Design of a Data Acquisition System for USB Devices over Gigabit Ethernet

Ansiya Eshack
Department of Electronics & Communication
KMEA Engineering College
Ernakulam, India

Jagadeesh Kumar
Department of Electronics Engineering
Govt. Model Engineering College
Ernakulam, India

ABSTRACT
The Universal Serial Bus (USB) is one of the most widespread technical innovations in personal computer and home consumer applications over the past few years. Keyboards, mouse devices, printers, webcams, and several other computer peripherals are available with this type of connection. This success is due to some bus characteristics, such as simplicity, plug & play features, hot plug support, and, particularly, the possibility of supplying power to the devices. USB offers good noise immunity due to its differential type signal transmission. The high data transfer rate of USB interface makes it a suitable choice for data acquisition and processing. This paper proposes a system which acquires data from multiple USB devices simultaneously and transfers this data at the speed of Gigabits per second to PCs connected to the LAN network. This system shall act as an interface between the USBs and the PCs. As the FPGA provides a high level of flexibility and also contains a number of modules on chip, which can operate independently, and at very high speeds, this data acquisition system shall have FPGA chip as the main processing unit.

General Terms
IEEE 802.3 protocol, MAC Core, Xilinx Platform Studio, LAN

Keywords
Data acqusition, high-speed USB, Gigabit Ethernet, FPGA.

1. INTRODUCTION
Universal Serial Bus (USB) is a specification to establish communication between devices and a host controller (usually personal computers), developed and invented by Ajay Bhatt while working for Intel. The first silicon for USB was made available by Intel in 1995. USB was intended to make it fundamentally easier to connect external devices to PCs by replacing the multitude of connectors at the back of PCs, addressing the usability issues of existing interfaces, and simplifying software configuration of all devices connected to USB, as well as permitting greater bandwidths for external devices. 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. USB was designed for personal computers, but it has become commonplace on other devices such as smartphones, PDAs and video game consoles, and as a power cord.

Gigabit Ethernet is a data transmission standard based on the IEEE 802.3 protocol. This standard is the next step of the Ethernet and Fast Ethernet standards that are widely used today. With transmission rates of up to 1000 Mbps, Gigabit Ethernet is faster than IEEE 1394a and IEEE 1394b. As Gigabit Ethernet is backwards compatible, it reduces the infrastructure investment that an organization must make. Gigabit Ethernet was first standardized only for fiber. This meant that network managers wanting to deploy Gigabit were often faced with costly re-wiring of their buildings in order to upgrade the infrastructure to fiber cable. The situation was alleviated when IEEE standardized Gigabit over Cat-5 copper cabling. Other cable standards for Gigabit Ethernet include Cat-5e, Cat-6 and Cat-6a. Widespread deployment of Gigabit Ethernet became possible at less expense over the existing copper infrastructure. Gigabit connections can be established at lengths of up to 70Km. And, for the future growth, with 10 Gbps and 100 Gbps Ethernet standards on their way, along with Cat-7 cables, the communications can easily scale to use the higher transmission rates available.

The objective of this paper is mainly to design a system to acquire data from multiple USB2.0 devices and transmit the data over the LAN. An interface between USB controller and an FPGA through the Host Port Interface is enabled and a MAC Core is designed to interact with the PHY and to provide data transmission over the LAN at Gigabit rates. All interactions are controlled with the help of Microblaze soft microprocessor core implemented in FPGA.

2. PROPOSED ARCHITECTURE
The proposed data acquisition system model that connects USB peripheral devices to different PCs over a Local Area Network is depicted in Figure 1.

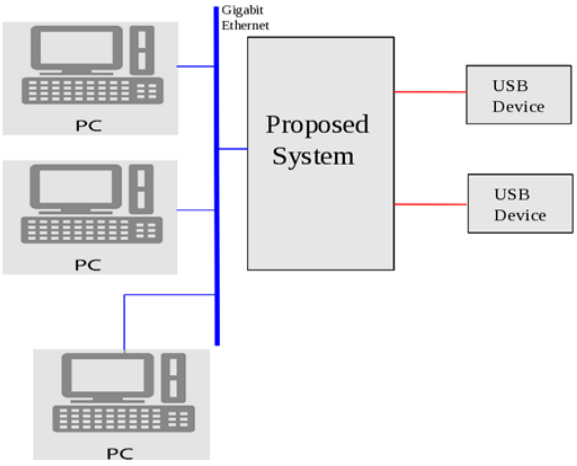

Fig 1: Proposed data acquisition system model

The block diagram of the proposed system architecture is shown in Figure 2. The system shall acquire data from the USB devices, convert to IP packets, and send it to PCs over Gigabit Ethernet Networks. Data from the USB devices are to be streamed continuously over inexpensive Cat-5 copper LAN cable at 1 Gbps line rate. At the PC, the Cat-5 cable plugs into a standard Gigabit Ethernet Network Interface Card. The

application software will receive the network packets and display the data. The system shall provide interface to all devices which comply with the USB 2.0 Standard.

