Lab 1: Combinational Logic Design

Chaiwat Ekkaewnumchai

ECE Box #693

Jan 21, 2018

Introduction

The purpose of this lab is to review and practice Verilog on Basys3 board which is completed earlier in ECE2029. In addition, this lab focus mainly on combinational logic. Also, most regular ports on the board are involved in this lab. We can accomplish the final result set in lab manual.

Discussion

In this lab, the main objective is to review what have been accomplished on ECE2029, the previous course.

In the first part, all ports and seven segment display need to be connected to inputs and outputs in verilog and constrain files. Slide switches and buttons are simple to be connected; they can be binded to the ports specified on a board directly. On the other hand, seven segment display has more complicated system. First, only one pattern can be shown at specific blocks; pattern is specified by 7 bits called cathodes, while sections that the pattern will appear are specified by 4 bits called anodes. For patterns, bits are hard-coded so that desired leds are lit properly. Unlike switches and buttons, anode-and-cathode's logic zero indicates those ports are lit. For example, if the anode bits are 0110 from left to right, the most left and most right panels will show a pattern. Other than seven segment display, circuit is straightforward. Buttons are wired to choose which slide switches are selected and which panel is used to displayed. Then, those switches are evaluated to be displayed on seven segment display at the particular panel.

More precisely, `a`, `b`, `c`, and `d` are wired to the slide switches, 4 switches for each variable. `a` is wired to be displayed on the most right (least significant bit) panel of the seven segment display, while `d` is the one to show on the most left (most significant bit). Next, buttons are called `sel0` and `sel1` to represent to the left and right buttons on the board respectively. Then, they are wired together, `sel`, in order to make case to internally wire easier. Next, outputs can be determined by these inputs. `sel` is used to set anodes, which determine where the pattern is shown. Also, it is used to wire a variable to specify the pattern on the seven segment display, `bit`. Then, the variable is straightforwardly wired to particular pattern for all possible values. In addition, the xdc constraint file specifies which variables connect to physical ports on the board. Most of the used ports have labels near components; seven segment display does not, but can be found on the manual of the board. Moreover, the low and high voltages are specified to be 0 and 3.3 voltages.

In the second part, the first part function is instantiated to finally wire to outputs. In the intermediate step, new buttons are wired directly to decide which set of data will be connected to variables in the instantiation. There are two types of sets of data: bits that connect to slide switches as in the first part and bits that connect to constants (for course number and inbox number).

In the implementation, named association is used instead of positional association in instantiation of the previous module in the first part. Name association is easier to identify which wire or reg connects to specific port without confusion. Also, it can be assigned regardless of an original order of a base module differently from positional association. Then, buttons is a selector to wire regs (`a`, `b`, `c`, and `d`) which holds values to show on seven segment display.

Three of four cases are simple; particular data are connected to the regs directly, while the other is slightly complicated. The slide switches are calculated in base 10. Then, the results are added together and separated each digit in base 10 by using divider and modular. The verilog is intelligent enough to translate base-2 value from switches to base-10 value without loss of significant data. Hence, the value can be added directly and correctly.

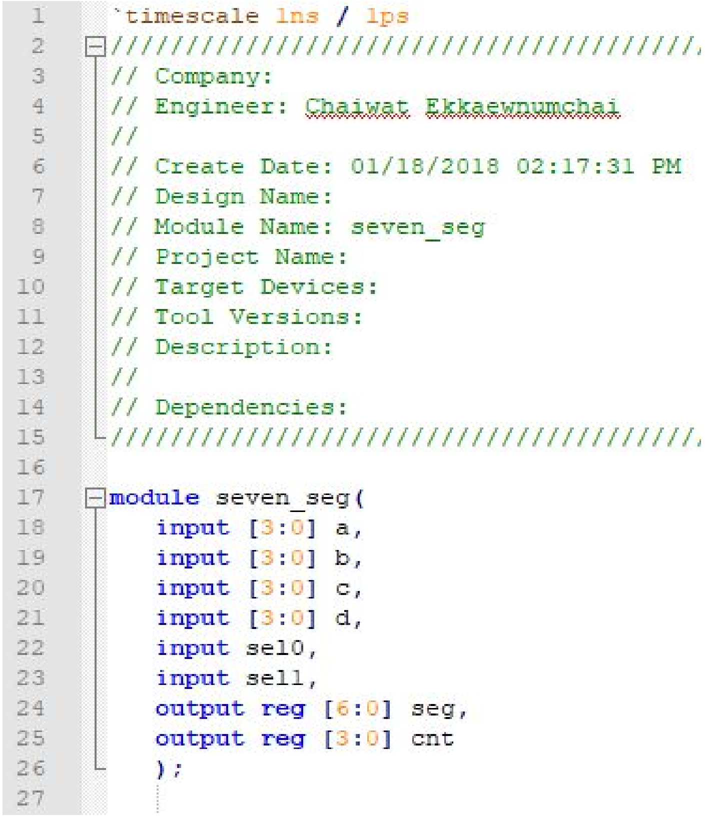
For result, the final code can accomplish the given specification. However, some problems are encouraged while implementation. For example, types of variable, wire and reg, cannot always be used interchangeably. Reg cannot be used in instantiation; it throws an error in implementation process and a warning in synthesis. Another example, wire cannot be used to assign value in always block. Also, reg can be used only in one always blocks.

