

Fall 2018 / EC311

Lab 3

Recommended Due Date: October 31, 2018

Actual Due Date: November 30, 2018.

1. Goals

- Introduction to sequential Verilog.
- Design of simple sequential circuits.
- Implementation of your designs on an FPGA

2. Overview

In Lab 3, you will design a simple 8-bit counter and implement it on the FPGA. This lab will introduce you to sequential logic and its Verilog implementation. Sequential logic, unlike combinational logic, is dependent on an external trigger. A periodic external trigger, called a clock, allows all parts of the system to be synchronized. In this lab, you will implement an 8-bit counter using behavioral Verilog, using push-buttons on the Nexys-3 FPGA board. Pushing the button may last several clock cycles, thus the counter may count several times for a single push. To prevent this, you will also write Verilog code to debounce the push-buttons of the FPGA board. The debouncer should output 1 for a single clock cycle once no matter how long the button lasts.

2.1. Counter I/O

The top level module must have a clock, reset and increment inputs. The top level module should output the 8-bit counter value. The increment and reset inputs must be mapped to push buttons. The output should be displayed on the board LEDs named LD7 through LD0.

2.2. Components

Your design should have 3 modules:

1. debouncer - A module to clean noisy signals from the push button inputs. Pressing a push button may not cleanly transition from 0 to 1 or 1 to 0. The switch may mechanically bounce between the two states. A debounce circuit will detect transitions and prevent rapid switching between 1 and 0.

As a very basic debouncer, we can implement a counter that increments every clock cycle after the push-button was pressed. Only once the counter has reached its maximum value, we determine that the push-button has changed state. At this point we can also reset the counter. This technique utilizes the fact that our global clock is significantly faster than the time it takes a person to press the push-button repetitively. Your debouncer module should have global clock and reset inputs, as well as a push-button input. Its output should be the clean push-button signal. A sample algorithm for implementing the debouncer is shown in Figure 1 below. You should test this module in a test bench.

2. counter8 - An 8-bit counter with clock, reset and increment inputs. The output of the counter8 module should be the value of the counter. At the positive edge of the clock, if the increment signal is high then the counter is incremented. At the positive edge of reset, the counter should be reset to zero. You should test this module in a test bench.

3. lab3 - The Top Level module where you would instantiate the debouncer and counter8 modules. Connect the modules as seen in Figure 2. You must demonstrate the working lab3 module on the FPGA to the TA. Refer to Lab B and the schematic capture tutorial for directions on implementing the design on the FPGA. The pin name for the FPGA clock input is designated as v10.

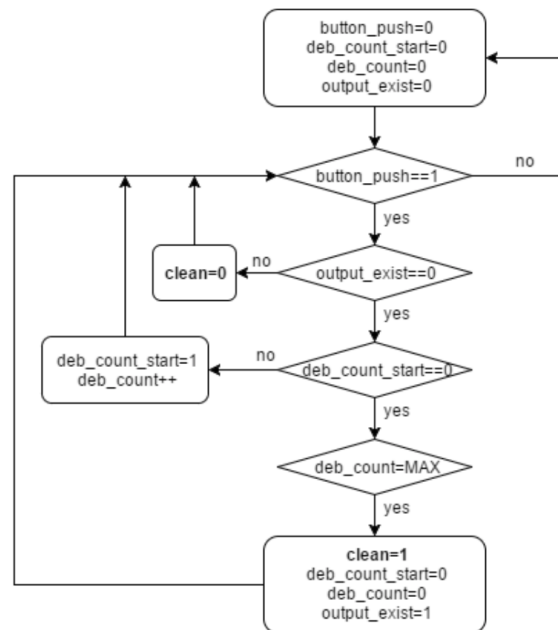


Figure 1: Sample debounce algorithm.

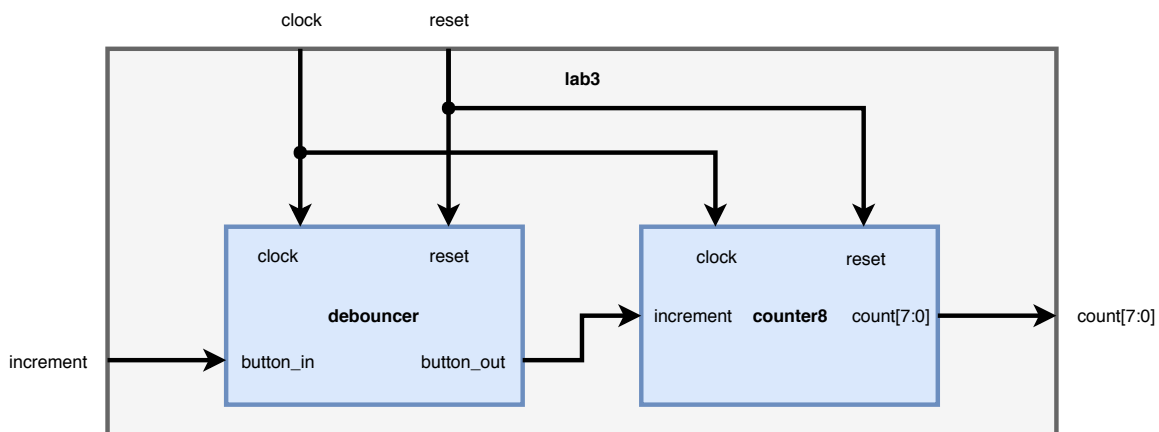


Figure 2: Lab 3 top level module.

3. 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.

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 the debouncer and counter8 modules. 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 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. 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.

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

4. Due date

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