EC605: Computer Engineering Fundamentals - Fall 2024

Lab 3: Finite State Machines

Sequential Logic

Goals

- Introduction to sequential logic in Verilog.
- Introduction to state machine design in Verilog.
- Utilizing the FPGA clock and clock dividers with the Seven-Segment Display.

Overview

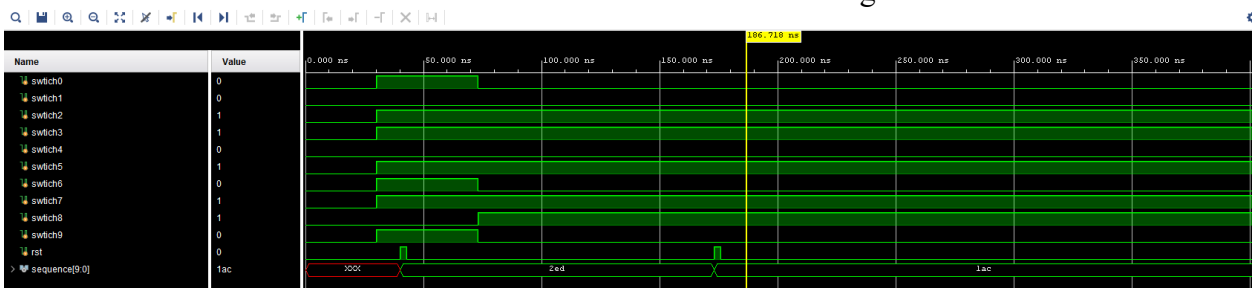
In this lab, the input provided by a set of switches is stored in a serial register. Then, you will design a Moore state machine to implement a counter for a sequence detector with overlap.

Tasks

Task 1: Store Switch Inputs

1. Your input will be a 10-bit binary number, provided via ten switches (each representing one bit). You should store this 10-bit number in a 10-bit serial register. The register will only update its storage after pressing reset button.

2. Implement a testbench and simulate your design. Screenshot the waveform and save it to a PDF for submission. Your waveform should resemble the following:



Task 2: Clock Divider

We would like to utilize the clock provided in the FPGA board to control our circuit. However, the frequency of this clock is too fast when used as the input clock for the seven-segment display. A clock divider takes an input clock of a given frequency and generates an output clock of a different frequency.

1. Implement a clock divider that receives clk and rst as inputs, and outputs a clock with a frequency of 1Hz. The input clock frequency is 100MHz.

2. Implement a testbench and simulate your design. Screenshot the waveform and save it to the PDF for submission.

Task 3: 101 Sequence Detector with Overlap

1. Draw a state diagram for a Moore-type state machine with a minimum number of states according to the following inputs, outputs and specifications:
   - Inputs: clock and reset (button0), ten switches (switch0 to switch9) for sequence input, one switch10 for pause.
   - Outputs: One seven-segment output.
   - The state machine operates as follows:
     - On reset, the state machine moves to state 0, the initial state, with the seven-segment displaying “0”.
     - At any state, if switch10 is switched to low level, the state machine should remain in its current state. When switch10 is switched back to high level, the state machine should go back to work.
     - When the state machine is working, the seven-segment should display the number of “101” dynamically. For example, if the sequence is “1010101010”, when it processed the third digit, the seven-segment should display “1”, then when it processed the fifth digit, the seven-segment should display “2”.
     - Note: You may use several LEDs to indicate the current state, which may help with debugging.
     - Note: The state machine only updates the sequence after pressing the reset button. When it is working, changing the input won’t affect the result.
     - Hint: You may use shift operators to read the sequence from the shift register. Read the sequence from the most significant bit (i.e., switch9).
     - Extra Challenge: Use three seven-segments on the left to display “OFF” when the state machine is paused and to display “ON” when it is working. You may need to implement one more clock divider to display 4 different characters on the 4-digit seven-segments display. See the tutorial at the following link:

https://www.fpga4student.com/2017/09/seven-segment-led-display-controller-basys3-fpga.html

2. In a new file, implement the above state machine.
   Note: Refer to lecture notes for a Verilog state machine implementation example.

3. Implement a testbench and simulate your design. Screenshot the waveform and save it to the PDF for submission.

4. Implement a Top_module.v to program the FPGA board. Instantiate the three modules you have already designed, connecting the clock divider, convertor, and sequence detector state machine. The top module should contain the following inputs and outputs which will be connected to the FPGA board:
   - Inputs:
     - clk – 100 MHz input clock from board
     - rst – push button which resets state machine elements and updates binary sequence
     - switches – ten switches which set the binary sequence
     - button – turn on/off the state machine
   - Outputs:
     - seven-segment – to display the number of "101"
     - Optional: three additional seven-segment displays on the left showing the state machine on/off state

   - Note: the clock input to the state machine is the 1Hz output of the clock divider.
   - Remember: the state machine receives a 10-bit serial register and one switch as input and generates 1 (or 4) seven-segment outputs and several LEDs outputs.

Deliverables

1. Submit your Verilog code and testbench for each task on Blackboard:

   - **Convertor:** code and testbench .v files
   - **Clock Divider:** code and testbench .v files
   - **State Machine:** code and testbench .v files
   - **Top Module:** code .v file

2. Submit a PDF including a waveform and a description of the tests done for each task, as well as the state machine drawing, on Blackboard:

   - **Convertor:** waveform and description

- **Clock Divider:** waveform and description
- **State Machine:** waveform and description, drawing of state diagram
- **Top Module:** waveform and description

3. Sign-up to demo your design on the FPGA board to a TA. Come prepared to answer questions on your design.