ECE 3544: Digital Design I
Project 3 (Part B): Design and Synthesis of a Synchronous Finite State Machine

Student Name: Colin Grundey

Honor Code Pledge: I have neither given nor received unauthorized assistance on this assignment.

____

Grading: The design project will be graded on a 100 point basis, as shown below:

Manner of Presentation (30 points)

______ Completed cover sheet included with report (5 points)

______ Organization: Clear, concise presentation of content; Use of appropriate, well-organized sections (15 points)

______ Mechanics: Spelling and grammar (10 points)

Technical Merit (70 points)

______ General discussion: *Did you describe the objectives in your own words? Did you discuss your other conclusions and the lessons you learned from the assignment?* (5 points)

______ State machines: *Does your state diagram for the* `buttonpressed` *module correctly represent its behavior? Did you address the questions on structuring state machines in Verilog?* (10 points)

______ Implementation discussion: *Did you address the question of how the* `buttonpressed` *module should be used appropriately? Did you discuss the approach you took to modifying the counter logic?* (10 points)

______ Testing discussion: *What was your approach to formulating your test benches? How did you verify the correctness of the modules you designed?* (5 points)

______ Supporting figures: *Waveforms showing correct operation of the top-level module.* (10 points)

______ Supporting files: *Do the modules pass any tests applied by the grading staff? Modules that do not conform to the requirements of the project specification will receive no credit.* (10 points)

______ Validation of the final design on the DE1-SOC board (20 points)

______ **Project Grade**

ECE 3544: Digital Design I
Project 3B Validation Sheet

GTA Validation Instructions:
Program the FPGA on the DE1-SoC board. When the programming has successfully completed, press and release KEY1 to reset the design. Record the value of the seven-segment displays:

______
Press KEY1

Set SW[6:0] to 1000000. (The switches are 0 when they are down.) Press and release KEY0. Record the values of the seven-segment displays.

SW[6:0] = 1000000 ______
Press KEY0

If the value does not match the last four digits of the student's Student ID Number, stop the validation.

DO NOT RESET THE DESIGN. Set SW[6:4] to 010. Choose a non-zero value for SW[3:0]. Record the value.

______
SW[3:0] value

Press and release KEY0 five times. After each release, record the value of the seven-segment displays.

SW[6:4] = 010					
	1st Press	2nd Press	3rd Press	4th Press	5th Press

Reset the design using KEY1. Set SW[6:4] to 011. Choose a new non-zero value for SW[3:0]. Record the value.

______
SW[3:0] value

Press and release KEY0 five times. After each release, record the value of the seven-segment displays.

SW[6:4] = 011					
	1st Press	2nd Press	3rd Press	4th Press	5th Press

DO NOT RESET THE DESIGN. Set SW[6:4] to 000. Press and release KEY0 five times. After each release, record the value of the seven-segment displays.

SW[6:4] = 000					
	1st Press	2nd Press	3rd Press	4th Press	5th Press

DO NOT RESET THE DESIGN. Set SW[6:4] to 001. Press and release KEY0 five times. After each release, record the value of the seven-segment displays.

