IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 4, Issue 6, Ver. III (Nov - Dec. 2014), PP 41-49
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Implementation Of Under Water System Application in Power Pc Processor Based Using Fpga Ip Cores

R.Aditya, P.Kartik, K.Suhasan, R.V.Ch.Sekhar Rao
[123],IV B.TECH (2011 -2015), [4]Asst.professor
[1234], Dept of Electronics & Communication Engineering, Lendi institute of engg & technology – 535005 AP India (Affiliated to JNTU Kakinada, A.P., India)

Abstract: *Under water systems use processor based rooted systems to provide control and guidance to the under water vehicles. They obtain target and vehicle dynamics data from sensors and gyros, and process this data as per control and guidance algorithms to generate control and guidance parameters to the actuation system. Traditionally x86 families are being used in these systems in amassing to memory, I/Os and other peripherals being on the card. The recent developments in FPGA (Field Programmable Gate Array) technology has made pavement to use superior FPGAs with IP cores to develop under water systems. The modern FPGA devices include 32-bit Power PC processor, memory blocks and programmable area to comprise peripheral blocks. Under water systems developed out of the FPGA cores are definitely have several advantages like, saving the card size (FPGA accommodates several of the components in addition to the processor), flexibility to adopt changes in design (as FPGA can be programmed by the end user), preventing obsolescence of components.*

Building an under water systems based on FPGA IP cores is an innovative and hottest technological demonstration with several advantages to prophesy. The present work describes the dwindling in power consumption and size of the Under Water System.

The present work describes the mellowness of under water system relevance in Power PC Based FPGA using FPGA IP Cores. This work includes understanding the design flow of EDK and learns about various IP Cores Provided by Xilinx EDK 10.1. The underwater system application has been implemented in 'C' language by using Xilinx Device Drivers. A custom logic in VHDL has been developed for truncating extra bits of ADC. In Xilinx ISE10.1 project navigator the developed VHDL Code has been integrated with C. The combined bit file generated has been downloaded into Xilinx Virtex-II Pro FPGA Proto Board.

Keywords: *CPLD, Field Programmable Gate Array, Embedded Development Kit, Xilinx Platform Studio, IP Core, VHDL, Platform FPGA, UART, Embedded Processor, RS232 Cable, MEMS*

I. Introduction

Higher density programmable logic devices, including complex PLD's (CPLD), and FPGAs can be used to integrate hefty amounts of logic in a single IC.

An escalating number of digital designs which use to build in custom silicon have been implemented in Field Programmable Gate Arrays (FPGAs). A typical FPGA is an IC that contains an array of identical logical cells with programmable interconnections. The User can line up the functions realized by each logical cell and the connections between the cells.

Currently in many General and naval applications the Hardware is designed by using Embedded Processors (Such as Power Pc 405) along with FPGA. Most of the processors available in market are designed based on the application requirement. Many of the naval applications are highly constrained in space and power consumption, which forces the designer to make hardware highly compact and stumpy power consuming.

The main design complexity of MDAC card is space and processing capability. Due to these reasons hardware compactness is more grave issue.

Using conformist design approaches it may not be possible to develop such hardware with required specifications. Focal bottlenecks in the design are processor supporting hardware (i.e. pic, timers, wdt, glue logic circuit and other interfaces support hardware) which occupies most of the space in the board. Best approach to solve the above problem is "Hardware Designs with reprogrammable chips (FPGA)". Reprogrammable Hardware is a flexible hardware. Designer can change the hardware configuration of the chips without physically changing the hardware. So FPGA is chosen as a main computational element.

IP Core refers to a hardware module, usually written in VHDL, which can be added to a system through Xilinx Platform Studio. Each hardware core also has an associated "driver", which refers to a library of software functions more specifically to interact with a particular hardware core. Device Driver module to be the only piece of software in the entire system that reads or writes that particular device's control and status registers directly. If the device generates any interrupts, the ISR that responds to them should be an integral part of the device driver.

