

Lab 4: BCD Counters

Submission Due Dates:

Demo: 2023/10/17 17:20
 Source Code: 2023/10/17 18:30
 Report: 2023/10/22 23:59

Objective

- Getting familiar with the 7-segment display and pushbuttons on the FPGA board.
Note: Signals from pushbuttons should be processed by debouncing and one-pulse converters properly.
- Getting familiar with the counter designs and finite-state machines in Verilog.

Action Items

1 lab4_1.v (60%)

In this lab, you must design a 3-digit BCD reaction timer. This timer must be control by a finite-state machine (FSM) which controls the LEDs and 7-segment display based on the inputs from the pushbuttons.

a. I/O list

I/O	Connected to	Definition
clk	W5	Clock signal with the frequency of 100MHz
rst	SW0	Asynchronous positive reset signal, reset the counter back to the INITIAL state
stop	BTNC	Stop the counter in the COUNTING state and then enter the RESULT state.
start	BTNU	Changing the INITIAL state to the PREPARE state.
direction	BTND	Switching the counting direction between counting down and counting up.
LED[9:0]	LD9~LD0	Lighting up the LEDs based on the current counter's hundreds digit.
DISPLAY[6:0]		Controlling the seven LED segments of the 7-segment display.
DIGIT[3:0]		Enabling one of the 7-segment digits.

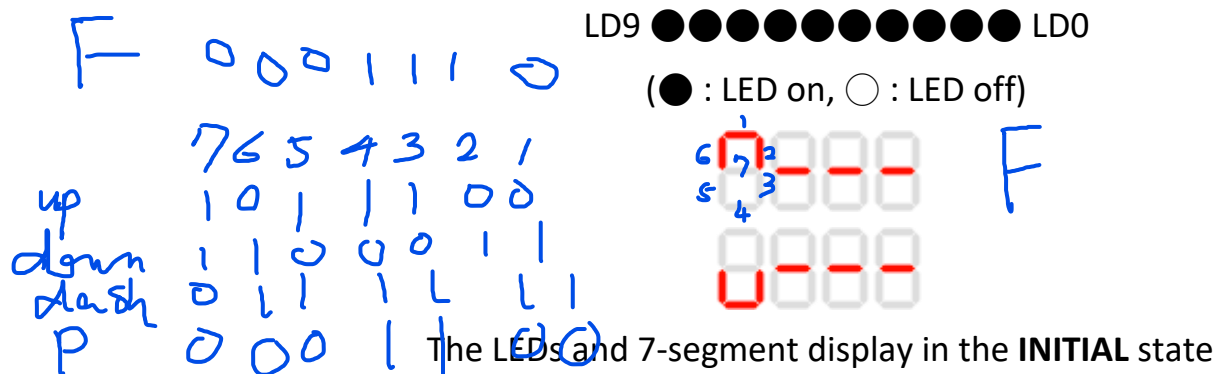
b. Finite state machine (FSM)

The FSM may consist of an **INITIAL** state to initialize the counter and set direction, a **PREPARE** state before entering the **COUNTING** state, a **COUNTING** state to increase or decrease the counter value based on the current direction (you may store the direction information in a register), and a **RESULT** state for showing the reaction

time. You can design your own variant instead.

✓ INITIAL state

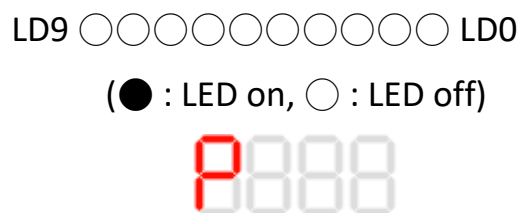
- Pressing the **direction** button can change the counting direction between UP and DOWN. After reset, the direction will be set as **UP**.
✧ NOTE: the direction button will only work in the **INITIAL** state.
- First, it initializes the current BCD counter **000 (UP)** or **999 (DOWN)** (it will be 000 after reset); the **LD9~LD0** are all **on**; the current direction will stay the same as the previous setting. Then, the 7-segment display will show the current direction and three dashes as below. The FSM can enter the **PREPARE** state by pressing the **start** button.



✓ PREPARE state

- After entering the **PREPARE** state, the counter will wait for **3 seconds**. After this time period, it will enter the **COUNTING** state automatically.
- In this state, all the LEDs will turn off and the 7-segment display will show a single "P" as in the following.

EX:



✓ COUNTING state

- In this state, the counter will start counting up/down from 000/999 (depending on the direction). The 7-segment display will display the current direction and counter value. When the **stop** button is pressed, the counter will stop and enter the **RESULT** state. Also, if the counter counts to 999 (when counting UP) or 000 (when counting DOWN), it will also stop counting and enter the **RESULT** state. That is, the counter will stop at 999 and enter the **RESULT** state eventually when the direction is UP; the counter will stop at 000

and enter the **RESULT** state when the direction is DOWN.

