ECE 3544: Digital Design I
Project 4 – Design and Synthesis of a Synchronous Finite State Machine

Student Name: Sefunmi Ashiru

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

Sefunmi Ashiru

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

Manner of Presentation (25 points)

_____ / 5 Completed cover sheet included with report

_____ / 10 Organization: Clear, concise presentation of content; Use of appropriate, well-organized sections

_____ / 10 Mechanics: Spelling and grammar

Technical Merit (75 points)

_____ / 5 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?*

_____ / 10 Design discussion: *Did you discuss the approach you took to designing and implementing the modules that make up your system, and how you synthesized the system from its components?*

_____ / 10 System controller state diagram: *Does your state diagram model a system that performs the required tasks? Connect this discussion to your design discussion*.

_____ / 10 System block diagram: *Connect the system block diagram to your design discussion – specifically, to a discussion of how your system employed communicating state machines to implement the tasks required by the specification.*

_____ / 5 Testing discussion: *What was your approach to formulating your test benches? How did you verify the correctness of the modules you designed? What were the results of the test of your counter's accuracy? Did you comment on the significance of these results?*

_____ / 10 Supporting figures: *Waveforms showing correct operation of the top-level module.*

_____ / 25 Validation of the final design on the DE1-SoC board

_____ **Project Grade**

Purpose

In this project, I will be design and implementing a synchronous finite state machine module for a stopwatch. I will implement my top-level module on the Altera DE1-SoC board. This will provide me with practice in assigning pins of your FPGA to the module's input and output ports, and synthesizing modules. Through this project I have practice skills in deriving state machine models from natural language specifications, using state machine models to produce synthesizable Verilog modules, designing and implementing synchronous counters having various control and output functions, designing interacting synchronous finite state machines, and implementing larger-scale synchronous systems via synthesis from smaller finite state machines. This project is by far one of my favourites so far due to its applicability in common devices like digital watches and clock design and control.

Technical Approach

When it came to designing my system i went through multiple interaction. My first step was designing a finite state machine that replicated the behaviour correctly. For this step i utilized the fsm already provided to use with a slight modification to identify when the state the down-count-ready state needs to set a value. This state is based on weather ready state has been pressed once to set the value to zero and then action is pressed again to increment the value from zero. Other than that, the majority of implementing my fsm was a standard practice. Once I have determined the state then out put the related LED outputs as well as the control setting for the clock modules so that I may transfer my states to a given action. Originally I added a separate function using conditional logic but i found it to be much learner to set the controls as outputs of the fsm. The main thing I found difficulty with was handling action that can occur in a constant state in a synchronous manner. I solved this by using procedural assignment to check when certain values changed and those respond appropriately to this change.

Results

Testbench

The waveforms from testbench can be found in Appendices

When designing the test for my top-level module I followed a similar test using timing delays and manually setting the inputs using procedural assignments. To get my tests to function I started by activating the reset state and causing my number stored to be set to zero. From this point, I was able to test all functions and modes but in a non-exhaustive manner. To check my logic, I programmed my board with stable versions of my design.

After testing and debugging I was able to find some issues with my design. The majority were fixable and due to my design overlooking some cases. However, I was not able to get my hex display to run so that is showed the output count in real time. My current design requires me to press pause to see and update in the Hex display values.

Conclusions

In Conclusion, I feel that I have learned a lot about Verilog's design implementation and fsm modelling. I have also gained a lot of understanding of synchronous behaviour. Through this project I have practice skills in deriving state machine models from natural language specifications, using state machine models to produce synthesizable Verilog modules, designing, and implementing synchronous counters having various control and output functions, designing interacting synchronous finite state machines, and implementing larger-scale synchronous systems via synthesis from smaller finite state machines. This project is by far one of my favourites so far due to its applicability in common devices like digital watches and clock design and control. If I were to proceed again, I would attend office hours earlier to solve issues I am unable to do on my own.

