Lab 03: Clock Divider and LED Controller

(Oct 9, 2018)

Submission deadlines:

Source Code:	18:30, Oct 9, 2018
Report	23:59, Oct 14, 2018

Objective

Be familiar with the clock divider and LED control on the FPGA demo board.

Action Items

1.

Write a Verilog module for the clock divider that divides the frequency of the input clock by 2^{26} to get the output clock.

You should use the following template for your design and name the file as "clock_divider.v".

(The parameter statement is required.)

```
module clock_divider(clk, clk_div);
    parameter n = 26;
    input clk;
    output clk_div;

    // add your design here

endmodule
```

2.

Write a Verilog module of the LED Controller which is synchronous with the clock whose frequency is obtained by dividing the frequency of Basys3's clock,100MHz, by 2^{26}. Also, program your LED Controller to the FPGA demo board. You should use the clock divider you designed in action item 1.

Here are the inputs and outputs constraints:

✓ When the rst_n is 0, the left-most LED (L1) is ON, and others are OFF.

✓ When the en is 1 and dir is 1, the LED will begin to shift (turn on/off) from left to right synchronized to the clock.

✓ When the en is 1 and dir is 0, the LED will begin to shift (turn on/off) from right to left synchronized to the clock.

✓ When the en is 0, hold the LEDs.

IO Connection:

clk	connected to W5
rst_n	connected to W16
en	connected to V17
dir	connected to V16
led	connected to LD15-LD0

You should use the following template for your design and name the file as "lab3_1.v".

Demo: https://youtu.be/ynIQ3fyP_6o

```
module lab3_1(clk, rst_n, en, dir, led);
    input clk;
    input rst_n;
    input en;
    input dir;
    output[15:0] led;

    //add your design here

endmodule
```

3.

Write a Verilog module of the LED Controller which is synchronous with the clock whose frequency is obtained by dividing the frequency of Basys3's clock,100MHz,by 2^{23} or 2^{26}.

Divide the LEDs on the FPGA board into two parts, the left part (LD15-LD8) and the right part (LD7-LD0). LEDs will begin to shift (turn on/off) from left to right synchronized to the clock.

Here are the inputs and outputs constraints:

✓ When the rst_n is 0, the left-most LED (L1) of the left part is ON; the left-most LED (V14) of the right part is ON; others are OFF.

✓ When the en is 1 and the speed is 0, the LED will begin to shift (turn on/off) from left to right synchronized to the clock. The left part operates at the divided-by-2^{26} clock, and the right part operates at the divided-by-2^{23} clock.

✓ When the en is 1 and the speed is 1, the LED will begin to shift (turn on/off) from left to right synchronized to the clock. But the left part operates at the divided-by-2^{23} clock, and the right part operates at the divided-by-2^{26} clock.

✓ When the en is 0, hold the LEDs.

✓ After the left part runs 5 laps, it will hold on LD8.

IO Connection:

clk	connected to W5
rst_n	connected to W16
en	connected to V17
speed	connected to V16
led	connected to LD15-LD0

You should use the following template for your design and name the file as "lab3_2.v".

Demo: https://youtu.be/8rrH6RldmQE

```
module lab3_2(clk, rst_n, en, speed, led);
    input clk;
    input rst_n;
    input en;
    input speed;
    output[15:0] led;

    //add your design here

endmodule
```

4.

Bonus:

Write a Verilog module for a clock divider that divides the frequency of the input clock by 7 to get the output clock. (Only the simulation is needed.)

You should use the following template for your design and name the file as "lab3_3.v".

```
module lab3_3(clk, rst_n, clkDiv7);
    input clk;
    input rst_n;
    output clkDiv7;

    // add your design here

endmodule
```

Attention

1. You may have to change your runtime (ns) in "Simulation Settings" to fit your testbench settings before you run the simulation.
2. You should hand in clock_divider.v, lab3_1.v, lab3_2.v, (and lab3_3.v, optional) and any other design files you create (you should describe it clearly in the report if you need additional design files). (DO NOT hand in a compressed ZIP file!)
3. You should also hand in your report as lab03_report_StudentID.pdf (i.e., lab03_report_105080001.pdf).
4. You should be able to answer questions of this lab from TA during the demo.
5. You need to generate bitstream before demo.(lab3_1 and lab3_2).