead to significant EMI noise generation. Scrambling spreads the concentrated energy over a frequency range which helps in reducing the average EMI noise generated.

5.1.2 GEN1DataScrambling

On the Transmitter side, scrambling is applied to only the data bits prior to the 8b/10b encoding. On the receiver side, descrambling is applied to the data bits after the 10b/8b decoding. As shown in Figure11, the LFSR is scrambled or descrambled by serially XORing the8-bit (D0-D7) data with the16-bit (D0-D15) output of the LFSR. The output of the LFSR D15bit is first XORed with D0bit of the actual data to be processed. Then the output of the LFSR and data registerare advanced serially and the output processing is repeated from D1 toD7.

The implemented polynomial by LFSR for Gen1scrambling: G(X)= X16+X5+X4+X3+1.

Then except the SKP symbol, each symbol of LFSR shall be advanced with serially eight shifts. The LFSR (D0-D15) is initialized from FFFFh. The LFSR on the transmitter side is initialized whenever COM leaves the transmitter LFSR. The LFSR on the receiver side is initialized whenever COMenters the Receiver LFSR.

5.1.3 GEN2DataScrambling

The process used for scrambling in Gen 2 is different than that of Gen1.

The implemented polynomial by LFSR for Gen 2 scrambling: G(X)= X23+ X21+ X16+ X8+ X5+ X2+1.

The scrambler can operate on 3 different modes:

1. 1.The scrambler advances and is also XORed with the data, i.e. the scrambling take splace.
2. 2.The scrambler advances and is bypassed (withoutbeing XORed with the data).
3. The scrambler does not advance and is bypassed (without being XORed with the data).

There are some rules for scrambling in Gen2.The 4bits, Block Header are always by passed without advancing the scrambler. The Block Header indicates whether it's a data block or control block. The manner how the control blocks and their corresponding symbols work, have been described in the table below.

Table1.Description of Control Blocks

Controlblocks	Symbol	Advances	Bypass
TS1,TS2,TSEQ	0	YES	YES
	1TO13	YES	NO
(Whenused forDC balance)	14,15	YES	YES
(Whennot usedfor DCbalance)	14,15	YES	NO
SKP,SKPEND	ALL	NO	YES
SDS	ALL	YES	YES
SYNC	ALL	YES	YES

The LFSR on the transmitter side is initialized whenever the last Symbol of a SYNC Ordered Set is sent from transmitter and on the receiver side it is initialized whenever the last Symbol of a SYNC Ordered Set is received by the receiver. Receiver checks Symbol 0 of Control Blocks to find out whether to advance the LFSR or not. Except when Symbol 0 is SKP or SKPEND, for all other conditions, the LFSR is advanced. Scrambling for a symbol is executed from the least significant bit up to the most significant bit. The LFSR is initialized from 1 DBFBCh. After every16384TSEQ sets, a SYNC Ordered Set shall be inserted to reset scrambler and to improve the block alignment. For data blocks, all the 16 symbols (i.e.128bit, as1symbol=8bit) will be passed after scrambling.

5.3 Elastic Buffer

The term “elastic” refers to elasticity of time. It is the maximum allowable changes in timing parameters, not properties of materials changes. It is a type of a synchronous buffer which gives more time for buffering of data to beat transaction state, instate of resending the same data again by the sender.

VI. USB 3.1 PHY RTL AND SIMULATION WAVE FORM

As shown in Figure4.30 and Figure4.31, the USB3.1 PHY block which is the complete system has been shown in two different simulation outputs. Here the 8 bit data to be transmitted is given to the USB3.1 PHY input port tx_data_in and the corresponding 8 bit data is received at the 8 bit output portrx_data_out. Here the 2 bit input signal named “rate” decides the rate of transmission. If the rate value is 00, then the GEN1 PHY is selected and if the rate value is 01, then the GEN 2 PHY is selected.

