Dylan Thornburg & Alan Rieger
ELEN 21L Lab 4
5/2/23

Introduction

In this lab, we made a full adder using 2:1 multiplexers. In the prelab leading up to the lab, we made our equations for the inputs we would use in the circuit. Our circuit adds our inputs together and then the outputs (which are in binary) tell the seven segment display what to display.

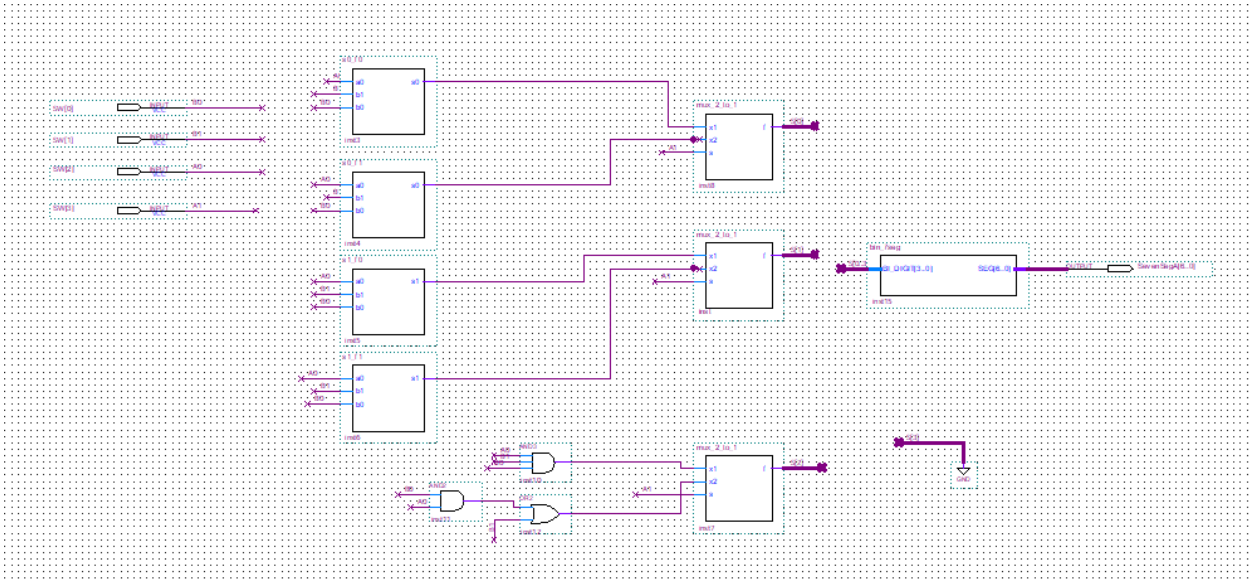
Procedures

First, we wrote verilog code in Quartus for each logic function in S0 and S1 along with a multiplexer and a seven segment display. Next, we converted those into symbols which we used in our circuit schematic. Next, we created the circuit with three inputs and one of them (A1) being the selection signal into the 2:1 multiplexers. Those outputs were connected to the seven segment display symbol along with LEDs that gave the output in binary. Lastly, we complied and uploaded the schematic onto the FPGA and tested that it worked in adding the binary numbers both on the seven segment display and the LEDs.

Conclusion

This lab was rather challenging. The prelab was longer than usual and we had to discuss answers with each other during the lab to make sure we were on the same page. We had to make numerous changes along the way due to errors and we also detected problems in our circuit when we uploaded to the FPGA (nothing displayed due to wire naming and wire type being incorrect). Quartus even crashed on us and kept moving/losing files but we persisted and got it to work finally.