Conclusion

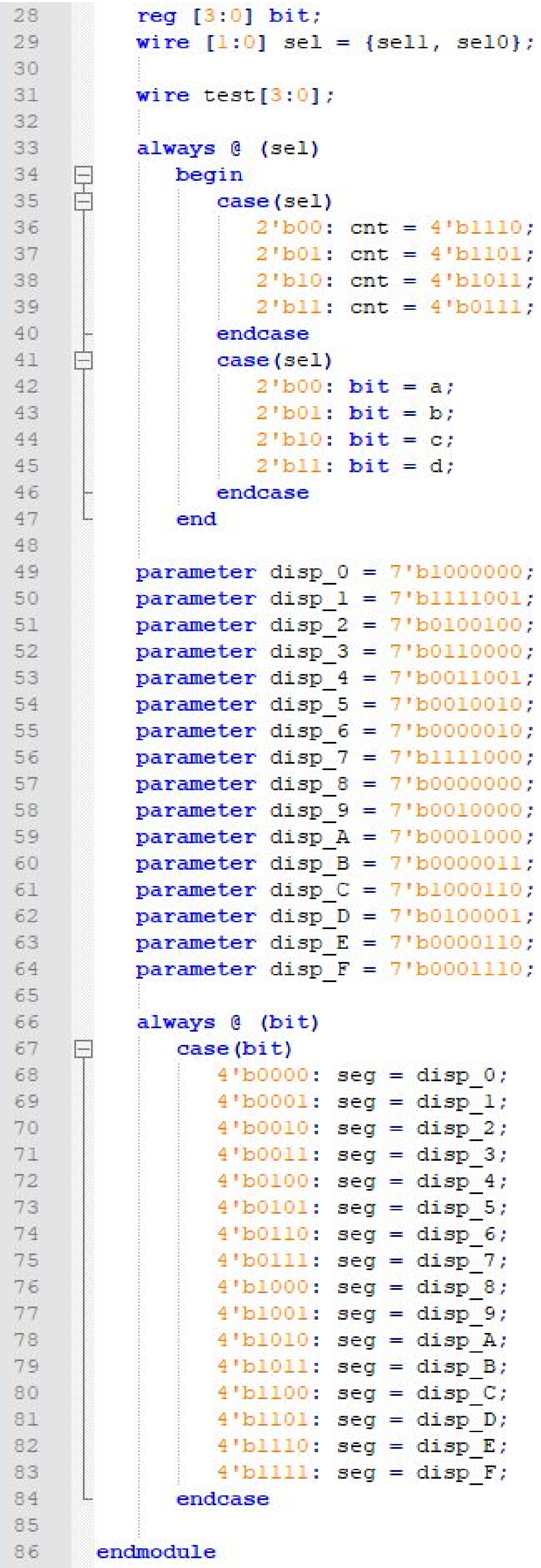
We can accomplish a simple implementation on combinational logic and port usages on the board: buttons, switches, and seven segment display. Also, we also learned how to instantiate a separate module in this lab and how to decide when to use wire and reg to hold variables. In conclusion, the designated board is achieved at the end of the lab; we can show constants, dynamic values given by slide switches, and combination of values on seven segment display at designed panel.