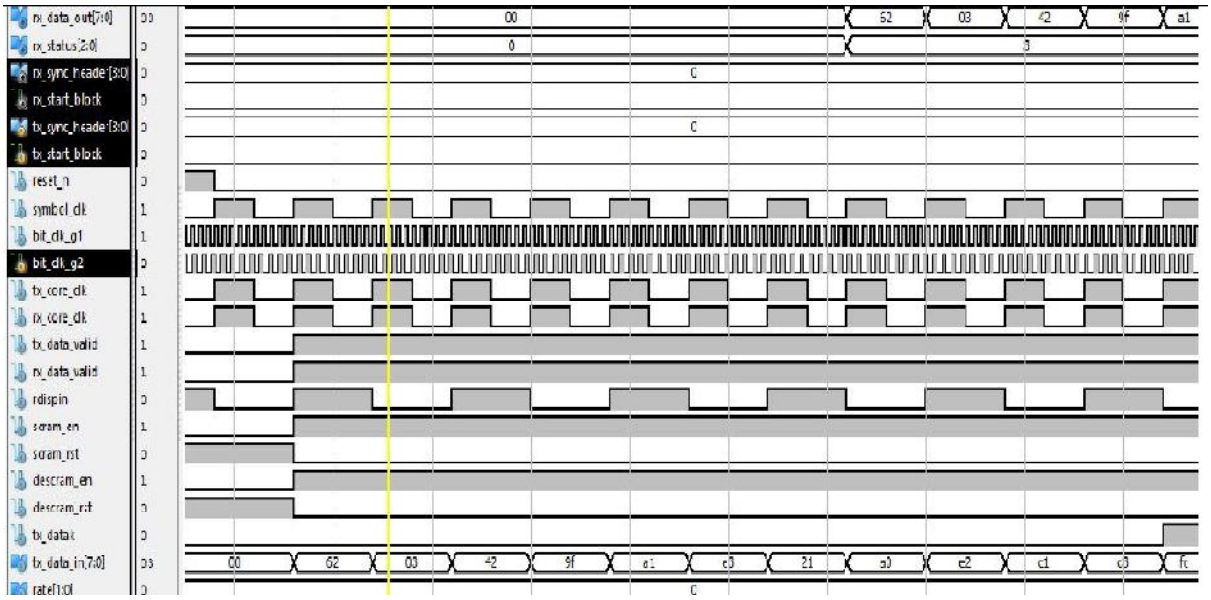

Figure 9. USB 3.1 GEN 1 PHY Simulation

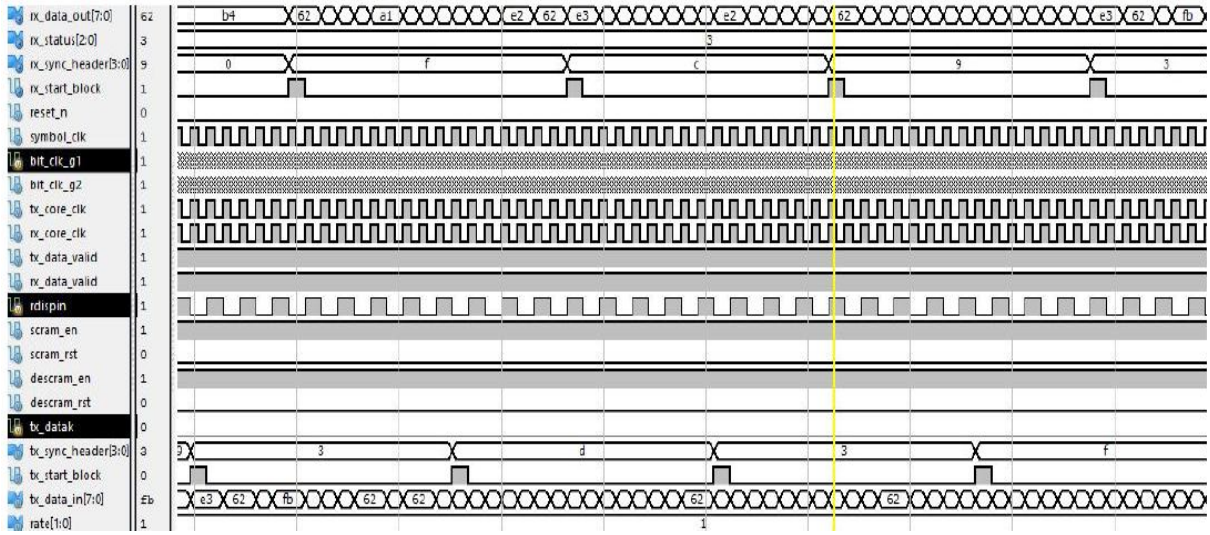

Figure 10. USB 3.1 GEN 2 PHY Simulation

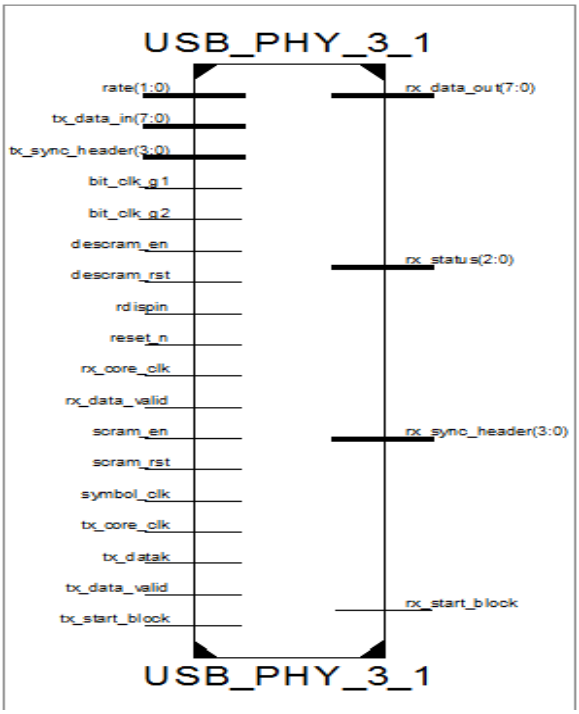

Figure 11. USB 3.1 PHY RTL

VII. HARDWARE UTILIZATION

After synthesizing the entire system in Xilinx ISE, the report is generated for resource utilization of the synthesized module for the target device XC7Z100-1-FFG900. Table 4.7 has the hardware utilization by different modules synthesized. The table gives slice register, slice LUTs and LUT-FF pair utilization by different units.

Table 2. Hardware Utilization by USB 3.0 PHY and USB 3.1 PHY System.

Module	Number of slice registers	Number of slice LUTs
Complete system of USB 3.0 PHY	159	2437
Complete system of USB 3.1 PHY	1110	3373

7.1. Calculations of hardware utilization and hardware overhead

The target device XC7Z100-1-FFG900, is configured with 6-input LUT.

1 register = 4 gates
1 LUT= 6 inputs= 64registers= 256gates

Hardware utilization by USB3.0 PHY (In terms of logic gates)

= (159*4)+ (2437*256)= 62508

Hardware utilization by USB3.1 PHY (In terms of logic gates)

= (1110*4)+ (3373*256)=867928

Hardware over head (%) USB3.1 to USB3.0

= (867928-62508)/ 62508×100=38.978

CONCLUSION

In this paper, various scenarios including the implementation and the functionality verification of the USB3.1 PHY has been done using Verilog HDL language in Xilinxtool. The RTL schematic and the Simulated Behavioral model wave form for USB3.1 PHY has been obtained according to the available methods in the literature. This design is validated in Xilinx ISIM simulator and synthesized using Xilinx ISE14.7Design suite. The number of clock cycles required for the transmission of a signal from transmitter to receiver has been reduced. From the design summary generated for resource utilization of the synthesizedmoduleforthetargetdeviceXC7Z100-1-FFG900, the number of slice register and slice LUTs utilization by different units has been obtained. The calculation for hardware utilization forUSB3.0PHYandUSB3.1PHYdesignhave been done. The percentage hardware overhead of38.978has been obtainedinUSB3.1 PHY with respect to USB 3.0PHY with doubling the clock frequency rate. The efficiency is ensured by the accurate results.

