International Journal of Current Trends in Engineering & Research (IJCTER)
e-ISSN 2455–1392 Volume 5 Issue 2, February 2019 pp. 1 – 5
Scientific Journal Impact Factor : 4.23
http://www.ijcter.com

VLSI IMPLEMENTATION OF UNIVERSAL SERIAL BUS NRZI ENCODING AND DECODING

Dr.S.R.RUCKMANI

LECTURER, CIT SANDWICH POLYTECHNIC COLLEGE,
COIMBATORE-641014, TAMILNADU, INDIA

Abstract—In this paper the hardware design approach for implementing of universal serial bus NRZI encoding and decoding in chip. Communication plays an important role in day to day life. The information or data is transmitted through various techniques and line coding is one of the finest techniques for sending data. The selection of these techniques depends on the bandwidth requirement, DC level, bit error rate performance and the inbuilt error detection property. In this line coding techniques whose encoder and decoder have been designed and analyzed are NRZI, NRZI coding technique can be access with the mode of selection. Switching activity is the one of the main factors that is responsible for the dynamic power dissipation. Power consumption by the encoders and decoders is directly proportional to switching activity. To optimize the power of the universal serial bus encoder – decoder, Shift coding scheme is applied that circularly shifts the data to minimize the transition. . Verilog programming language is used for the coding and the final design will be downloaded to the Xilinx-Spartan3 FPGA toolkit.

Keywords— VLSI, USB, NRZI ENCODING, NRZI DECODING

I. INTRODUCTION

An aspect of the USB communication is line states and encoding, which is generally ignored by the system designers as a USB device can be designed without knowing the Signaling and Encoding schemes. The signaling and encoding schemes are implemented at the lower levels (usually on the controller IC) and usually gets abstracted during system designing. However, understanding the terms related to bus states and encoding would help in getting a comprehensive view of the USB System[1].

Bus States

The Bus state refers to the signal voltage level on the USB bus or the conditions that these voltages signify -:

Differential 0 and Differential 1: The Differential 0 state refers to the condition when D+ data line is logic low and D- is logic high. The Differential 1 state occurs when D+ is logic high and D- is logic low.

Single Ended Zero: This state is also known as SE0. It occurs when both D+ and D- are logic low. The SE0 state is used by bus when entering the Disconnect, Reset and EOP (End of Packet) states.

Single Ended One: Also known as SE1. It occurs when both D+ and D- are logic high. This state can be taken as a compliment of the SE0 state. SE1 should never occur as it is an invalid bus State.

Idle: This state occurs when bus remains idle or there is no exchange of packet. For a full speed device, the voltage at D+ will be more positive than D- while for a low-speed device, the voltage at D- will be more positive than D+

Data J and Data K: Data J and K states are used to define the Data States like the Differential 0 and 1 that refers to the bus states. They are defined by Differential 0 or 1 state and the speed of the bus.

The data states are used to describe an event even when the voltages at data lines are differential. For example, a Start of Packet state exists when the bus changes from Idle to the K state.

Start of Packet: Also known as SOP. This state occurs when there is a change from Idle to K state. Every transmission of the packet begins with SOP.

End of Packet: Also known as EOP. This state occurs when SEO state occurs for two-bit times, followed by a J state for 1-bit time.

Disconnect: A downstream port at which the device is connected enters disconnect state when an SE0 state has lasted for at least 2.5 uS.

Connect: A downstream port enters connect state when there is an Idle state for a minimum 2.5 uS and not more than 2 mS

Some Other States and Signals

Keep Alive Signal: This signal is represented by a low-speed EOP. The main motive of this signal is to keep the device in an idle state. This signal is sent at least once every millisecond, to keep the device from entering suspend state.

Suspend State: The device enters Suspend state when there is no exchange of packet or the bus remains idle for 3 mS. This state has been incorporated in USB for power conservation. In Suspended state, the device must not draw more than 500 uA of current. A suspended device must recognize the reset and resume signal.

Resume: Resume signal is used to wake the device from Suspend state. The Host wakes the device from suspend state by keeping the bus in K state for at least 20 ms followed by a low-speed EOP.

Reset State: A device enters Reset state if SE0 is applied for more than 2.5 uS. The reset sets the device to its default unconfigured state.

Detached State: This state occurs when the USB device is detached from the Host. In this state, the Host sees both data lines low.

Attached State: This state occurs when the USB device is attached to the Host. The host recognizes this state if it sees either D+ or D- high.

