International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 04 Issue: 07 | July -2017 www.irjet.net p-ISSN: 2395-0072

Implementation on Transmission and Reception of Data Through Normal USB & Parallel USB in VHDL

Mangesh M. Sahare[1], Prof. Vinay Keswani[2]

[1]M.Tech. II[nd] year, Dept. of Electronic & Communication, Vidarbha Institute of Technology, Umrer, Maharashtra, India
[2]Assistant Professor, Dept. of Electronic & Communication, Vidarbha Institute of Technology, Umrer, Maharashtra, India

---

Abstract - USB system is used to communication with the help of computer peripherals and the host computer. Xilinx device programming takes our design from design entry to perform software simulation. The ISE 13.2 project navigator processes our design entry in various steps to perform ISE design simulation. In our project Xilinx 13.2 is used ISE13.2 simulator contains the signals in the test bench waveforms. In the current work normal USB and the parallel USB are used for communication. Parallel USB reduces the delay of processing and there by improve the speed of communication. In our proposed work, we are implementing a parallel processing unit inside the USB host, which will enable the host to reduce the delay of processing, and increase overall communication speed. The existing USB stack is implemented as it is, but the processing unit of the stack is modified and made parallel. The internal DMA engine and the wish bone architectures are the components in which we are performing parallel processing to improve the results.

Keywords: USB, Xilinx ISE (integrated software environment) 13.2**,**

1. INTRODUCTION

USB means Universal Serial Bus. It is the attachment networking and connected to host computer. These attachments are two types. They are known as Function and Hubs. Function contains the peripherals such as mice, printers etc. Hubs contain the peripherals such as double adapter does on power point, converting one socket components on the USB. Each device contains the number of endpoint. These are the collection of sources and destination for the communication between the host and the device. Each function has to know the piece of data and the host computer needs to know where the signals are coming from. Hence numbers are assigned to each component on the USB. Each device contains the number of endpoint. These are the collection of sources and destination for the communication between the host and the device. Hubs and Functions are commonly called as device. The combinations of address, end point number are arrangement for traditional PC layout. Needs to know where the signals are coming from.

Types of Data transfer

In communication process through USB we need to understand the different types of data which travels across the USB.

i. Control Transfers*:*

The function of control transfer data type are configuring, controlling and checking the status of a USB device. Host send the request to the device and appropriate data transfers is follows in appropriate pipes.

ii. Isochronous Transfers*:*

USB has enough time to handle the maximum data flow. USB provides special type of data transfer. It gives the guarantee of constant transmission rate with the required bandwidth. This Isochronous transfer method uses unidirectional pipes with no error handling procedures.

2. LITERATURE SURVEY

A number of research papers of various journals and conferences were studied and survey of existing literatures in the proposed area is reported below:

2.1 Panday et al. (2013) designed USB 2.0 with great approach.

It contains both Low Level Programming (JTAG) capability as well as Application Level i.e. High Level Programming (Linux). Limitation of JTAG approach is based approach is suitable for low level programming. It will go expenditure, because of amount of code to be written for each feature. Thus States of system can be checked in intermediate stage are very slowly. Limitation of Linux based approach The Linux code is very bulky. It runs very fatly. It enables several modules in background. The number of retries is largely. Several issues are unnoticed or over written. The state of system cannot be checked in intermediate stage. It should be needed for debugging.

2.2 Przemys law et al. (2012) designed the concept of USB receiver/transmitter.
This concept is designed by implementing in hardware description language. It provides simulation model. The code is synthesizable simultaneously. It may be physically by its own finite state machine. Imaginable solution is splitting. The architecture in accordance with the direction of the data transmission.

2.3 Jolfaei et al. (2009) designed USB 2.0 for high speed and it is easy to use peripheral interface.
Spartan-3 FPGA hardware implementation is used. It has easily data handling capability

3. IMPLEMENTATION OF PROPOSED WORK

For our project several developmental tools are used for the implementation. Implementation of our project includes the generating the Test Bench signal waveforms, simulation, cycle and the design summary etc. Software ISE (Integrated Software Environment) 13.2 is used for designing circuit and verilog code. Development of the Test Bench and schematics of the module are also designed by our software. Xilinx device programming is performed by this software through design entry. Various steps are required in this ISE project navigator processes for the ISE design flow.

Table no. 1. Test Bench for Normal USB

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1677	42000	3%
Number of Slice LUTs	2180	21000	10%
Number of fully used LUT-FF pairs	1286	2571	50%
Number of bonded IOBs	235	210	111%
Number of BUFG/BUFGCTRLs	2	32	6%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos

Design Summary

Table no. 2. Test Bench for Parallel USB

Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1673	42000	3%
Number of Slice LUTs	2170	21000	10%
Number of fully used LUT-FF pairs	1283	2560	50%
Number of bonded IOBs	235	210	111%
Number of BUFG/BUFGCTRLs	2	32	6%

Detailed Reports

Report Name	Status	Generated	Errors	Warnings	Infos

Synthesized)

Xilinx ISE 13.2 simulator gives result in the form of the Test Bench containing the waveform signal. It can be used to simulate the modules. In the entire communication, the USB designers are worked on the serial processes. Due to USB is worked on serial protocol, we cannot change the communication process but we can change the internal processing of USB. Due to modifying internal processing of the USB, it reduces the delay of processing. Hence due to reduction of delay in processing, there is improvement in speed of communication. We are implementing the parallel processing unit inside the USB host. Therefore there is a reduction in the delay of processing and increase the overall communication speed. The existing USB stack is implemented as it is. But processing unit of stack is modified and made parallel. Wish Bone architecture and internal DMA engine is the components which helps for the performing parallel processing to improve the result.