REFERENCES

[1] “UniversalSerialBus3.1Specification,”Revision1.0,July 26,2013.

[2] “Universal PHY Interface for the PCI Express, SATA, and USB3.1 Architectures,”Version4.3,2014.

[3] MortenChristiansen,EricHuang,“USB3.1:Evolutionand revolution”, White paper of USB3.1, Feb,2014.

[4] “Universal Serial Bus 3.0 pecification, ”Revision1.0, November 12, 2008.

[5] "PHY Interface for the PCI Express, SATA and USB3.0" Version4.0, 2011.

[6] “Universal Serial Bus 2.0 Specification, ”Revision 2.0, March13, 2006.

[7] “8b/10b Encoder/Decoder, Technical Reference Manual,” Revision 1.4, LATTICE Semiconductor, January-2015.

[8] M. Maadi, “An8b/10b Encoding Serializer /Deserializer (SerDes) Circuit for High Speed Communication Applications Using a DC Balanced, Partitioned-Block,8b/10bTransmission Code,” International Journal of Electronics and Electrical Engineering Vol.3, No.2, April,2015.

[9] H. Trivedi, R. Kumar, R. Tank, C. Sundaresan & M. Madhushankara, “Implementation of USB 3.0 Super Speed Physical Layerusing Verilog HDL,” International Journal of Computer Applications, Vol.95, no.24, June-2014, pp.1-5.

[10] Wu, Yingand Yuehong Qiu,"The8-bitparallelCRC-32 research and implementation in USB3.0," Computer Science&Service System (CSSS), 2012 International Conference on. IEEE,2012.

[11] J. Cortadella, M. Kishinevsky, Bill Grundmann, “SELF: Specification and design of a synchronous elastic architecture for DSM systems, ”TAU’s2006: Hangouts of the international work shop on timing issues in the specification and synthesis of digital systems, 2006.

[12] J.Winkles, “Elastic Buffer Implementations in PCI Express Devices, ”http://www.docin.com/p-118906415.html, Nov2003.

[13] M.Kumar Chethan,Y.G.Praveen Kumar, andM.Z. Kurian, "Design and Implementation of Logical Scrambler Architecture for OTN Protocol, "International Journal of Advanced Research in Computer Engineering &Technology (IJARCET)Volume3 Issue4, April2014.

[14] Sobański, Ireneusz, and Wojciech Sakowski, "Hardware/software co-designinUSB3.0massstorage application," ICSES2010International Conference on Signals andElectronicCircuits,2010.

[15] J.Kopanski, WitoldPleskacz, and Dariusz Pienkowski."A 5Gb/sequalizer for USB 3.0receiverin 65nm CMOS technology, "Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2011 IEEE 14th International Symposium on. IEEE, 2011.

[16] Chen, Chung-Hao, Pujitha Davuluriand Dong-Ho Han."A novel measurement fixture for characterizing USB3.0 radio frequency interference." Electromagnetic Compatibility (EMC), 2013 IEEE International Symposiumon. IEEE,2013.

[17] Abba,A.,etal."ImplementationofUSB3.0buscontroller in FPGA for data transfer in multi-channel applications." Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC),IEEE,2013.

[18] Arvind, G.K., and S.Leonard Gibson Moses ."USB to USB Data Transfer without PC."Automation and Autonomous System7.1(2015):28-31.

[19] Basavana Gouda. k,A.B.Kalpana, "Design and Implementation of High Definition Multimedia Interface Transmitter and Receiver, "International Journal of Advanced Technology &Engineering Research (IJATER), Volume2, Issue5, Sept2012.

[20] G.Dimitrakopoulos, Anastasios Psarras, and Ioannis Seitanidis, “Microarchitecture of Network-on-Chip Routers,” Springer, 2014.

★★★