International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

Study and Implementation of Physical Layer Coding Used in Super Speed USB

S Monika[1], G Shruthi[2]

[1, 2]Assistant Professor, ECE, Padmasri Dr. B. V. Raju Institute of Technology, Hyderabad, India

Abstract: ***This paper implements the DC balanced 8B/10B coding in Super speed USB which employ a very fast FPGA from Xilinx family is proposed. This technique can be used by other high speed serial buses such as PCI Express, IEEE 1394b, Serial ATA, SAS, Fiber Channel, Gigabit Ethernet, Serial Rapid IO and HDMI (Transition Minimized Differential Signaling) that use the same coding. Using the look-up table and memory with fast technique made this design efficient to be implemented. Moreover, the proposed method has very low complexity and fast to execute with minimum logic and also easy to implement. The Scrambling and descrambling modules are added in the above modules to support USB 3 physical layer transactions.***

Keywords: 8B/10B coding, Super Speed USB, and Model Sim

1. Introduction

It is a specification to establish communication between devices and a host controller (usually a personal computer), developed and invented by Ajay Bhatt, while working for Intel. USB has effectively replaced a variety of interfaces such as serial and parallel ports [1]. USB can connect computer peripherals such as mice, keyboards, digital cameras, printers, personal media players, flash drives, Network Adapters, and external hard drives. For many of those devices, USB has become the standard connection method.

The Universal Serial Bus (USB) 3.0 specification is a new industry-standard peripheral connection technology, developed by USB Implementers Forum, for connecting peripherals to PCs and laptops [2]. The USB 3.0 specification draws from the same architecture of the wired USB specification and therefore is a backward-compatible standard with the same ease-of-use and plug and play capabilities of previous USB technologies [6], but with a 10X performance increase and lower power consumption. The USB 3.0 specification uses two additional high-speed differential pairs for Super Speed mode, which boosts its bandwidth to 5 GB/s.

The SuperSpeed USB specification is similar to earlier USB versions in terms of the connector and device drivers. The end-user and device driver engineer may find SuperSpeed USB similar to earlier versions, but it is significantly different to implementers of Super-Speed USB host and devices. At a mechanical level, the SuperSpeed USB specification supports dual-bus architecture for backward compatibility to a USB 2.0 device. This means that the SuperSpeed USB cable needs to support eight primary wires,

two wires for USB 2.0 connectors, two shared between the USB 2.0 and SuperSpeed USB specifications (PWR and GND) and four for SuperSpeed USB dual-simplex differential signals [2]. The SuperSpeed USB specification supports a dual-simplex data interface with four differential wires for simultaneous data flow in both directions. It should be noted that adding a bi-directional data interface was necessary to support the SuperSpeed USB specification's target speed because the half duplex, two wire differential signals of USB 2.0 and unidirectional data flow were not enough to support the SuperSpeed USB specification's high bandwidth.

2. USB 3.0 Architecture

The physical layer defines the PHY portion of a port and the physical connection between a downstream facing port (on a host or hub) and the upstream facing port on a device [5]. The Super Speed physical connection is comprised of two differential data pairs, one transmit path and one receive path. The physical layer receives 8-bit data from the link layer and scrambles the data to reduce EMI emissions. It then encodes the scrambled 8-bit data into 10-bit symbols for transmission over the physical connection.

The resultant data are sent at a rate that includes spread spectrum to further lower the EMI emissions. The bit stream is recovered from the differential link by the receiver, assembled into 10-bit symbols, decoded and descrambled, producing 8-bit data that are then sent to the link layer for further processing.

The link layer specifications for Super Speed are detailed. A Super Speed link is a logical and physical connection of two ports. The connected ports are called link partners. A port has a physical part and a logical part. The link layer defines the logical portion of a port and the communications between link partners.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

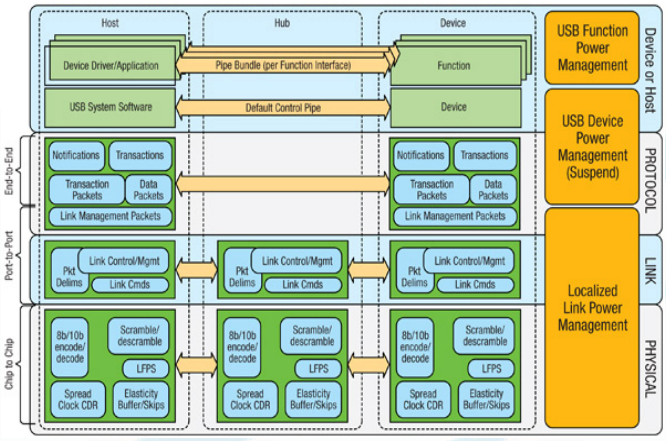

Figure 1: USB3.0 Logical Architecture

This protocol layer defines the "end-to-end" communications rules between a host and device. The Super Speed protocol provides for application data information exchanges between a host and a device endpoint [4]. This communications relationship is called a pipe. It is a host-directed protocol, which means the host determines when application data is transferred between the host and device.