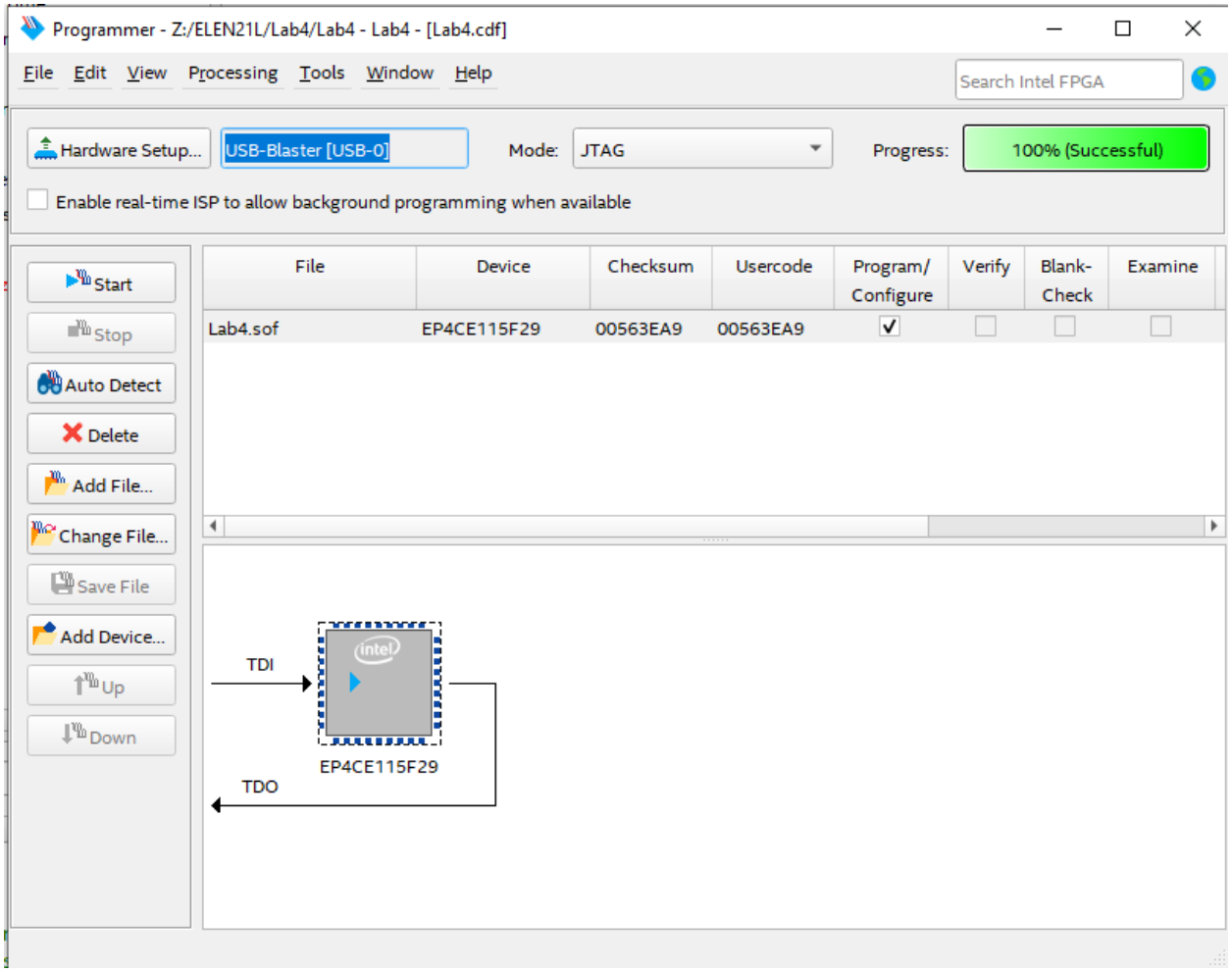
Circuit Schematic (Figure 1):



