# ENSC 254 Lab 4

Summer 2018

Group 81
Steven Luu
Zhe Chen

# Table of Contents

## Objective

The objective of this lab is to become familiar with working on the FPGA board and write code to interact with the LEDs of the board. The first part of the lab was to determine which bit in the 32-bit register corresponds to which LEDs on the board; once this was determined the leftmost LED of the board was used for the rest of the lab. Our next task was to have the LED blink with a 50% duty cycle and a period of 2 seconds. After this was done the rest of the lab was to observe and record the effects of the duty cycles and frequencies had on the LED. Most of the measurements of this lab are determined by using our eyes however it becomes impossible to confirm higher frequencies in later parts. To get an estimate of the frequencies and periods that we are inputting into our code we use the following formulas:

$$Frequency_{Unknown} = \frac{Period_{Known}}{Period_{Unknown}} \cdot Frequency_{Known} \quad (1)$$

$$Period_{Unknown} = \frac{Period_{Known}}{Frequency_{Unknown}} \cdot Frequency_{Known} \quad (2)$$

## Simple Blinking

In this part we programmed the LED light on the board to blink by having the LED turn on for 1 second and turn off for 1 second. This was done by having a register check the current state of the LED and branch to a section of code that turns it on or off. When the LED changes state we have two delay loops where the program will count up from zero to a specific constant to keep the LED in its current state. For our period of 2 seconds with a 50% duty cycle we had the delay loop count from 0 to 12,000,000 for both delay loops. We observed and timed the behavior of the LED several times before concluded on this number.

As noted in the lab we set the sliding switch of the FPGA board to essentially reset the program by having some code check for the state of the switch and then branching to the beginning of the program if it is enabled. This allowed us to better observe and measure the behavior of the LED in all parts of the lab.

## Duty Cycles

For this part of the lab we created signals of different duty cycles and observe the results. First a duty cycle of 80% was implemented with a period of 5 seconds with the LED lit for 4 seconds followed by 1 second of it off. We determined that the delay in both loops is close to the overall period. So we first multiplied the period of the previous section by 5 giving an overall delay value of 120,000,000 and then we multiple this value by the duty cycle percentage to place in the delay loops. After some tweaking we were able to get delay values for an 80% and a similar process was done to get a 5% duty cycle for a period of 10 seconds. The values are shown in table 1 below.

| Duty Cycle (%) | Period (Seconds) | Delay Value | Delay (On) | Delay (Off) |
|---|---|---|---|---|
| 80% | 5 | 122,000,000 | 100,000,000 | 22,000,000 |
| 5% | 10 | 254,000,000 | 14,000,000 | 240,000,000 |

*Table1: Delay Values for Different Duty Cycles*

## Persistence of Vision

For a duty cycle of 50% we modified the frequency of the signal to make the LED appear steady, using formula (1) to calculate the frequency. We used a 50% duty cycle with a period of 0.5 seconds as a known observable base signal. Since the frequency is the inverse of a period we determined that the frequency for our signal is 2 Hz. Using varying frequency values we can calculate the value of the period using formula (2), we started at 60 Hz at it is the common refresh rate of most computer monitors and went down. We observed that the LED started to appear steady at 44 Hz.

We discussed the logic behind the effect of duty cycles on frequencies for a steady LED light and came up with the hypothesis that the frequency should increase when the LED is lit for a shorter amount of time. In other words, the frequency decreases as the duty cycle increases for an LED to appear as a steady light. We tested frequencies for duty cycles between 0 and 100 and the trend is shown in Figure 1 below. As shown in our plot, our hypothesis is confirmed as the frequency decreases as the duty cycle increases in order for the LED to appear steady. The plot appears to be somewhat linear between duty cycle values of 1 and 90, with the frequency required dropping at a 95% duty cycle. This may be due to our eyes not being able to notice that the light is flickering at such a high duty cycle or it may be due to human error as we were observing the LED for an extended period of time. The values of our data points can be found in table 2 in the appendix.