A. Transmission

i) Scrambling: Scrambling reduces EMI problems associated with repeated patterns in the data being sent across an SS link. The scrambler output is simply XORed with each byte of data to eliminate the repeated patterns.

ii)8/10b Encoding: Every byte that traverses the link is first converted into a10-bit value called a symbol (this is a common encoding scheme in high-speed serial designs). Parallel/Serial Conversion — Bytes are converted to bit stream LFPS — Low Frequency Periodic Signaling is typically used in situations where the link is in an electrical idle state.

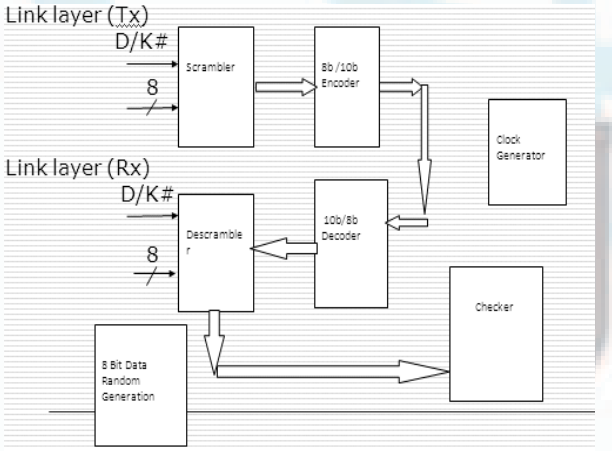

Figure 2: Physical layer Transmitter and Receiver

iii) *Differential Transmission:* Packets are clocked onto the link at a 5.0 Gbps rate.

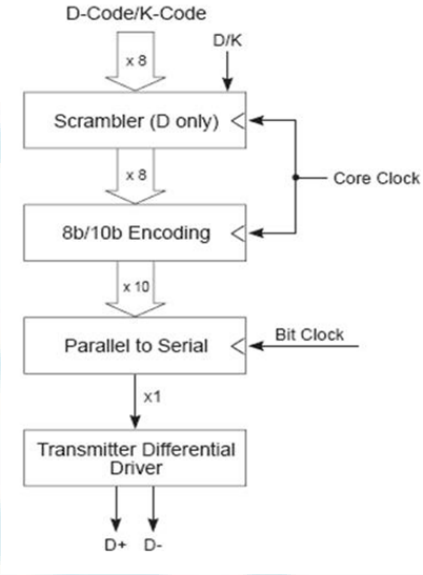

Figure 3: Transmitter

B. Reception

i) Differential Reception: the scrambled and encoded data is received and forwarded to the recovery blocks. Clock and Data Recovery — the clock is extracted from the bit stream and data is clocked into the serial/parallel converter. Serial/Parallel Conversion — data is clocked into the converter and 10-bit symbols are clocked into the elastic buffer.

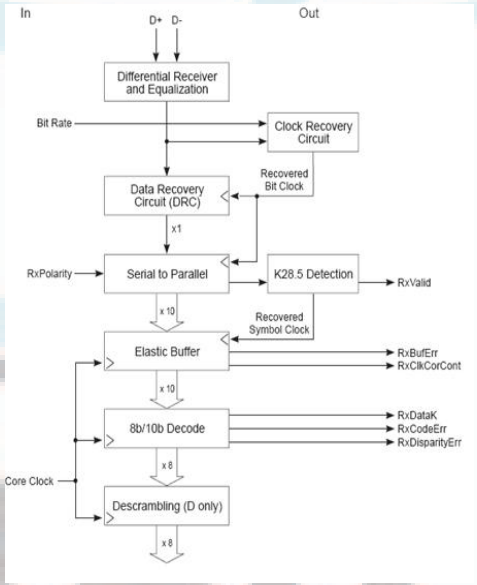

Figure 4: Receiver

ii) Elastic Buffer: The elastic buffer must absorb the worst-case clock variation between the transmitted lock frequency (recovered) and the local receive clock. The maximum variance is +300 to -300ppm. The buffer must also accommodate variations resulting from the Spread Spectrum clocking. Compensation is achieved via SKP ordered sets that are periodically inserted into the bit stream.

iii) 8/10b Decoding: 10-bit symbols are converted back to bytes. Un Scrambling — the same scrambling output is

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

XORed with the scrambled data a second time to recover the original data.

