**Laboratory 1**

**Multiplexed LED Display**

**Post-Lab**

**Team Number 5**

**Authors**

**Christopher Wilcox**

**James Nystrom**

**Christopher Wilcox**

**James Nystrom**

**LAB 1 – Multiplexed LED Display**

Lab 1 introduced our group to the usage of a hardware timer interrupt, and how to use the interrupt handler to control a multiplexed LED display. Code was produced to configure the Input and Output devices of the board, as well as to operate the Multiplexed LED display through the use of DIP switches. Later in lab we demonstrated the ability to use the DIP switches to control brightness of the multiplexed LED display.

This laboratory was performed using an IRQ interrupt, which we utilized to update the LED display at an 8kHz rate, and give the user the illusion that all 4 displays were simultaneously changed. We operated on the data given by the DIP switches in the main programming loop on the time in between the interrupts. This resulted in a satisfying result.

From this lab, we learned many things, but no more important than lab preparation. By spending time ahead of time, and generating code, we were able to greatly reduce the stress of being under a time restriction. Even with coding ahead of time, we still ran into our share of issues. We found that monitoring registers closely is an important skill, and the mismanagement of registers can cause much time to be spent debugging errors.

**Post-Lab Deliverable 5.A)**

It is requested by the lab manual, that we answer the following question:

Q: What is the measured peak current flowing in an illuminated display segment? How does this compare to what you had calculated in the pre-lab assignment? Can you explain any differences? What is the average current for an illuminated segment? How does this relate to the duty cycle, refresh rate, and number of multiplexed digits?

A: The measured peak currents as found in lab for an LED are:

ON: 10.675 mA OFF: -.260 mA

The measurements taken were with 1 digit illuminated, and a refresh rate of 8kHz for 4 segments. The measurement of 10.675mA for an active segment, was taken using the voltage when only that single segment is on.

In our pre-lab we said that the peak current through an LED would be 27.5 mA. This is very different from our value found in laboratory, as we apparently made in error in our pre-lab. We neglected that the LED has a zero voltage drop. In reconsidering the pre-lab question, this LED has a 1.8V typical to 2.3V max forward biased voltage. These numbers are stated in the datasheet. Assuming a voltage drop of 1.8V across the LED, we find that 1.8V/120ohm = 15mA, higher than 10.675mA. If you had a drop of 2.3V, the max, you would have 2.3V/120ohm = 19.2mA, almost double the lab measured value. In the pre-lab we made assumptions about the FET that could be creating a disparity in experimental and calculated values.

The display switches between 4 different segments, and is interrupted by timer1 and handled by IRQ at a rate of 8kHz. This means that the current is 0 for most of the time, and occasionally at 10.675mA. To be more exacting, the duty cycle is ¼, as only ¼ of the digits are ever on at the same time. Therefore, the LED has an average current of 10.675mA/4 = ~2.67mA . The refresh rate of 8kHz does not particularly affect this average current. The number of digits does affect this however as it relates to the duty cycle.

**We have attached all requested schematics, waveforms, and code. Below you will find a list of contents.**

1. Trace of Oscilloscope digital inputs DIGIT4-DIGIT1, SEG\_G-SEGA, as requested in step 4.
2. Oscilloscope trace as requested in step 5 to facilitate current measurements involving LED segments.
3. Oscilloscope traces of a dimmable multiplexed LED display, captured with inputs 2\_0011 and 2\_1110 as requested in step 6.A
4. Quartus pinout showing FPGA configuration
5. main.s – as used for steps 1-5
6. exceptions.s – as used for steps 1-5
7. IRQ\_Handler.s – as used for steps 1-5
8. arm\_bonus\main.s – as used for step 6
9. arm\_bonus\exceptions.s – as used for step 6
10. arm\_bonus\IRQ\_Handler.s – as used for step 6