**COE3DQ5 – Lab #1 Report**

**Group number**

**Mengjia Li**

**Nicole Wu**

[**lim14@mcmaster.ca**](mailto:lim14@mcmaster.ca)

**wuz78@mcmaster.ca**

**Sep 10th, 2019**

1. To achieve this effect, a reset select condition is added when assigning value\_7\_segment signals and led\_green signals to output SEVEN\_SEGMENT\_N\_O and LED\_GREEN\_O.

* In *always\_ff* statement (line 137 ~ 168), count\_down signal is decided to be HIGH when push button 2 and LOW when push 1 or resetn. When entering freezing mode and push button 3 is pushed to unfreeze the system, count\_down will be flipped to avoid overflow (00 → count up, 59 → countdown).
* To decide flag signal (when to freeze): an *always\_comb* combinational logic is used to decide if BCD0 and BCD1 is either 5/0 and 9/0, if flag will become HIGH, otherwise it always stays LOW. Count\_down is used to avoid falsely entering freezing mode right after dessert resetn (just have 00 and then freeze situation)
* Two 4 bit counter are used: BCD0 and BCD1. In an *always\_ff* statement, when stop\_count and flag signals are both LOW, the counter mechanism around BCD0 and BCD1 starts working. When count\_down is HIGH, BCD0 will add up to 9. When BCD0 reaches 9, it will be reverted back 0 meanwhile BCD1 counted up by 1. Similarly, when count\_down is LOW, BCD0 <= BCD0 - 1. Once BCD0 is 0 it will back up to 9 and BCD1 is subtracted by 1 at the same time. Otherwise, BCD0 and 1 will always hold their previous value.

1. The encoder reads the value from ports, starting from switch 0 to 15, in order to encode the position of the least significant switch. The hex value is converted to the leftmost 7-segment-display. *pushed\_button* stores the position value of the latest button pushed. It is updated once a valid pushed button is detected. if *pushed\_button* is still the initial value, 7-segment-display is not lighted.
2. For this question a new module called *switch\_to\_led* is created. Within this module a 5 bit switch input and 3 bit output are defined. *led\_out[2]* will be all 4 bits ORed together, *led\_out[1]* is all 4 bits ANDed together while *led\_out[0]* is all 4 bits XORed together to achieve an odd number of switch on assert LED. In top level file *experiment 5*, this module is called 3 times (set0, set1, set2) with different input and output.