VLSI IMPLEMENTATION OF PHYSICAL LAYER CODING USED IN SUPER SPEED USB USING VERILOG

K.V.Suresh babu1, Prof.A.S.Srinivasarao2, D.Srinivasa Rao3
1Student (M.Tech), Dept of ECE, AITAM, Tekkali, A.P
2Professor &H.O.D., Dept of ECE, AITAM, Tekkali, A.P
3Assistant. professor, Dept of ECE, AITAM, Tekkali, A.P

Abstract- This project implements the DC balanced 8B/10B coding used in Super speed USB with 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, SSA, Gigabit Ethernet, In fini Band, XAUI, Serial Rapid IO, DVI and HDMI (Transition Minimized Differential Signalling)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 30 physical layer transactions. The code is implemented in VERILOG

keywords— Usb, link layer, crc, ncverilog and cadence

I. 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. 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.

A. Version history:

• USB 0.7: Released in November 1994.
• USB 0.8: Released in December 1994.
• USB 0.9: Released in April 1995.
• USB 0.99: Released in August 1995.
• USB 1.0 Release Candidate: Released in November 1995.
• USB 1.0: Released in January 1996.Specified data rates of 1.5Mbit/s (Low-Bandwidth) and 12Mbit/s (Full Bandwidth). Does not allow for extension cables or pass-through monitors (due to timing and power limitations). Few such devices actually made it to market.
• USB 1.1: Released in September 1998.Fixed problems identified in 1.0, mostly relating to hubs. Earliest revision to be widely adopted.
• USB 2.0: Released in April 2000.Added higher maximum bandwidth of 480 Mbit/s [60 MB/s] (now called "Hi-Speed").

II. INTRODUCTION TO USB 3.0

This paper is a brief review of the USB 3.0 implementation, focusing on USB 2.0 backward compatibility and on the major features associated with the Super-Speed (SS) bus. The goal is to provide the reader with a short and concise description of USB 3.0, and enough detail to give a good feel for the technology, protocols, and techniques. Due to the limited scope of this paper, some terminology and concepts are introduced but not fully developed.

a) Motivation for USB 3.0:

USB 3.0 enables more demanding applications compared to USB 2.0 by addressing its limitations:

• Bandwidth - 5.0 Gb/sec Super Speed (SS) vs. 480 Mb/sec (High Speed) rate.
• Power Conservation - link power states (U0 - U3) and function power management.
• Data Flow Control - poll once versus poll multiple times
• Error Handling - End-to-end and port-to-port error detection and retries versus only end-to-end retries with USB 2.0.The additional bandwidth provided by USB SS transactions can benefit applications like real-time audio and video streaming that require higher bus bandwidth at regular intervals. Mass storage applications can also benefit from the SS bandwidth. For example, Table 2.1 lists approximate download times for the different transmission rates.

b) USB 2.0 Host:

• Hubs power downstream ports when directed by software (USB 2.0) with Super Speed support disabled.
• Device connects at high-speed.
• Host system begins device enumeration at high-speed. When the host is powered off, the hub does not provide power to its downstream ports unless the hub supports charging applications. When the host is powered on and there is no Super Speed hardware support, the following is the typical sequence of events:
• Hub detects VBUS and connects as a high-speed device.
Host system begins hub enumeration at high-speed.

III. PHYSICAL LAYER

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. The Super Speed physical connection is comprised of two differential data pairs, one transmit path and one receive path. The nominal signalling data rate is 5 Gbps.

![Figure 1: Super Speed Physical Layer](image)

The electrical aspects of each path are characterized as a transmitter, channel, and receiver; these collectively represent a unidirectional differential link. Each differential link is AC-coupled with capacitors located on the transmitter side of the differential link. The channel includes the electrical characteristics of the cables and connectors. At an electrical level, each differential link is initialized by enabling its receiver termination.

a) Physical Layer Overview

The physical layer defines the signalling technology for the Super Speed bus. This chapter defines the electrical requirements of the Super Speed physical layer. This section defines the electrical-layer parameters required for interoperability between Super Speed components. The PHY Interface for the PCI Express and USB Super Speed Architectures (PIPE) is intended to enable the development of functionally equivalent PCI Express and USB Super Speed PHY’s. Such PHY’s can be delivered as discrete IC’s or as macro cells for inclusion in ASIC designs. The PIPE specification is defined to allow various approaches to be used. Where possible the PIPE specification references the PCI Express base specification or USB 3.0 Specification rather than repeating its content. In case of conflicts, the PCI-Express Base Specification and USB 3.0 Specification shall supersede the PIPE spec. This spec provides some information about how the MAC could use the PIPE interface for various LTSSM states and Link states.

b) PCI Express PHY Layer

The PCI Express PHY Layer handles the low level PCI Express protocol and signalling. This includes features such as: data serialization and de-serialization, 8b/10b encoding, analog buffers, elastic buffers and receiver detection.

