

# FPGA Low Cost Microcontroller circuit with USB Interface for embedded System Applications

M.Aldalbahi ,M.Aboueela, B.Almashary

**Abstract** — The universal serial bus (USB) has largely replaced the standard serial port to connect devices to a PC because it is simpler, more foolproof, and faster. But while USB makes connection easier for the user, it adds more design challenges. Many designers continue to use the UART (Universal Asynchronous Receiver Transmitter) with the standard serial port, waiting for products that make USB easier to implement. Some microcontroller providers integrate the USB hardware within the microcontroller chip but most of the commercial low cost microcontroller families are still not equipped with a USB. External arrangements can be applied to add USB interface to a given microcontroller circuit. This may be done through either, serial or parallel communication with the microcontroller. In this work, the design procedure for building  $\mu$ controller circuit, with USB interface, using FPGA technology is introduced and experimentally verified. The design is based on the PicoBlaze  $\mu$ -controller free core. The USB interface is implemented by direct parallel interfacing between the FPGA chip and the FTD 245L parallel to USB. The design procedure is simple and can be utilized by beginners having limited skills in VHDL and assembly programming languages as well.

**Keywords**— FPGA , microcontroller, UART,USB,VHDL.

## I. Introduction

Microcontrollers are widely used for embedded systems applications in automation, control, communication, M2M (Machine to Machine) and biomedical instrumentations [1, 2]. communications is the process of sending data one bit at one time, sequentially, over a communications channel or computer bus [3]. The UART has been the standard serial port framer ever since IBM's original PC motherboard used the Intel 8250 UART. Decades later, the UART continues to be widely used due to its reliability and simplicity. Most of the microcontroller chips available in the market today have at least one built-in UART [4]. The universal serial bus (USB) has largely replaced the standard serial port as the way to connect devices to a PC because it is simpler, more foolproof, and faster. But while USB makes connection easier for the user, it presents the designer with additional challenges [5]. Several microcontroller chips with built in USB interface are available in the market [6]. On the other hand an alternative solution seems to be easier and cost effective. By using USB Interface IC,s instead of wasting time learning a new architecture, one can use the existing designs, existing code and existing development tools. A USB interface IC,s can now be used in two ways:

-----  
 Department of Electrical Engineering, College of Engineering , King Saud University,POB 800 , 11421 Riyadh SA, [aboueela56@yahoo.com](mailto:aboueela56@yahoo.com).

1- Using a UART-to-USB converter, the designer works with the familiar UART frames of UART , A UART-to-USB converter IC chip is connected externally and an operating system driver take care of the complexities of USB SPI ( Serial Protocol ....)[7] .

2- Using a parallel to USB converter [8]. This solution has been available and proved better performance as regarding data transfer speed. The details of this solution will be given in details in section III where a complete microcontroller design with USB interface will be introduced and verified.

### I-2. Universal Serial Bus (USB) fundamentals

In 1996, the first commercial version of USB (USB 1.0) was released. It supported a maximum data rate of 1.5 Mbps. The second version, USB 1.1[9], was released in 1998 and supported a maximum data rate of 12 Mbps. The current version, USB 2.0 [10], was released in April 2000 and supports a maximum data rate of 480 Mbps. Most new computers and associated peripheral devices such as printers and scanners support USB. The USB interface may be either host or peripheral. There is only one host in any USB system. The USB interface to the host computer system is referred to as the Host Controller. The Host Controller may be implemented in a combination of hardware, firmware, or software. A root hub is integrated within the host system to provide one or more attachment points. USB peripherals are that attached to the host like mouse, flash memories, webcam...etc. This work deals only with peripheral devices interfacing.

## II. Microcontroller Implementation

The circuit designers of embedded systems containing microcontroller can implement their systems either by using off-shelf micro- controller chips available in the market, or by building exactly the optimized configuration that implements their system using FPGA.

### A. PicoBlaze core for Microcontroller

One of the available free cores for an 8-bits Microcontroller is the PicoBlaze  $\mu$ -controller core [11].It is optimized for efficiency and low deployment cost. The PicoBlaze Microcontroller is resource efficient. As shown in Fig.1, the PicoBlaze  $\mu$ -controller supports the following features:

- 16 byte-wide general-purpose data registers
- 1K instructions of programmable on-chip program store.
- Byte-wide Arithmetic Logic Unit (ALU) with CARRY and ZERO indicator flags
- 64-byte internal scratchpad RAM
- 256 input and 256 output ports
- Automatic 31-location CALL/RETURN stack
- Predictable performance, always two clock cycles per instruction, up to 200 MHz or

100 MIPS in a Virtex-II Pro FPGA

- Fast interrupt response; worst-case 5 clock cycles
- Optimized for Xilinx Spartan-3, Virtex-II, and Virtex-II Pro FPGA architectures .
- Assembler, instruction-set simulator support

In order to work with the PicoBlaze core, two codes have to be developed. The first one is written in assembly language (specially designed for the PicoBlaze) and represents the program structure for solving a given physical or mathematical problem. The assembler of this Microcontroller is used to change this code into ROM VHDL code. The second code is the top level code to connect the ROM to the Microcontroller and assign IN, OUT ports.

