CS220A Lab#3
Design of Sequential Logic

Mainak Chaudhuri
Indian Institute of Technology Kanpur

Sketch

- Assignment#1: blinking LED
- Assignment#2: rippling LEDs

Lab#3

- Make new folders Lab3_1, Lab3_2 under CS220Labs to do the assignments
- Refer to lab#1 slides for Xilinx ISE instructions
- Finish pending assignments from lab#1 and lab#2 first

Assignment#1

- Blinking LED
  - Goal is to make LED0 blink once every one second
  - The on-board clock of FPGA runs at 50 MHz
    - Read Chapter 3 of https://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf
    - The on-board clock at pin C9 will be the input clock to your design
  - The idea is to send an output of 1 to LED0 once every 50000000 cycles of the clock, keep it on for the next 25000000 cycles, then send an output of 0 to LED0

Assignment#1

- Blinking LED
  - Write a Verilog module that does the following
    - Input is clk and output is led0
    - On every posedge of clk, increment a counter by 1
    - If the counter has reached OFF_TIME, set led0 to 0
    - If the counter has reached ON_TIME, set led0 to 1, reset the counter back to 0
    - Initialize the counter to 0
    - led0 and counter should be declared as reg
    - Use sequential assignment for led0 and counter
    - `define OFF_TIME 25000000
    - `define ON_TIME (OFF_TIME*2)

Assignment#1

- Blinking LED
  - Write a Verilog test fixture for ISim simulation
  - Use PlanAhead to assign pins to clk and led0
    - Manually add the following line in the .ucf file

    NET "clk" PERIOD = 20.0ns HIGH 50%;
  - Synthesize the hardware
  - Try changing OFF_TIME and see the effect on blinking rate
- This is an example of an FSM where state sequencing is done using a counter as opposed to a PLA or a ROM

Assignment#2

- Rippling LEDs
  - The goal is to blink LED0, LED1, …, LED7 in that sequence every one second
    - The blinking LED will keep shifting to the left, wrap back, and keep rotating
    - It will look like a ripple of LED light rotating continuously

Assignment#2

- Rippling LEDs
  - Write a Verilog module that has clk as input and led0, led1, …, led7 as outputs
    - All outputs should be reg
    - Initialize led0 to 1 and counter to 0
    - On posedge clk, increment counter by 1
    - When the counter reaches SHIFT_TIME do the following and reset counter to zero

```
led1 <= led0;
led2 <= led1;
…
led7 <= led6;
led0 <= led7;
```

    - `define SHIFT_TIME 50000000

Assignment#2

- Rippling LEDs
  - Write a Verilog test fixture for ISim simulation
  - Use PlanAhead to assign pins to clk and led0, led1, …, led7
    - Manually add the following line in the .ucf file

    NET "clk" PERIOD = 20.0ns HIGH 50%;
  - Synthesize the hardware
  - Try changing SHIFT_TIME and observe its effect on the ripple rate
  - Try initializing multiple LEDs to 1 and see if the whole pattern is rotating
  - Try changing the direction of the ripple to right

Assignment#2

- Rippling LEDs
  - This is an example of an FSM where state sequencing is done using a counter as opposed to a PLA or a ROM