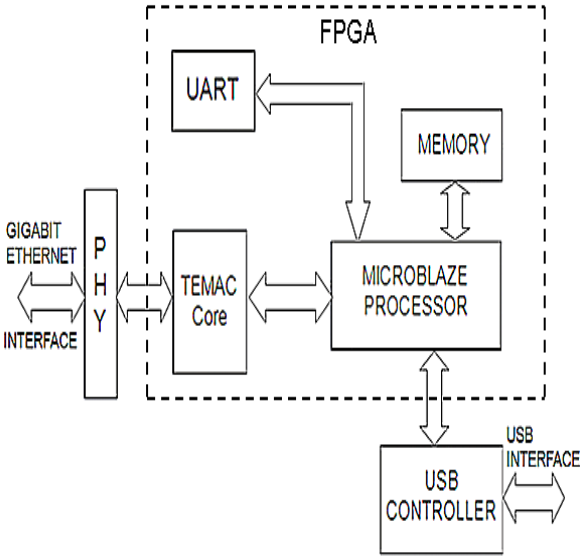

Fig 2: Block diagram of the proposed system

The system uses a USB controller (Cypress CY7C67300) to acquire data from the USB devices, Memory for intermediate data storage on the board, Microblaze Processor to control all the applications required for the data acquisition and transfer to LAN, Tri-mode Ethernet MAC Core (TEMAC) to provide all functions necessary to attach an Ethernet Physical layer to the host FPGA (Xilinx Virtex-4 FPGA, XC4VSX35-FF668-10), PHY (Marvell Alaska 8E1111) to encode/decode data and transmit over the coaxial cable and the UART to view and debug the transmission.

3. NETWORK COMPONENTS

The following sections describe the functions and connections of each component of the USB-Gigabit Ethernet network according to the proposed architecture.

3.1 USB controller connection to Microblaze processor

Figure 3 below shows the connection of the USB controller to Microblaze processor. The USB controller is interfaced to the Xilinx Platform Studio External Peripheral Controller (XPS EPC) through the Host Peripheral Interface (HPI). XPS EPC connects the Cypress CY7C67300 USB controller to the Microblaze processor as shown in Figure 3. The PLB DDR SDRAM is used as the external memory, for storage of the data to be transmitted onto the LAN. The reset signal of the USB controller is connected to the XPS GPIO Core (Push Buttons). Setting this bit to 1 resets the controller & setting it to 0 enables its normal operation. The XPS INTC core handles the interrupt signal from the Cypress CY7C67300 USB controller. The XPS UARTLITE core helps to view and debug the transmission.

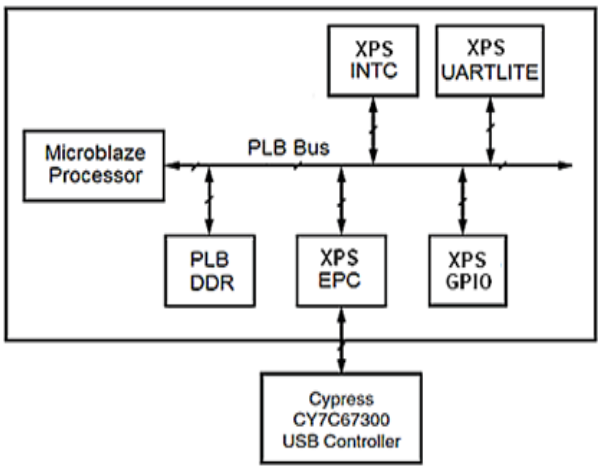

Fig 3: USB controller connection to Microblaze processor

3.2 Host Peripheral Interface

The HPI interface is a slave-only interface and provides a hardware interface into the CY16 processor of EZ-Host (USB controller) for coprocessor communications. The external master (FPGA) writes to the address register to setup a pointer into the internal memory of the EZ-Host. When the main CPU reads from, or writes to, the data register it is actually accessing internal memory locations.

