Lab 3: Clock Divider and LED Controller

Submission Due Dates:

Demo: 2021/10/12 17:20
Source Code: 2021/10/12 18:30
Report: 2021/10/17 23:59

Objective

1. Getting familiar with the clock divider and LED control on the FPGA demo board.
2. Practicing finite-state machines (FSMs) in Verilog.

Action Items

1 Module: clock_divider (10%)

Write a Verilog module for the clock divider that divides the frequency of the input clock by 2^{25} to get the output clock. Here is the template you should use:

```
module clock_divider #(parameter n=25) (
  input clk,
  output clk_div
);

   // add your design here

endmodule
```

2 lab3_1.v (30%)

Write a Verilog module of the LED Controller, which is synchronous with the positive clock edges. The clock frequency is obtained by dividing Basys3's onboard 100MHz clock by either 2^{24} or 2^{27}. Also, download and run your LED Controller on the FPGA board. You should use this clock divider in the following designs (e.g., lab3_2.v and lab3_3.v).

Here are the specification of inputs and outputs:

- ✓ If **rst** == **1**: all LEDs are ON.
  - The reset is positive-edge triggered.
- ✓ If **en** == **0**: hold the LEDs unchanged.
- ✓ If **en** == **1**: All LEDs will turn on and off repeatedly. To be specific, all LEDs will turn on for 1 cycle and off for another 1 cycle, and so on.
  - If **speed** == **1**: all LEDs turn on and off at the clock rate of (100 MHz / 2^{27}).
  - If **speed** == **0**: all LEDs turn on and off at the clock rate of (100 MHz / 2^{24}).

- ✓ There is a demo video for your reference.

IO Connection:

clk	connected to W5
rst	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_1.v".

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

   // add your design here

endmodule
```

Demo: https://youtu.be/Ybo8-2a5Z3w

3 lab3_2.v (60%)

Write a Verilog module of the LED Controller which is synchronous, with the positive clock edges. The clock frequency is obtained by dividing Basys3's onboard 100MHz clock by 2^{25}. This controller supports three different modes of LED light shows sequentially. In these three shows, LEDs behave as follows:

1. **Flashing mode**

   All LEDs will turn on and off repeatedly. After 6 on-off cycles, the controller goes to **Shifting mode**.

2. **Shifting mode**

   Initially, the odd-indexed LEDs (e.g., LD15, LD13, LD11, …, LD3, LD1) are ON; the even-indexed LEDs (e.g., LD14, LD12, LD10, …, LD2, LD0) are OFF.

   E.g., the initial state of Shifting mode: (●: LED on, ○: LED off)

(LD15) ●○●○●○●○●○●○●○●○ (LD0)

Then, the LED will begin to shift from left to right (when **dir==0**) or to shift from right to left (when **dir==1**) synchronized to the clock. When all LEDs are OFF, the controller goes to **Expanding mode**.

3. Expanding mode

Initially, the middle LEDs (LD8 and LD7) are ON, and others are OFF.

E.g., the initial state of Expanding mode: (●: LED on, ○: LED off)

(LD15) ○○○○○○○●●○○○○○○○ (LD0)

a. When dir==0, the LEDs will begin to expand at each clock edge. To be specific, the two LEDs next to the outermost lighted LEDs will turn on. So, the lightbar will expand gradually.

E.g., the 2nd to 4th states of Expanding mode if dir==0: (●: LED on, ○: LED off)

(LD15) ○○○○○○●●●●○○○○○○ (LD0)

(LD15) ○○○○○●●●●●●○○○○○ (LD0)

(LD15) ○○○○●●●●●●●●○○○○ (LD0)

b. When dir==1, the LEDs will begin to shrink at each clock edge. To be specific, the outermost lighted LEDs will turn off. So, the lightbar will shrink gradually until all LEDs are OFF.
E.g., the next four following states of Expanding mode if dir==1:

(LD15) ○○○○○●●●●●●○○○○○ (LD0)

(LD15) ○○○○○○●●●●○○○○○○ (LD0)

(LD15) ○○○○○○○●●○○○○○○○ (LD0)

(LD15) ○○○○○○○○○○○○○○○○ (LD0)

After all the LEDs go OFF, they will keep OFF. Afterward, if the dir switches to 0, the controller will jump to the initial state of Expanding mode.

The controller will go to **Flashing mode** only after all LEDs are ON.

Here are the specification of inputs and outputs:

- ✓ If **rst** == **1**: all LEDs are ON.
- ✓ If **en** == **0**: hold the LEDs unchanged.
- ✓ If **en** == **1**:
  - If in **Flashing mode**: all LEDs turn on and off repeatedly.
  - If in **Shifting mode**:
    - ➢ If **dir == 0**: the LED will begin to shift from left to right.
    - ➢ If **dir == 1**: the LED will begin to shift from right to left.
  - If in **Expanding mode**: the LEDs will begin to expand from middle.
    - ➢ If **dir == 0**: the LEDs next to the outermost lighted LEDs will turn on.
    - ➢ If **dir == 1**: the outermost lighted LEDs will turn off.
- ✓ There is a demo video for your reference.

IO Connection:

clk	connected to W5
rst	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_2.v”.

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

   // add your design here

endmodule
```

Limit:

1. You must have one finite state machine (FSM) in this design.
2. The FSM should have at least three states. Additional states are acceptable. You

may also design multiple FSMs. But remember to explain your design in the report.