Xilinx EDK Delivers Processing Peripheral IP Cores at no Additional Cost. The EDK includes over 40 popular cores for licensing at no charge that can be used for a wide range of applications for PowerPC and Micro Blaze processor systems. Using the EDK suite of no-fee IP cores, designers can select and build an unlimited number of configurations to meet their needs and not incur the additional costs associated with traditional IP license agreements. EDK version 10.1 includes the Micro Blaze v7 processor core with new optional memory management unit (MMU), Xilinx Platform Studio (XPS) 10.1 tool suite, software drivers, power pc 405 processor core. XPS10.1 supports Micro Blaze and PowerPC processing design for Virtex-5, Virtex-4, Virtex-II Pro, and Spartan-3 FGPAs. XPS10.1 supports a broad range of computing platforms, including Windows XP, Linux Red Hat Enterprise as well as Solaris [5].

II. Advantages Of Reuse Of Ip Cores

The advantages of reuse of IP Cores are enormous. They include:

1 Offer a modular concept
2 Offer fast development life cycles
3 Excellent cores are available from specialized core developers

A. IP's Used in This Work

1 GPIO IP Core
2 Timer IP Core
3 UART IP Core
4 Delta Sigma ADC IP Core
5 Delta Sigma DAC IP Core
6 INTC IP Core

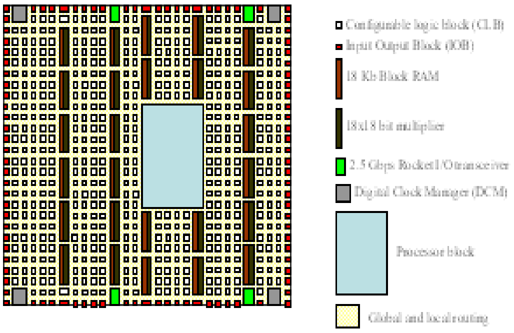
B. Platform FPGA's

A Platform FPGA is a comparatively recent category of chip that combines several different types of programmable components. A platform FPGA has all the components necessary to build a complete embedded system and should require few, if any, additional chips. Platform FPGA's clearly including FPGA fabrics. They may also include CPU's, embedded memory, memory interfaces, high speed serial interfaces and bus interfaces. In this project Xilinx Virtex-II Pro Platform FPGA has been utilizing.

C. Advantages of Platform FPGA's

1 High levels of integration
2 Smaller Physical Size
3 Low Power Consumption
4 Higher Reliability
5 They can more efficiently implement many system level functions

III. Virtex-II Pro Fpga Architecture



A. Uses and Applications of Virtex-II Pro

FPGA Standalone Board

The Board provides easy to use development platform, useful to physically verify

1 DSP and Image Processing Algorithms

2 High Speed Data Transfer
3 Embedded PowerPC based Applications
4 Simple Digital Designs Around Virtex-II Pro FPGA

B. Components Present In Virtex-II Pro Based FPGA Standalone Board

The Board includes the following components
1 Virtex-II Pro FPGA
2 Analogue Interface(14 bit ADC9240,12 bit AD7541)
3 SRAM Interface(4 SRAM's of 1M*16bits)
4 Flash Interface(2 Flash of 512k*16 bits size)
5 DIP Switches(8 No's)
6 LEDS(8 O/P LEDS,1 Done LED)
7 LCD Interface(16 character 2 row LCD)
8 RS232 Serial Interface
9 RS422 Serial Interface
10 JTAG Port
11 Free I/O's(120/98 true bidirectional I/o's)
12 Push Button Switches(4 momentary Contact switches)

C. Reasons for Why Virtex-II Pro FPGA Devices Are Chosen For This Project Implementation

1 It is used for designs that are based on IP cores and customized modules.
2 It incorporates multi-gigabit transceivers and PowerPC CPU cores.
3 These devices are optimized for high-density and high-performance system designs.
4 These devices support very high frequency designs.

D. Established System Design

In traditional system design hardware and software flows are decoupled. De-coupled HW and SW flows are inefficient. Results are not optimal because software and hardware are specified, urbanized and optimized individually.

E. Advantages of Using EDK

1 Integration - Less IC's on your board.
2 Modifiable/ customizable your system in silicon - nothing more, nothing less
3 Performance-Some functions are easier to implement in software. Processor intensive functions can be Off-Loaded to hardware integration.

