ECEN 749 Lab 1 Report

Tong Lu (UIN 621007982)

September 18, 2019

TEXAS A&M
UNIVERSITY®

Introduction

In Lab 1 I learned the basics of Xilinx Vivado, from creating projects to write program into the FPGA board. After trying the whole process using LED switch project as an example, I also finished a counter module as well as a jackpot game module.

Procedure

1. Create ”Counter” project on Vivado.
2. Finished ”Counter.v” as the module source file, as well as the ”counter.xdc” as the constraint file. The source code is attached in the appendix.
3. Run synthesis for the counter module, generate bitstream with on Vivado, turn on the FPGA borad and write the bitstream to it.
4. Check and make sure the program is working as required on FPGA board.
5. Apply the same steps to Jackpot project.

Result

All three designed was finished and demostrated to TA. The programs are working well and meet all the requirement on lab manual.

Conclusion

In this lab I learned the basics of Vivado and finished two moudles by myself. During this lab I refreshed my knowledge on Verilog and learned how to deug Verilog code. This is a good introduction for me and it will help me a lot in the coming projects.

Question

(a) Based on the reference manual, BTN0 correspond to R18, BTN1 correspond to P18, BTN2 correspond to V18, BTN3 correspond to Y18.
(b) Using edge detection circuit can help synchronize the circuit behavior on each clock rising/falling edge. For example, if we are using level detection on counter, we won’t be able to know how many times it will count on each clock cycle, but if we use edge detection circuit, the behavior of the circuit will be predictable.

Appendix

```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 09/07/2018 10:47:37 AM
// Design Name:
// Module Name: switch
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module switch(SWITCHES , LEDS);
    input [3:0] SWITCHES;
    output [3:0] LEDS;

    assign LEDS[3:0] = SWITCHES[3:0];

endmodule
```

src/switch.v

```
## Switches

set_property PACKAGE_PIN G15 [get_ports {SWITCHES[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[0]}]

set_property PACKAGE_PIN P15 [get_ports {SWITCHES[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[1]}]

set_property PACKAGE_PIN W13 [get_ports {SWITCHES[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[2]}]

set_property PACKAGE_PIN T16 [get_ports {SWITCHES[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[3]}]

##LEDs

set_property PACKAGE_PIN M14 [get_ports {LEDS[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[0]}]

set_property PACKAGE_PIN M15 [get_ports {LEDS[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[1]}]

set_property PACKAGE_PIN G14 [get_ports {LEDS[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[2]}]

set_property PACKAGE_PIN D18 [get_ports {LEDS[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[3]}]
```

src/switch.xdc

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 09/07/2018 10:47:37 AM
// Design Name:
// Module Name: switch
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

// This is the module for module7: counter
module clock_divider(clk, out_clk);
    input clk; // Input clock, which is the fast clock signal
    output reg out_clk; // The output clock, the slower clock signal
    reg [23:0] clk_counter; // The counter is used to count the cycle of the fast clock

    initial begin
        out_clk = 0; // Initalize output clock to zero
    end

    always @(posedge clk) begin
        if (clk_counter == 24'hFFFFFF) begin // If the number of fast clock cycles reach 2^23-1
            out_clk <= ~out_clk; // Toggle the clock output
            clk_counter <= 0; // Clear the counter
        end
        else begin
            clk_counter = clk_counter + 1; // If not, keep counting
        end
   end
endmodule

module counter(BUTTONS, LEDS, CLOCK); // This is the counter module
    input [2:0] BUTTONS; // First define three button unputs: up, down, and reset
    input CLOCK; // Input clock signal
    output [3:0] LEDS; // The output of the counter are four LEDs
    reg [3:0] counter; // The LEDs are controlled using a 4-bit register
    wire my_clock; // Output of the clock divider, which is the slower clock
    clock_divider div(CLOCK, my_clock); // Instancation
    always @ (posedge my_clock) begin
        if (BUTTONS[0]) // If count up button is pressed
            counter <= counter + 1; // Register count up 1
        if (BUTTONS[1]) // If count down button is pressed
            counter <= counter - 1; // Register count down1
        if (BUTTONS[2]) //If reset button is pressed
            counter <= 0; // Set counter register to zero
    end

    assign LEDS[3:0] = counter[3:0]; // Connect LEDS with counter register
endmodule
```

src/counter.v

```
## Switches

set_property PACKAGE_PIN G15 [get_ports {SWITCHES[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[0]}]

set_property PACKAGE_PIN P15 [get_ports {SWITCHES[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[1]}]

set_property PACKAGE_PIN W13 [get_ports {SWITCHES[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[2]}]

set_property PACKAGE_PIN T16 [get_ports {SWITCHES[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[3]}]