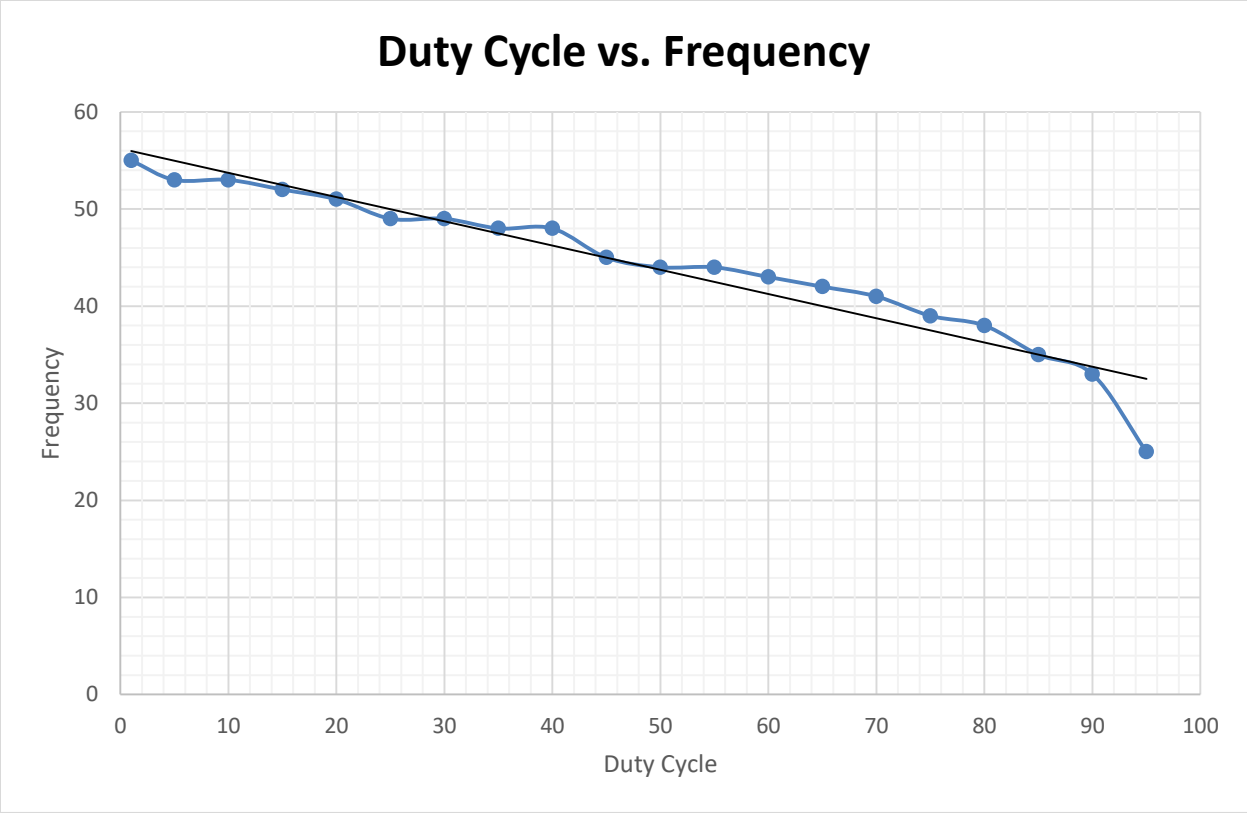


*Figure 1: Duty Cycle vs. Frequency Plot for a Steady LED Light*

## Intensity of Light

As observed in the previous part of the lab, the LED appears dimmer at lower duty cycles compared to higher values. We observed the effects of duty cycles on the brightness of the LED light by selecting a high frequency so the LED will always appear steady and changing the duty cycles. Again we used formula (2) to get a period for a 60 Hz signal and then multiply the value for varying duty cycle percentages. We observed the brightness using duty cycles from 0 to 100 as shown in Figure 2.

From our data we see that the brightness increases as the duty cycle increases, the plot we get from our data shows a linear relationship for duty cycle values between 5 and 90. With our sliding switch enabled the LED will stay lit at a 100% duty cycle and is used to compare the brightness between the 100% and the set value. The LED appears to be completely lit with a brightness value of 1 at a duty cycle of 93% with no observable differences with a 100% duty cycle for both group members. A very small difference is observed once the duty cycle is set to 92% and the brightness starts decreasing. So the LED appears to be fully lit once the duty cycle values are in the range of 93% to 100%. For a brightness half-way between the two extremes, brightness value of 0.5, we found that a duty cycle of 45% yielded this result.