SW[6:4] = 001					
	1st Press	2nd Press	3rd Press	4th Press	5th Press

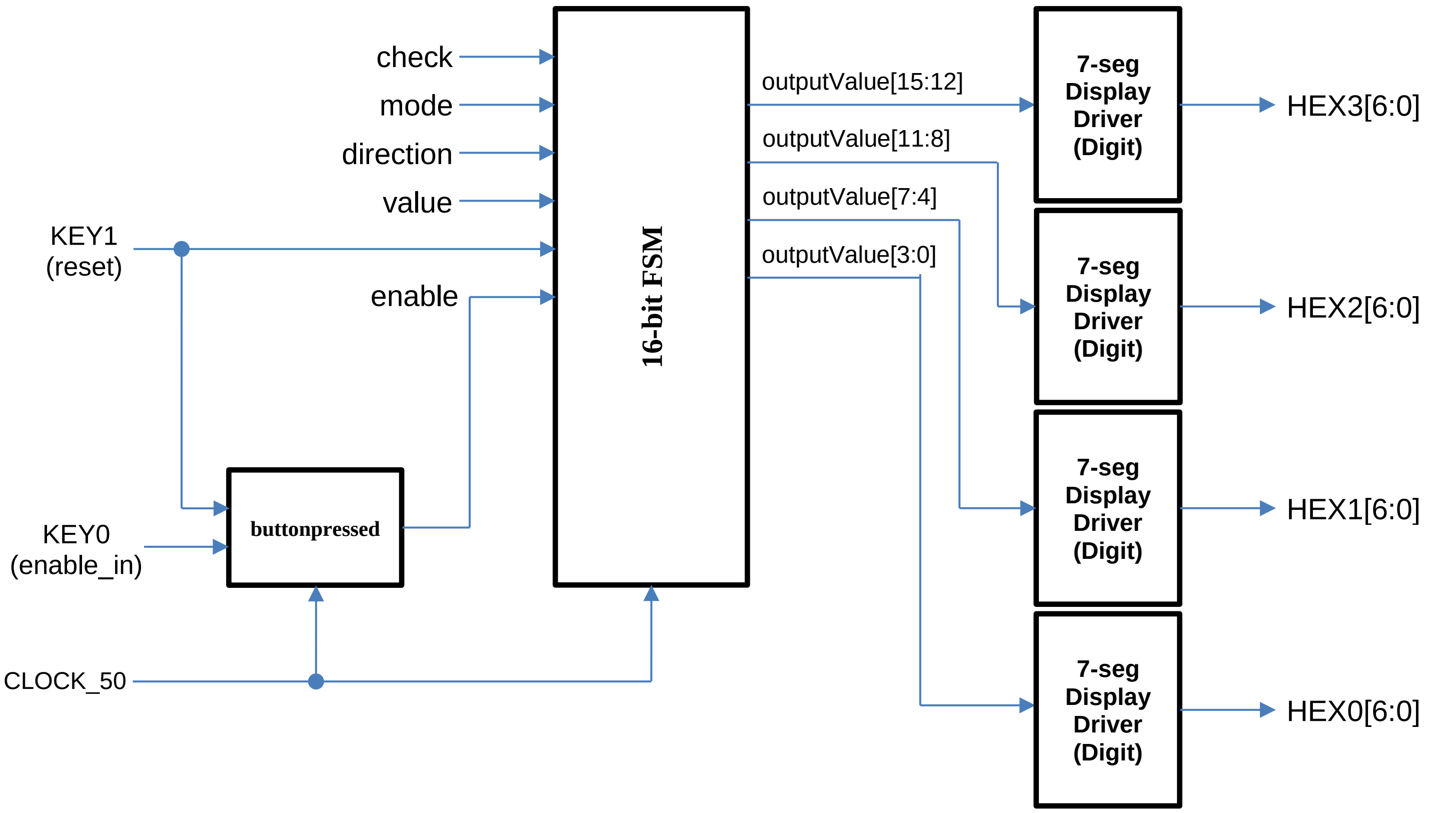
______
GTA Printed Name

______
GTA Signature

______
Date and Time of Validation

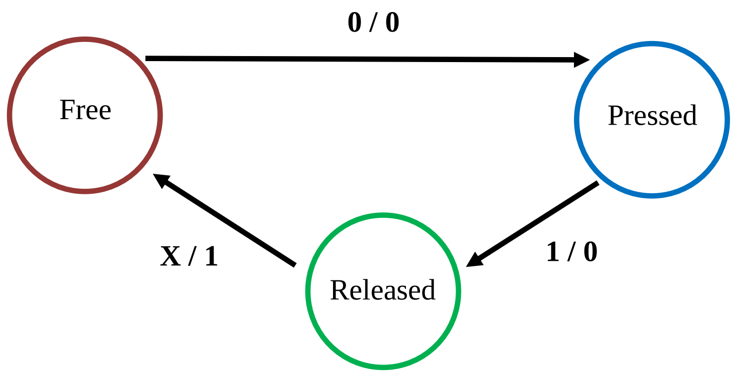
Objective:

Project 3B describes a synchronous finite state machine to be implemented and run on the DE1 board. Inputs for the state machine include the switches and push buttons. The switches represent various control signals and binary value inputs while the push buttons control the enable and reset for the state machine. Having the enable controlled by a button press allows the state to be easily see on the hex displays so the value does not display . This state machine controls whether or not the next output will be a result of a shift register function or a counter function. There is also a mode to display the last four digits of my PID. A system diagram below shows the design fully. The objective is to gain experience in designing, simulating, and testing on the DE1 a synchronous FSM.



