Lab 6
Quimpie Tuada
Lab partner: Adrian Insingo

## 1. **7-seg decoder using a PLD**

The following is the code we used for our .PLD:

```
.PLD
ARTNO       PLD01;
NAME        CODECONV;
DATE        10/28/2018;
REV         01;
DESIGNER    Quimpie Tuada;
COMPANY     CSUSB;
DEVICE      G16V8A;
ASSEMBLY    BREADBOARD;
LOCATION    JBH-356;
/* INPUTS */
PIN 1 = W;
PIN 2 = X;
PIN 3 = Y;
PIN 4 = Z;
/* OUTPUTS */
PIN 12 = a;
PIN 13 = b;
PIN 14 = c;
PIN 15 = d;
PIN 16 = e;
PIN 17 = f;
PIN 18 = g;
/* BOOLEAN FUNCTIONS */
a = !X&!Z # !W&Y # X&Y # W&!Z # !W&X&Z # W&!X&!Y;
b = !W&!X # !X&!Z # !X&!Y&!Z # !W&Y&Z # W&!Y&Z;
c = !W&!Y # !W&Z # !Y&Z # !W&X # W&!X;
d = W&!Y # !W&!X&!Z # !X&Y&Z # X&!Y&Z # X&Y&!Z;
e = !X&!Z # Y&!Z # W&Y # W&X;
f = !Y&!Z # X&!Z # W&!X # W&Y # !W&X&!Y;
g = !X&Y # Y&!Z # W&!X # W&Z # !W&X&!Y;
```

**Discussion:**

My partner and I programed our chip using the code above. What we first did was finish the truth table that we had in our notes from lecture. When we completed our truth table, it basically shows which sides should be on (1) or off (0) when we want to display numbers 1-9 and letters A-F. We then used a K-map based on our truth table to obtain the equations for each sides (a-g on our code). We saved our .PDL in a usb and programmed our code in our chip. With our lab assistant’s help, we programmed our chip with our code. After successfully programing our chip. We built the circuit in which we are to display numbers 1-9 and letters A-F. We successfully built the circuit, which displayed the numbers and letters correctly according to their binary codes from our truth table.

2. **Spartan 3 code**

**.v file**

```
module lab6(
        input W,
        input X,
        input Y,
        input Z,
        output a,
        output b,
        output c,
        output d,
        output e,
        output f,
        output g,
        output [3:0]EN
        );
wire a0, b0, c0, d0, e0, f0, g0;
  assign a0 = ~X&~Z | ~W&Y | X&Y | W&~Z | ~W&X&Z | W&~X&~Y;
  assign b0 = ~W&~X | ~X&~Z | ~X&~Y&~Z | ~W&Y&Z | W&~Y&Z;
  assign c0 = ~W&~Y | ~W&Z | ~Y&Z | ~W&X | W&~X;
  assign d0 = W&~Y | ~W&~X&~Z | ~X&Y&Z | X&~Y&Z | X&Y&~Z;
  assign e0 = ~X&~Z | Y&~Z | W&Y | W&X;
  assign f0 = ~Y&~Z | X&~Z | W&~X | W&Y | ~W&X&~Y;
  assign g0 = ~X&Y | Y&~Z | W&~X | W&Z | ~W&X&~Y;

  assign a = ~a0;
  assign b = ~b0;
  assign c = ~c0;
  assign d = ~d0;
  assign e = ~e0;
```

```
   assign f = ~f0;
   assign g = ~g0;

   assign EN = 4'b1110;

endmodule
```

**.UCF file**

```
# digit enable
NET EN[0]  LOC = D14;
NET EN[1]  LOC = G14;
NET EN[2]  LOC = F14;
NET EN[3]  LOC = E13;    #disable

#inputs
NET Z LOC = F12;
NET Y LOC = G12;
NET X LOC = H14;
NET W LOC = H13;

#NET "point"  LOC = P16; # decimal point
NET a  LOC = E14; # segment a
NET b  LOC = G13; # segment b
NET c  LOC = N15; # segment c
NET d  LOC = P15; # segment d
NET e  LOC = R16; # segment e
NET f  LOC = F13; # segment f
NET g  LOC = N16; # segment g
```

**Discussion:**

For this part of the lab we used the same equations as we did when we used the truth table and for the program on our chip. We used the same equations and implemented a code for our Spartan 3 in order to come up with the same results when we used our breadboard with a 7-segment LED display. This time we will be using the 7-seg LED display on our Spartan 3 FPGA. First we implemented a code that uses the equations we produced from the truth table for the Spartan 3. Then we assigned the designated locations for our outputs on the LED. After doing so, we compiled and tested if the LED on the Spartan 3 displays our desired numbers and letters. Using the switches, we were able to display the correct numbers and letters. We showed our lab instructor and we successfully did the lab.