Appendices

seven_seg.v

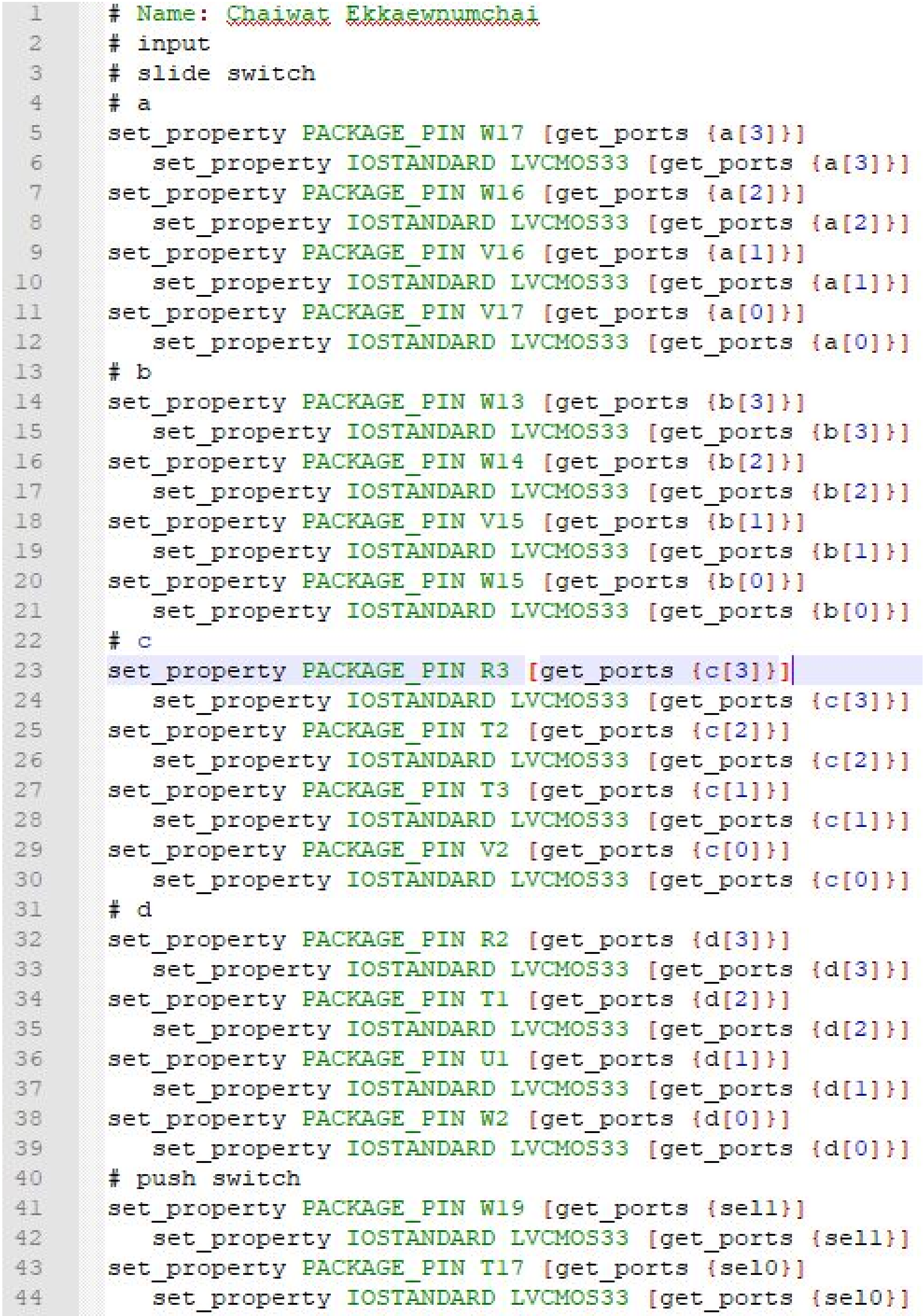


```
`timescale 1ns / 1ps
////////////////////////////////////////.
// Company:
// Engineer: Chaiwat Ekkaewnumchai
//
// Create Date: 01/18/2018 02:17:31 PM
// Design Name:
// Module Name: seven_seg
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
////////////////////////////////////////.

module seven_seg(
    input [3:0] a,
    input [3:0] b,
    input [3:0] c,
    input [3:0] d,
    input sel0,
    input sel1,
    output reg [6:0] seg,
    output reg [3:0] cnt
    );

