Lab 1. LED Display

I. Purpose

This lab aims to learn a whole process of FPGA programming with a simple example of LED display. We will implement LED display modules using FPGA pins such as LEDs and buttons.

II. Problem Statement

Problem 1A (Week 3): Implement an LED blinking system

1) Implement an LED blinking system with the following details.

   A. Input

      i. BTN0 (rst): button for LED off

      ii. BTN1 (en): button for LED on

   B. Output

      i. LD4: LED to be turned on or off

   C. Scenario

      i. When BTN0 is pressed, LD4 is turned off. (All LEDs are turned off by default.)

      ii. When BTN1 is pressed, LD4 is turned on. (Other LEDs must not be turned on.)

Problem 1B (Week 4): Implement an LED counter system

1) Implement an LED counter system with the following details.

   A. Input

      i. BTN0 (rst): button to turn off all RGB LEDs

      ii. BTN1 (en): button to increase the binary number represented by RGB LEDs

      iii. BTN2 (ch_color): button to change the RGB LED color

   B. Output

      i. LD3 (MSB 4), LD2 (MSB 2), LD1 (MSB 1), LD0 (MSB 0): LEDs to represent 4-bit binary integer.

      (e.g. a binary integer 1010_2 → LD3 on, LD2 off, LD1 on, LD0 off)

C. Scenario

i. When BTN0 is pressed, LD3~0 are turned off. (All LEDs are turned off by default.) This represents a binary integer 0000_2.

ii. When BTN1 is pressed for the first time, red LED of LD0 is turned on. (Other LEDs must not be turned on.) This represents a binary integer 0001_2.

iii. When BTN1 is pressed after ii, the binary integer represented by LD3~0 is increased by 1. (Please ignore the overflow. If BTN1 is pressed when the integer is 1111_2, it becomes 0000_2)

iv. When BTN 2 is pressed, the color of LD3~0 is changed to the next color. (Red → Green → Blue → Red → ···) The value of the binary integer is not changed.

III. Backgrounds

SystemVerilog Syntax

Please refer to the lecture material and the SystemVerilog cheet sheet.

IV. Lab Procedure

Problem 1A (Week 3): Implement an LED blinking system

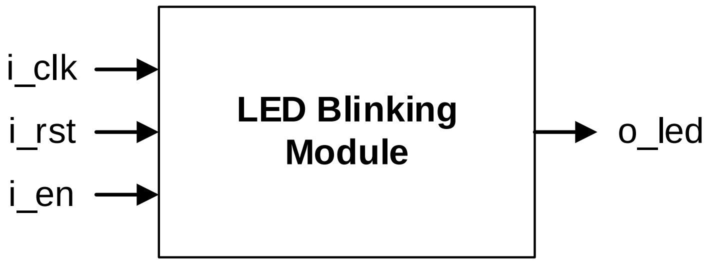

Figure 1. LED Blinking Module

1) Design an LED blinking module (Figure 1) with SystemVerilog.

A. Input

i. i_clk: clock signal

ii. i_rst: user input signal to turn off LED (1: turn off LED, 0: hold)

iii. i_en: user input signal to turn on LED (1: turn on LED, 0: hold)

B. Output

i. o_led: LED display signal (1: LED on, 0: LED off)

2) Map the module ports to FPGA pins.

Please follow up the experiment guide file.

3) Synthesize and implement your module on FPGA

   Please follow up the experiment guide file.

4) Verify your module on FPGA.

Problem 1B (Week 4): Implement an LED counter system

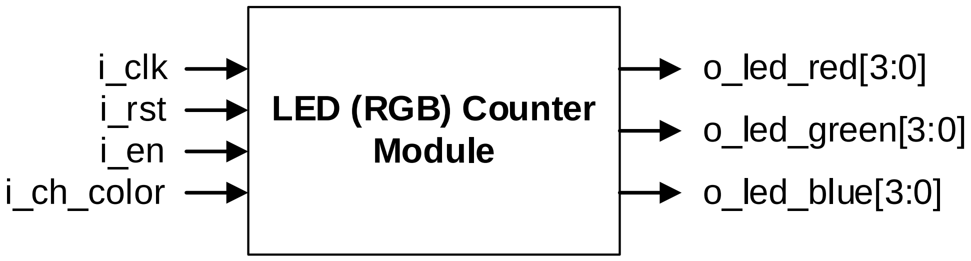

Figure 2. LED Counter Module

1) Design an LED counter module (Figure 2) with SystemVerilog.

   A. Input

      i. i_clk: clock signal

      ii. i_rst: user input signal to turn off all RGB LEDs

          (1: turn off all RGB LEDs, 0: hold)

      iii. i_en: user input signal to increase a counter by 1

          (1: increase counter by 1, 0: hold)

      iv. i_ch_color: user input signal to change LED color

          (1: change LED color, 0: hold)

   B. Output

      i. o_led_red[3:0]: Red LED display signal

          (e.g. o_led_red=1010 → Red LED 3 and 1 are turned on, and red LED 2 and 0 are turned off)

      ii. o_led_green[3:0]: Green LED display signal

      iii. o_led_blue[3:0]: blue LED display signal

2) Map the module ports to FPGA pins.

   Please follow up the experiment guide file.

3) Synthesize and implement your module on FPGA

   Please follow up the experiment guide file.

4) Verify and demo your system on FPGA.