Demo: https://youtu.be/vCKf_2zAeO8

4 Bonus: `lab3_3.v` (Extra 10%)

Write a Verilog module of the LED Controller, synchronous at different clock rates dividing from Basys3's 100MHz clock.

There are two LED runners, called Mr. 1 and Mr. 3.

- ✓ Mr. 1 and Mr. 3 race on the 16 LEDs of the FPGA Demo board.
- ✓ Use one single LED to represent Mr. 1.
- ✓ Use three consecutive LEDs to represent Mr. 3. (His position is determined by the middle LED.)
- ✓ If **rst** == **1**:
  - Mr. 1's position is at LD15.
  - Mr. 3's position is at LD1.

    E.g., the initial state: (●: LED on, ○: LED off)

    Mr. 1: (LD15) ●○○○○○○○○○○○○○○○ (LD0)

    Mr. 3: (LD15) ○○○○○○○○○○○○○●●● (LD0)
- ✓ If **en** == **0**: Mr. 1 and Mr. 3 will all hold at their current positions.
- ✓ If **en** == **1**:
  - ■ **Mr. 1** will begin to rotate from left to right. Its position after LD0 is LD15.
  - ■ **Mr. 3** will begin to rotate from right to left. Its position after LD15 is LD0.
  - If **speed** == **0**: **Mr. 3** runs at the clock rate of (100 MHz / 2^{25}); **Mr. 1** runs at the clock rate of (100 MHz / 2^{23}), respectively.
  - If **speed** == **1**: **Mr. 3** runs at the clock rate of (100 MHz / 2^{26}); **Mr. 1** runs at the clock rate of (100 MHz / 2^{24}), respectively.

- ✓ When they collide with each other, Mr. 1 will change its direction, but Mr. 3 will not.
  - ➢ Collision Situation 1 (Touch): Mr. 1 touches one end of Mr.3

E.g.

Mr. 1: (LD15) ○○○○○⇨●○○○○○○○○○○(LD0)

Mr. 3: (LD15) ○○○○○○●●●○○○○○○○(LD0)

  - ➢ Collision Situation 2 (Overlap): Mr. 1 overlaps one end of Mr. 3

E.g.

Mr. 1: (LD15) ○○○○○○⇨●○○○○○○○○○ (LD0)

Mr. 3: (LD15) ○○○○○○●●●○○○○○○ (LD0)

(Hint: At what condition will Collision Situation 2 happen? Given the different clock rates of the two runners. You need to answer this question during demo.)

➢ Collision Situation 3 (Overlap): Mr. 1A overlaps the middle of Mr. 3.

E.g.

Mr. 1: (LD15) ○○○○○○○●○○○○○○○○ (LD0)

Mr. 3: (LD15) ○○○○○○●●●○○○○○○○ (LD0)

✓ There is also a demo video for your reference.

IO Connection:

clk	connected to W5
rst	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_3.v”.

```
module lab3_3(
    input clk,
    input rst,
    input en,
    input speed,
    output [15:0] led
);

    // add your design here

endmodule
```

Demo: https://youtu.be/4VmqjTtBUts

Attention

- ✓ Please refer to Lab 0 and Lecture 03 for the FPGA implementation flow using Vivado.

- ✓ You may need to change your runtime (ns) in “Simulation Settings” to fit the testbench settings before the simulation.

- ✓ You should hand in **lab3_1.v, lab3_2.v (and lab3_3.v optionally)**. If you have multiple modules for an Action Item, you must integrate them into a single Verilog file (e.g., lab3_1 and clock_divider in lab3_1.v; lab3_2 and clock_divider in lab3_2.v, etc.).
  - ■ **Upload each source file directly, DO NOT hand in a compressed ZIP or RAR file!**

- ✓ **DO NOT** copy-and-paste code segments from the PDF materials. Occasionally, it will also paste invisible non-ASCII characters and lead to hard-to-debug syntax errors.

- ✓ You should also hand in your report as **lab3_report_StudentID.pdf** (i.e., lab3_report_108080001.pdf).

- ✓ 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 lab3_1, lab3_2 (and lab3_3.v optionally), respectively, to make the demo process smooth.