Appendices

Appendix 1.1 - Testing hex display

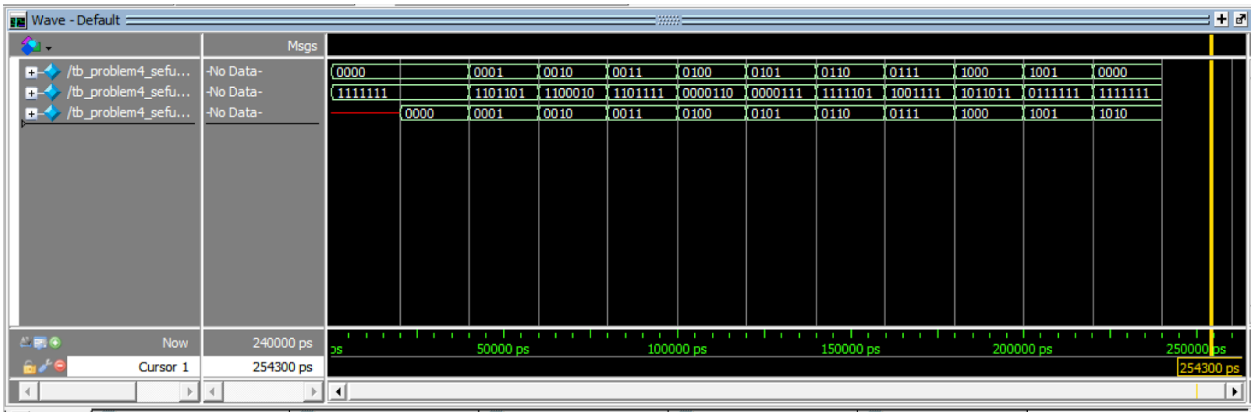


Diagram of full circuit Implementation

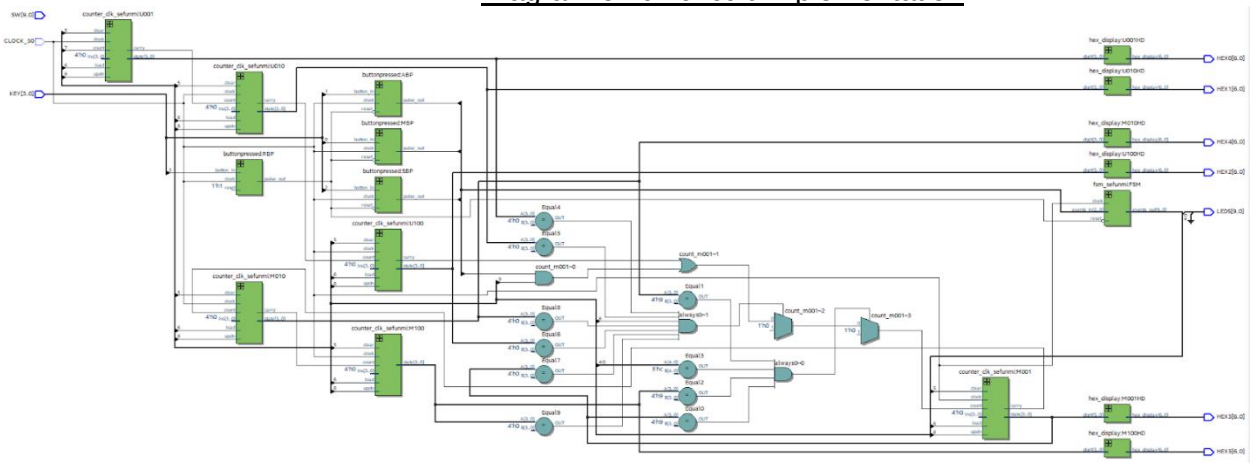


Diagram of full circuit Implementation

