

Fall 2018 / EC311

Lab 5

Recommended Due Date: November 14, 2018

Actual Due Date: November 30, 2018.

1. Goals

- Implement a Finite State Machine in Verilog
- Design simple sequential circuits
- Implement your design on an FPGA

2. Overview

In Lab 5, you will build on Lab 4 to create a counter controlled by an finite state machine. For this lab you must implement a 16-bit counter that counts up or down. The counter value must be displayed on the four 7-segment displays as in Lab 4. The counter should count at 1Hz. The direction of the counting should be toggled by a push button. When your design is reset or powers on for the first time, it must display four zeros and start counting up. The counter should wrap around when it reaches 0x0000 or 0xFFFF. At the press of a button (at any time), the counter must be set to the value input on the 8 switches. The 8-bit value from the switches must be zero extended to 16-bits. The direction of counting (up or down) should not change when a new user selected value is set.

3. Lab 5 I/O Requirements

The top level module must have clock, reset, switch, direction_toggle_button and set_value_button inputs. The top level module must output a 4-bit digit_select signal to select a digit to light up (only 1 can be lit at a time) and the 7-bit encoded digit display value. The direction_toggle_button, set_value_button and reset inputs must be mapped to push buttons. The switch input must be 8-bits and mapped to the Nexys-3 switches. The counter value must be displayed in hex on the 4 digit 7-segment display.

4. Components

Your design should have 7 modules:

1. seven_segment_decoder - Same binary to 7-segment decoder as in Lab 4.
2. display_control - Same as in Lab 4.
3. counter16 - A counter module to count up or down. This module will be similar to the counter in Lab 4 but it will have extra features. The counter module must allow a user to set an arbitrary value to the counter. This will take two extra inputs: 1) the arbitrary value and 2) a set_value input to indicate the arbitrary value input is valid and should be latched. You will also need an extra input to control the direction of the counting.
4. clock_divider - The same 1Hz and 1kHz clock dividers used in Lab 4.

5. control_FSM - A Finite State Machine (either Mealy or Moore, the choice is up to you) to implement the appropriate control signals for the counter module. The finite state machine should take two buttons (direction_toggle_button and set_value_button) as inputs. The outputs should be the count_direction and set_value_counter. The finite state machine design and state encoding is up to you. Your report must include a state diagram of the finite state machine.
6. debouncer - Same as in Lab 4.
7. lab5 - A top level module to connect all of the sub-modules.

5. Deliverables

You need to show your design hierarchy (under Design → Implementation window), your Verilog code, and FPGA implementation (on the FPGA board) to the TA. The TAs do not need to check off your test benches before you move on to other modules or the FPGA implementation.

Your design must abide by the following requirements:

1. Use multi-bit vector/bus/array (just different ways of naming) instead of 1-bit signals for all multi-bit signals.
2. Follow the input/output specifications.
3. Create test benches for all modules. Modules re-used from Lab 4 can re-use test benches from Lab 4 (you may have to make minor changes). The test benches should be thorough but do not need to provide complete coverage. Complete test coverage in sequential designs can be difficult to achieve.
4. Be hierarchical and use a separate file for each module. Test each module and demonstrate that each works as a standalone unit. Generally, it is good engineering practice to test submodules before going to the next level. You must turn in a test bench for each module.

You must turn in a lab report describing what you did in this lab, how you did it and what happened when you did it. The lab report should be broken up into these sections: Objective, Methodology, Observation (can include screenshots timing diagrams, schematics, etc.) and Conclusion. The report must include a state diagram for your finite state machine. Your report should contain enough detail such that any other engineer can replicate your experiments after reading your lab report. Please include all the required details in your lab report. The lab report should be 2-3 pages long with figures and diagrams. Your report must be written in paragraph form with complete sentences.

You must submit a PDF to TurnItIn. The PDF must contain your lab report and all of your Verilog code (including test benches).

6. Due date

Your lab report and lab must be checked off by a TA and submitted to TurnItIn by November 30.