Plan of work

1. Creation of CRC modules

2. Development of Wishbone architecture

3. Development of internal DMA

4. Development of standard USB

5. Improving USB using parallel processing

6. Result analysis and comparison

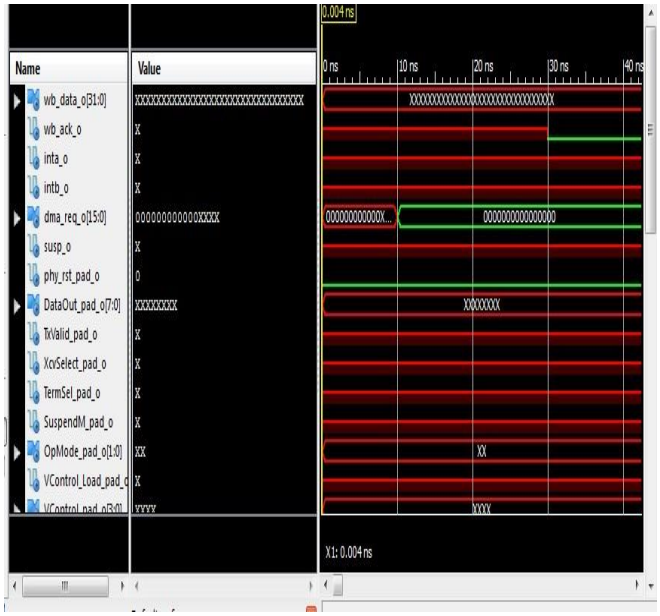

Fig. no. 4.1 Signal generated in Normal USB

In order to build the USB High Speed core, we had to implement the USB 2.0 specification, which only specifies the language that high speed USB speaks but provides no details of implementation. Therefore, our first target was to to complete the VHDL code that implements the USB High Speed protocol specification. As part of the specification, our core is supposed to be backward compatible with all three speeds of USB devices.

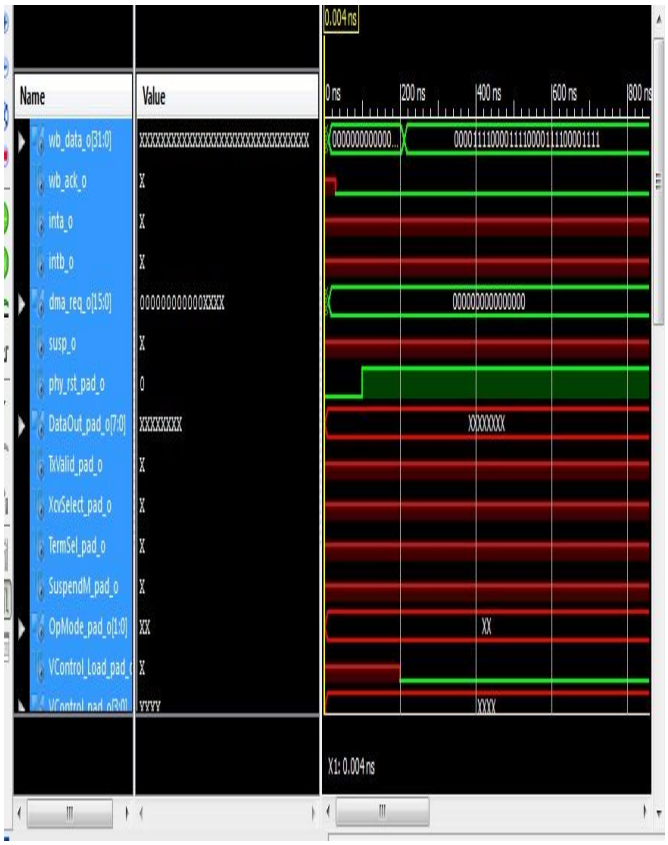

Fig. no. 4.2 Signal generated in parallel USB

4. CONCLUSION

Universal Serial Bus (USB) is developed to connect to the number of peripheral devices to the PC. USB 2.0 has applications in industry wide. It is host oriented protocol. It is utilizing physical serial bus. It is host controlled. Basic data transferring is the essential specification of USB 2.0. The USB specification defines three data speeds. I.e. low speed. Full speed and High speed. We implement the simulation of Normal USB and Parallel USB on Xilinx ISE 13.2 software. The result is verified with test bench generation and signaling. We are implementing a parallel processing unit inside the USB host, which will enable the host to reduce the delay of processing, and increase overall communication speed.

5. REFERENCES

[1] Pandey, M. K., Shekhar, S., Singh, J., Agarwal, G. K., & Saxena, N. (2013, July). A novel approach for USB2. 0 validation on System on Chip. In 2013 Fourth International

Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1-4). IEEE

[2] Jolfaei, F. A., Mohammad izadeh, N., Sadri, M. S., & FaniSani, F. (2009, December). High speed USB 2.0 interface for FPGA based embedded systems. In Embedded and Multimedia Computing, 2009. EM-Com 2009. 4th International Co

[3] Babulu, K., & Rajan, K. S. (2008, July). FPGA Implementation of USB Transceiver Macrocell Interface with USB2. 0 Specifications. In Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on (pp. 966-970). IEEE.

[4] USB 2.0 Transceiver Macrocell Interface (UTMI): Retrieved from http://www.intel.com/technology/usb/download/2_0_xcvr_macrocell_1_05.pdf

[5] Jan Axelson, USB Complete - Everything you need to develop custom USB peripherals, Lakeview research, 3rd Edition, Aug 2005

[6]"Spartan-3 FPGA Family: Complete Data Sheet", Xilinx Corp., Aug 2005