CSCI 210
HOMEWORK ASSIGNMENT 04

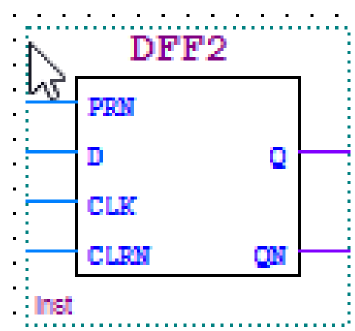
COMBINATION LOCK CIRCUIT/STATE MACHINE

Description

You will design, starting from description, to state–machine, to state table, to Karnaugh maps to equations to implementation in Quartus a combination lock that works as follows. The combination is a three–digit code (“1” “3” “0”) (i.e. a 1 followed by a 3 followed by a 0). To enter the number, you will enter the number on SW[1..0] and then press KEY0 (i.e. the first push–button on the right). Keep in mind that the push button may be active low (i.e. negative logic). Once the correct sequence has been entered, the system will cause HEX0 to display an “F”, otherwise a “0”. HEX5 and HEX4 should display the current state of the machine. HEX3, HEX2, and HEX1 should all be blank (no lights on). Note: the SSD from HW02 has no input corresponding to all segments off.

Use KEY[3] as a reset that will send the flip flops back to state 0 when pressed. Again, make sure that the sense (active high versus low) matches with the reset line on the D flip flops.

For this, you will want to use the “dff2” flip–flop. The PRN line is an active–low preset line (that sets the flip flop to 1), and the CLRN is an active–low reset line (that sets the flip flop to 0). You shouldn’t need the PRN line, but will need the CLRN for resetting (assign to KEY3). The positive–edge triggered CLK line will be hooked to the push–button KEY0.



Use state 0 as the initial state. The state machine should be designed so that if any mistake is made in entering the combination, the system will return to state 0. The reset button can be pressed at any time to “start over” the process of entering the combination.

When the last number is entered correctly, the system should transition to the final state where the output of the system will assert TRUE on a wire that, using a SSD, will cause HEX0 to display an “F”, otherwise, it will display a “0”. This is how we know that the correct combination has been entered.

Part 1: (design circuit)

Draw a state machine and state table which completely describe the behavior of this device. Create the appropriate Karnaugh maps, and derive the correct equations for the outputs. Write these neatly and cleanly for turning in. The state machine, state table, Karnaugh maps, and equations will be due in class on Friday prior to the program submission.

Part 2: (implement in Quartus)

Using Quartus, implement the system using the equations derived in Part 1. Compile and deploy to DE0 board. Verify the design and the implementation actually works by toggling through all possible input combinations.

Submission: (no late work accepted)

Prior to the due date and time posted in Moodle, upload your project into Moodle. This will be a zip file which contains the entire project directory structure. Your submission should be named userid-210-HW04.zip, where userid is your userid. Make sure to check two things after submitting your assignment:

(1) Your file has successfully been uploaded into Moodle.
(2) Download your file from Moodle, unpack the file, load and execute it on your board. If this doesn't work for me, you will loose significant credit on the assignment.