A master CPU write to the mailbox register will generate an interrupt to the EZ-Host CPU informing it that it should come and read the command that the main CPU has sent. When the EZ-Host has completed its command, it writes a response into the mailbox register, and this generates an interrupt to the master CPU. When the master CPU reads this response from the mailbox the EZ-Host CPU will be alerted via a separate interrupt. The master CPU can also read a status register that summarizes the state of the pending interrupts of the EZ-Host.

The HPI Interface is shown in Figure 4 below. It shows the signals used to transmit signals between the USB Controller (EZ-Host) and the external master (FPGA).

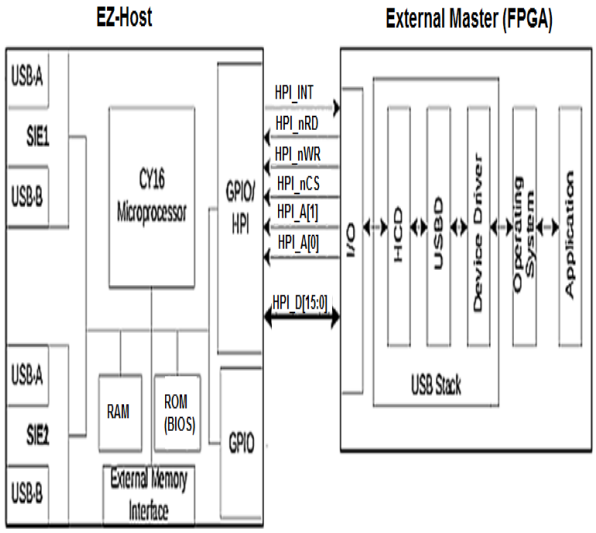

Fig 4: HPI Interface

3.3 Tri-mode Ethernet MAC Core

The Tri-mode Ethernet MAC (TEMAC) core is used to connect the FPGA to the Ethernet PHY device, which

performs the BASE-T standard at 1 Gbps, 100 Mbps, and 10 Mbps speeds. It implements the full 802.3 specification with preamble and Start of Frame Delimiter generation, frame padding generation, CRC generation and checking on transmit and receive respectively. The TEMAC core is configurable for 10/100/1000 Mbps MAC with full and half duplex modes of operation as specified in the IEEE 802.3.

The TEMAC is shown in Figure 5 and contains a Transmit Engine, Receive Engine, Flow Control, Gigabit Media Independent Interface/Media Independent Interface (GMII/MII) block and Client Side Interface block. The transmit engine accepts Ethernet frame data from the Client Transmitter Interface, adds the preamble field to the start of the frame, adds padding bytes (if required) to ensure that the frame meets the minimum frame length requirements, and adds the frame check sequence. The transmitter also ensures that the inter-frame spacing between successive frames is at least the minimum specified. The frame is then converted into a format that is compatible with GMII/MII and sent to the GMII/MII Block. The receive engine accepts Ethernet frame data from the GMII Block, removes the preamble field at the start of the frame, removes padding bytes and Frame Check Sequence. The receiver also performs error detection on the received frame using information such as the frame check sequence field, received GMII error codes, and legal frame size boundaries. The flow control is designed according to specifications of the IEEE 802.3-2005 standard. The MAC may be configured to send pause frames and to act upon their reception during full-duplex operation. These two behaviors can be configured independently. The GMII/MII block implements GMII/MII style signalling for the physical interface of the core and is typically attached to a physical