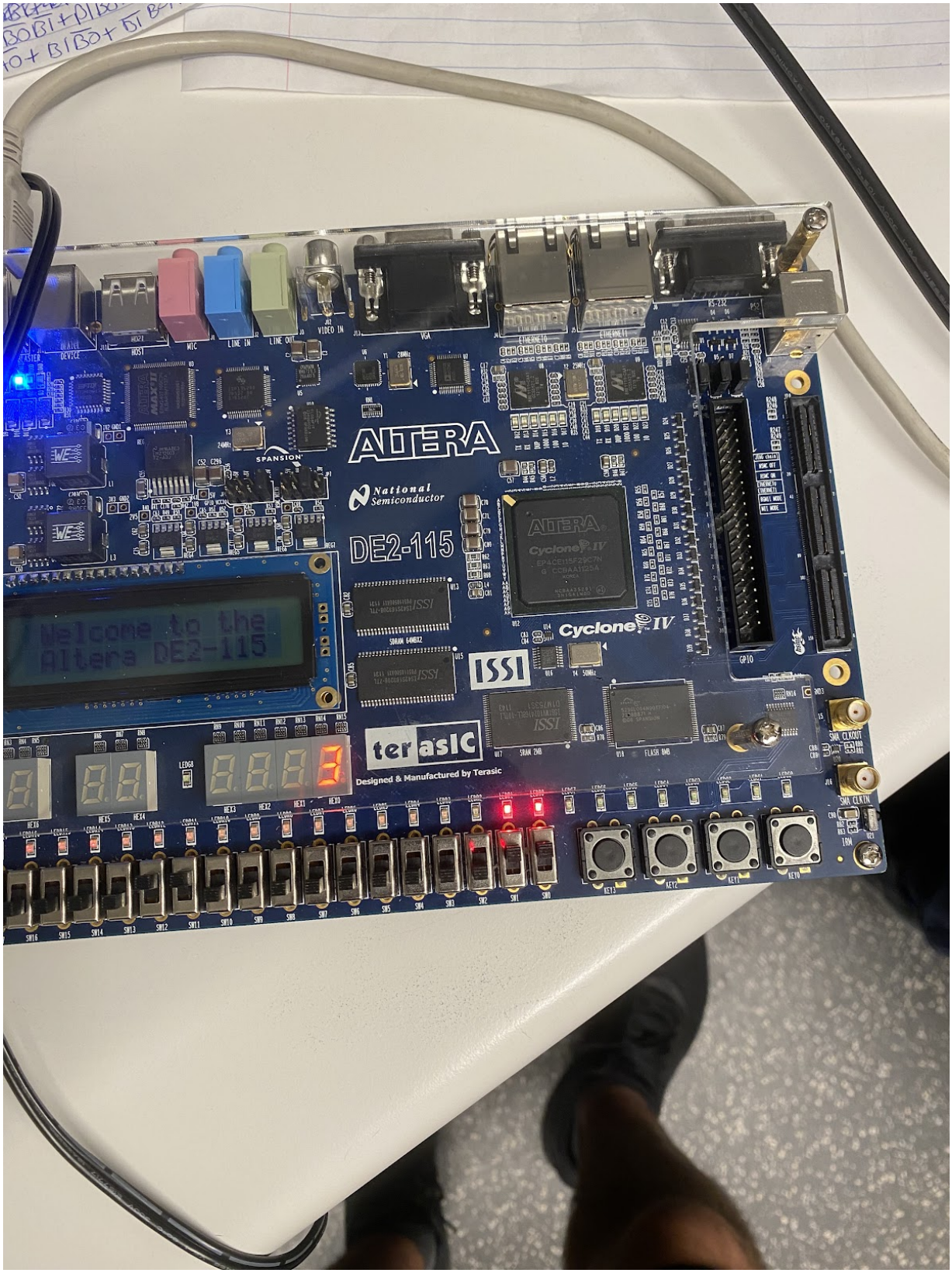
Verilog Code (Figure 2):

