# Lab 4: BCD and Pingpong Counters

## Due: 18:30, October 18, 2016

---

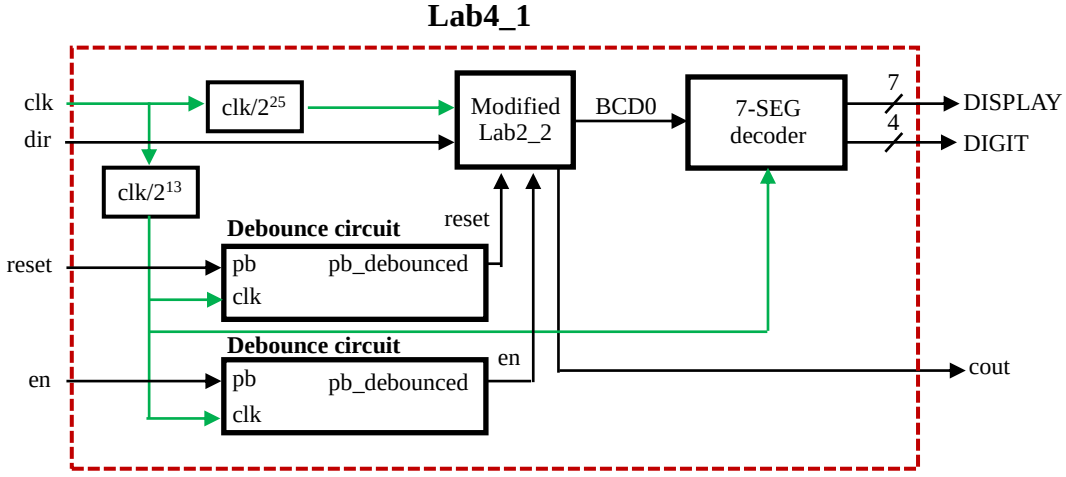
## Objective

- To be familiar with the FPGA design flow and the demo-board I/Os.

## Action Items

1. Modify and program the 1-digit BCD up/down counter (Lab2_2) to the FPGA on the demo board. You must follow the block diagram shown below to write your Verilog code. The I/O signals of the counter are described below.

   I/O signal specification:
   **clk**: clock signal with the frequency of 100MHz (connected to pin **W5**)
   **dir**: 1 to count up, and 0 to count down (connected to **SW0**)
   **reset**: asynchronous active-high reset (connected to **BTNC**); the counter is reset to 0 when the **reset** signal is 1
   **en**: pushing the pushbutton odd times to start/resume counting, and even times to pause (connected to **BTNU**)
   **DISPLAY[6:0]**: signals to show the BCD digit of the counter on the rightmost 7-segment display
   **DIGIT[3:0]**: signals to enable one 7-segment display
   **cout**: the carry-out bit of the counter (connected to **LD0**); the LED is turned on when cout is 1, and is turned off otherwise



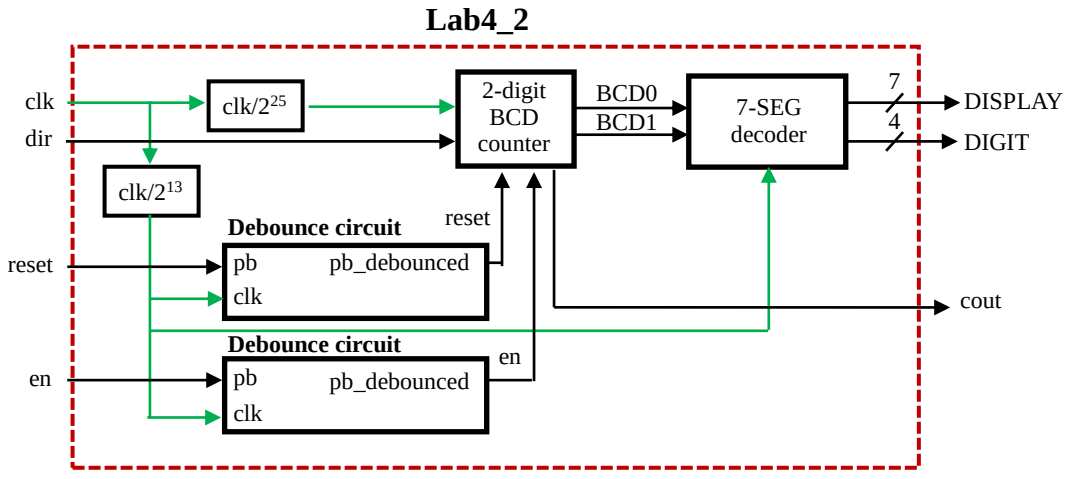
You have to use the following template for your design.

```
module Lab4_1(DIGIT, DISPLAY, cout, en, reset, dir, clk);

input en;
input reset;
input dir;
input clk;
output [3:0] DIGIT;
output [6:0] DISPLAY;
output cout;
      // add your design here
endmodule
```

2. Design and program a 2-digit BCD up/down counter to the FPGA on the demo board. The I/O signals and the block diagram of the counter are shown below.

