COMPENG 3DQ5: Digital Systems Design Take-Home Exercise 1

09/22/2021
Thursday - Group 55

William Siddeley 400245905
Mohamed Al-Asfar 400262489

This take home exercise was completed in two parts. The first part was a modification to the counter in experiment 5 that allows the counter to count upwards and downwards, as well as having the counter be displayed in base 5 rather than base 10. The second part of the exercise was to display the green LEDs on the board according to the logic that was specified in the lab outline.

For the first part of the exercise, we added a new logic variable “up_count” that controls whether or not the counter will count upwards or downwards. If up_count is a 0 then the counter counts up, and if it is a 1 then it counts down. We set up_count to a 0 by default, and if the user presses Push Button 2 on the board, then up_count will change to a 1. Similarly, if the user presses Push Button 1 on the board, then up_count will change to a 0. We also used two variables called “value” and “value_1”. The value variable holds the value to be displayed on the first seven segment display, and value_1 holds the second (most significant) digit. Inside the always_ff block where the values are incremented, we used an if statement to check if the value of up_count is a 0 or 1, so we know whether or not to increment or decrement our two value variables. For changing the seven segment displays to display in base 5, we used if statements to check if our two value variables are greater than 4 or less than 0. If this is true, then the values are reset to a 0 (if counting up) or a 4 (if counting down). This gives us the “roll over” effect that was needed when counting in base 5 (for example, counting up to 44 and rolling over to 00).

For the second part of the exercise, we immediately identified that in order to manipulate the LEDs we would have to implement several different logic operators. So our first course of action was noting down which logic operators we required for each specification. For instance, LED 7 required the use of the XNOR operator since the amount of switches in the high position had to be an even number, which matches the operator’s truth table. In certain instances, more than one logic operator was required due to the complexity of the specification. For example, LED 6 had to be split into two sections, the first requiring the AND operator, the second requiring the NOR operator, and both of these operators were AND’d together. With regards to the final two specifications, we implemented a priority encoder that covered both the high and low cases for each switch. This allowed us to manipulate LEDs 3 down to 0 by ensuring the binary representation of only the least significant switch’s index was displayed in the simulation. Finally, we assigned the logic variables that we used to the LED_GREEN_O variable to display on the board.