C. Various blocks

i) Scrambler: Scrambling reduces repeated patterns in the bit stream and lowers EMI by preventing the concentration of emitted energy at only a few frequencies. Scrambling works by generating a pseudo-random data pattern that is XORed with the outgoing bit stream. The algorithm used for scrambling data is expressed as a polynomial implemented as a linear feedback shift register (LFSR).

ii) Linear feedback shift register (LFSR): The purpose of the scrambler is to reduce repeated patterns in order to prevent concentration of emitted energy at only few frequencies. It validates the analytical assumption that all symbols have equal probability to be transmitted. The scrambler is implemented using the linear feedback shift registers (LFSR).Input sequence is XORed with a PN (pseudorandom) sequence, to produce a seemingly random sequence. XOR the output with the same PN sequence to recover the data in the receiver.

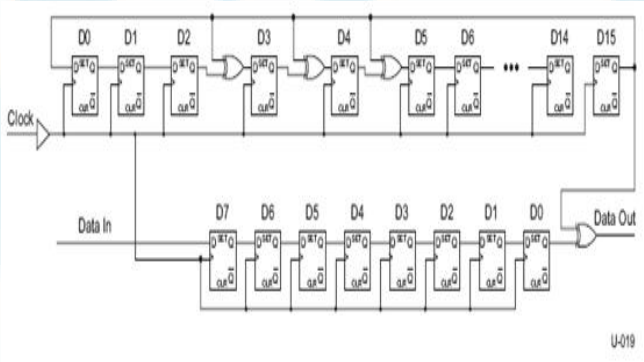

Figure 5: LFSR

iii) Decoder: The 8b/10b Decoder uses two lookup tables (the D and K tables) to convert the 10-bit symbol stream back into bytes. Each symbol value is submitted to both lookup tables but only one of the tables will find a match for the symbol. The state of the D/K# signal indicates that the received symbol is a:

- Data (D) Symbol — a match for the received symbol is located in the D table. D/K# is driven high.
- Control (K) Symbol — a match for the received symbol is located in the K table. D/K# is driven Low.

iv) Descrambler: Disabling scrambling is intended to help simplify test and debug equipment. Control of the exact data patterns is useful in a test and debug environment. Since scrambling is reset at the physical layer, there is no reasonable way to reliably control the state of the data transitions through software. The Disable Scrambling bit is provided in the training sequence for this purpose.

3. Tests and Results

The architecture of FPGA implementation of 8b/10b coding used in super-speed USB is proposed and designed for digital hardware implementation. All individual modules have been designed individually and verified functionally using random test bench using Modelsim 6.3f. It is observed that the simulation results for the 8b 10b encoder, 8b 10b, decoder, scrambler and descrambler generated were satisfactory and also the interconnections among all the modules are perfect. A priority encoder method is used for 8b 10b encoder and decoder. The symbol errors for both D and K symbols are verified in both encoder and decoder. The above modules are simulated using VERILOG hardware description language.