```

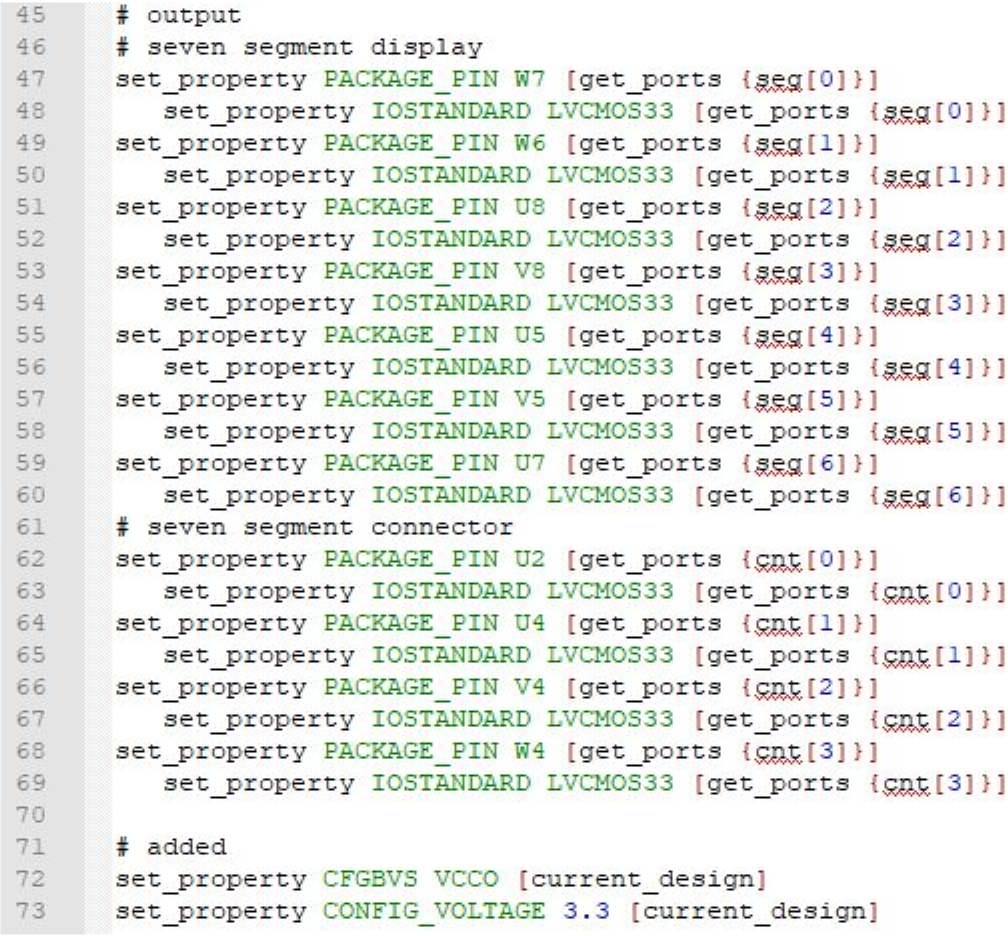


```
    reg [3:0] bit;
    wire [1:0] sel = {sel1, sel0};

    wire test[3:0];

    always @ (sel)
        begin
            case(sel)
                2'b00: cnt = 4'b1110;
                2'b01: cnt = 4'b1101;
                2'b10: cnt = 4'b1011;
                2'b11: cnt = 4'b0111;
            endcase
            case(sel)
                2'b00: bit = a;
                2'b01: bit = b;
                2'b10: bit = c;
                2'b11: bit = d;
            endcase
        end

    parameter disp_0 = 7'b1000000;
    parameter disp_1 = 7'b1111001;
    parameter disp_2 = 7'b0100100;
    parameter disp_3 = 7'b0110000;
    parameter disp_4 = 7'b0011001;
    parameter disp_5 = 7'b0010010;
    parameter disp_6 = 7'b0000010;
    parameter disp_7 = 7'b1111000;
    parameter disp_8 = 7'b0000000;
    parameter disp_9 = 7'b0010000;
    parameter disp_A = 7'b0001000;
    parameter disp_B = 7'b0000011;
    parameter disp_C = 7'b1000110;
    parameter disp_D = 7'b0100001;
    parameter disp_E = 7'b0000110;
    parameter disp_F = 7'b0001110;

    always @ (bit)
        case(bit)
            4'b0000: seg = disp_0;
            4'b0001: seg = disp_1;
            4'b0010: seg = disp_2;
            4'b0011: seg = disp_3;
            4'b0100: seg = disp_4;
            4'b0101: seg = disp_5;
            4'b0110: seg = disp_6;
            4'b0111: seg = disp_7;
            4'b1000: seg = disp_8;
            4'b1001: seg = disp_9;
            4'b1010: seg = disp_A;
            4'b1011: seg = disp_B;
            4'b1100: seg = disp_C;
            4'b1101: seg = disp_D;
            4'b1110: seg = disp_E;
            4'b1111: seg = disp_F;
        endcase

endmodule
```

