**Laboratory 0**

**Hardware Introduction**

**Post-Lab**

**Team Number 5**

**Authors**

**Christopher Wilcox**

**James Nystrom**

**Christopher Wilcox**

**James Nystrom**

**LAB 0 - Hardware Introduction**

**During Lab 0, we acquainted ourselves with laboratory hardware, as well as the software tools Quartus, and Keil Microvision. We learned how to program the FPGA in the board to connect different components together. As a final step, we programmed a seven segment display to display the hexadecimal character for a binary input through the boards dip switches.**

**The lab required us to read the AduC7026 – Cyclone Board Manual as well as to familiarize ourselves with the corresponding circuit schematics. It also required us to use board schematic diagrams to determine characteristics of a device such as if it is active high or active low.**

**Below are responses to the various questions posed during the laboratory. They are enumerated according to Lab #0 as shown in the ECE 315 Lab Manual.**

1. **The LEDs in the LED bar are active low. When a 0 is driven to the LEDS current flows from the 3.3V power rail through the LED.**
2. **The name of the FPGA pin of the leftmost LED is LED\_BAR9. The schematic diagram is attached as Figure 1.**
3. **The pushbuttons on the board are active-low. The voltage at the FPGA-pin when the switch is on(closed) is 0V and when the switch is off(open) is 3.3V.**
4. **The name of the FPGA pin for the Left pushbutton is PB2. The schematic diagram is attached as Figure 2.**
5. **If you were to mistakenly set the FPGA pin connected to the pushbutton as an output, you would see a worse case current flow of 3.3V/ 120Ω = 27.5mA. This occurs when the current flows into the switch. When current flows into the FPGA, and out of the switch you would see a worst case of 3.3V/(10K + 120)Ω = .326mA. The FPGA current rating is 24mA thus the first case where current flows into the switch would be a problem, but the current flowing out of the switch would not be.**
6. **You don't want to configure P0.0 as an output as it controls the boot mode of the board and could cause you not to be able to reset the device. P0.3 controls the JTAG emulator pod, so changing the configuration on this FPGA pin could cause you to not be able to reset the board, thus not allowing you to program it.**
7. **The DIP switches on the board are active-low.**
8. **The corresponding diagram is attached as Figure 3.**
9. **The FPGA pins for the rotary encoder are ENC\_A, ENC\_B, ENC\_SW.**
10. **clockwise counter-clockwise**

**A B A B**

**0 0 0 0**

**1 0 0 1**

**1 1 1 1**

**0 1 1 0**

1. **From output A and B of the encoder you can determine the direction the rotary encoder is being twisted. You would do this by using a lookup table comparing the current A and B output values to that of the previous values. From this you could different things depending on the direction such as a volume control.**
2. **The third encoder output is used for the pushbutton on the rotary encoder. You could use this as a power on/off button.**
3. **The FPGA pin associated with Segment A is pin 98 and has the label SEGA. The FPGA pin associated with the decimal point is pin 96 and has the label SEGP. The leftmost digit enable has the name DIGIT1 and is assigned to pin 85.**
4. **The digit enable and segment lines are active-high.**