### III. Implementation of Microcontroller with USB

#### Interface

Recently, introduced in the market some IC chips that works as USB peripheral controller with parallel interface such as the PDIUSB12 from Philips semiconductor [12] and the FT245R from FDTI [13]. Since we have parallel interface (ports) in PicoBlaze Microcontroller Core that may provide high connection speed (compared to Serial) FT245R parallel to USB chip will be used. Fig.2 gives the block diagram of this circuit. The software driver for this chip is available from the vendor and can be installed on the host computer. The design procedure for a given application using USB interface may be summarized in the following steps:

- 1- Define the function of the peripheral device needed for a given application, for example Programmable Logic controller board that receives commands from host computer through the USB interface .
- 2- Develop the assembly program using the PicoBlaze instruction set .and represents the program structure.
- 3- Using the assembler of this Microcontroller to change this assembly code into ROM VHDL code.
- 4- Developing the top level code to connect the ROM to the Microcontroller and assign IN, OUT ports.
- 5- Verifying the design using software tools
- 6- Circuit implementation and hardware verification using hardware emulator.

### IV. Experimental Work

The block diagram given in Fig. 3 is proposed to test the design procedure described above .The test is divided into three steps:

**Step1:** testing the VHDL code as written to implement the Microcontroller.

**Step 2:** Integrate the hardware built in step1 with the FT245R chip and prepare the USB physical connection.

**Step 3:** Testing the overall circuit by establishing a serial communication connection with the USB interface of any computer host using familiar communication package like Hyper Terminal under windows.

#### A. Hardware verification of PicoBlaze testing code for the microcontroller with USB hardware

The basic program for PicoBlaze  $\mu$ -controller core was tested by simulation using Aldec tool, [14], now we need to test this program in real signal environment. Using the block diagram given in Fig. 4 a circuit was developed to test the functionality of the USB interface .The circuit is composed of three parts:

- 1- The Spartan III FPGA Chip mounted on NEXYSII
- 2- The FT245R chip
- 3- The USB jack

A PicoBlaze assembly program was developed with Top level VHDL and downloaded onto the FPGA chip. This program reads the status value of 8 switches in hex format. This value is then transferred to the attached PC through the built USB interface and displayed using the Hyper Terminal under windows as the ASCII equivalent of read HEX value. On the same time any character entered by the key board of the PC through the Hyper Terminal window will be displayed on the 8 LED display of the NEXYSII board. A reset function of all LED is also available.

The following code written in assembly is a simple subroutine that tests the status of two switches and transmits their status to other two output pins on the test board.

```

CONSTANT sws_port, 02           ; Read switches on this
port
CONSTANT leds_port, 08         ; Update LEDs at this
port
;----- - Loop reading Switches from the board and
display it in board LED's
start:  INPUT s0, sws_port
        OUTPUT s0,leds_port
        JUMP start

```

To impalement step1 above a new version for the FPGA test and development board NEXYSII was chosen and supplied from Digilent [9], Since Xilinx FPGA will be used here, another tools from Xilinx (ISE 9.1) is also used.

### V. Conclusion

Most of the Microcontrollers available in the market are equipped with built in UART circuit. USB is a highly growing technology applied for implementing high speed serial data transfer. The availability of a Microcontroller with a USB interface is limited. Several techniques may be used to build a microcontroller circuit with USB. The circuit designers of systems containing Microcontroller can implement their system. In two ways: Using off-shelf Microcontroller chips or builds exactly the optimized configuration that impalement their system using FPGA. One of the available free cores for an 8-bits  $\mu$ -controller is the PicoBlaze  $\mu$ -controller core which had been used effectively throughout this work. Integrating the built Microcontroller with the FT245R IC Chip provides an optimized solution for our application. The built circuit had been used successfully to establish USB serial

communication with the laptop as a host computer where data can be transferred. The experimental work confirms exactly the work objectives.

**References**

[1] M. Abou Elela ,” Programmable Logic Remote Controller Using GSM Network “Proceeding of MMS 2003 , 6-9 May 2003, Ain Shams University , Cairo, Egypt.  
 [2]M. Abou Elela, “GSM Network Based PLC Systems ”, Proceedings of the 2nd IEC-2004, December 19-21, 2004, Riyadh, Kingdom of Saudi Arabia.  
 [3]Nebojsa Matic, “PIC microcontrollers, for beginners too” ebook., www.mikroe.com/en/books/picbook.  
 [4]Paul H. Young ,”Electronic communication techniques” ,PrintcHall,1996.  
 [5]www.usb.org.org/developers/usb\_20.zip

