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 DE10-Lite's 7-segment displays. 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. [3 pts] Perform and document a preliminary design including:
   - a block diagram,
   - a timing diagram for Counter1,
   - a timing diagram for Counter2, and
   - very 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 these four cases:
   1) one least-significant digit increment, and
   2) through all of its possible values when:
   a) *divideby* = 000001, and
   b) *divideby* = 110010

II. Counter1: 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. Output *go* is always low.
  - `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. Counter2: 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 Counter1
• *up-down*	SW1	input, counter increments count when high, decrements when low
• *free-run*	SW2	input, counter runs forever looping from its maximum value back to zero when this input is high; counter halts at half of maximum count when this input is low
• *display*	HEX5–HEX0	24-bit output of the counter's state

IV. Top Design

Create a top level module with an instance of Counter1 and Counter2 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 the earlier lab, so that *display* is shown as 6 hexadecimal digits.

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

V. Testing in Simulation

Design and write a testbench for your Counter1 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: implement multiple independent tests that are shorter in length and focus on specific features.

VI. Demonstration on the DE10-Lite board

After your design is working correctly in simulation, download it onto the DE10-Lite board and verify it works there.

Submitted Work [100 pts total]

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

[5 pts] **Prelab**

[85 pts] **Lab Checkoffs**

- [10+25 pts] Demonstrate your ModelSim simulation to your TA
- [50 pts] Demonstrate your design compiling and operating on the DE10-Lite board, to your TA
- Your TA will ask you to generate and will record a `certutil` hash of your top-level files. This hash must match the hash of the files you upload to canvas so no file modifications are possible after your demo. The hash is calculated using the following commands:
  1. in Quartus, double click a few key modules determined by your TA, into the Project Navigator Hierarchy pane to open its Verilog
  2. right click the file's tab and select Copy Full Path
  3. open a Command Prompt window (by typing "cmd" into the Windows search bar)
  4. type "`certutil -hashfile` *`filepath`*" where *`filepath`* is the file path that was copied earlier

[10 pts] **Lab Report**

Upload the following two files to Canvas by the end of your lab session—this is essential to receive any credit for the entire lab.

I. A single Zip file. Submit all Verilog hardware and testbench code that you wrote in a single zip file. Do not include any code that you did not write such as files generated by Quartus or IP components.

- Make a folder on your computer
- Copy all files to be submitted into the folder
- "zip" the folder into a single .zip file
- Log onto Canvas, click Assignments, find the correct lab number
- Upload the .zip file

II. A single PDF or Word file.

1. Any required diagrams
2. Required hardware verilog
3. Required testbench verilog
4. Text and/or waveform printouts