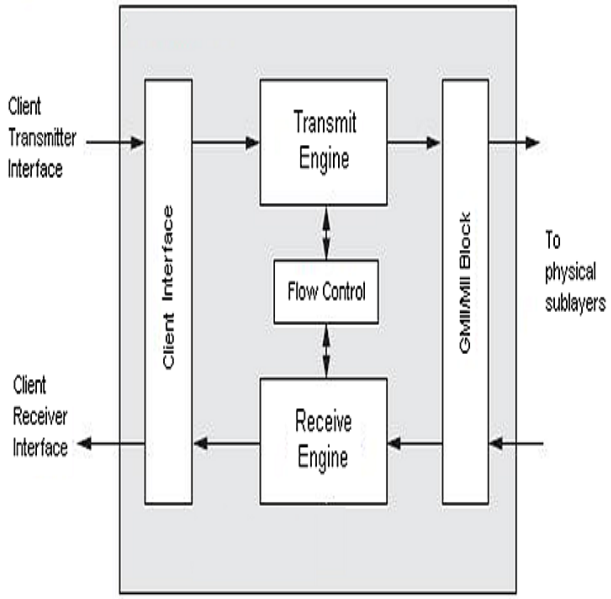

Fig 5: TEMAC Core

layer device (PHY), either off-chip or internally integrated. When connecting to external PHY devices, either of GMII/MII or RGMII interfaces can be used. Finally in the client side interface, the client-side transmitter signals of the core are used to transmit data from the core to the client. The client-side receive signals of the core are used to receive data from the client to the core.

4. RESULTS

4.1 HPI transactions to enumerate USB devices

The followings steps are needed to enumerate a USB device connected to the EZ-Host USB Controller. These steps are initiated by the FPGA and sent to the EZ-Host, when an interrupt occurs from the EZ-Host indicating that a device has been connected to one of its ports. The steps are as follows:

1. EZ-Host Configuration
2. USB Reset
3. USB Set Address
4. USB Get Device Description

Figure 6, Figure 7, Figure 8 and Figure 9 show the simulation results of these above four steps.