E.g., the counter counts up to 053

E.g., the counter counts down to 855

- The counter value will increase or decrease by one every **0.01 seconds** (that is, exactly 0.01 seconds) based on its current direction, UP or DOWN, respectively.
- The LEDs will light up based on the current counter's hundreds digit.

EX:

When the counter counts to 089, the LEDs will light up as follows:

LD9 ○○○○○○○○● LD0

When the counter counts to 199, the LEDs will light up as follows:

LD9 ○○○○○○○○●○ LD0

When the counter counts to 200, the LEDs will light up as follows:

LD9 ○○○○○○○○●○○ LD0

When the counter counts to 876, the LEDs will light up as follows:

LD9 ○●○○○○○○○○ LD0

(● : LED on, ○ : LED off)

✓ RESULT state

- When entering this state, the **LD9~LD0** will be flashing for 5 seconds (lighting up first for one second then off). After flashing, the LEDs will be all **on**. Therefore, the on-off sequence will be "**on off on off on**".
- Pressing the **start** button can go back to the **INITIAL** state with the counter paused.
- In this state, the leftmost digit of the 7-segment display will show the direction; the other three digits will show the counter value.

EX:

If you pressed the **stop** button in the **COUNTING** state when the counter counted down to 127, the 7-segment will be as follows in the **RESULT** state:

If you pressed the **stop** button in the **COUNTING** state when the counter counted up to 039, the 7-segment will be as follows in the **RESULT** state:

c. DEMO video

<https://youtu.be/Xe-hGBvvEck>

d. You have to use the following template for your design:

```
module lab4_1 (
  input wire clk,
  input wire rst,
  input wire stop,
  input wire start,
  input wire direction,
  output reg [3:0] DIGIT,
  output reg [6:0] DISPLAY,
  output reg [9:0] led
);
  /* Note that output ports can be either reg or wire.
   * It depends on how you design your module. */
  // add your design here
endmodule
```

2 lab4_2.v (40%)

In this lab, you need to design a Timing Game with a BCD timer. In this game, the timer value will be hidden and the user needs to press the **stop** button in a certain time interval.

a. I/O list:

I/O	Connected to	Definition
clk	W5	Clock signal with the frequency of 100MHz.
rst	SW0	Asynchronous positive reset signal; reset the machine back to the INITIAL state.
Digit_1	SW1	Select the units digit when configuring the counter.
Digit_2	SW2	Select the tens digit when configuring the

		counter.
Digit_3	SW3	Select the hundreds digit when configuring the counter.
stop	BTNC	Stop the counter in the COUNTING state.
start	BTNU	Changing the machine's state from the INITIAL state to the COUNTING state or from the FAIL/SUCCESS state to the INITIAL state.
increase	BTNR	Increase the chosen digit(s) by 1 in the INITIAL state.
decrease	BTNL	Decrease the chosen digit(s) by 1 in the INITIAL state.
direction	BTND	Switching the counting direction between counting down and counting up.
LED[15:0]	LD15~LD0	Lighting up based on the current state.
DISPLAY[6:0]		Controlling the seven LED segments of the 7-segment display.
DIGIT[3:0]		Enabling one of the 7-segment digits.

b. Finite state machine (FSM)

The FSM may consist of an **INITIAL** state to initialize the counter and set direction, a **COUNTING** state to increase or decrease the counter value based on the current direction (you may store the direction information in a register), and a **SUCCESS** state and a **FAIL** state for showing the pressing time. You can design your own variant instead.

✓ INITIAL state

- Pressing the **direction** button can change the counting direction between UP and DOWN. After reset, the direction will be set as **UP**, and the counter will be set as **000**. If the FSM enters the **INITIAL** state not because of the reset, it initializes the **counter to the value that you previously set**. The **LD15~LD0** are **all on**, and the **current direction** will be the same **as the previous setting**. The 7-segment display will show the current direction and the value that you previously set, as shown below. The FSM can enter the **COUNTING** state by pressing the **start** button.

LD15 ●●●●●●●●●●●●●●●● LD0

The LEDs in the **INITIAL** state

(● : LED on, ○ : LED off)



The 7-segment display in the **INITIAL** state

- NOTE: the direction button will **only** work in the **INITIAL** state. When the system is in the **INITIAL** state, you can use the switch (**Digit_1~Digit_3**) to select one or more desired digits to adjust by increasing or decreasing the selected digit(s) by one when the **increase** button or **decrease** button is pressed, respectively.

For example, if the **Digit_1** and **Digit_2** are switched to high and the current counter value is **523**, pressing the **increase** button will change the counter value to **534**. By contrast, pressing the **decrease** button will change the counter from **523** to **512**.

- When the selected digit's value is 9, pressing the **increase** button will reset its value to 0. For example, if **Digit_1** is selected and the current counter value is **529**, pressing the **increase** button will change the value to **520**. Similarly, when the selected digit's value is 0, pressing the **decrease** button will set its value to 9. For example, if **Digit_2** is selected and the current counter value is **902**, pressing the **decrease** button will change the value to **992**.