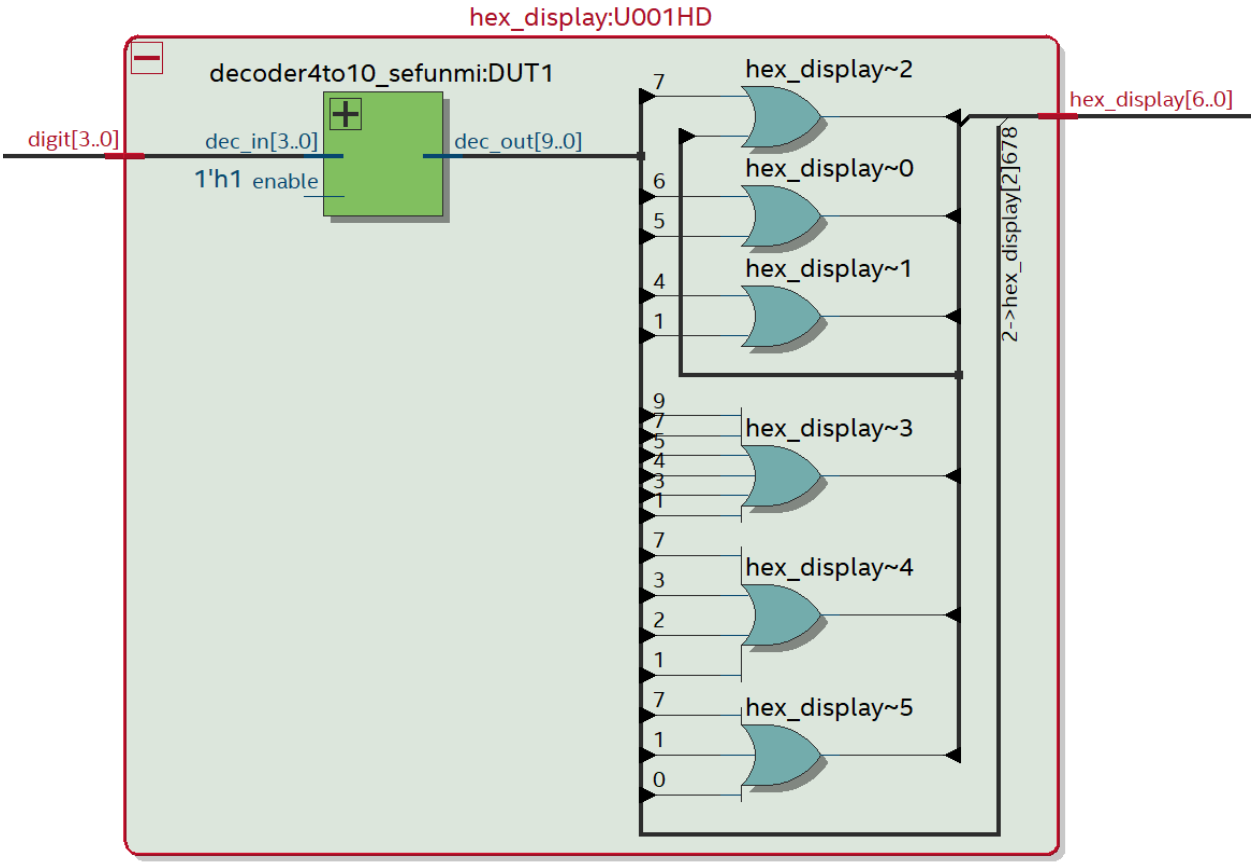


Diagram of Button pressed module

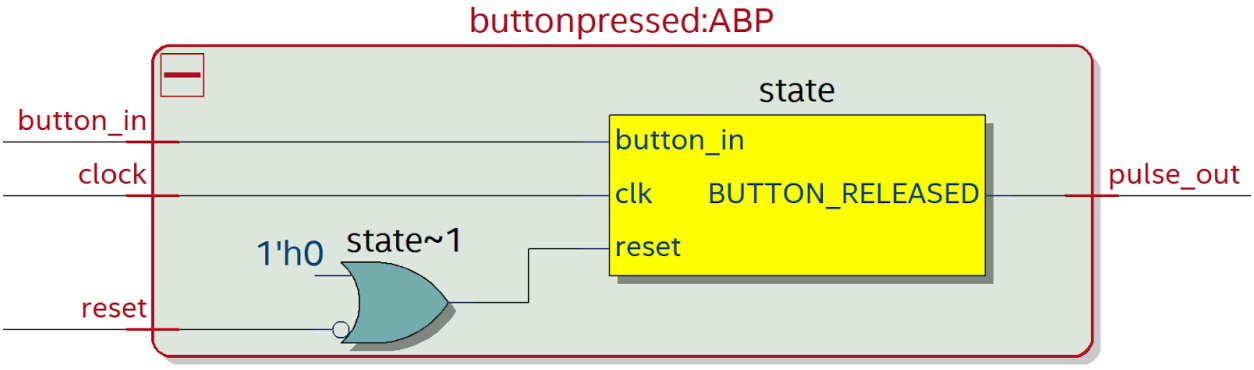


Diagram of stopwatch counter

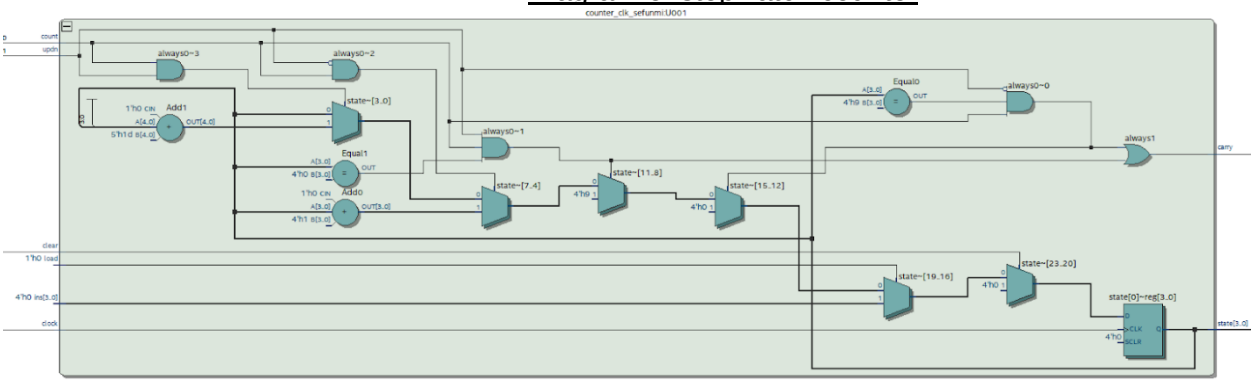
