# Lab 4: Counters

Submission Due Dates:

| | | |
|---|---|---|
| Demo: | 2021/10/26 | 17:20 |
| Source Code: | 2021/10/26 | 18:30 |
| Report: | 2021/10/30 | 23:59 |

## Objective

1. Getting familiar with the 7-segment display and pushbuttons on the FPGA board.
   Note: Every signal connected to the pushbutton (except the reset signal) should be properly processed with the debouncing and/or one-pulse converters.
2. Getting familiar with the counter designs and finite-state machines in Verilog.

## Action Items

**1. `lab4_1.v` (40%)**

Design a 2-digit BCD up/down counter which also displays the counting direction.

a. I/O list:

✔ Inputs: clk, rst, en, dir, speed_up, speed_down

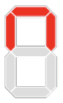
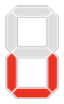
✔ Output: DIGIT[3:0], DISPLAY[6:0], max, min

b. **clk** (connected to **W5**): the clock input with the frequency of 100MHz.

c. **rst** (connected to **BTNC**): the positive-edge-triggered (active-high) reset signal to reset the counter value to decimal 00.

d. **en** (connected to **BTNU**): the control signal to toggle between the *start/resume* and *paus*e mode.
Press the button to start the counting. Press it again to pause the counting. Press it one more time to resume the counting, and so on. After the reset, the counter is initialized to 00 and remains unchanged. That is, the counter will be in the pause mode initially. The counter will stop at 99 eventually when counting up; the counter will stop at 00 at the end when counting down.

e. **dir** (connected to **BTND**):
After the reset, the counter is counting up. Pressing and holding the **dir** button will change the direction to count down. Release the button to change the direction to count up again. Pressing the **dir** button in the pause mode will **not** change the counting direction. Show the current counting direction on the second digit (from the left) of 7-segment. Use the up arrow to indicate the up direction; use the down arrow to indicate the down direction.

Up Arrow:  Down Arrow: 

f. **speed_up** (connected to **BTNR**):
After being reset, the counter changes every two seconds (e.g., it will take two seconds to go from 1 to 2). Press the **speed_up** button to increase the speed. The counting speed will become twice as fast every time you press it. The maximal speed is four times the original speed (i.e., the counter changes every 0.5 seconds). The speed will not increase any further after it reaches the maximum.

The leftmost digit of the 7-segment indicates the counting speed. After being reset, the digit is set to 0. Pressing the **speed_up** will add one to the value until the maximal speed (i.e., when the leftmost digit is 2).

g. **speed_down** (connected to **BTNL**):
On the contrary, pressing the **speed_down** can slow down the counter. The counting speed will become twice as slow every time you press it. The minimal speed is the original speed. The speed will not decrease any further after it reaches the minimum.

Since the leftmost digit of the 7-segment indicates the counting speed, press the **speed_down** will subtract one from the value until the minimal speed (i.e., when the leftmost digit is 0).

h. **DISPLAY[6:0]**: the signals to control the seven LED segments of the 7-segment display.

i. **DIGIT[3:0]**: the signals to control the four digits of the 7-segment display.
In this lab, the two rightmost digits are used to display the counting number; the two leftmost digits are used to display the counting speed and direction, respectively.

j. **max** (LED0): 1 if the counter reaches the largest number (99), and 0 otherwise.

k. **min** (LED1): 1 if the counter reaches the smallest number (00), and 0 otherwise.

You have to use the following template for your design:

```
module lab4_1 (
    input clk,
    input rst,
    input en,
    input dir,
    input speed_up,
    input speed_down,
    output [3:0] DIGIT,
    output [6:0] DISPLAY,
    output max,
    output min
  );

   // add your design here

  endmodule
```

Demo:[https://reurl.cc/EZ1OoA](https://reurl.cc/EZ1OoA)

## 2. `lab4_2.v` (60%)

Implement a timer with its four digits representing “0:00.0” and the rightmost digit representing 0.1 seconds. You can ignore the colon “:” and decimal point “.”. They are only to help you understand the format. You don’t need to show them on the 7-segment display.

a. I/O list:

- ✔ clk, rst, enter, input_number, count_down, en
- ✔ Output: DIGIT[3:0], DISPLAY[6:0], led0

b. There are three states of the machine: direction-setting state, number -setting state, and counting state.

c. **Direction-setting state**: After reset, the timer goes into the direction-setting state.
In this state, pressing the count_down, the timer will count down in the counting state. Otherwise, the timer will count up. Press again to dismiss the count-down. Press one more time to set the count-down again. Display ▬ ▬ ▬ ▬ on the 7-segment display in this state.