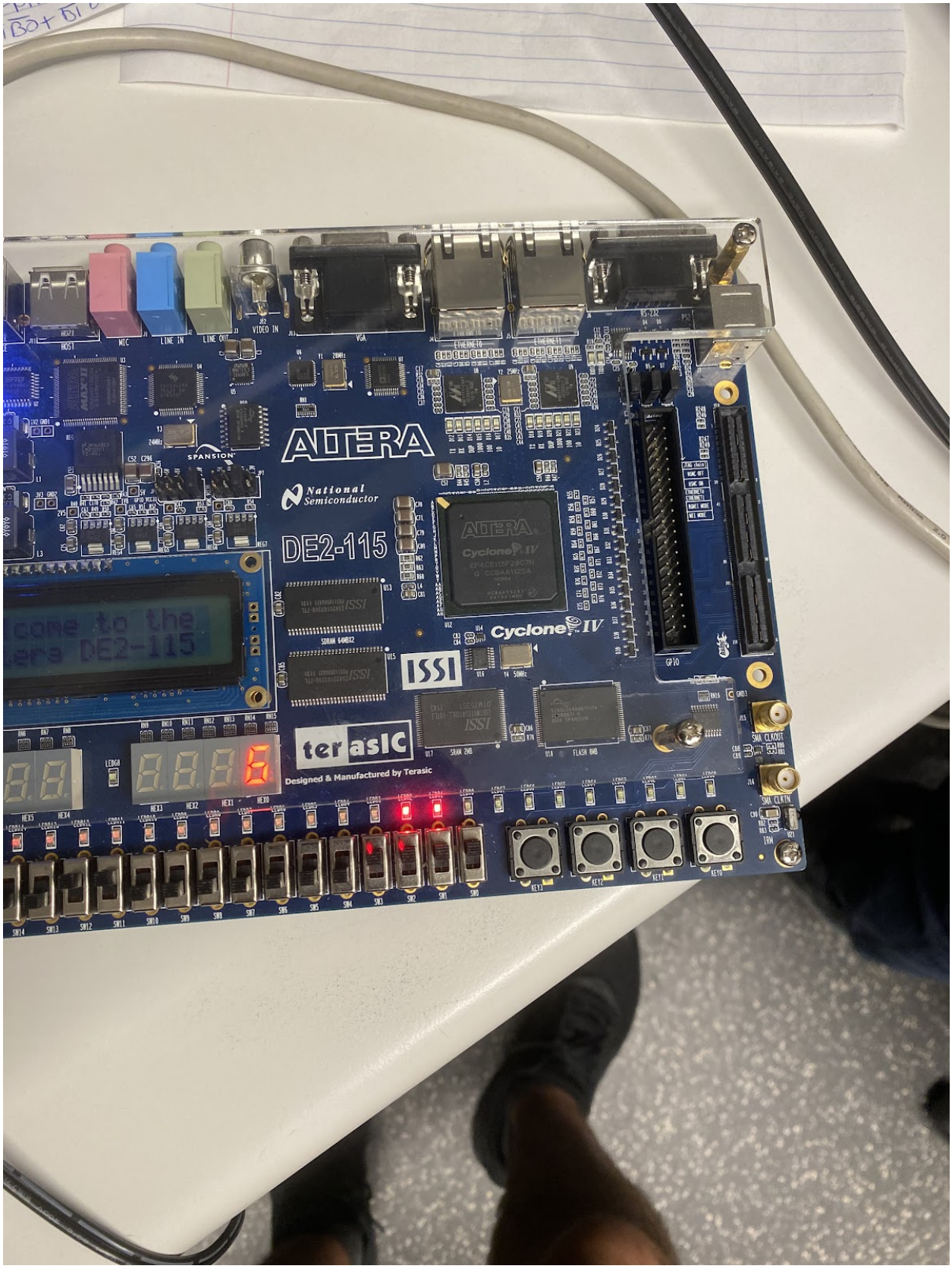
```
module bin_7seg(BI_DIGIT,SEG);
    input [3:0] BI_DIGIT;
    output [6:0] SEG;
    reg [6:0] SEG;

    // seg = {g,f,e,d,c,b,a};
    // ---a----
    // |      |
    // f      b
    // |      |
    // ---g----
    // |      |
    // e      c
    // |      |
    // ---d----

    always @(BI_DIGIT)
        case (BI_DIGIT)
            4'h0: SEG = ~7'b0111111;
            4'h1: SEG = ~7'b0000110;
            4'h2: SEG = ~7'b1011011;
            4'h3: SEG = ~7'b1001111;
            4'h4: SEG = ~7'b1100110;
            4'h5: SEG = ~7'b1101101;
            4'h6: SEG = ~7'b1111101;
            4'h7: SEG = ~7'b0000111;
            4'h8: SEG = ~7'b1111111;
            4'h9: SEG = ~7'b1100111;
            4'ha: SEG = ~7'b1110111;
            4'hb: SEG = ~7'b1111100;
            4'hc: SEG = ~7'b1011000;
            4'hd: SEG = ~7'b1011110;
            4'he: SEG = ~7'b1111001;
            4'hf: SEG = ~7'b1110001;
        endcase
endmodule

module mux_2_to_1 (x1, x2, s, f);
   input x1, x2, s;
   output f;
   assign f=(~s&x1)|(s&x2);


endmodule


module s0_f0(a0, b1, b0, s0);
  input a0, b1, b0;
  output s0;
  assign s0=b0^a0;

endmodule

module s0_f1(a0, b1, b0, s0);
  input a0, b1, b0;
  output s0;
  assign s0=b0^a0;

endmodule


module s1_f0(a0, b1, b0, s1);
  input a0, b1, b0;
  output s1;
  assign s1=(~a0&b1)|(b1&~b0)|(a0&~b1&b0);

endmodule

module s1_f1(a0, b1, b0, s1);
  input a0, b1, b0;
  output s1;
  assign s1=(~a0&~b1)|(~b1&~b0)|(a0&b1&b0);

endmodule
```