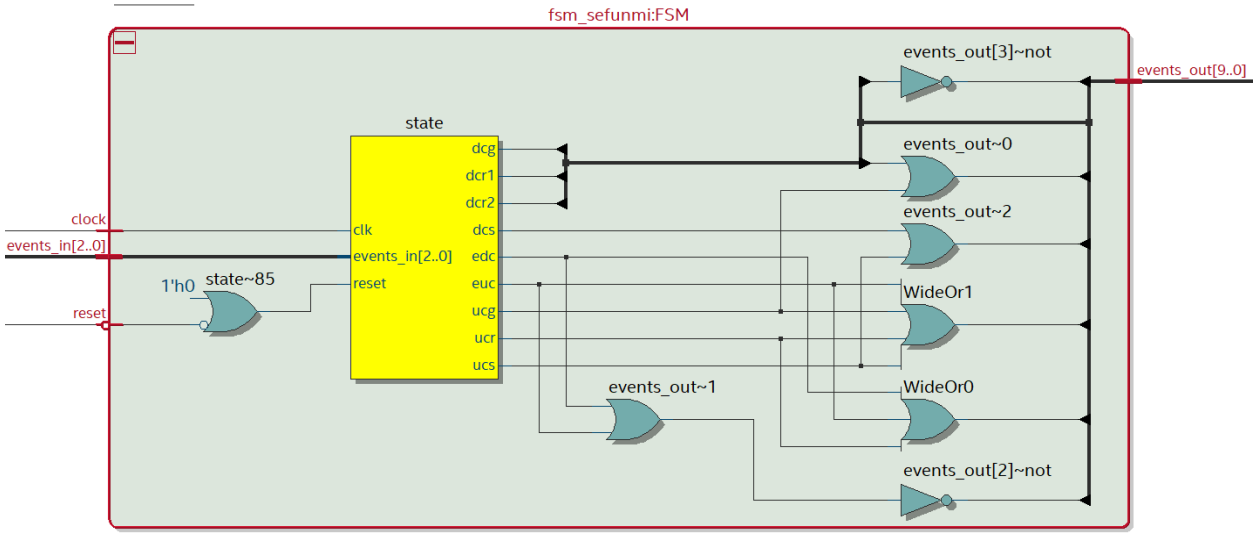


Diagram of finite state machine module



My finite state machine for stopwatch

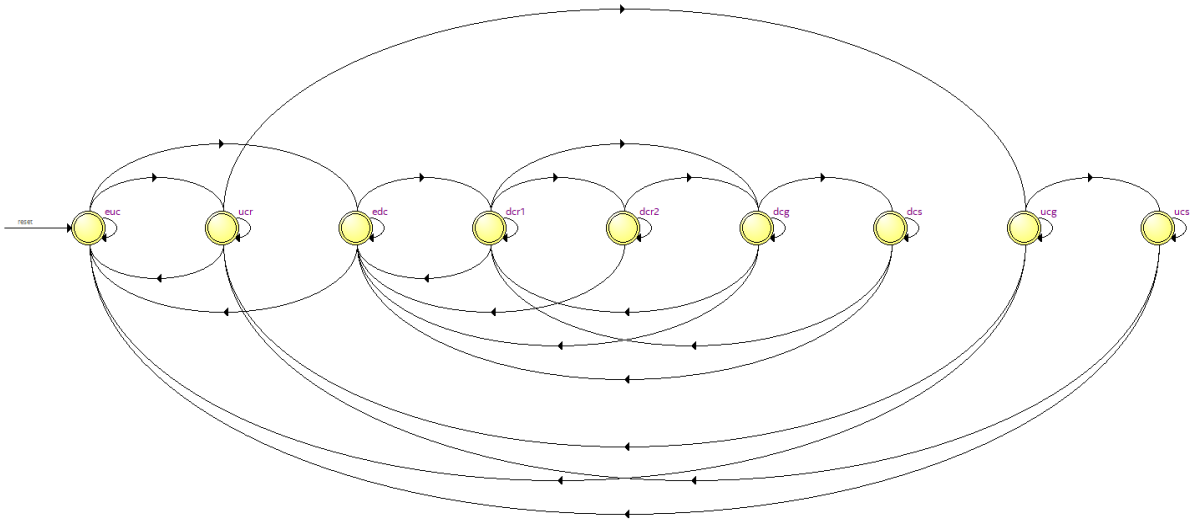


Diagram of Button pressed Finite State Machine