Design:

The button-pressed and seven-segment display driver modules already were implemented so the focus of this project was the 16-bit FSM module. This state machine operates on the positive edge of the clock signal or the negative edge of the reset signal. I designed a series of if statements that represent the function described in the spec. The enable is the output of the button-pressed module because it allows the signal to go high when the button is pressed and released. The reset however does not use this module because a reset should occur instantaneously on the negative edge of the signal as the always block defines. A diagram of the button-pressed module is shown below.

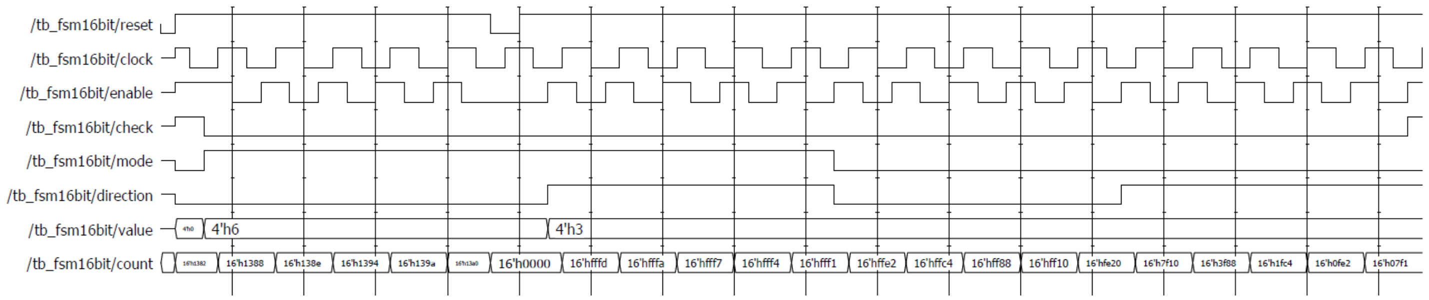


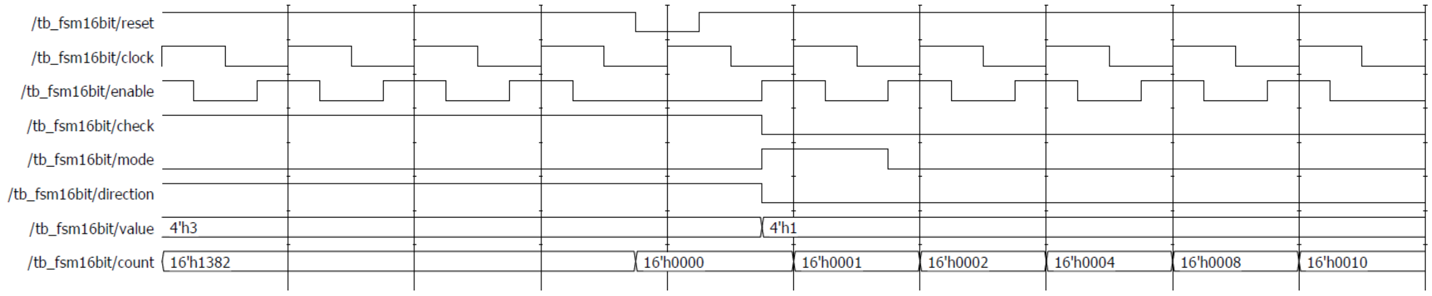
Implementation:

First, I check to see if the reset signal is low, which means the "reset button" has been pressed and that the state should be reset to 0. If that statement is not executed, then my next if-statement in the always block ensures a high enable signal because that is when the functions of the FSM operate. The logic for the counter and shift register modes are implemented using ternary operators for simplicity. As the specification states, the counter either increments or decrements by the input value and the shift register shifts in either direction once.

Testing:

For testing my design, I wrote a test-bench file and simulated it in ModelSim with a generated clock signal for the synchronous elements. I used the validation procedure as a starting point and added my own scenarios to prove proper functionality. After debugging my simulations, programming the board was the next step. I assigned pins on my DE1 and programmed my board to test with switches, push buttons, and hex displays. Everything worked as intended and I ran through the validation procedure to double check and ensure successful output for a variety of inputs. The first waveform below shows proper output of a simulated version of the validation sheet while the second waveform shows my custom tests.





Conclusion:

In conclusion, I have learned about the design process for a synchronous finite state machine. I was also able to practice creating a state machine diagram from a Verilog model. Additionally, I improved my simulation techniques and was able to produce good results when testing my model. Using state machine design principles, ModelSim simulation tools, and the DE1 board allowed me to experience a more full and hands-on process in this project.