IV. Under Water System Design Using Xilinx Edk

Xilinx Embedded Development Kit 10.1(EDK 10.1) accelerates System Design. In EDK the hardware and software flows are coupled very accurately. Coupling of custom HW processing system and its associated SW platform ensures HW/SW integration. Through EDK it is possible for Rapid development using common tool chain, libraries and bus structure.

The UWS of an under water vehicle process target parameters and vehicle parameters from various on-board sensors to control and guide the UWV to reach the target in real time.

In order to test the functionality of UWS in laboratory, the required parameters are stimulated through a test system such as OBC.

The total under water system (UWS) is implemented in FPGA Proto Board. The LED's on the UWS are connected to relays.If you turn on all relays simultaneously it will take large amount of power and looks like sluggish.So in this project, the relays have been turn on one by one and provide a delay of 5 seconds between each relay. The delay of 5 seconds is achieved by using XTmrCtr IP.So after 40 seconds all the relays are in ON condition. After turning all relays ON, it is also possible to turn ON/OFF a particular relay by sending commands from OBC to UWS.According to power requirements of sub system and flow we can turn ON or turn OFF that particular relay.

A. Functions Of Under Water System (UWS)

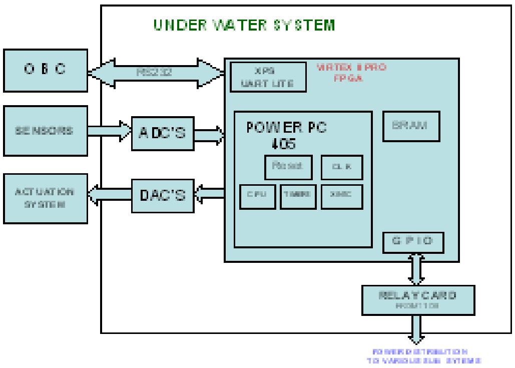
1 UWS basically does the operation of power distribution to various on-board sub systems by using relays.

2 Data Acquisition from different sensors by using ADC's of FPGA Proto Board
3 Controlling of the underwater vehicle by giving commands to actuation system as per commands from OBC.
4 Communication to OBC.

B. Miniaturized Electro Mechanical System

The analog outputs from MEMS are read through ADC's on the FPGA Proto Board and the embedded application on FPGA converts the values to proportionate angles in degrees.
MEMS witness the development of miniaturized systems which can measure physical quantities.
Example: ADXL203.
ADXL203 is a dual axis tilt sensor.

C. UWS Block Diagram



D. Inputs and Outputs of UWS

The analog input parameters for controlling the UWS are given in the form of commands like Pitch, Roll and Yaw along the 3 axis of UWV, which are generated manually using the MEMS which is a two/three axis sensor. By rotating the MEMS (ADXL203) for different orientations i.e. –X, 0, +X and –Y, 0, +Y the value of the analog parameters Pitch and Yaw change where Roll is kept constant. In addition to these parameters, the other analog input parameters to UWS like Pitch Rate, Roll Rate, Yaw Rate, and Depth Deep, Depth Shallow for different values are generated by using Digital to Analog Converter (DAC) of Board.

The analog output commands generated by the UWS are Yaw top, Yaw bottom, Pitch left and Pitch right corresponding to the input analog parameters given. These analog outputs are connected to actuators. These analog voltages are observed using the hyper terminal with the help of UART and serial communication circuit.

At another input side, the discrete inputs related to UWS are generated from the test set up using switches and corresponding discrete outputs are generated from UWS will be observed on the test system using LED's.

Along with Analog Parameters there are input discrete parameters to UWS for controlling the UWV. The discrete output parameters generated by the UWS for the discrete input have to be monitored with the help of Test Set Up. The discrete Input/Output parameters to UWS are listed below

Discrete Input's
1 FF – Firing Free
2 ACC – A Cable Cut
3 WC – Water Contact
4 Cage Ind
5 UnCage Ind
6 Gyro Rdy-1

Discrete Output's
1 PCO – Power Change Over
2 C-ON – Contact On
3 T-OFF1
4 T-OFF2

5 MDRST
6 V-Ok – Voltages Ok
7 Uncage Control
8 Cage Control

The discrete input parameters are generated on the test set up using a DIP switch having 8 parallel lines all with +5v input supply. Depending on the switches ON/OFF position the input supply voltage is connected to the outputs of the switches. The outputs from the switches are given to FPGA. The C code is written in such a way that it reads each switch individually and drives the output pins high which are assigned for the selection. The UWS gives the discrete outputs with respect to the discrete inputs from the test setup.