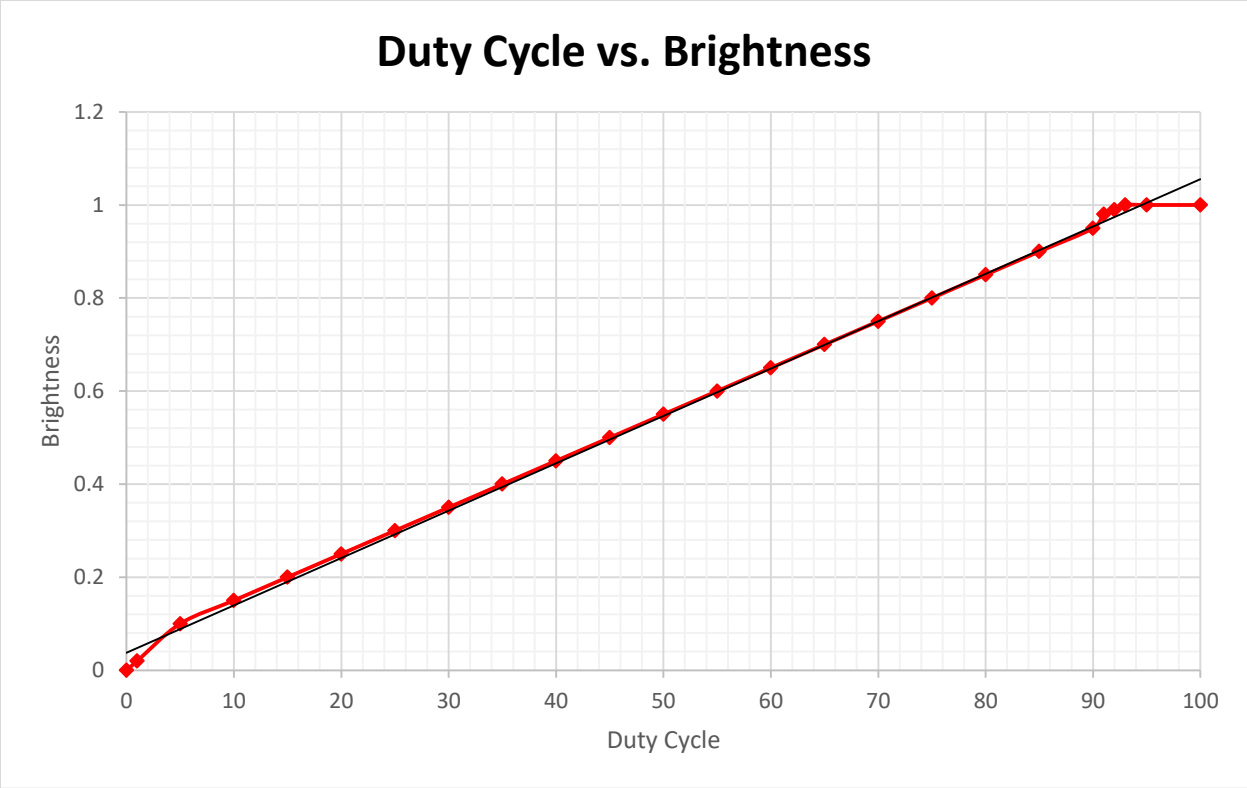


*Figure 2: Duty Cycle vs. Brightness Plot for an LED at 60Hz*

We tested the LED with a duty cycle of 1% and were able to notice that the LED was lit though it was difficult to see. We also tested the LED at the smallest possible value for the delay loop of 1 and the LED appeared as a tiny red dot, however we need to look extremely closely to see this and a person observing the LED from a comfortable distance would not be able to notice.

We found observing the LED brightness to be somewhat difficult was we were not entirely sure how bright the LED is at various duty cycles. We can determine that the LED grew dimmer but we were not sure at what magnitude the LED is dimmer compared the when it is fully lit, which resulted in the range where the LED appeared to be fully lit. As such the values of the brightness may not be precise but the trend remains, and shows that there is a linear relationship between the duty cycle and brightness.