II. RELATED WORKS

Mohammed Alamgir et. al [2] purposed bus shift (BS) coding scheme that circularly shifts the data to minimize transition. Power saving of the bus invert is poor in average cases while BS scheme better in both maximum and average cases of power save 14% for a 32 bit bus. Shankaranarayana Bhat et. al [3] purposed a technique for bus encoding which reduce number of transition on data bus, acts better than the current methods such as bus invert coding and shift invert coding for random data in terms of switching activity. Here reduction in switching activity irrespective of bus width and more power efficient. Amrinder Kaur et. al [4] describes the implementation of various line encoders and decoders using VERILOG on a single chip. Anjali.v et. al [5] in this the implementation of Manchester encoder and decoder circuit along with irrational detection and clock recovery unit (CRU) using HDL. This encoding technique is broadly used in fields like biomedical applications, satellite communication etc.Wei Dang [6] purposed the rules based on the HDB3 encoder and decoder. For the detection of four zero's sequence in source information, there are four modules in encoder. For the generator of "V" and generator of "B", obeying the alternate mark inversion (AMI) rule. Simulated output of HDB3 encoder & decoder proved that results are identical with theory analysis.

III. ENCODING SCHEME

The USB employs NRZI (Non Return to Zero Inversion) encoding mechanism to encode the data on the bus. In NRZI encoding, a '1' is represented by no change in level while a '0' is represented by change in level. Together with NRZI encoding, bit stiffing and SYNC field is used for synchronization between host and device.

NRZ coding and decoding

The NRZ encoding scheme is actually not a coding scheme at all. It simply states that a '0' is transmitted as a '0' and a '1' is transmitted as a '1'. It is only worth mentioning because a designer may see the term NRZ and assume that a specific encoder or decoder was required, whereas in fact this is not the case. It is also worth noting that there are some significant disadvantages in using this simple approach. The first disadvantage, especially when compared to the Manchester coding scheme is that long sequences of '0's or '1's give effectively DC values when transmitted, that are susceptible to problems of noise and also make clock recovery very difficult. The other issue is that of bandwidth. Again if we compare the coding scheme to that of the Manchester example, it is obvious that the Manchester scheme requires quite a narrow bandwidth (relatively) to transmit the data, whereas the NRZ scheme may require anything from DC up to half the data rate (Nyquist frequency) and anything in between. This makes line design and filter design very much more problematic.

IV. VLSI IMPLEMENTATION OF NRZI CODING AND DECODING

In the NRZI scheme, the potential problems of the NRZ scheme, particularly the long periods of DC levels are partially alleviated. In the NRZI, if the data is a '0', then the data does not change, whereas if a '1' occurs on the data line, then the output changes. Therefore the issue of long sequences of '1's is addressed, but the potential for long sequences of '0's remains. It is a simple matter to create a basic model for a NRZI encoder using the following VHDL model:

Bit stiffing

When long series of zeros are transmitted using NRZI, it causes a transition in the levels. But when long series of one's is transmitted, no transition takes place as per NRZI encoding scheme. No transition in levels for a long time can confuse the receiver and makes it desynchronized.

Bit stiffing is a process in which a zero is inserted in raw data after every six consecutive ones. The inserting of zero causes transition in level. The receiver must recognize the stuffed bits and discard them after decoding the NRZI data.

In case if no transition takes place in NRZI signal after six consecutive one's, then the receiver decides that bit stiffing has not been done and discards the data