V. Xilinx Platform Studio (XPS)

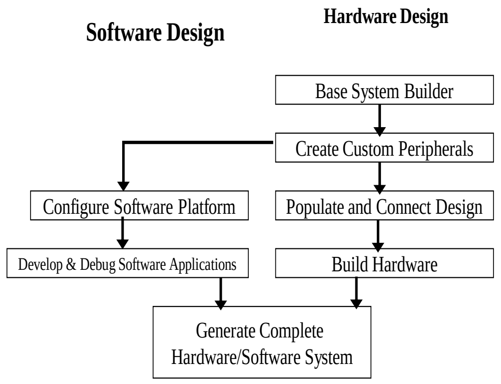
XPS is the design development software provided in Xilinx EDK.
XPS enables you to design a complete embedded processor system for implementation with Xilinx FPGA's.
XPS consists of an interface and all the underlying tools needed to develop the hardware and software components of an embedded processor system.

A. Features of XPS

1 XPS generates a Microprocessor Hardware Specification File
2 It also generates a Microprocessor Software Specification File.
3 Through XPS it is possible to generate and view system block diagram.
4 Through XPS it is also possible to add, edit core parameters and make bus and signal connections.
5 XPS is Intuitive design environment for Xilinx Platform FPGA's.

B. Steps for XPS Embedded design Flow

XPS Embedded Design Flow


Software Design
Hardware Design
Base System Builder
Create Custom Peripherals
Configure Software Platform
Populate and Connect Design
Develop & Debug Software Applications
Build Hardware
Generate Complete Hardware/Software System

C. Hardware Design Flow Steps

1 By using BSB generate the hardware system required by selecting processor, memory, clock frequency, IO interfaces, and peripherals.
2 Change the MHS file and save.
3 Generate the Hardware Net lists.
4 Change the UCF file and enter the pin constraints.
5 Generate the Bit stream.

D. Software Design Flow Steps

1 Software Platform Settings.
2 Generate libraries and BSPs.
3 Write the C code.
4 Build All User Applications.

E. UWS Software Specifications

Initially when power ON, the system would turn on discrete outputs from S.No. 1 to 8.Then the system waits for the commands from OBC (On-Board Computer) on RS232.

Obtaining a command, it turns ON or OFF the corresponding discrete output. It is like that of programming the relays.

The system waits and read discrete inputs no s 1 and 2 till they occur and sends the same to OBC for further action.

For a command from OBC, it will display the status of DIP Switches (Discrete Inputs) on the screen.

For one more command from OBC, it will display the status of all LEDS (Discrete Outputs) on the Screen

For next command from OBC, the FPGA Sub System starts ADC read. After ADC read is completed the analog input voltage value is displayed on the OBC screen with the help of HyperTerminal.

For another command from OBC, the FPGA sub system complete DAC Write and displays the corresponding analog output value on OBC screen with the help of HyperTerminal.

On another command from OBC, it will be observed that LED flasher on the FPGA subsystem, and corresponding statuses of LEDS is shown on the HyperTerminal screen. The delays between each LED are obtained by using Timer Interrupt IP Core (XTmr Car) Component [1].

VI. Project Implementation

As the application requirement is a combination of both Hardware and Software, it is recommended to implement the application on a FPGA proto board.