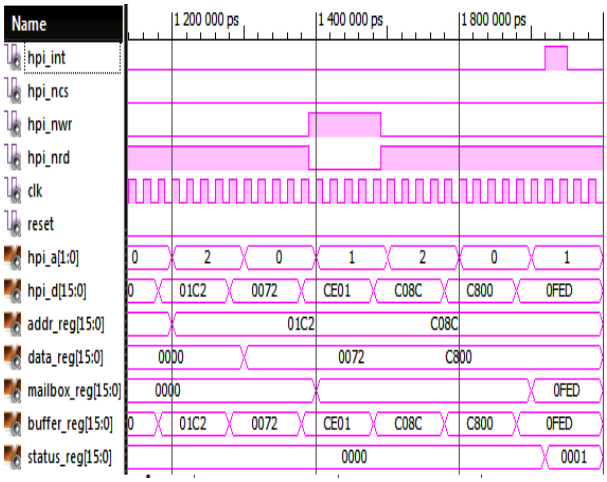

Fig 6: EZ-Host Configuration

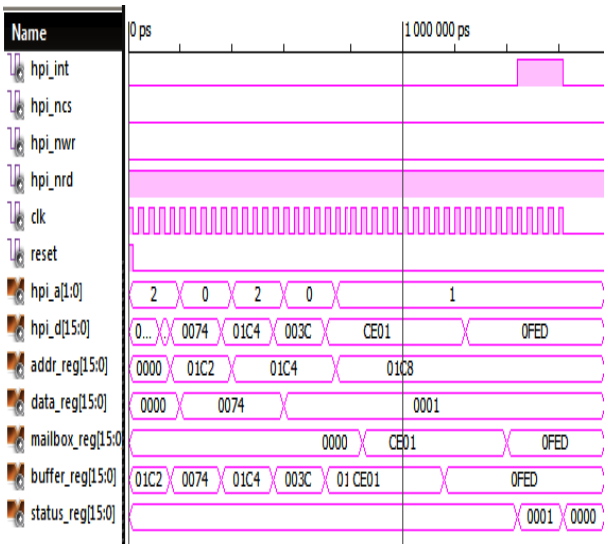

Fig.7: USB Reset

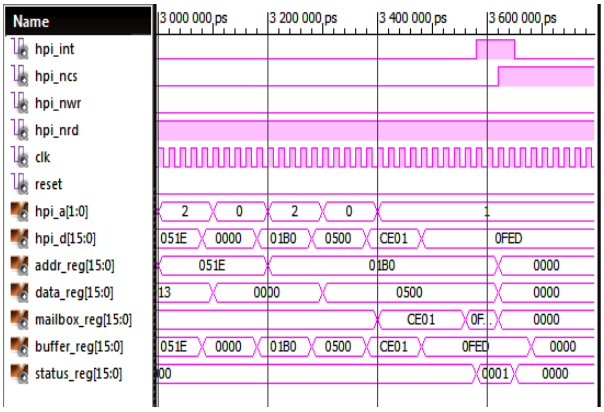

Fig 8: USB Set Address

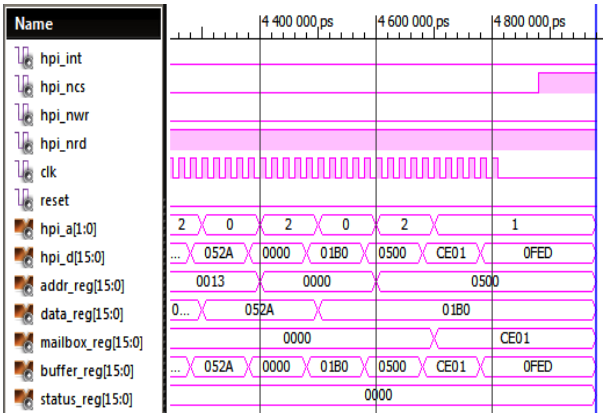

Fig 9: USB Get Device Descriptor

4.2 Cypress USB controller connection in XPS

The Cypress USB Controller was attached to the Microblaze processor in the FPGA through the XPS_EPC. Figure 10 shown below is the result of this connection implemented through Xilinx Platform Studio.

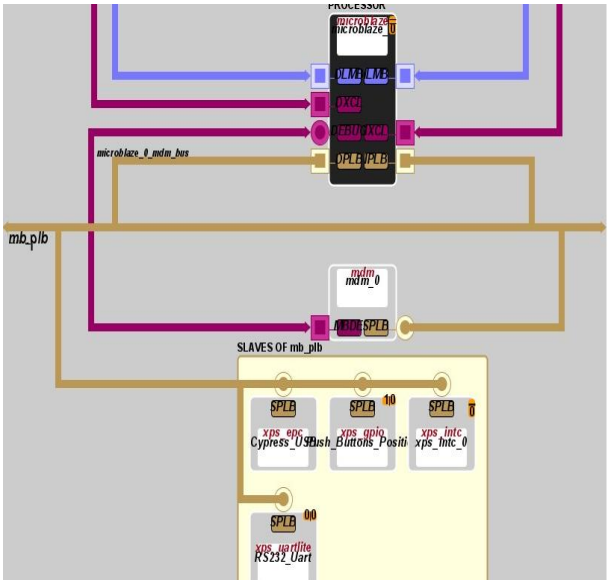

Fig 10: USB Controller connected to Microblaze processor

4.3 Tri-mode Ethernet MAC Core

Figure 11 and Figure 12 show the simulation results of the Receive and Transmit Engines of the TEMAC core.

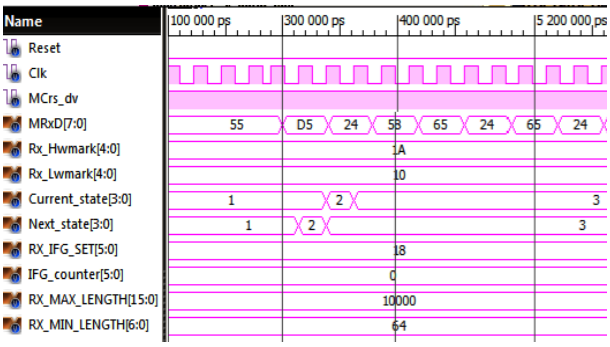

Fig 11: TEMAC Receiver module

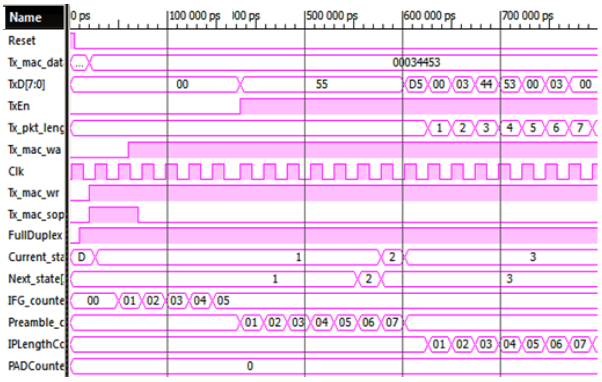

Fig 12: TEMAC Transmitter module

4.4 TEMAC connection in XPS

The block diagram shown below is the result of the connection of both USB Controller and the TEMAC core to the Microblaze processor implemented using Xilinx Platform Studio.

