COE3DQ5 – Lab # 1 Report

Group # 33
Fahad Mahmood – 001414984 – mahmof4@mcmaster.ca
Wei Che Kao – 001328256 – kaow@mcmaster.ca

September 19, 2018

Exercise 1:

To keep Green LED 8 off, we did not assign any variables to it. We assigned AND gate for Green LED 7, OR gate for Green LED 6 when switches were ON. For Green LED 5 we assigned AND gate, for Green LED 4 we assigned OR gates when switches were turned OFF. For Green LED 3, we did some trail and errors for the Logic Gates and finally come up with a solution of ORed switch 11 and 10 and a AND Gate in between with ORed switch 10 and 9 and another AND Gate in between with ORed switch 11 and 9 to meet the requirements (all switches equal to 1). For Green LED 2. Since the requirement is just the opposite for the requirement of Green LED 3, we just flip the sign of the whole expression for Green LED 3. For Green LED 1, we also did trial and error with the Logic Gates and come up with an expression to meet the requirements: (Switch 5 XNOR switch 4) XOR (switch 3). For Green LED 0, the opposite sign of even number will be odd number, so we just flip the entire expression for Green LED 1 to meet the requirement. To keep the two rightmost 7-seven segment displays to turn off when not a single switch is turn on we need to assign a new logic (turn_off) and make all the switches to turn off. Similarly, for the left most 7-segment displays we assigned a logic (turn_on) so that the display doesn't turn on when all the switches are turned on.

Exercise 2:

For this part of our lab, we changed our experiment 4 and experiment 5 code to achieve the task. Firstly, instead of counting till 59 as we did in experiment 4 we changed our code so that we count till 99. One of the requirement of the exercise 2 was that the counter counts up on the power-up so we used an if statement to see the counting up and counting down input and appropriately decided whether to increment or decrement the counter using if and else statement. Secondly, to output the rollover on the left most 7-segment display we created another counter called counter1 which incremented only when counter counted to 99 from 00 or down to 00 from 99. Lastly, the requirements for push-button 3 were; 1) turn off all 7-segment displays 2) reset all BCD counters to 00 3) start counting up only after the release. To program this we used an OR statement to check if button 3 was pressed and released at buffer 0 and 1. When the button was pressed and held, button_3 (a signal) got a value of 1 which we used to turn off 7 segment displays using if statements. We added another if statement in counter to check when button_3 was pressed and held to reset the BCD counters to 00. For counting up we assigned values of 1 and 0 to our counting signals, counter_up and counter_down.