Lab 3: Clock Divider and LED Controller

Due: 18:30, Oct 11, 2016

Objectives

- To be familiar with how to design a designated frequency divider.
- To be familiar with the FPGA design flow and the demo-board I/Os.

Action Items

1. Write a Verilog module for a clock divider that divides the frequency of the input clock (**clk**) by **(2^26)** to get the frequency of the output clock (**clk26**).

   You have to use the following template for your design.

```
module clock_divider(clk, clk26);
input clk;
output clk26;
   // add your design here
endmodule
```

2. Write a Verilog module for an LED controller which is synchronous with a clock whose frequency is obtained by dividing the frequency of Basys3's clock (**clk**), 100MHz, by **(2^26)**. Also program your LED controller to the FPGA on the demo board.

   The I/O signals of the LED controller are explained below:

   clk: positive-edge triggered clock signal with the frequency of 100MHz (connected to pin **W5**)

   reset: asynchronous active-high reset (connected to **V17**)

   LED[15:0]: signals to turn on/off the 16 LEDs **LD15-LD0** (connected to 16 LED pins)

   The 16 LEDs on the demo board are evenly divided into the left part and the right part such that each part has 8 LEDs. To turn on or turn off an LED is determined by the LED controller and is synchronous

with a clock of frequency **100MHz/(2^26)**. More details are described below:

- Whenever the reset is 1, all the LEDs are turn off.
- When the reset is 0, the 8 LEDs of the left part will be turned on one by one from right to left, and then turned off one by one from left to right.
- After each LED of the left part is turned on once and turned off once, the 8 LEDs of the right part will be turned on one by one from left to right, and then turned off one by one from right to left.
- The turn-on/turn-off of LEDs will be alternate between the left part and the right part until the reset becomes 1.

You have to use the following template for your design. You may reuse the module clock_divider from Action Item 1.

```
module lab3_1(clk,reset,LED);
input clk, reset;
output [15:0]LED;
// add your design here
endmodule
```

Note: You can download the file "*Basys3_Master.xdc*" from iLMS, and then edit it to get the required *.xdc* file for your design.

3. Modify the module of your LED controller by adding an input **sel_clk** (connected to pin **V16**) to select the clock frequency for turning on/off LEDs. When **sel_clk** is 1, the LED controller selects the frequency of 100 MHz divided by **(2^19)**. When **sel_clk** is 0, the LED controller selects the frequency of 100 MHz divided by **(2^26)**.

   You have to use the following template for your design.

```
module lab3_2(clk,reset,sel_clk,LED);
input clk, reset, sel_clk;
output [15:0]LED;
// add your design here
endmodule
```

Reference Videos

- lab3_1: https://youtu.be/jMheQuNoIUs
- lab3_2: https://youtu.be/2wlaIOo6YBE