seven_seg.xdc

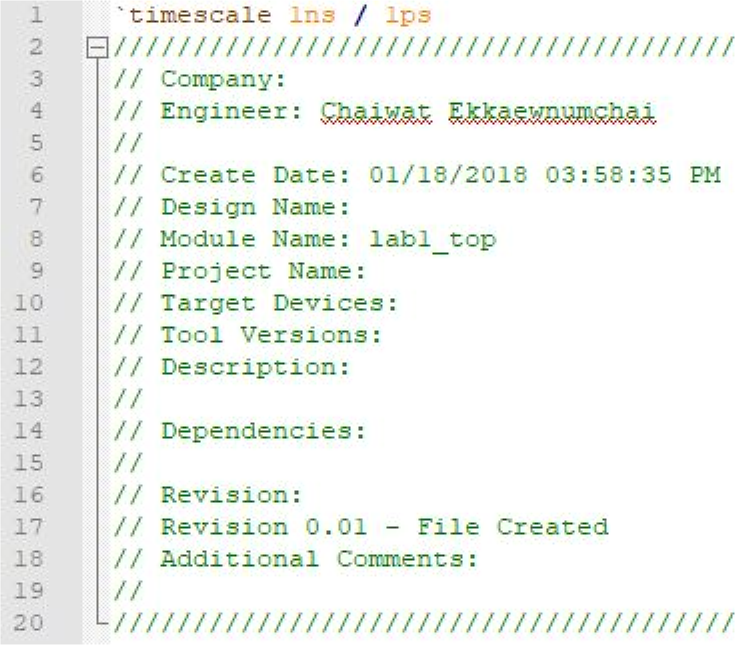


```
# Name: Chaiwat Ekkaewnumchai
# input
# slide switch
# a
set_property PACKAGE_PIN W17 [get_ports {a[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {a[3]}]
set_property PACKAGE_PIN W16 [get_ports {a[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {a[2]}]
set_property PACKAGE_PIN V16 [get_ports {a[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {a[1]}]
set_property PACKAGE_PIN V17 [get_ports {a[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {a[0]}]
# b
set_property PACKAGE_PIN W13 [get_ports {b[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {b[3]}]
set_property PACKAGE_PIN W14 [get_ports {b[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {b[2]}]
set_property PACKAGE_PIN V15 [get_ports {b[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {b[1]}]
set_property PACKAGE_PIN W15 [get_ports {b[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {b[0]}]
# c
set_property PACKAGE_PIN R3 [get_ports {c[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {c[3]}]
set_property PACKAGE_PIN T2 [get_ports {c[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {c[2]}]
set_property PACKAGE_PIN T3 [get_ports {c[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {c[1]}]
set_property PACKAGE_PIN V2 [get_ports {c[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {c[0]}]
# d
set_property PACKAGE_PIN R2 [get_ports {d[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {d[3]}]
set_property PACKAGE_PIN T1 [get_ports {d[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {d[2]}]
set_property PACKAGE_PIN U1 [get_ports {d[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {d[1]}]
set_property PACKAGE_PIN W2 [get_ports {d[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {d[0]}]
# push switch
set_property PACKAGE_PIN W19 [get_ports {sel1}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel1}]
set_property PACKAGE_PIN T17 [get_ports {sel0}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel0}]
```