##LEDs

set_property PACKAGE_PIN M14 [get_ports {LEDS[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[0]}]

set_property PACKAGE_PIN M15 [get_ports {LEDS[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[1]}]

set_property PACKAGE_PIN G14 [get_ports {LEDS[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[2]}]

set_property PACKAGE_PIN D18 [get_ports {LEDS[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[3]}]

##Buttons
set_property PACKAGE_PIN R18 [get_ports {BUTTONS[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {BUTTONS[0]}]

set_property PACKAGE_PIN P16 [get_ports {BUTTONS[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {BUTTONS[1]}]

set_property PACKAGE_PIN V16 [get_ports {BUTTONS[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {BUTTONS[2]}]

##CLOCK
set_property PACKAGE_PIN L16 [get_ports CLOCK]
set_property IOSTANDARD LVCMOS33 [get_ports CLOCK]
```

src/counter.xdc

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
```

```
//
// Create Date: 09/08/2018 01:31:37 PM
// Design Name:
// Module Name: jackpot
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////

module clock_divider(clk_in, clk_out); // same clock divider from step 6
    input clk_in; // The input clock signal, fast clock
    output reg clk_out; // The output clock signal, the slower clock
    reg [23:0] clk_counter; // The counter is used to count the clock cycles

    initial begin
        clk_out = 0; // Initalize output clock signal to zero
    end

    always @(posedge clk_in) begin // at positive edge of input clock
        if (clk_counter == 24'hFFFFFF) begin // Count clock cycle up to 2^23-1
            clk_out <= ~clk_out; // Toggle output clock signal
            clk_counter <= 0; // Clear the counter register
        end
        else begin
            clk_counter = clk_counter + 1; // If clock cycle haven't reach 2^23-1, keep counting
        end
   end
endmodule

module jackpot(SWITCHES, LEDS, CLOCK, RESET); // jackpot implemented from state machine
    input [3:0] SWITCHES; // Initalize switches input
    input CLOCK; // Initalize clock signal
    input RESET; // Initalize reset button
    output reg [3:0] LEDS; // Initalize output LEDs
    wire my_clk; // This is the slow clock signal
    reg [1:0] state; //
    reg prev_state;
    reg [3:0] led_pattern;
    parameter IDLE = 2'b00, PUSHED = 2'b01, MISSED = 2'b10, WIN = 2'b11;
    clock_divider clk_div(CLOCK, my_clk);

    always @ (posedge my_clk or posedge RESET) begin
        if (RESET) led_pattern <= 4'b0001;
        else led_pattern <= led_pattern[3] ? 4'b0001 : led_pattern << 1;
    end

    always @ (posedge CLOCK) begin // jackpot logic (FSM)
        case(state)
            IDLE: begin
```

```
                LEDS <= led_pattern;
                if (SWITCHES) state <= PUSHED;
            end
            PUSHED: begin
                if (SWITCHES != LEDS)
                    state <= MISSED;
                else state <= WIN;
            end
            MISSED: begin
                if (!SWITCHES) state <= IDLE;
                LEDS <= led_pattern;
            end
            WIN: begin
                LEDS <= 4'b1111;
                if (RESET) state <= SWITCHES? MISSED: IDLE;
            end
        endcase
    end
endmodule
```

src/jackpot.v

```
## Switches

set_property PACKAGE_PIN G15 [get_ports {SWITCHES[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[0]}]

set_property PACKAGE_PIN P15 [get_ports {SWITCHES[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[1]}]

set_property PACKAGE_PIN W13 [get_ports {SWITCHES[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[2]}]

set_property PACKAGE_PIN T16 [get_ports {SWITCHES[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {SWITCHES[3]}]

##LEDs

set_property PACKAGE_PIN M14 [get_ports {LEDS[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[0]}]

set_property PACKAGE_PIN M15 [get_ports {LEDS[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[1]}]

set_property PACKAGE_PIN G14 [get_ports {LEDS[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[2]}]

set_property PACKAGE_PIN D18 [get_ports {LEDS[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {LEDS[3]}]

##CLOCK
set_property PACKAGE_PIN L16 [get_ports CLOCK]
set_property IOSTANDARD LVCMOS33 [get_ports CLOCK]

##Reset button
set_property PACKAGE_PIN R18 [get_ports RESET]
set_property IOSTANDARD LVCMOS33 [get_ports RESET]
```

src/jackpot.xdc