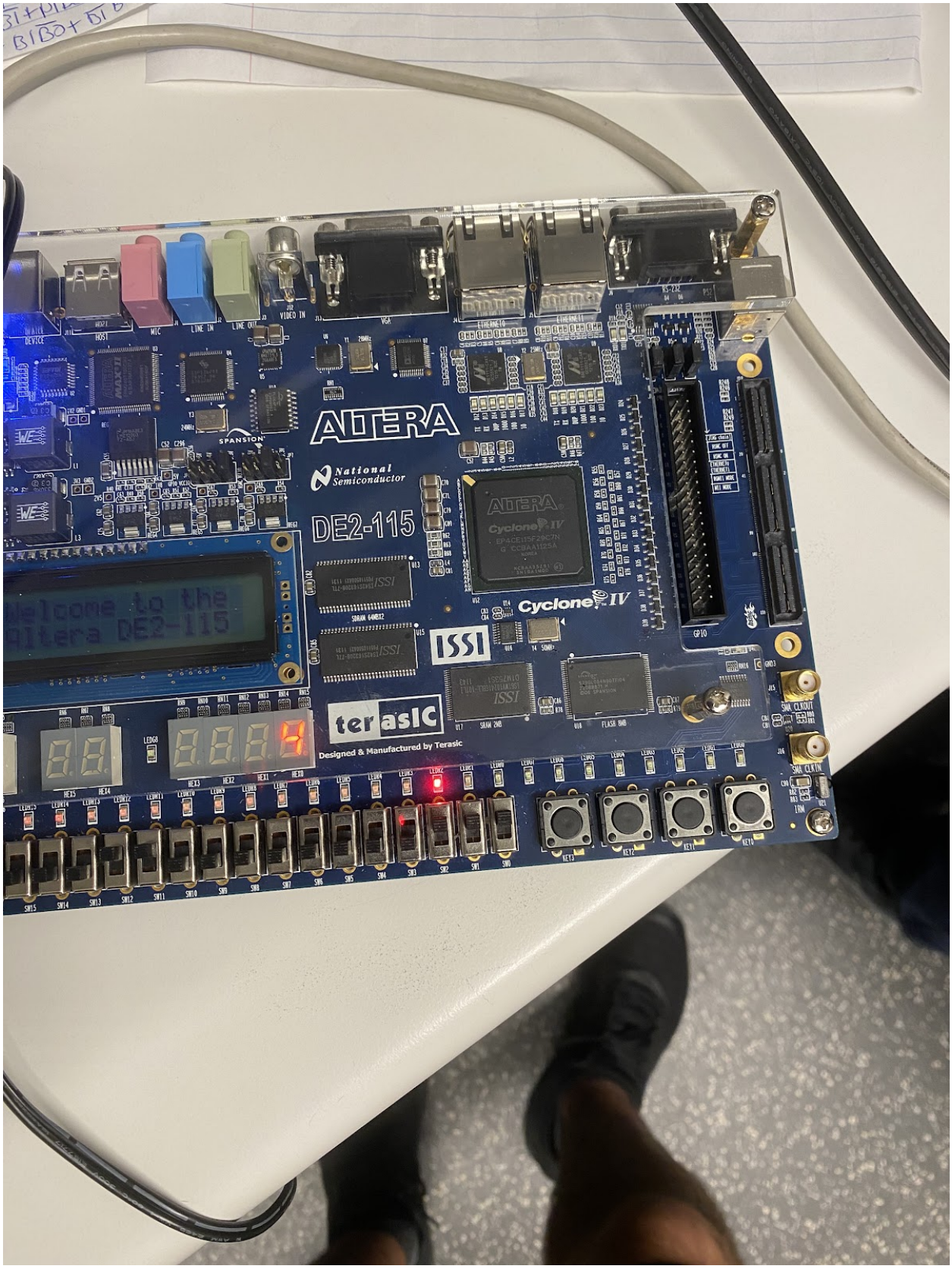
Successful Upload to FPGA (Figure 3):



FPGA Working (Figures 4-6):




ALTERA
DE2-115
National Semiconductor
SPANSION
ISSI
Cyclone IV
terasIC
Designed & Manufactured by Terasic
MIC
LINE IN
LINE OUT
VIDEO IN
VGA
HOST
GPIO
SMA CLKOUT
SMA CLKIN


ALTERA
National Semiconductor
DE2-115
Welcome to the
Altera DE2-115
Cyclone IV
ISSI
terasIC
Designed & Manufactured by Terasic

Logic Functions (Figure 7):

$$S0F0 = A0 \oplus B0$$

$$S0F1 = A0 \oplus B0$$

$$S1F0 = \overline{A0} \cdot B1 + B1 \cdot \overline{B0} + A0 \cdot \overline{B1} \cdot B0$$

$$S1F1 = \overline{A0} \cdot \overline{B1} + \overline{B1} \cdot \overline{B0} + A0 \cdot B1 \cdot B0$$

$$S2F0 = A0 \cdot B1 \cdot B0$$

$$S2F1 = B1 + A0 \cdot B0$$