EEC 180B DIGITAL SYSTEMS II Spring Quarter 2018

University of California, Davis

Department of Electrical and Computer Engineering

Lab 5: Counter Design

This lab involves the design and implementation of counting circuits that display values on the six 7-segment displays on the DE10-Lite FPGA board. All circuits are synchronously clocked by only the 50 MHz standard clock. Details not specified in this lab should be chosen by you and stated in your lab report.

I. Prelab

Complete the following and submit your work at the beginning of your lab session.

1. [8 pts] Perform and document a preliminary design including a block diagram and preliminary verilog including items such as wire and reg declarations and basic code blocks.

2. [2 pts] Calculate how many seconds are required for counter2 to count:
   1) one least-significant digit increment, and 2) through all possible values when:
   a) *divideby*=000001, and b) *divideby*=110010

Hint: for the fewest possible problems, write your combinational logic as a function of register (FF) outputs.

II. Counter 1: Pulse Generator

The first counter contains 6 bits and counts in a circular sequence from all zeros to *divideby* minus one. It has the following I/O ports:

reset	KEY0	input, sets count to 000000 and halts
enable	SW0	input, disables operation when low
divideby	SW9–SW4	6-bit input, specifies the length of the circular sequence 000000: halt counter with value 000000 light all ten LEDR9–LEDR0 LEDs force output *go* to low value 000001: count 0 → 0 → 0 → 0 ... 000010: count 0 → 1 → 0 → 1 ... 000011: count 0 → 1 → 2 → 0 ... ... 111111: count 0 → 1 → ... → 61 → 62 → 0 → 1 ...
go	-	output, is high when the state of the counter is 000000 and the input *divideby* is not 000000

III. Counter 2: Main Counter

The second counter contains 24 bits and has the following I/O ports:

reset	KEY0	input, sets count to zero and halts
enable1	SW0	input, disables operation when low
enable2	-	input, disables operation when low connects to the *go* output of Counter 1
up-down	SW1	input, counter increments count when high, decrements when low
free-run	SW2	input, counter runs forever when high, halts at half of maximum count when low
display	HEX5–HEX0	24-bit output, displays the counter state

IV. Top Design

Create a top level module with an instance of counter 1 and counter 2 that are connected to each other, switches, keys, LEDs, and displays as described.

Connect the 24-bit counter2 *display* output to the board's 7-segment displays using your seg7.v module from Lab 3 so that *display* is shown as a 6-hexidecimal-digit value.

Register (with FFs) all inputs as soon as they enter and immediately before they leave the top level module.

V. Testing in Simulation

Design and write a testbench for your counter 1 module that exercises all functions.

Design and write a testbench for your top-level module that exercises all functions.

Once your test benches are working correctly, demonstrate them to your TA and have them checked off.

Suggestion: consider implementing multiple independent tests that are shorter in length and focus on specific features.

VI. Implementation and Verification on the DE10-Lite

Download your design onto the DE10-Lite board and verify it works correctly.

Demonstrate it compiling, downloading, and operating to your TA and have it checked off. Your TA will record a `certutil` hash of your top-level files and this must match the hash of the files you upload to canvas so no file modifications are possible after your demo.

Submitted Work [100 pts total]

With the exception of instructor-provided code, all work must be yours alone.

[10 pts] **Prelab**

[80 pts] **Lab Checkoffs**

1) [10+20 pts] Demonstrate correct operation in simulation using your testbenches
2) [50 pts] Demonstrate correct compiling, downloading, and operation on the FPGA board

[10 pts] **Lab Report**

Submit all Verilog hardware and testbench code that you wrote. Do not include any code that you did not write such as files generated by Quartus or IP components.

a) [10 points] Print and submit a paper copy during your lab session.

b) Upload a copy to Canvas by performing the following steps by the end of your lab session—this is essential to receive credit for the entire lab.

   1. Make a folder on your computer
   2. Copy all verilog files you wrote into the folder—only the ones you wrote
   3. "zip" the folder into a single .zip file
   4. Log onto Canvas, click Assignments, find the correct lab number
   5. Upload the .zip file

2018/05/06	Added point breakdown for simulation checkoffs
2018/05/04	Posted