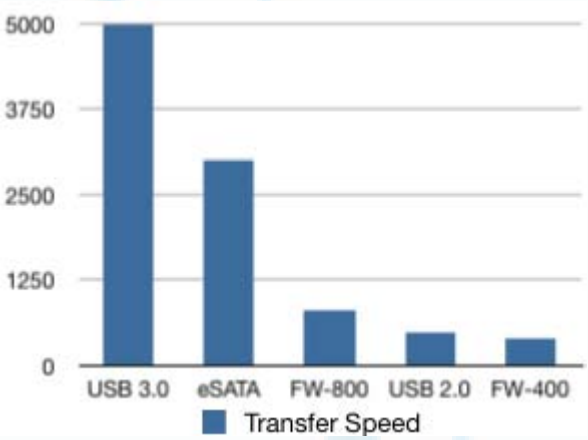

Figure 6: Transfer Speed

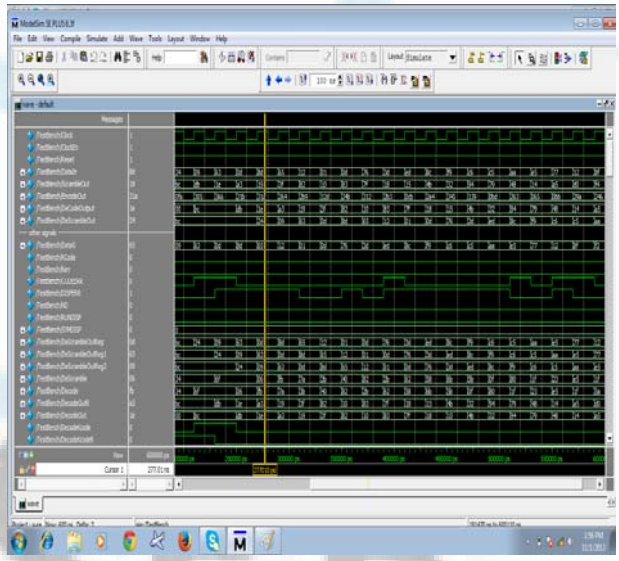

Figure 7: Test bench simulated results

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

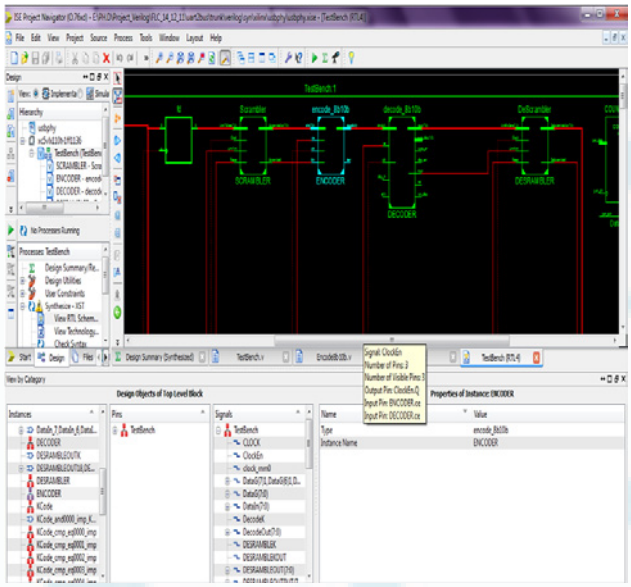

Figure 8: Synthesis Results

4. Conclusion

This work can be extended by connecting this total module in between Link layer and Physical analog layer of USB 3.0 architecture and transferring USB3.0 packets rigorously from link layer to physical layer. Also the work can be extended to do the FPGA implementation by using SPRTAN 3E or Vertex V XILINX FPGA's. This papers best suits as an IP (Intellectual Property) core of Physical Coding Layer in USB 3.0 specification.

The Design of Physical layer coding can be rigorously tested if soft cores of link layer, physical analog layer are available. This can be an extended work for the present paper. Complete USB 3.0 project can be done if one has finished soft cores of link, protocol, application layers in addition to our paper. If analog PHY's are available in the market hardware level validation can be done to our paper.

References

[1] A.X. Widmer and P.A. Franaszek, "A dc-balanced, partitioned block, 8B/10B transmission code, "IBM Journal of Research and Development, vol.27, no.5, pp. 440-451, Sep 1983.
[2] Universal Serial Bus 3.0 Specification Revision0.9, July30, 2008.
[3] Ravi Budruk, Don Anderson & Tom Sanely, 2004. PCI Express System Architecture - Mindshare Inc., pp 419-434.
[4] Thatcher, Jonathan (1996-04-01). "Thoughts on Gigabit Ethernet Physical". IBM Retrieved on 2008-08-17.
[5] Jenming Wu & Yu-Ho Hsu, 8B/10B Codec for Efficient PAPR Reduction in OFDM Communication Systems. International technology roadmap for Semiconductors (ITRS).
[6] K.V.Suresh Babu, Prof.A.S.Srinivasarao, D.Srinivasa Rao "VLSI implementation of physical layer coding used in super speed usb using verilog"

Author Profile

Ms. S. Monika is working as Assistant professor in Padmasri Dr.B.V.Raju Institute of Technology. She completed her M.Tech in Embedded system from Padmasri Dr. B.V.Raju Institute of Technology and B.Teh from TRR College of Engineering. Area of interest is Embedded system and VLSI Design.

Mrs. G. Shruthi completed her M.Tech-Embedded systems and B.Tech -Biomedical Engineering from Padmasri Dr. B. V. Raju Institute of Technology. Area of interest includes Embedded systems.