[6] Behrouz A Forouzan,Data Communications and Networking , McGraw-Hill Forouzan Networking , 2006 .  
 [7] <http://www.maxim-ic.com/>  
 [8] Compaq, Intel, Microsoft, NEC, “Universal Serial Bus Specification Revision 1.1” September 23, 1998.  
 [9] Steven McDowell & Martin D.Seyer, “USB Explained”, 1999.  
 [10] [www.xilinx.com](http://www.xilinx.com).  
 [11] <http://digilentinc.com/>  
 [12][http://www.nxp.com/acrobat\\_download/datasheets/PDIUSB12\\_9.pdf](http://www.nxp.com/acrobat_download/datasheets/PDIUSB12_9.pdf).  
 [13][http://www.ftdichip.com/Documents/DataSheets/DS\\_FT245R.pdf](http://www.ftdichip.com/Documents/DataSheets/DS_FT245R.pdf).  
 [14] [www.aldec.com](http://www.aldec.com).

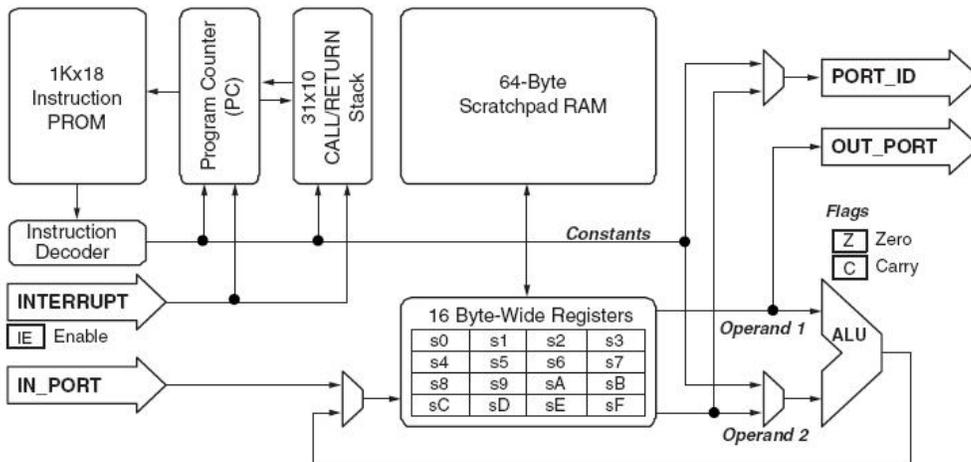


Fig.1. The PicoBlaze  $\mu$ -controller block diagram

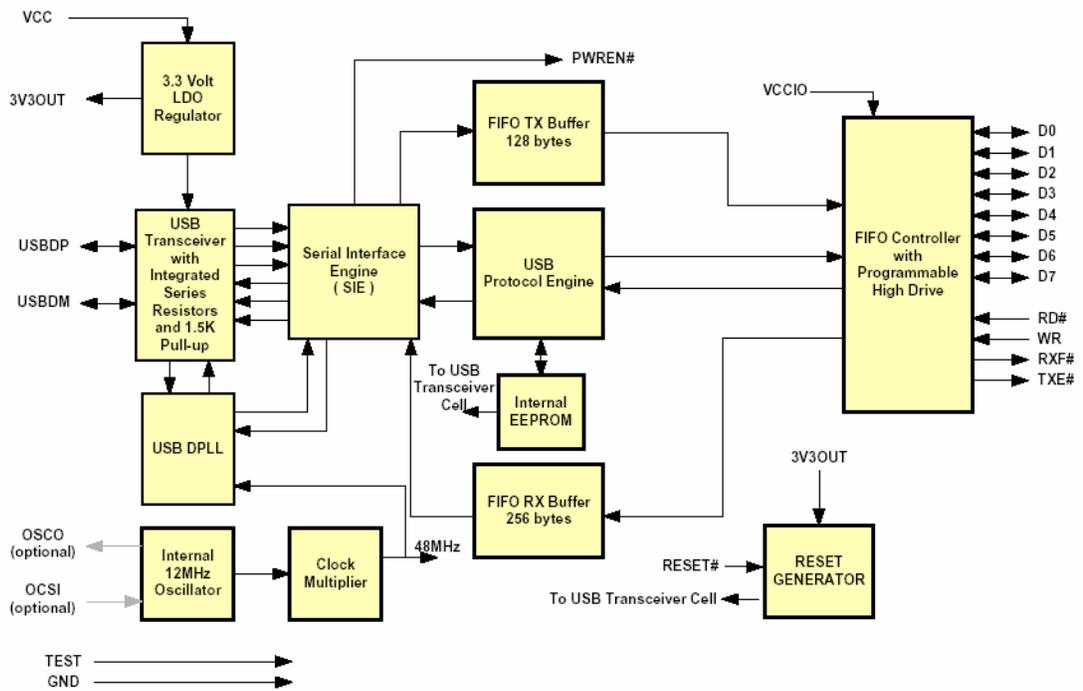


Fig.2. block diagram of the FT245LR IC chip



Fig. 4. proposed block diagram for Microcontroller with USB