   I/O signal specification:
   **clk**: clock signal with the frequency of 100MHz (connected to pin **W5**)
   **dir**: 1 to count up, and 0 to count down (connected to **SW0**)
   **reset**: asynchronous active-high reset (connected to **BTNC**); the counter is reset to 00 when the **reset** signal is 1
   **en**: pushing the pushbutton odd times to start/resume counting, and even times to pause (connected to **BTNU**)
   **DISPLAY[6:0]**: signals to show one of the two BCD digits of the counter on one of the two rightmost 7-segment displays
   **DIGIT[3:0]**: signals to enable one 7-segment display
   **cout**: the carry-out bit of the counter (connected to **LD0**); the LED is turned on when cout is 1, and is turned off otherwise



You have to use the following template for your design.

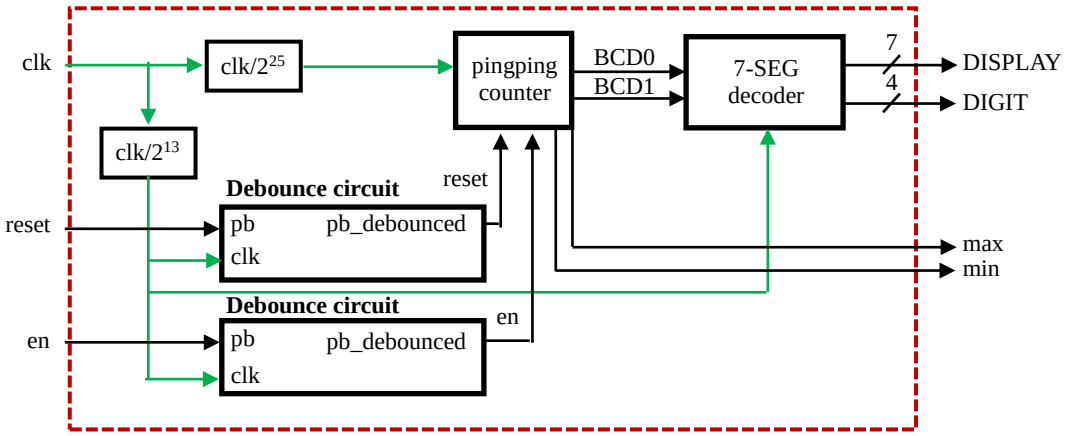
```
module Lab4_2(DIGIT, DISPLAY, cout, en, reset, dir, clk);

input en;
input reset;
input dir;
input clk;
output [3:0] DIGIT;
output [6:0] DISPLAY;
output cout;
    // add your design here
endmodule
```

3. Modify your 2-digit BCD up/down counter to model a negative-edge triggered 2-digit pingpong counter. The counter counts down from 99 to 00 and then counts up from 00 to 99. The counter will repeatedly count down and count up until it is disabled. The I/O signals and the block diagram of the counter are shown below.

   I/O signal specification:
   **clk**: clock signal with the frequency of 100MHz (connected to pin **W5**)
   **reset**: asynchronous active-high reset (connected to **BTNC**); the counter is reset to 99 when the **reset** signal is 1
   **en**: pushing the pushbutton odd times to start/resume counting, and even times to pause (connected to **BTNU**)
   **DISPLAY[6:0]**: signals to show one of the two digits of the counter on one of the two rightmost 7-segment displays
   **DIGIT[3:0]**: signals to enable one 7-segment display
   **max**: 1 if the counter reaches the largest number 99, and 0 otherwise (connected to **LD0**); the LED is turned on when **max** is 1, and is turned off otherwise
   **min**: 1 if the counter reaches the largest number 00, and 0 otherwise (connected to **LD1**); the LED is turned on when **min** is 1, and is turned off otherwise

**Lab4_3**




You have to use the following template for your design.

```
module Lab4_3(DIGIT, DISPLAY, max, min, en, reset, clk);

input en;
input reset;
input clk;
output [3:0] DIGIT;
output [6:0] DISPLAY;
output max;
output min;
      // add your design here
endmodule
```