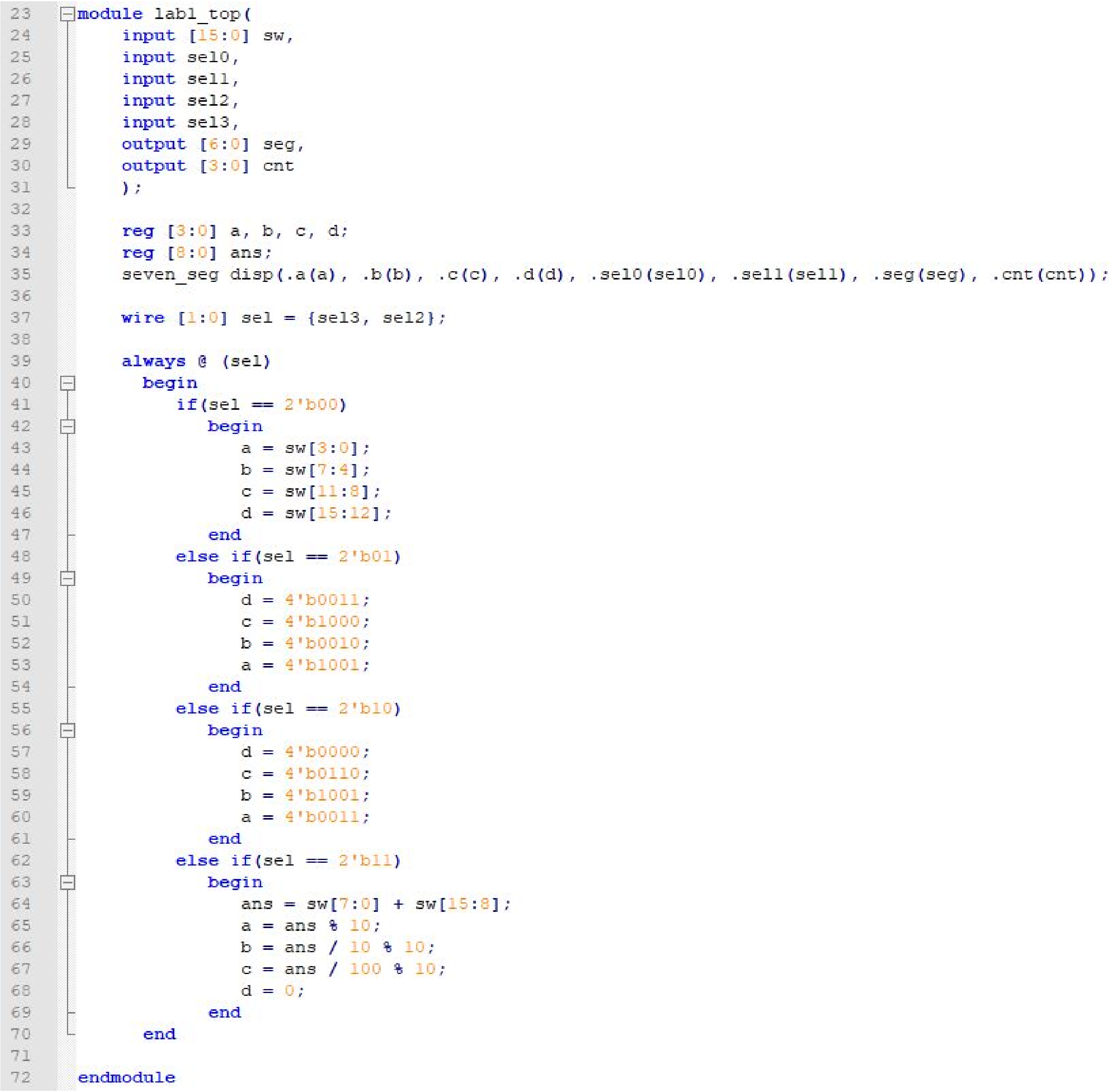


```
# output
# seven segment display
set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]
# seven segment connector
set_property PACKAGE_PIN U2 [get_ports {cnt[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[0]}]
set_property PACKAGE_PIN U4 [get_ports {cnt[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[1]}]
set_property PACKAGE_PIN V4 [get_ports {cnt[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[2]}]
set_property PACKAGE_PIN W4 [get_ports {cnt[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[3]}]

# added
set_property CFGBVS VCCO [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
```

lab1_top.v

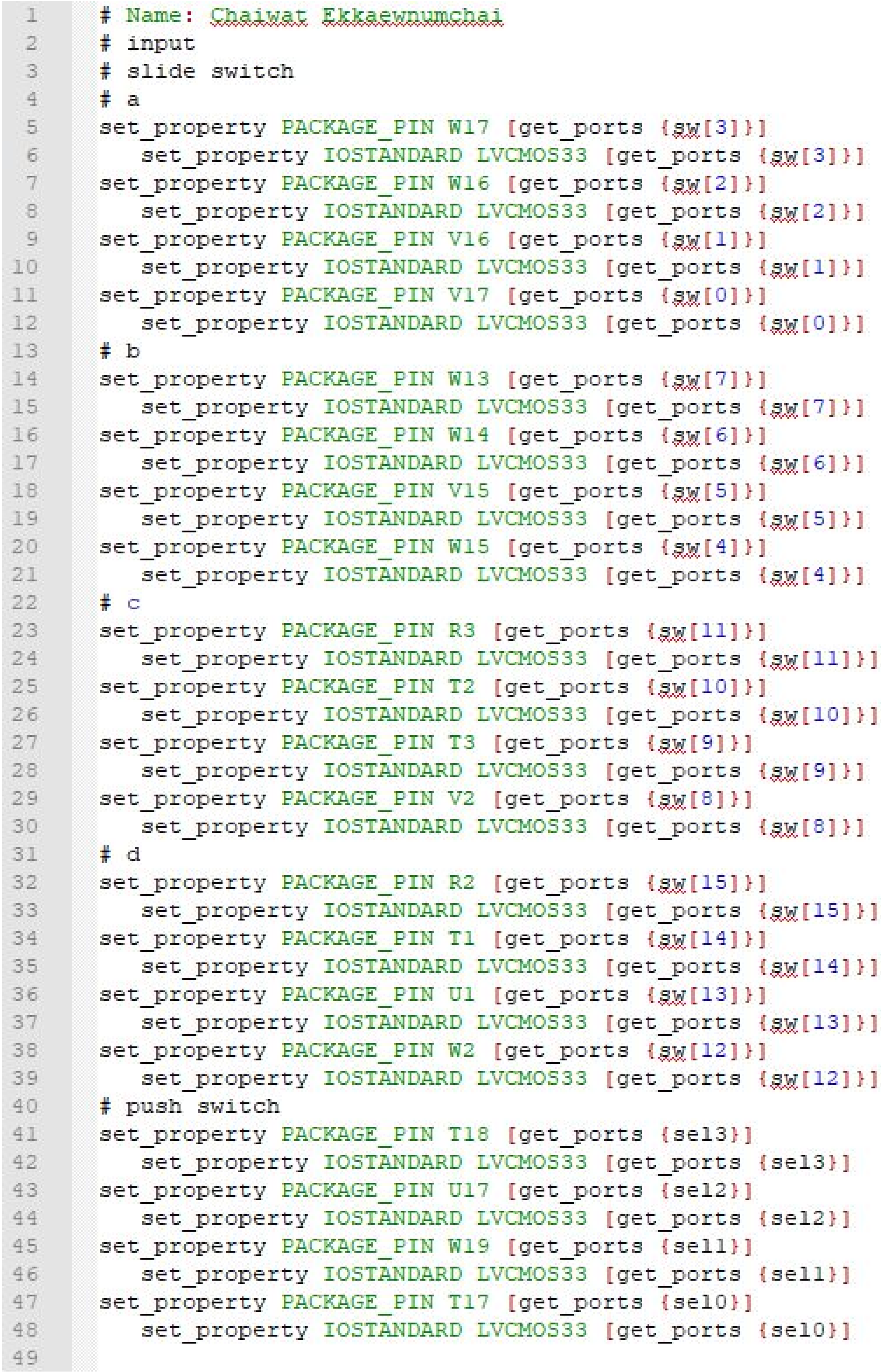


```
`timescale 1ns / 1ps
//////////////////////////////////////////
// Company:
// Engineer: Chaiwat Ekkaewnumchai
//
// Create Date: 01/18/2018 03:58:35 PM
// Design Name:
// Module Name: lab1_top
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////
```