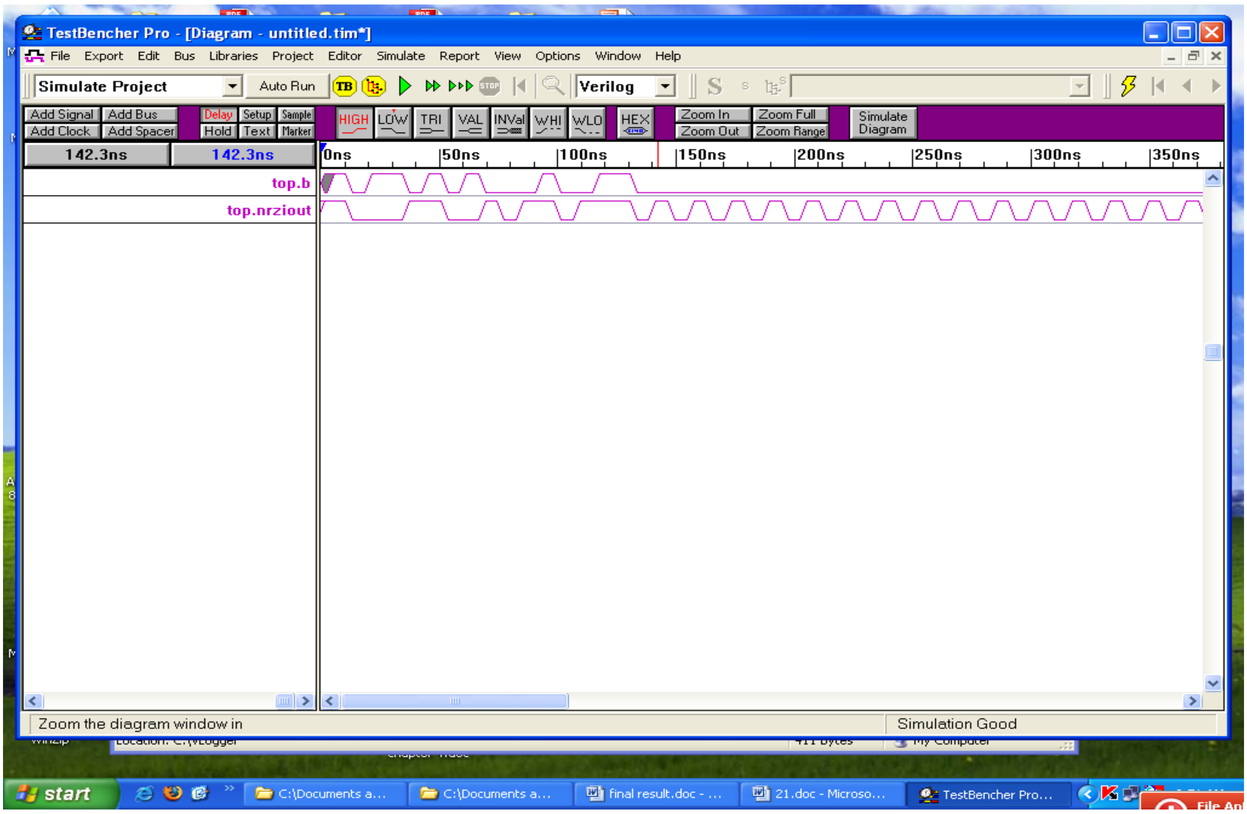
V. RESULTS FOR FPGA IMPLEMENTATION

Once the correctness of the design was verified, performance was analyzed. All FPGA designs were implemented using the Verilog HDL language in Xilinx ISE software.

Conventional model	Power analysis	Designed USB model	Power analysis
encoder	6500	encoder	2400
Decoder	5010	Decoder	1016

Table 1 performance analysis report

The power analysis of designed Universal serial bus (USB) encoder and decoder by compared with the conventional encoder and decoder. Performance analysis report table 1 shows that power saving margin of 21% in average cases of USB encoder while the saving margin of 32% in average cases of USB decoder by using bus shift coding technique.