✓ COUNTING state

- In this state, the counter will start counting up/down from 000/999 (depends on the direction). The 7-segment display will show the current direction and counter value, and the LEDs will all light up in the first three seconds (**UP**:000~299, **DOWN**:999~700). After three seconds, the 7-segment display will show the direction and three dashes; the LEDs will all turn **off**. When the **stop** button is pressed, the counter will stop and enter the **SUCCESS** or the **FAIL** state. Also, if the counter counts to 999 (UP) or 000 (DOWN), it will also stop counting and enter the **FAIL** state automatically.

LD15 ●●●●●●●●●●●●●●●● LD0

The LEDs in the first three seconds

LD15 ○○○○○○○○○○○○○○○○○ LD0

The LEDs after three seconds

(● : LED on, ○ : LED off)



The counter counts to 053 with the up direction



The counter counts to 300~999 with the up direction



The counter counts to 855 with the down direction



The counter counts to 699~000 with the direction

- The counter value will be increased or decreased by one every **0.01 seconds** (that is, exactly 0.01 seconds) based on its current direction, UP or DOWN, respectively.
 - If the user presses the **stop** button between “**the number you set in the INITIAL state**” ± 100 , the counter will stop and enter the **SUCCESS** state. Otherwise it will enter the **FAIL** state.
- ✓ **FAIL state**
- When entering this state, the **LD15~LD0** will be flashing for 5 seconds (lighting up first for one second then off for another second). After flashing, the LEDs will be all **off**. Therefore, the on-off sequence will be “**on off on off on off**”.
 - Pressing the **start** button can go back to the **INITIAL** state.
 - In this state, the leftmost digit of the 7-segment display will show “F”, and the three other digits will show the current counter value.

EX:

If you pushed the **stop** button in the **COUNTING** state when the counter counted to 127 and 127 is not in the time period that you set in the **INITIAL** state, the 7-segment will be as follows in the **FAIL** state:



If you pushed the **stop** button in the **COUNTING** state when the counter counted to 039 and 039 is not in the time period that you set in the **INITIAL** state, the 7-segment will be as follows in the **FAIL** state:



- ✓ **SUCCESS state**
- When entering this state, the **LD15~LD0** will be flashing for 5 seconds. (lighting up first for one second then off). After flashing, the LEDs will be all

on. Therefore, the on-off sequence will be “on off on off on”.

- Pressing the **start** button can go back to the **INITIAL** state with the counter paused.
- In this state, the leftmost digit of the 7-segment display will show “S”, and the three other digits will show the current counter value.

EX:

If you pushed the **stop** button in the **COUNTING** state when the counter counted to 177 and 177 is in the time period that you set in the **INITIAL** state, the 7-segment will be as follows in the **SUCCESS** state:

S177

If you pushed the **stop** button in the **COUNTING** state when the counter counted to 839 and 839 is in the time period that you set in the **INITIAL** state, the 7-segment will be as follows in the **SUCCESS** state:

S839

c. DEMO video

https://youtu.be/1KL1Dq_5KHg

d. You have to use the following template for your design:

```
module lab4_2 (
  input wire clk,
  input wire rst,
  input wire Digit_1,
  input wire Digit_2,
  input wire Digit_3,
  input wire stop,
  input wire start,
  input wire increase,
  input wire decrease,
  input wire direction,
  output reg [3:0] DIGIT,
  output reg [6:0] DISPLAY,
  output reg [15:0] led
);
  /* Note that output ports can be either reg or wire.
  * It depends on how you design your module. */
```



```
// add your design here  
endmodule
```

3 Questions and Discussion

Please answer the following questions in your report.

- A. Why do we need the debounce and one-pulse modules? What is the relation between the FSM's clock rate and the one-pulse module's clock rate? Explain your reason. And what will happen when we implement with a wrong clock rate? (Please draw waveforms to explain)
- B. Please propose two different design methods to manipulate two or more clocks in one module.

4 Guidelines for the report

Refer to the guidelines in the report template (or in the previous lab assignments).

Grading policy (subject to change): Part (A): 35%; Part (B): 50%; Part (C): 10%; (D): 5%

Attention

- ✓ DO NOT copy-and-paste code segments from the PDF materials. It may also paste invisible non-ASCII characters, leading to hard-to-debug syntax errors.
- ✓ In this lab, we provide you **onepulse.v**, **debounce.v** and **clock_divider.v**. Do not include them in your submissions. We will take care of them when verifying your source code.
- ✓ If you have two or more modules used for any specific lab, merge them into one Verilog file before the submission.
- ✓ You should submit **two** source files, including **lab4_1.v** and **lab4_2.v**. **Upload each source file individually. DO NOT hand in any compressed ZIP files, which will be considered an incorrect format.**
- ✓ You should also hand in your report as **lab4_report_StudentID.pdf** (i.e., lab4_report_111456789.pdf).
- ✓ You should be able to answer questions about this lab from TA during the demo.
- ✓ You need to prepare the bitstream files before the lab demo to make the demo process smooth.
- ✓ Feel free to ask any questions about the specification on the EECLASS forum.