### 3. Verilog decoder

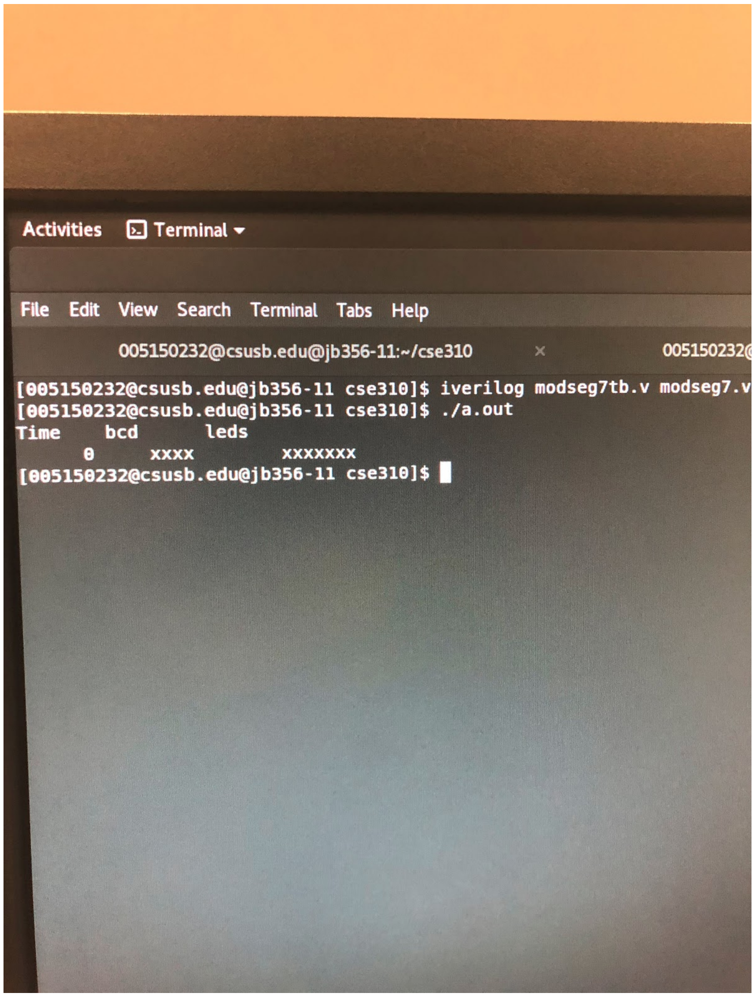
### Module segment 7.v

```
timescale 1ns / 1ps

module seg7(input [3:0] bcd,output reg [1:7] leds);

        always @(bcd)
        case (bcd)              //abcdefg
                0: leds = 7'b1111110;
                1: leds = 7'b0110000;
                2: leds = 7'b1101101;
                3: leds = 7'b1111001;
                4: leds = 7'b0110011;
                5: leds = 7'b1011011;
                6: leds = 7'b1011111;
                7: leds = 7'b1110000;
                8: leds = 7'b1111111;
                9: leds = 7'b1111011;
                10: leds = 7'b1110111;
                11: leds = 7'b0011111;
                12: leds = 7'b1001110;
                13: leds = 7'b0111101;
                14: leds = 7'b1001111;
                15: leds = 7'b1000111;
       endcase

initial begin

#160 $finish;
end
endmodule
```

### Module 7 test bench

```
module seg7tb;
//input
reg [3:0] bcd;

//output
wire [1:7] leds;
```

```
//Instatiate the Unit Under Test (UUT)
seg7 uut(.bcd(bcd), .leds(leds));

//set up monitoring
initial begin
        $display("Time    bcd      leds");
        $monitor("%3g     %b        %b", $time,bcd,leds);

initial begin
        bcd=4'b0000; leds=7'b0000000;
        #160 bcd=4'b0100; leds = 7'b1111110;

#160 $finish;
end
endmodule
```



```
[005150232@csusb.edu@jb356-11 cse310]$ iverilog modseg7tb.v modseg7.v
[005150232@csusb.edu@jb356-11 cse310]$ ./a.out
Time    bcd      leds
      0     xxxx         xxxxxxx
[005150232@csusb.edu@jb356-11 cse310]$ 
```

**Discussion:**

Another part of the lab is to implement a code on verilog that displays the same results as the previous code. Unfortunately, my partner and I weren't able to implement a code that displays the desired output. We created a module for the segment 7 decoder and attempted a stimulus based on previous labs. The test bench we created weren't able to run our module for segment 7 decoder. We aren't sure how to implement the code correctly and we were running out of time. Because of this, I don't think I deserve a perfect score but we still attempted to code. Therefore, I think we deserve a score of 18, overall. We were able to demonstrate the Spartan 3 and breadboard circuit flawlessly, but had coding issues when it came to the verilog code. I think we performed the majority of the lab well.