Some key features of the PCI Express PHY are:

- Standard PHY interface enables multiple IP sources for PCI Express Logical Layer and provides a target interface for PCI Express PHY vendors.
- Supports 2.5GT/s only or 2.5GT/s and 5.0 GT/s serial data transmission rate.
- Utilizes 8-bit, 16-bit or 32-bit parallel interface to transmit and receive PCI Express data.
- Allows integration of high speed components into a single functional block as seen by the End point device designer.
- Data and clock recovery from serial stream on the PCI Express bus.
- Holding registers to stage transmit and receive data.
- Supports direct disparity control for use in transmitting compliance pattern(s).

c) Physical Layer Transmission and Reception:

Transmission:

- 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.
- 8/10b Encoding — every byte that traverses the link is first converted into a 10-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.
- Differential Transmission — Packets are clocked onto the link at a 5.0 Gb/s rate.

Reception:

- 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.
- Elastic Buffer — The elastic buffer must absorb the worst-case clock variation between the transmitted clock 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.
- 8/10b Decoding — 10-bit symbols are converted back to
bytes. Un Scrambling — the same scrambling output is XORed with the scrambled data a second time to recover the original data Rx Ck.

d) **VLSI Implementation of physical coding layer**

**USB 3.0 ARCHITECTURE**

**Test bench:**

The term “test bench” usually refers to simulation code used to create a predetermined input sequence to a design, then optionally to observe the response.

Today, in the era of multi-million gate ASICs and FPGAs, reusable intellectual property (IP), and system-on-a-chip (SoC) designs, verification consumes about 70% of the design effort. Design teams, properly staffed to address the verification challenge, include engineers dedicated to verification. The number of verification engineers can be up to twice the number of RTL designers. Higher abstraction levels are usually accompanied by a reduction in control and therefore must be chosen wisely. These higher abstraction levels often require additional training to understand the abstraction mechanism and how the desired effect can be produced. If a verification process uses higher abstraction levels by working at the transaction- or bus-cycle levels (or even higher ones), it will be easier to create large amount of stimulus. But it may be difficult to produce a specific sequence of low-level zeroes and ones.

**ENCODER:**

One of the major goals of 8b/10b encoding is to embed a clock into the serial bit stream before transmission across the link. This eliminates the need for a high frequency 5.0 GHz clock signal on the link that could generate significant EMI. Every byte to be sent is converted to a 10-bit value, called a symbol a look-up table associated with the encoder.

- Data bytes — consisting of every byte send across the link except ordered Sets. The data lookup table must support the 256 possible input values.
- Control bytes— used in ordered sets. The 8b/10b encoding scheme is also designed to prevent “DC wander”, meaning the possibility that too many bits of one polarity could interfere with the ability of the receiver to properly see them.

**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.

**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).

**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. The mechanism(s) and/or interface(s) used to notify the physical layer to disable scrambling is component implementation specific and beyond the scope of this specification.

**IV. RESULTS**

**TOOLS USED FOR EXPERIMENTAL ANALYSIS:** SIMULATOR- MODELSIM 6.3F, SYNTHESIS- ISE 9.2i, DEBUGGING TOOL- CHIPSCOPE 9.2i, BOARD- SPARTAN 3E.
An synchronous clock with a total period of 20 ns or 50MHz was generated in the top level block for synchronous operation. Asynchronous reset is generated for negative edge Reset.

Using random different values for input is generated and it is sent via top module to Scrambler module in the Architecture as shown in fig 3. The results for randomly generated input is shown above.

Generated 8 bit data from scrambler is given as input encoder module and is observed as din signal, kin signal, the result will be observed in dout which a 10 bit output from the above figure.
VI CONCLUSIONS

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 entire above module are coded in VERILOG hardware description language.

This work can 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. The above project is best suits as an IP (Intellectual Property) core of Physical Coding Layer in USB 3.0 specification.

VII FUTURE SCOPE

The Design Physical coding layer can be rigorously tested if soft cores of link layer, physical analog layer are available. This can be an extended work for the present project.

Complete USB 3.0 project can be done if one has finished soft cores of link, protocol, application layers in addition to our project. If analog PHY’s are available in the market hardware level validation can be done to our project.

REFERENCES

[7] International technology roadmap for semiconductors(ITRS)