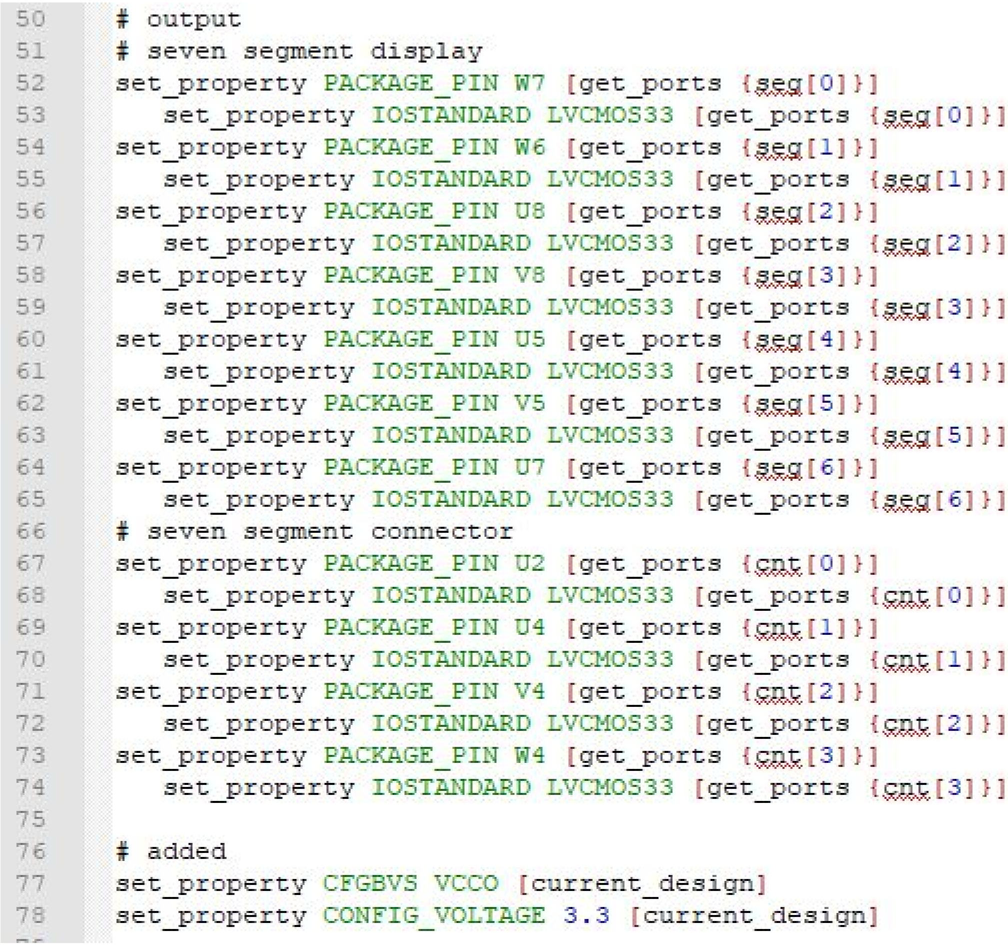


```
module lab1_top(
    input [15:0] sw,
    input sel0,
    input sel1,
    input sel2,
    input sel3,
    output [6:0] seg,
    output [3:0] cnt
    );

    reg [3:0] a, b, c, d;
    reg [8:0] ans;
    seven_seg disp(.a(a), .b(b), .c(c), .d(d), .sel0(sel0), .sel1(sel1), .seg(seg), .cnt(cnt));

    wire [1:0] sel = {sel3, sel2};

    always @ (sel)
      begin
         if(sel == 2'b00)
            begin
               a = sw[3:0];
               b = sw[7:4];
               c = sw[11:8];
               d = sw[15:12];
            end
         else if(sel == 2'b01)
            begin
               d = 4'b0011;
               c = 4'b1000;
               b = 4'b0010;
               a = 4'b1001;
            end
         else if(sel == 2'b10)
            begin
               d = 4'b0000;
               c = 4'b0110;
               b = 4'b1001;
               a = 4'b0011;
            end
         else if(sel == 2'b11)
            begin
               ans = sw[7:0] + sw[15:8];
               a = ans % 10;
               b = ans / 10 % 10;
               c = ans / 100 % 10;
               d = 0;
            end
      end

endmodule
```