Board Supplier: Mechatronics Test Equipment Limited
Board Model : MXV2p4/7-RKPC
Revision : 002
Architecture : Virtex2 Pro
FPGA Device : XC2VP7
Package : FF672
Speed Grade : 5
Clock Freq. : 32MHz

A. Specifications of FPGA Proto Board

Detailed study of the virtex II pro FPGA board has been done. As most of the hardware specifications exist on the FPGA board, the existing hardware is made use to develop the application [7].The FPGA will be programmed to have IP cores of UARTs (1 no. s), IP cores of timers (1 no .s), 8 no. s discrete outputs (LEDS), 8 no. s of discrete inputs (Diop Switches).The other hardware ADC, DAC, and LCD are available on the board, external to FPGA. The PowerPC core of the FPGA is chosen for implementing the application. The hardware logic will be implemented in VHDL and the other logic will be written in 'C' language. The entire application will be developed in EDK and tested. When creating the instance of design in FPGA,

1 The software part is loaded in the processor program memory.
2 The Hardware part is at the periphery of the processor.
3 Wrappers are created by the tool and implemented in hardware to connect the user application to the processor.

B. Hardware Specifications

The underwater system named MDAC has the following requirements:
1 Discreet outputs (5v level) ----- 8 no's.
2 Discreet Inputs (5v level) -------8 no's.
3 RS422 communication port------1 no.
4 RS232 communication port------1 no.
5 14 bit ADC.
6 12 bit DAC.
7 16*2 LCD Drivers.

These are the specifications of the MDAC sub system. This sub system communicates with OBC of the Underwater Vehicle on RS232 communication channel.

The FPGA subsystem basically does the operation of power distribution, Data Acquisition from different sensors (Gyros, inclinometers, pressure transducers), controlling of the underwater vehicle by giving commands to the Actuation System as per the commands from OBC.

The discrete inputs related to OBC are generated from test set up using switches, and corresponding discrete outputs are observed on test system using LED's. These are handled by GPIO core. GPIO Core consists of registers and multiplexers for reading and writing the XPS GPIO channel registers.

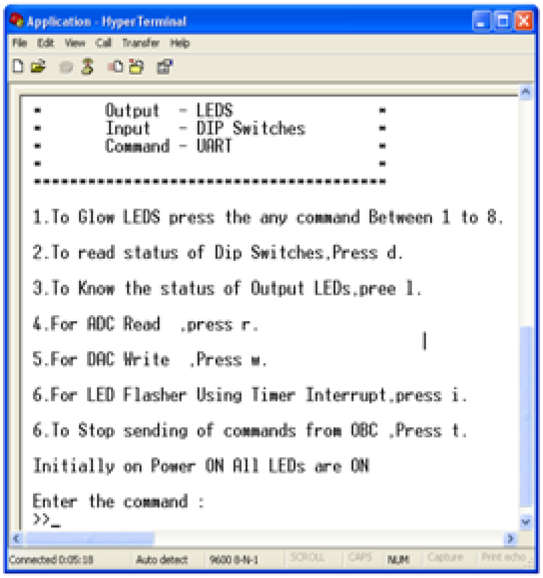
The UartLite controller provides an interface to the serial Port and RS232 cable. Serial communication in this project will involve sending text information to the hyper terminal of the connected pc (OBC) concerning the status of operations on Virtex-II Pro proto Board.

Timer/Counter Peripheral on the PLB, used to generate interrupts at varying intervals.

PC Serial port is used for display of analog outputs from UWS.PC Parallel port is used for JTAG connection and downloading.