Press **enter** to go into the number setting state.

d. **count_down** (connected to BTNU):
In the **direction setting state,** press **count_down** can make the timer count down in

the **counting state**. Press it again to dismiss the countdown.

e. **led0** (connected to **LD0**):
The signal to control the rightmost LED. Light on the LED if the direction is to count down; light off otherwise.

f. **enter** (connected to **BTNR**):
In the **direction setting state,** press **enter** to go into the **number setting state.**
In the **number setting state,** press **enter** to adjust the next digit. After pressing the **enter** for four times, the timer will go into the **counting state.**
In the **counting state,** the **enter** doesn't work.

g. **Number setting state**: After pressing the enter in the direction setting state, the number goes into the number setting state. In this stage, you can set the goal for the timer.

Initially, the timer displays 0 0 0 0 on the 7-segment in this state. You can adjust numbers from the leftmost digit. Press the **input_number** button to increase the value by one for the current digit. The value goes back to 0 after reaching the max number. Press the **enter** to adjust the next digit to the right. After adjusting the rightmost digit, press the **enter** to go into the **counting stage**.

The four digits represent the minutes, 10 seconds, seconds, and 0.1 seconds from left to right, respectively. The timer can count up to 1:59.9 maximally (that is, the max number on the 7-segment display is 1599).

h. **input_number** (connected to **BTND**):

In the **number setting state,** press the **input_number** to increase the current digit number by one. After reaching the maximal value, the number goes back to 0.

Ex: For the first (leftmost) digit, the number is initially 0. Pressing the input_number will set it to 1. Since 1 is its maximal value, pressing the input_number again will set the number to 0.

i. **Counting state:**

After adjusting the rightmost digit and pressing enter, the timer goes into the counting stage. If the count_down is set in the **direction setting state, the** timer counts down from the goal (set in the number setting state) to zero**.** Otherwise, the timer counts up from zero to the goal**.** After reaching the goal (or reaching zero), the timer stops.

j. **clk** (connected to **W5**):
The clock input with the frequency of 100MHz.

k. **rst** (connected to **BTNC**):
The asynchronous active-high reset. The counter is reset to the **direction setting**

**state.**

l. **en** (connected to **V17**):
The control signal to toggle between the counting mode and pause mode. Switch on to start counting. Switch down to pause counting. In other states(that is not a counting state), en can be either functional or not (won't be tested during demo)

m. **DISPLAY[6:0]** :
The signals to control the seven LED segments on the 7-segment display.

n. **DIGIT[3:0]**:
The signals to control the four digits on the 7-segment display.

You have to use the following template for your design:

```
 module lab4_2 (
   input clk,
   input rst,
   input en,
   input input_number,
   input enter,
   input count_down,
   output [3:0] DIGIT,
   output [6:0] DISPLAY,
   output led0
);

// add your design here


endmodule
```

Demo:https://reurl.cc/1oGr39

**3. Bonus (5%)**

Create the exact 0.1-second timesteps for the lab4_2 to count up. You must explain your implementation both during the demo and in the report to get the extra points. Integrate the bonus feature in lab4_2, instead of having a separate design.

## Attention

- ✔ You should hand in **lab4_1.v, lab4_2.v**. If you have multiple modules for an Action Item, you must integrate them into a single Verilog file.
  - **Upload each source file directly. DO NOT hand in a compressed ZIP or RAR file!**

- ✔ You should also hand in your report as **lab4_report_StudentID.pdf** (i.e., lab4_report_108456789.pdf).

- ✔ Your report should include diagrams to explain your design (such as FSM, block diagram, etc.).

- ✔ You should be able to answer questions of this lab from TA during the demo.

- ✔ You need to generate the bitstream files before the lab demo. Prepare separate bitstream files for lab4_1, lab4_2, respectively, to make the demo process smooth.