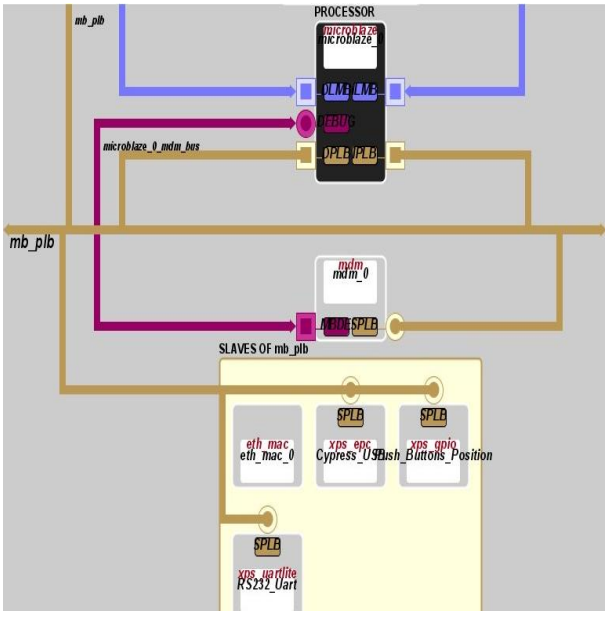

Fig 13: TEMAC and USB Controller connected to Microblaze processor

5. CONCLUSION

The hardware model of a system to acquire data from USB2.0 devices and transmit the data at Gigabit rate over the LAN is designed. An interface between USB controller and the FPGA through the Host Port Interface is designed for transmission of control and data between FPGA and the USB Controller. A TEMAC Core required to connect to the PHY and for data transmission over Gigabit Ethernet has been designed and implemented. A Microblaze processor system is designed to control the peripherals required for the data acquisition system.

The TEMAC Core was successfully attached as an external peripheral to the Microblaze and the bit stream generated from this design was downloaded to Xilinx ML402 FPGA development board. Hyper Terminal was used to debug the design software.

Cypress USB controller CY7C67300 can be configured using the Microblaze processor, and is designed using Xilinx XPS and the design is supported by Xilinx ML403 evaluation board. Keeping this in mind, the design was done to connect Cypress USB and the TEMAC Core to Microblaze processor in FPGA, choosing ML 403 as the development board.

This design is purely hardware oriented and hence can be used to develop a single chip which can be used widely for data acquisition, data storage and data transmission applications. This design is based on USB 2.0 standard but now due to the release of USB 3.0 standard with upto 5Gbps, faster data acquisition and transfer is possible. This chip can be used to transmit the stored data at speeds even greater than 1Gbps by slight modification in the TEMAC Core.

6. REFERENCES

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

[2] Fatemeh Arbab Jolfaei, Neda Mohammadizadeh, Mohammad Sadegh Sadri, Fatemeh FaniSani, "High Speed USB 2.0 Interface for FPGA Based Embedded systems", Embedded and Multimedia Computing, 2009, 4th International Conference, December 2009, pp.1-6.

[3] Bi Bo, Sun Shuying, Wang Chunping, "Design of Data Acquisition Equipment Based on USB", Electronic Measurement and Instruments, Eighth International Conference, pp.866-869.

[4] Weihua Zhang, Zhonghui Wei, Xin He, Peiyu Qiao, Guolong Liang, "The design of high speed image acquisition system over Gigabit Ethernet," IEEE Intl Conference on Wireless Communications, Networking and Information Security, pp 111-115, 2010.

[5] CY7C67300 EZ-Host Programmable Embedded USB Host/Peripheral Controller, Cypress Semiconductor Corp., Sep 2003.

[6] Using HPI in Coprocessor Mode with OTG-Host, Cypress Semiconductors, Application Note, April 2006

[7] IEEE Std 802.3™-2005, Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications

[8] Virtex-4 FPGA Family Overview: Product Specification, Xilinx Corp., Sept 2007.

[9] XPS External Peripheral Controller (EPC) v1.02a, DS581, Sep 2009

[10] ML401/ML402/ML403 Evaluation Platform User Guide, UG080 (v2.5), May 2006

[11] Marvell Alaska, 88E1111 Single-Port Gigabit Ethernet Transceiver, Available: www.marvell.com

[12] John Hyde, USB Multi-Role Device Design by Example, Cypress Semiconductors, 2003

[13] "USB in a nutshell, Making sense of the USB standard", Craig Peacock, Nov 2002,Available:www.beyondlogic.org