# CDA 4203L Spring 2022

**Computer System Design Lab**

**Lab 3 – Programmable Counter, BCD Converter and FPGA Synthesis**

***Assigned on Wednesday, 2nd Feb***

***Due Date: 11:59 PM, Sunday, 13th Feb***

## Teaming allowed. No more than three (3) members per group.

**Objective:** To learn and practice advanced sequential hardware design in Verilog and synthesizable Verilog based circuit implementation on FPGA board.

**Note:** Before you read this, you need to perform the Tutorial for Lab 3 (one PDF description which works on one included Verilog and one user constraints file (UCF)).

**Table 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Port | Width | I/O | Function |
| max\_count | 7 | I | Set the maximum  counter value |
| Run | 1 | I | Set run mode;  0 = set, 1=run |
| digit\_2 | 4 | O | Tens digit BCD  output |
| digit\_1 | 4 | O | Ones Digit BCD  output |

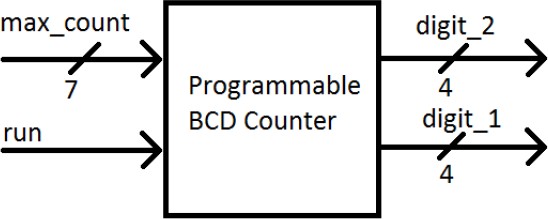


Figure 1: Counter Top-Level Block Diagram

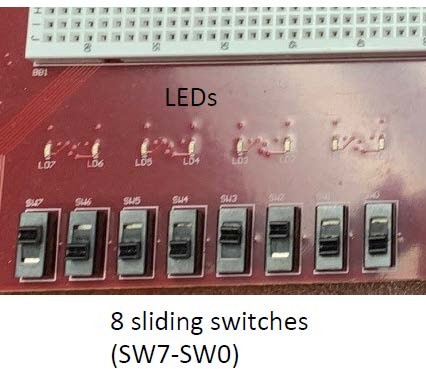
**Table 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Port | Width | I/O | Component | Mapping |
| Run | 1 | I | Sliding Switches | SW7 |
| Max\_Count | 7 | I | Sliding Switches | SW6 (MSB) – SW0 (LSB) |
| CLK | 1 | I | - | USER\_CLK |
| Digit\_1 | 4 | O | LEDs | LED3 (MSB) – LED0 (LSB) |
| Digit\_2 | 4 | O | LEDs | LED7 (MSB) – LED4 (LSB) |

## Description:

Design a programmable counter that outputs BCD and has a range of 0-99. The counter must count from zero to a user-input value. The valid input range is 0-99. The output from your design should be in binary-coded decimal. You will need to design a binary counter that counts at least to 99 (decimal), and a module which converts the binary output to two-digit BCD. A single-bit input, called run, determines the operation of the counter. When run is ‘0’, the counter output is set to zero and does not count. The user can set the maximum value up to 99 when run is ‘0’, using an input called max\_count. When run is set to ‘1’, the counter starts counting from zero, and stops when it reaches the maximum. Any changes to max\_count is ignored when run is set to ‘1’. Your design will have two 4-bit outputs; digit 1 and digit 2. Each is used to output one BCD digit. Digit 2 is the tens digit, digit 1 is the ones digit.

The Anvyl board includes a single 100MHz Crystal oscillator connected to pin D11. In this lab, we use 8 out of 14 LEDs (LED[0]-LED[7]), all the eight slide switches (SW[0]-SW[7]). Use SW[7]

(pin P8) for run and the other six switches for max\_count. The master UCF file and that of the tutorial make writing your UCF easy. This is a picture for the I/O you use in this lab.

## Included Verilog files:

You can modify these as you wish but some of them are already complete if you connect/use them as intended.

1. Binary\_bcd.v (complete)

This is for binary to BCD conversion of your count\_out, e.g., 00011001 in binary to its BCD which is 19. It uses an algorithm for conversion. No change needed as this file is already written for you.

1. Counter.v (complete)

Source template for a simple 7-bit binary counter with enable and synchronous reset. No change needed as this file is already written for you.

1. Prog\_counter.v (you need to complete)

Wrapper to add programmability to a 7-bit counter. This module contains the logic to stop a counter when it reaches a designated value. The maximum value is 99 (decimal).

1. Final\_bcd\_counter.v (you need to complete)

This is the top module for the programmable BCD counter. It instantiates your programmable 7-bit counter and your binary-to-bcd converter that can output two digits.

1. Lab\_board.v (complete) Connections to test the design.

Example waveform and testbench are also added in a folder for assisting you.

1. (30 pts.) Create a Verilog design (See Figure 1) which will provide the required functionality.
2. (20 pts.) Exhaustively test the design to ensure that it works under all conditions.
3. (20 pts., demonstration mark) Synthesize and map the design on ANVYL board using Table 2. Demonstrate to the TA, get your demonstration mark and add that page to the final PDF report.

**Deliverables (Only one .zip file per group)** A zipped file (.zip) which includes these two items:

1. Your group design files (Verilog Models and test benches). Organize your files in folders named Problem1, Problem2, etc. need to be in .zip file.
2. A concise PDF group report (that includes your Verilog code and simulation results as well as the graded demonstration page) needs to be included in the submitted .zip file.

**Who submits?** Only one of group members needs to submit. Choose one student among your group members (it does not matter who) and that person submits. TAs will mark all group members based on that equally. Do not submit the same .zip file for each student.

## PDF Report Organization to be included in your ZIP submission (A template is provided on Canvas):

* + Cover sheet
  + Marked and signed group demonstration page by the TA
  + Problem 1 and 2: Behavioral Verilog Code, Test Bench, and Simulation Results (Waveforms)
  + Problem 3: Pin Mapping file (UCF file that you used)
  + Feedback: Hours spent, Exercise difficulty (Easy, Medium, Hard)

## Important:

* + Do not discard your designs. They may be used as starting point for subsequent Labs.
  + You need to submit your group report on Canvas in ZIP format (only one .zip file which includes all the above).