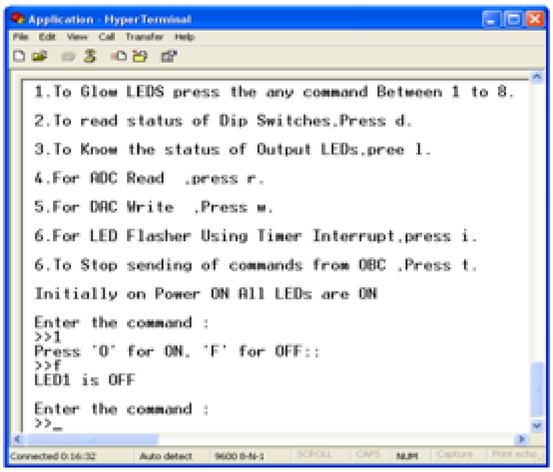
VII. Results

Screenshots



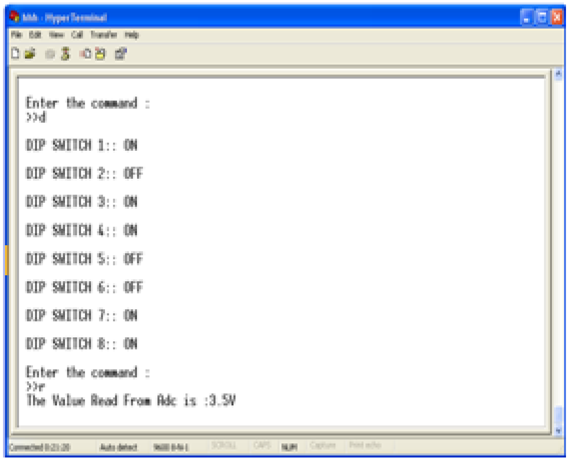
```
*       Output  - LEDS                 *
*       Input   - DIP Switches         *
*       Command - UART                 *
*                                      *
****************************************

1.To Glow LEDS press the any command Between 1 to 8.

2.To read status of Dip Switches,Press d.

3.To Know the status of Output LEDs,pree l.

4.For ADC Read  ,press r.

5.For DAC Write  ,Press w.

6.For LED Flasher Using Timer Interrupt,press i.

6.To Stop sending of commands from OBC ,Press t.

Initially on Power ON All LEDs are ON

Enter the command :
>>_
```

Figure 1: Initial HyperTerminal Output on Power On



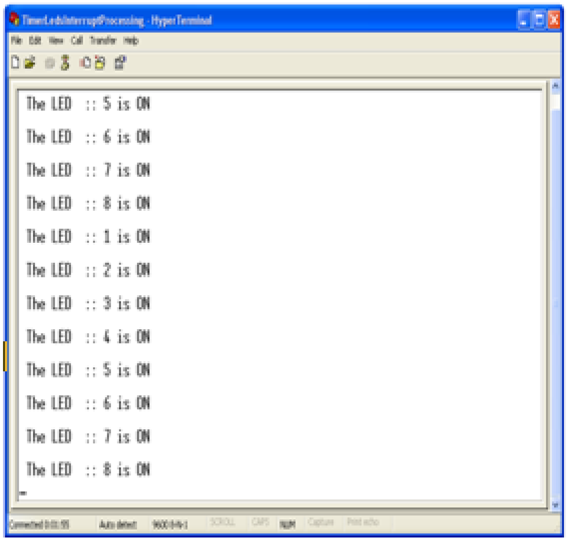
```
1.To Glow LEDS press the any command Between 1 to 8.

2.To read status of Dip Switches,Press d.

3.To Know the status of Output LEDs,pree l.

4.For ADC Read  ,press r.

5.For DAC Write  ,Press w.

6.For LED Flasher Using Timer Interrupt,press i.

6.To Stop sending of commands from OBC ,Press t.

Initially on Power ON All LEDs are ON

Enter the command :
>>1
Press 'O' for ON, 'F' for OFF::
>>f
LED1 is OFF

Enter the command :
>>_
```

Figure 2: Command Screen for Relay ON/OFF



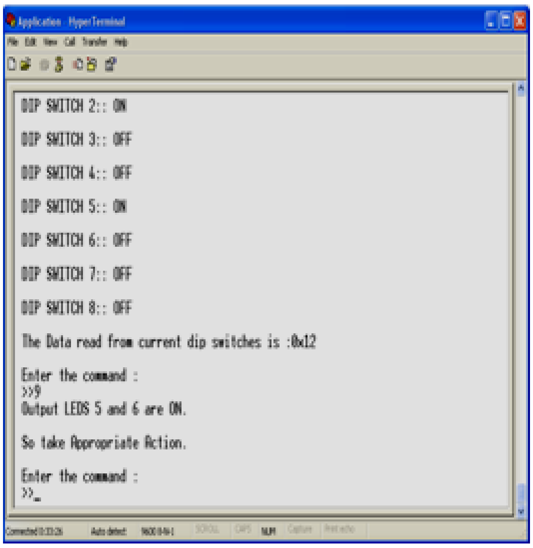
```
Enter the command :
>>d

DIP SWITCH 1:: ON

DIP SWITCH 2:: OFF

DIP SWITCH 3:: ON

DIP SWITCH 4:: ON

DIP SWITCH 5:: OFF

DIP SWITCH 6:: OFF

DIP SWITCH 7:: ON

DIP SWITCH 8:: ON

Enter the command :
>>r
The Value Read From Adc is :3.5V
```

