Lab 04: BCD Counter (Oct 15, 2019)

Submission deadlines:

Source Code:	18:30, Oct 22, 2019
Report	23:59, Oct 27, 2019

Objective

To be familiar with the 7-segment display and switches on the FPGA board using the counter designs with simple finite-state machine (FSM)

Action Items

1. (30%) Use the switches to provide binary inputs. Display the decimal values on the 7-segment display. Only the numbers of 0 to 9 are supported. If the input is greater than 9, simply display 9. The four decimal digits from left to right on the 7-segment display should correspond to SW[15:12], SW[11:8], SW[7:4], and SW[3:0], respectively.

   Note：

   - You should use the clock frequency of clk/2^{13} to update the 7-segment display.

   ### I/O signal specification

   - **clk**: (connected to pin **W5**) the clock input with the frequency of 100MHz
   - **SW[15:0]**: the binary input of the four digits.
     For example, if SW =16'b0001_1000_0000_0101, the 7-segment display should show 1805.
   - **reset**: (connected to **BTNC**) the **asynchronous** active-high reset. The 7-segment display should show 0000 when being reset.
   - **DISPLAY[6:0]:** the signals to control the seven LED segments on the 7-segment display.
   - **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_1 (
    input wire [15:0] SW,
    input wire clk,
    input wire reset,
    output wire [3:0] DIGIT,
    output wire [6:0] DISPLAY
);
    // add your design here
endmodule
```

Demo

https://youtu.be/TGoO76aMAPQ

2. (35%) Design the 2-digit BCD up/down counter with a record button to record the counting number.

Note：

- The counter will stop at 99 eventually when counting up; the counter will stop at 00 eventually when counting down.
- Every signal connected to the push button (except the reset signal) should be properly processed (with debounce and/or one pulse).
- You should use the clock frequency of clk/2^{23} to update your FSM, and use the clock frequency of clk/2^{25} to update counting number.

I/O signal specification

- **clk**: (connected to pin **W5**) the clock input with the frequency of 100MHz
- **en:** (connected to **BTNU**) the control signal to toggle between the *start/resume* and *pause* 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 and so forth.

  After the reset, the counter stays at **00** (that is, the pause mode).
- **dir**: (connected to **BTND**) after the reset, the counter is to count up. Press the dir button to change the direction to count down; Press it again, the counter will change the direction to count up.

Note: Pressing the dir button in the pause mode will change direction to count as well.

- **record:** (connected to **BTNL**) press the record button will make the **two leftmost** digits display the current counting number. If you press the record button and hold for a long time, the two leftmost digits only record the counting number at the moment when you pressed. In other words, only one number will be recorded, no matter how long you press.
  After the reset, the two leftmost digits on the 7-segment display is 00 initially.
- **reset:** (connected to **BTNC**) the **asynchronous** active-high reset. The counter is reset to **00** when being reset.
- **DISPLAY[6:0]:** the signals to control the seven LED segments on the 7-segment display.
- **DIGIT[3:0]**: the signals to control the four digits on the 7-segment display.
  In this lab, the **two rightmost** digits are used to display the counting number, and the **two leftmost** digits are used to display the recorded number.
- **max**: (connected to **LD0**) 1 if the counter reaches the largest number (99), and 0 otherwise. The LED is turned on when max is 1, and is turned off otherwise.
- **min**: (connected to **LD1**) 1 if the counter reaches the smallest number (00), and 0 otherwise. the LED is turned on when min is 1, and is turned off otherwise.

You have to use the following template for your design.

```
module lab4_2 (
    input wire en,
    input wire reset,
    input wire clk,
    input wire dir,
    input wire record,
    output wire [3:0] DIGIT,
    output wire [6:0] DISPLAY,
    output wire max,
    output wire min
);
        // add your design here
endmodule
```

Demo

https://youtu.be/7DHQAWoHb6E

3. (35%) Implement a count-down counter. The two leftmost digits represent the **minute** and the two rightmost digits represent the **second**. It has two mode, the **setting mode** and **counting mode.** You can use the **switch** to switch between the two modes.

   Note:

   - The range of the counter is **00:00~59:59**. If you press **sec_plus** button when the counter is 59:59, it should become 00:00.
   - When switching from the setting mode to the counting mode, the counter is paused.
   - When switching from the counting mode to the setting mode, the counter is set to **0000**.
   - If the counter is in the setting mode, pressing **en** button will not change anything.
   - If the counter is in the counting mode, pressing **min_plus or sec_plus** button will not change anything

- You can choose a proper clock divider to approximate the 1 second timescale as close as possible. (Hint: use a faster frequency during your coding to speed up the development, just as the demo clip shows.)

I/O signal specification

- **clk:** (connected to pin **W5**) the clock input with the frequency of 100MHz
- **mode:** (connected to pin **V17**) when the mode is 0, the counter is in the setting mode; when the mode is 1, the counter is in the counting mode.
- **min_plus**: (connected to pin **BTNL**) in the setting mode, pressing the min_plus button will increase the **minute** value by 1. Even if you press and hold the button for a long time, the number will increase by 1.
- **sec_plus:** (connected to pin **BTNR**) in setting mode, pressing the sec_plus button will increase the **second** value by 1. Even if you press and hold the button for a long time, the number will increase by 1.
  If you press the sec_plus button when the **second** value is **59**, the minute value will increase by 1 and the second value will return to 00.
- **en:** (connected to **BTNU**) the control signal to toggle between the count-down mode and pause mode. Press the button to start the count down. Press it again to pause the counting. Press it one more time to resume the counting, and so on and so forth. The counter is reset to the pause mode.
- **reset**: (connected to **BTNC**) the **asynchronous** active-high reset. The counter is reset to **00:00**.
- **DISPLAY[6:0]:** the signals to control the seven LED segments on the 7-segment display.
- **DIGIT[3:0]**: the signals to control the four digits on the 7-segment display.
  In this lab, the **two leftmost** digits are used to display the minute value, and the **two rightmost** digits are used to display the second value.
- **stop:** (connected to **LD0**) 1 if the counter counts down to 00:00 in counting mode (mode = 1), and 0 otherwise. The LED is turned on when the stop is 1, and is turned off otherwise.

You have to use the following template for your design.

(Hint：You can maintain one major counter for the counting down, and convert its value to minutes and seconds. In the setting mode, you can add proper minute/second value to this counter accordingly.)

```
module lab4_3 (
    input wire en,
    input wire reset,
    input wire clk,
    input wire mode,
    input wire min_plus,
    input wire sec_plus,
    output wire [3:0] DIGIT,
    output wire [6:0] DISPLAY,
    output wire stop
);
    // add your design here
endmodule
```

Demo

https://youtu.be/BkezQs7vSBE

Bonus

1. (10%) Create the exact 1 second timestep for lab4_3 to count down. You have to explain the design in your report.

   You have to use the following template for your design.

```
module lab4_bonus (
    input wire en,
    input wire reset,
    input wire clk,
    input wire mode,
    input wire min_plus,
    input wire sec_plus,
    output wire [3:0] DIGIT,
    output wire [6:0] DISPLAY,
    output wire stop
);
    // add your design here
endmodule
```

Attention

✓ You should also hand in your report as **lab04_report_*StudentID*.pdf** (i.e., lab04_report_107080001.pdf).

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