**LAB1**

This lab was an introduction to using QSYS System Integration tool, Altera Monitor Program, and downloading the program onto the DE1-SOC board. Before starting lab1, our group read through some parts of the DE1-SOC manual to understand some features of the board and how the internal components of the board are connected.

In the first part of the lab, we used the Qsys tool to design the digital system by inter-connecting some Qsys components. By using Qsys tool, we put together a mini-SOC computer by setting up Qsys components.

We opened up the Qsys tool and instantiated the ARM Cortex A9 HPS, JTAG UART, On-Chip Memory and PIO (Parallel I/O) for LED’s, HEX3\_HEX0, HEX5\_HEX4, Slider Switches and Pushbuttons. For this part of the lab, we followed the instructions step-by-step. There were some new terminologies that we didn’t know about. For example, we had to uncheck the “Enable MPU standby and event signals.” We learned that MPU standby and event signals are notification signals to the FPGA that the MPU is standby, and we had to uncheck it in our design.

One of the important steps was to add JTAG communication between the PC and the system. In this way, we were able to see the address map of the system. We added PIO port for driving LEDs, switch inputs, HEX3\_HEX0, HEX5\_HEX4, and push buttons. Then we connected the system to the clocks, reset, and data lines as the system is synchronous.

Then, we assigned addresses to each peripheral in order to be able to access the appropriate peripheral later in our C code. The addresses of the Peripherals are as followings:

LED Base Address: 0xFF200040, Switch Base Address: 0xFF200030, HEX3\_HEX0 Base Address: 0xFF00020, HEX5\_HEX4 Base Address: 0xFF200010, Push Button Base Address: 0xFF200000

After generating HDL, we got some connection errors. We went back and checked all of the connections for the clock, reset and data items and found out that some of the connections were missing. We generated the file again and included the .qip file in the project directory. After following the rest of the instructions, such as instantiating in the Verilog and connecting the pin assignment, we finished designing a basic DE1-SOC computer which was downloadable onto the FPGA.

In the second part of the lab, we were introduced to Altera Program Monitor. We followed all the steps in this part, and we didn’t encounter any difficulties in setting up and adding the led.c code to our program. After finishing part 2, we designed a mini-computer that had the Verilog code programmed and C code in FPGA and HPS of the DE1-SOC board.

In the third part of the lab assignment, we had to implement C code to display two light patterns:

1. HELLO UUORLD, 2)A customized pattern using the switches

Since we only had seven HEXs and there were more letters to display, we had to implement horizontal scroll movement to our code.

We used :

1. The first push button, KEY0, to load the pattern to the display.
2. The second push button, KEY1, as the speed up mode.
3. The third push button, KEY2, as the slow down mode,
4. The fourth push button, to paus the movement.

The major obstacle in this lab was that when we toggled between the push buttons, ex. Pressing the pause button, we had to press the key multiple times in order to activate the mode. In order to fix this problem, we added more polling; however, it created the problem of a single press registering as multiple presses since there was so little time between the checking. In order to overcome this problem, we added in a small delay immediately after the switching logic.

After several, testing and changing the C code, we successfully created the system with the required “four-mode feature.” We downloaded it onto the board and verified its functionality successfully.