## Conclusion

From this lab we were able to gain experience in working with the FPGA board and learned the effects of duty cycles and frequency on the LED light. We determined the frequency and periods used in our testing with formulas (1) and (2) that we created. Using the values that we calculated we were able to see the effect of duty cycles on frequencies for a steady LED and the effects of duty cycles on the LED brightness. As the duty cycles increases the frequency required for the LED light to appear steady decreases with a somewhat linear relationship. The brightness of the LED increase as the duty cycles increase with a linear relationship for most of the duty cycle range.

# Appendix

## Data Points

| Duty Cycle | Frequency | Delay1 | Delay2 |
|---|---|---|---|
| 95 | 25 | 912000 | 48000 |
| 90 | 33 | 654545 | 72727 |
| 85 | 35 | 582857 | 102857 |
| 80 | 38 | 505263 | 126315 |
| 75 | 39 | 461538 | 153846 |
| 70 | 41 | 409756 | 175609 |
| 65 | 42 | 371428 | 200000 |
| 60 | 43 | 334883 | 223255 |
| 55 | 44 | 300000 | 245454 |
| 50 | 44 | 270000 | 270000 |
| 45 | 45 | 229787 | 280851 |
| 40 | 48 | 200000 | 300000 |
| 35 | 48 | 175000 | 325000 |
| 30 | 49 | 146938 | 342857 |
| 25 | 49 | 122448 | 367346 |
| 20 | 51 | 94117 | 376470 |
| 15 | 52 | 69230 | 392307 |
| 10 | 53 | 45283 | 407547 |
| 5 | 53 | 22641 | 430188 |
| 1 | 55 | 4363 | 432000 |

*Table 2: Duty Cycle vs. Frequency Data Points*

| Duty Cycle | Brightness | Delay1 | Delay2 |
|---|---|---|---|
| 100 | 1 | 400000 | 0 |
| 95 | 1 | 380000 | 20000 |
| 93 | 1 | 372000 | 28000 |
| 92 | 0.99 | 368000 | 32000 |
| 91 | 0.98 | 364000 | 36000 |
| 90 | 0.95 | 360000 | 40000 |
| 85 | 0.9 | 340000 | 60000 |
| 80 | 0.85 | 320000 | 80000 |
| 75 | 0.8 | 300000 | 100000 |
| 70 | 0.75 | 280000 | 120000 |
| 65 | 0.7 | 260000 | 140000 |
| 60 | 0.65 | 240000 | 160000 |
| 55 | 0.6 | 220000 | 180000 |
| 50 | 0.55 | 200000 | 200000 |
| 45 | 0.5 | 180000 | 220000 |
| 40 | 0.45 | 160000 | 240000 |
| 35 | 0.4 | 140000 | 260000 |
| 30 | 0.35 | 120000 | 280000 |
| 25 | 0.3 | 100000 | 300000 |
| 20 | 0.25 | 80000 | 320000 |
| 15 | 0.2 | 60000 | 340000 |
| 10 | 0.15 | 40000 | 360000 |
| 5 | 0.1 | 20000 | 380000 |
| 1 | 0.02 | 4000 | 396000 |
| 0 | 0 | 0 | 400000 |

*Table 3: Duty Cycle vs. Brightness Data Points at 60 Hz*

## Code

```
.global asm_main

asm_main:
  LDR R0, =0x41210000       // LED
  LDR R1, =0x41200000       // Buttons
  LDR R2, =0x41220000       // Switches
  LDR R3, =#128             // 128 Correlates to Leftmost LED
  STR R3, [R0,#0]
  B CheckLED

CheckLED:
  LDR R5, [R2, #0]
  CMP R5, #1                // Button to reset program
  BEQ asm_main
  LDR R4, =#0
  CMP R3, #128              // Check if LED is lit
  BNE TurnON                // Branches to Turn On
  BEQ TurnOFF               // Branches to Turn Off

TurnON:
  LDR R3, =#128             // 1000 0000
  STR R3, [R0, #0]
  B Delay1

TurnOFF:
  LDR R3, =#0               // 0000 0000
  STR R3, [R0, #0]
  B Delay2

Delay1:
  LDR R5, =#300000          // Delay Value for LED ON
  ADD R4, R4, #1
  CMP R4, R5
  BNE Delay1
  BEQ CheckLED

Delay2:
  LDR R5, =#245454          // Delay Value for LED OFF
  CMP R4, R5
  ADD R4, R4, #1
  BNE Delay2
  BEQ CheckLED
```