Figure 3: HyperTerminal Output Screen Showing status of Dip Switches and ADC Read Value



```
The LED  :: 5 is ON

The LED  :: 6 is ON

The LED  :: 7 is ON

The LED  :: 8 is ON

The LED  :: 1 is ON

The LED  :: 2 is ON

The LED  :: 3 is ON

The LED  :: 4 is ON

The LED  :: 5 is ON

The LED  :: 6 is ON

The LED  :: 7 is ON

The LED  :: 8 is ON
-
```

Figure 4: Hyper Terminal Output Which Shows LED Flasher Using Timer Interrupt



```
DIP SWITCH 2:: ON

DIP SWITCH 3:: OFF

DIP SWITCH 4:: OFF

DIP SWITCH 5:: ON

DIP SWITCH 6:: OFF

DIP SWITCH 7:: OFF

DIP SWITCH 8:: OFF

The Data read from current dip switches is :0x12

Enter the command :
>>9
Output LEDS 5 and 6 are ON.

So take Appropriate Action.

Enter the command :
>>_
```

Figure 5: HyperTerminal Screen Showing Status of Certain Dip Switches and Indicating Necessary Actions Taken

VIII. Conclusions

From Figure 1, it can be observed that initially when power on all LED's are ON; this was observed on HyperTerminal screen of OBC. Figure 2 specify that how to ON/OFF particular relays or LEDs and the corresponding status is also observed on HyperTerminal. From Figure 3, it can be pragmatic that the status of Dip switches and the value read from ADC. These values will be empirical on OBC HyperTerminal Screen. Figure 4, depict that the status of LED Flasher which glows LEDs at different times and the corresponding led flasher rate was set by timer interrupt. Figure 5, exemplify that the status of certain Dip switches and it also illustrate corresponding actions taken when the particular switches are ON.

The FPGA and MEMS part of UWS abridged the size of UWS by 25%, thus sinking the weight of UAV and in turn reduces the consumption of power required for UAV. This System replaces the earlier bulkier section and save sizeable amount of space. The use of MEMS circuit enables to provide the physical pitch and roll signals to UWS. This can be done by physical movement of circuit board thus simulates the real time signals. There is litheness in changing the types of ADC's, DAC'c and their resolution which was not possible with discrete IC's on the add-on card. Also the cost of this test board is less than that of add-on card.

IX. Future Expansion And Enhancement

In the present work, all the software program is resides on the Block RAM's (BRAM) of FPGA.BRAM's are of limited size. Due to this reason in the future implementation of the project it is required to implement the SRAM interface, a FLASH interface and a SDRAM interface in order to handle large program sizes. It is also required to develop the application based on auxilary crystal oscillator clock source. In the present work, the proto type board was used and the application developed was also of a prototype. This Prototype application was used as reference to develop the bigger under water system applications.

In this work, a basic set of IP Cores such as XGPIO, XTMRCTR, XINTC, XUARTLITE etc are used to develop the application. In future the application might also includes functionality of some advanced IP Cores such as XEMACLITE,USB Controller, Communication Area Network(CAN), Multi port Memory Controller(MPMC) etc and they are used to develop under water system application with more advanced features.

References

[1]. http://www.fpgadeveloper.com/2008/04/timer-with-interrupts.html

[2]. http://forums.xilinx.com/xlnx/board/message?board id=EDK&thread.id=12471

[3]. http://www.xilinx.com/ise/embedded/edk_docs.htm.

[4]. http://www.eda- publishing.org/ewme2008/htmls/Pdfs/P209.pdf.

[5]. http://www.xilinx.com/ipcenter/

[6]. http://www.cse.sc.edu/~gquan/CSCE313/Docs/Lab2. Pdf.

[7]. http://www.mte-india.com/DownloadNew.aspx#.

[8]. http://www.xilinx.com/support/documentation/White_papares/wp213.pdf.