Input/Output Devices in DE-Series Computer

Objective: This lab presents an introduction to I/O devices in the computers designed for DE-Series boards. Each I/O device is driven by a parallel port which is assigned a base address and contains up to four 32-bit registers. Through the lab exercises, you will learn how to access LEDs, pushbuttons, slider switches and seven-segment displays through individual parallel ports.

There are two basic techniques for dealing with I/O devices: program-controlled polling and interrupt-driven approaches. In this lab exercise you will use both approaches in C programs. Your programs will be executed on a Nios II processor in a DE-series Computer.

Intel FPGA Monitor Program is used to download the computer system to a DE-series board. Monitor Program is also used to compile, debug, load and run the C programs.

Resources: In order to carry out this lab, you will require the following resources. More details of the software resources to be used in the lab can be found on Intel FPGA website and the class webpage associated with this lab.

1. A computer running Linux, preferably RHEL 6.8 or RHEL 7.3. Any of the computers provided in the Computer Engineering Labs can be used for carrying out the work associated with this lab.
2. Intel FPGA Development and evaluation board: The process of downloading and debugging a Nios II program requires the presence of an FPGA board to implement the Nios II hardware system. In this lab, DE2-115 Development and Education board has been used to run this lab. The board can be obtained from the Support Engineers. Other DE-series boards (DE1-SoC, DE2, etc.) can also be used.
3. Intel Quartus: The subscription editions of Intel Quartus has been used to run this lab.
4. Intel Nios II Embedded Design Suite (EDS): Nios II EDS is released as part of Quartus software.
5. Intel FPGA Monitor Program: Intel FPGA Monitor Program is released as part of University Program Design Suite (UPDS).

Preparation:

1. Review DE2-115 Computer system
2. Review Intel FPGA Monitor Program
3. Review C language

Procedure:

This section describes the specific steps you are to take in carrying out this lab. Your mark will be based on the contents of your lab report and the evaluations of your in-lab work by the person(s) running the lab session. Be complete in your lab report: include the output produced by the run steps (either by hand-written entries or by printing) and clearly write all required analyses and modifications. Submit your lab report and C programs to the "de2io_lab" hand-in folder through Black Board/PAWS course tools.

Part I: Seven Segment Displays and Slider Switches

Parts I and II of this lab is to investigate the use of seven-segment displays and slider switches in DE2-115 Computer.

Perform the following:

1. Obtain a suitable computer running Linux.

2. Create a new folder, named de2io_part1 to hold your Monitor Program project for this part.
3. Create a file called de2io_part1.c.
4. Write a C program that makes the seven segments on HEX5 display the binary input SW_{7-4} in hexadecimal format. You may want to refer to the examples of C language codes for DE2-115 Computer about displaying numbers on seven-segment displays.
5. Make a new Monitor Program project in the folder where you stored the de2io_part1.c file. Use the DE1-115 for this project, and select Nios II as the target processor architecture.
6. Compile, download, and test your program.
7. Modify the C program to also make HEX2-0 display the binary input SW_{15-8} in decimal format.
8. Compile, download, and test your program.
9. Compress your source code into one zip file named <nsid>_de2io_part1.zip. Hand in the zip file to the "de2io_lab" hand-in folder through Black Board course tools.

Part II Pushbuttons

Part II of this lab is to continue investigating the use of seven-segment displays and pushbuttons in DE2-115 Computer. In your program, **use polled I/O** to read the Data register to see when a button is being pressed. **You should not use the *Interruptmask* or *Edgecapture* registers for this part of the exercise**.

Perform the following:

1. Create a new folder, named de2io_part2 to hold your Monitor Program project for this part. Create a file called de2io_part2.c.
2. Write a C program that that displays a decimal digit on the seven-segment display HEX0. The other seven-segment displays on your DE-series board should be blank.
   - Initially the number displayed on HEX0 should be 0.
   - Pressing KEY_1 to increment the displayed number.
   - Pressing KEY_2 to decrement the displayed number.
   - Pressing KEY_3 should reset the display to 0.
3. Make a new Monitor Program project in the folder where you stored the de2io_part2.c file.
4. Compile, download, and test your program.
5. Compress your source code into one zip file named <nsid>_de2io_part2.zip. Hand in the zip file to the "de2io_lab" hand-in folder through Black Board course tools.

Part III: Pushbuttons with Interrupt

Part III of this lab is to implement similar functions of Part II using the interrupt-driven approach instead.

Perform the following:

1. Create a new folder, named de2io_part3 to hold your Monitor Program project for this part.
2. Make a new Monitor Program project named de2io_part3. At the step of "Specify a program type":
   - Check "Include a sample program with the project"
   - Select Interrupt Example to copy the following files to de2io_part3 project.
     C:\<interrupt example folder>\exception_handler.c
     C:\<interrupt example folder>\interrupt_example.c
     C:\<interrupt example folder>\interval_timer_ISR.c

C:\<interrupt example folder>\address_map_nios2.h
C:\< interrupt example folder>\pushbutton_ISR.c
C:\<interrupt example folder>\globals.h
C:\<interrupt example folder>\nios2_ctrl_reg_macros.h

3. Modify the C program to display a decimal digit on the seven-segment display HEX0. The other seven-segment displays on your DE-series board should be blank.
- Initially the number displayed on HEX0 should be 0.
- HEX0 should increment every 1 second. Rather than using a delay loop, the Interval Timer in the DE2-115 computer must be used to measure an exact of 1 second.
- HEX0 rolls back to 0 after reaches 9.
- Pressing any KEY should trigger an interrupt to reset the display to 0.

4. Compile, download and test your code.

5. Compress your source code into one zip file named <nsid>_de2io_part3.zip. Hand in the zip file to the "de2io_lab" hand-in folder through Black Board course tools.