lab1_top.xdc



```
# Name: Chaiwat Ekkaewnumchai
# input
# slide switch
# a
set_property PACKAGE_PIN W17 [get_ports {sw[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[3]}]
set_property PACKAGE_PIN W16 [get_ports {sw[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[2]}]
set_property PACKAGE_PIN V16 [get_ports {sw[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[1]}]
set_property PACKAGE_PIN V17 [get_ports {sw[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[0]}]
# b
set_property PACKAGE_PIN W13 [get_ports {sw[7]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[7]}]
set_property PACKAGE_PIN W14 [get_ports {sw[6]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[6]}]
set_property PACKAGE_PIN V15 [get_ports {sw[5]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[5]}]
set_property PACKAGE_PIN W15 [get_ports {sw[4]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[4]}]
# c
set_property PACKAGE_PIN R3 [get_ports {sw[11]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[11]}]
set_property PACKAGE_PIN T2 [get_ports {sw[10]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[10]}]
set_property PACKAGE_PIN T3 [get_ports {sw[9]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[9]}]
set_property PACKAGE_PIN V2 [get_ports {sw[8]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[8]}]
# d
set_property PACKAGE_PIN R2 [get_ports {sw[15]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[15]}]
set_property PACKAGE_PIN T1 [get_ports {sw[14]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[14]}]
set_property PACKAGE_PIN U1 [get_ports {sw[13]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[13]}]
set_property PACKAGE_PIN W2 [get_ports {sw[12]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sw[12]}]
# push switch
set_property PACKAGE_PIN T18 [get_ports {sel3}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel3}]
set_property PACKAGE_PIN U17 [get_ports {sel2}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel2}]
set_property PACKAGE_PIN W19 [get_ports {sel1}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel1}]
set_property PACKAGE_PIN T17 [get_ports {sel0}]
   set_property IOSTANDARD LVCMOS33 [get_ports {sel0}]

```



```
# output
# seven segment display
set_property PACKAGE_PIN W7 [get_ports {seg[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[0]}]
set_property PACKAGE_PIN W6 [get_ports {seg[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[1]}]
set_property PACKAGE_PIN U8 [get_ports {seg[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[2]}]
set_property PACKAGE_PIN V8 [get_ports {seg[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[3]}]
set_property PACKAGE_PIN U5 [get_ports {seg[4]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[4]}]
set_property PACKAGE_PIN V5 [get_ports {seg[5]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[5]}]
set_property PACKAGE_PIN U7 [get_ports {seg[6]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {seg[6]}]
# seven segment connector
set_property PACKAGE_PIN U2 [get_ports {cnt[0]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[0]}]
set_property PACKAGE_PIN U4 [get_ports {cnt[1]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[1]}]
set_property PACKAGE_PIN V4 [get_ports {cnt[2]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[2]}]
set_property PACKAGE_PIN W4 [get_ports {cnt[3]}]
   set_property IOSTANDARD LVCMOS33 [get_ports {cnt[3]}]

# added
set_property CFGBVS VCCO [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
```