Zoom the diagram window in

Simulation Good

Fig. 1 Simulation timing diagram for NRZI

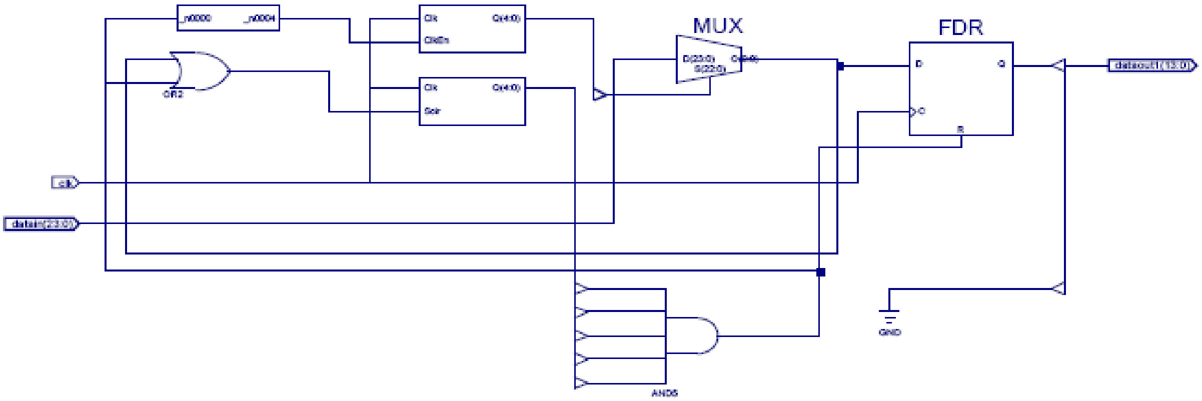


Fig. 2 Synthesis block for NRZI

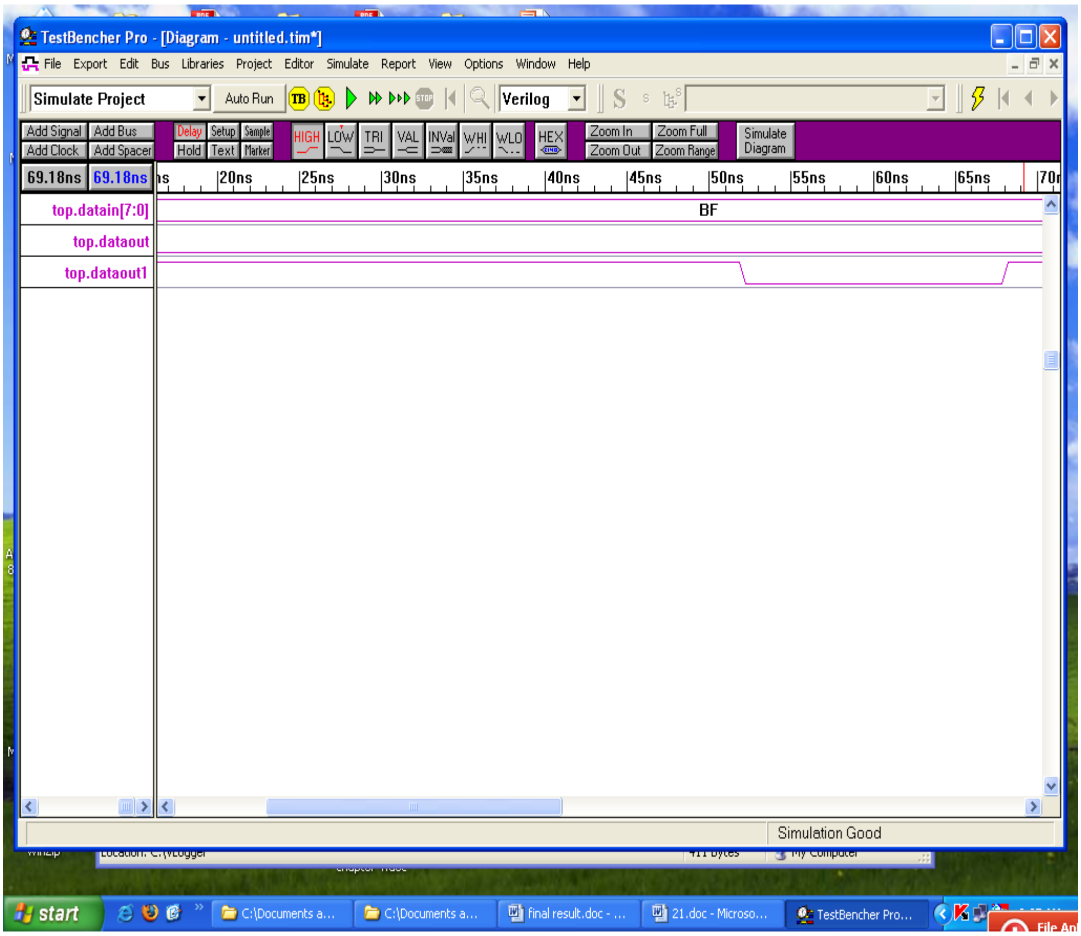


Fig. 3 Simulation timing diagram for bit stuffing

Fig. 2 shows the simulation timing diagram for NRZI , Fig. 3 shows synthesis block for EOP and Fig. 4 Simulation timing diagram for bit stuffing

VI. CONCLUSION

In this work, universal serial bus encoder and decoder has been designed and analyzed by NRZI coding techniques. The designed universal serial bus encoder and decoder were compared with the conventional encoder & decoder, which shows the power saving margin of 21% in case of encoder while 32% in case of decoder. The design has low power and efficient for all the encoding and decoding schemes. In this paper the hardware design approach for implementing NRZI encoder and decoder s function for USB in chip and this is the best solution, which reduces power requirement, size and also cost to approximately 1/N of the conventional approach.

REFERENCES

[1] USB specification 1.1, September 1998; http://www.usb.org

[2] M. Alamgir, I. I. Basith, T. Supon and R. Rashidzadeh, "Improved bus-shift coding for low-power I/O," 2015 IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, 2015, pp. 2940-2943.

[3] Shankaranarayana Bhat M and D. Yogitha Jahnavi “Universal rotate invert bus encoding for low power VLSI”, International Journal of VLSI design & Communication Systems (VLSICS) Vol.3, No.4, pp.97-109, August 2012..

[4] Amrinder Kaur, Mandeep Singh, Balwinder Singh “VHDL Implementation of Universal Encoder for communication”, ISP Journal of Electronic Engineering, Vol.1, Issue.2, ISSN 2250-0537(online), pp. (37-41). Dec 2011.

[5] Anjali. V, P. Satishkumar, "Manchester Encoder and Decoder with Clock Recovery Unit and Invalid Detector”, ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online) Vol.6, No.2, 2015

[6] Wei Dang, "Implement of HDB3 coder in base-band system," in Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference, pp.3273-3274, 21-23 April 2012