

EEC 180 Lab 2

Bradley Manzo

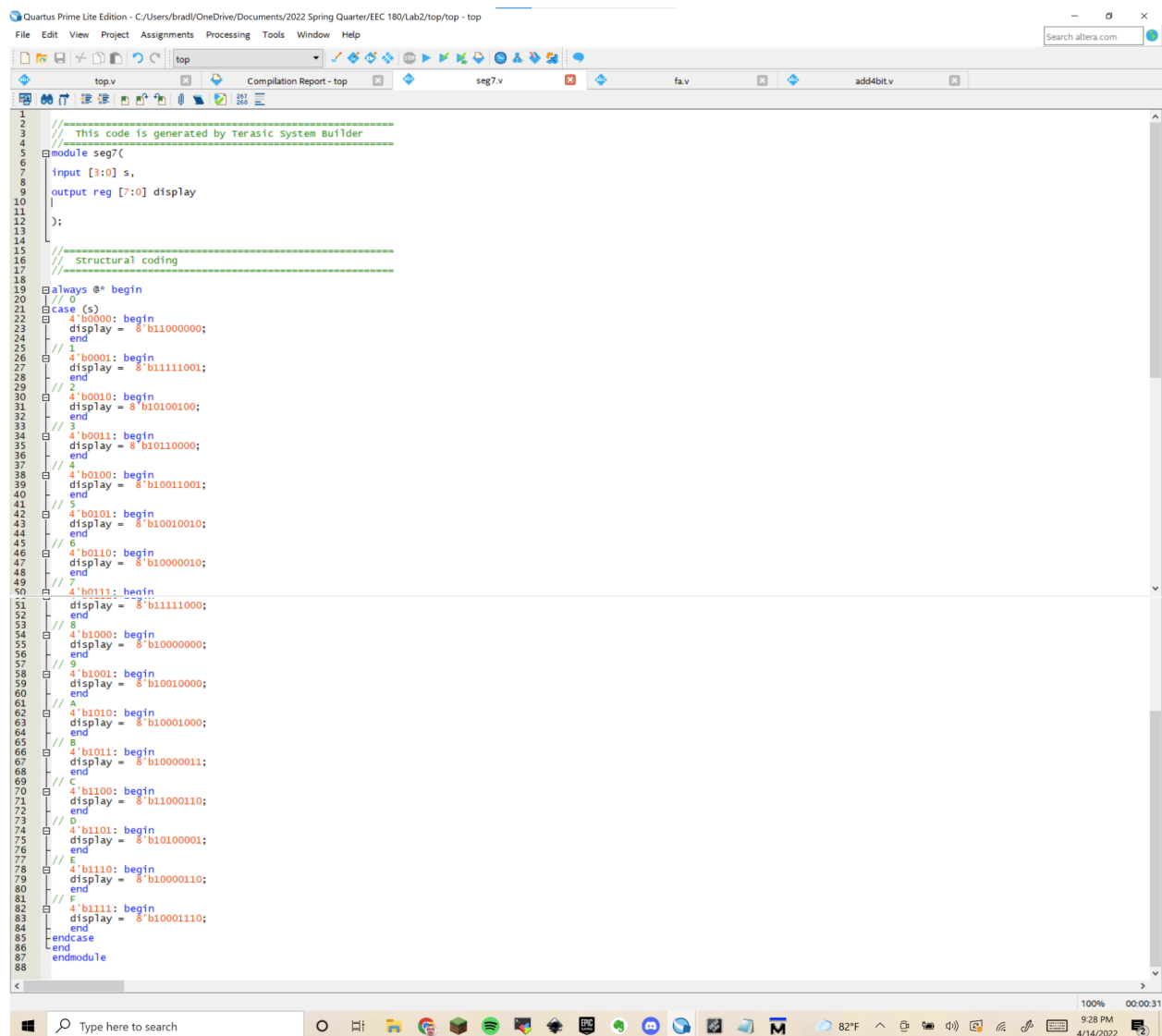
916953788

Objective and Introduction:

The objective of this lab is to synthesize a top-level design including two four-bit adders. This will be achieved by designing lower level modules such as full adders, four-bit adders, and seven-segment display drivers and instantiating the proper amount of modules throughout the hierarchy.

Experiment:

Part II. A Hexadecimal display driver



```
1 // This code is generated by Terasic System Builder
2
3
4
5 module seg7(
6     input [3:0] s,
7     output reg [7:0] display
8 );
9
10
11 // structural coding
12
13
14
15
16
17
18 always @* begin
19     case (s)
20     // 0
21     4'b0000: begin
22         display = 8'b11000000;
23     end
24     // 1
25     4'b0001: begin
26         display = 8'b11111001;
27     end
28     // 2
29     4'b0010: begin
30         display = 8'b10100100;
31     end
32     // 3
33     4'b0011: begin
34         display = 8'b10110000;
35     end
36     // 4
37     4'b0100: begin
38         display = 8'b10011001;
39     end
40     // 5
41     4'b0101: begin
42         display = 8'b10010010;
43     end
44     // 6
45     4'b0110: begin
46         display = 8'b10000010;
47     end
48     // 7
49     4'b0111: begin
50         display = 8'b11111000;
51     end
52     // 8
53     4'b1000: begin
54         display = 8'b10000000;
55     end
56     // 9
57     4'b1001: begin
58         display = 8'b10010000;
59     end
60     // A
61     4'b1010: begin
62         display = 8'b10001000;
63     end
64     // B
65     4'b1011: begin
66         display = 8'b10000011;
67     end
68     // C
69     4'b1100: begin
70         display = 8'b11000110;
71     end
72     // D
73     4'b1101: begin
74         display = 8'b10100001;
75     end
76     // E
77     4'b1110: begin
78         display = 8'b10000110;
79     end
80     // F
81     4'b1111: begin
82         display = 8'b10001110;
83     end
84     endcase
85 end
86 endmodule
87
88
```

Figure 1: seg7.v

Part III. A Combinational 4-bit Adder

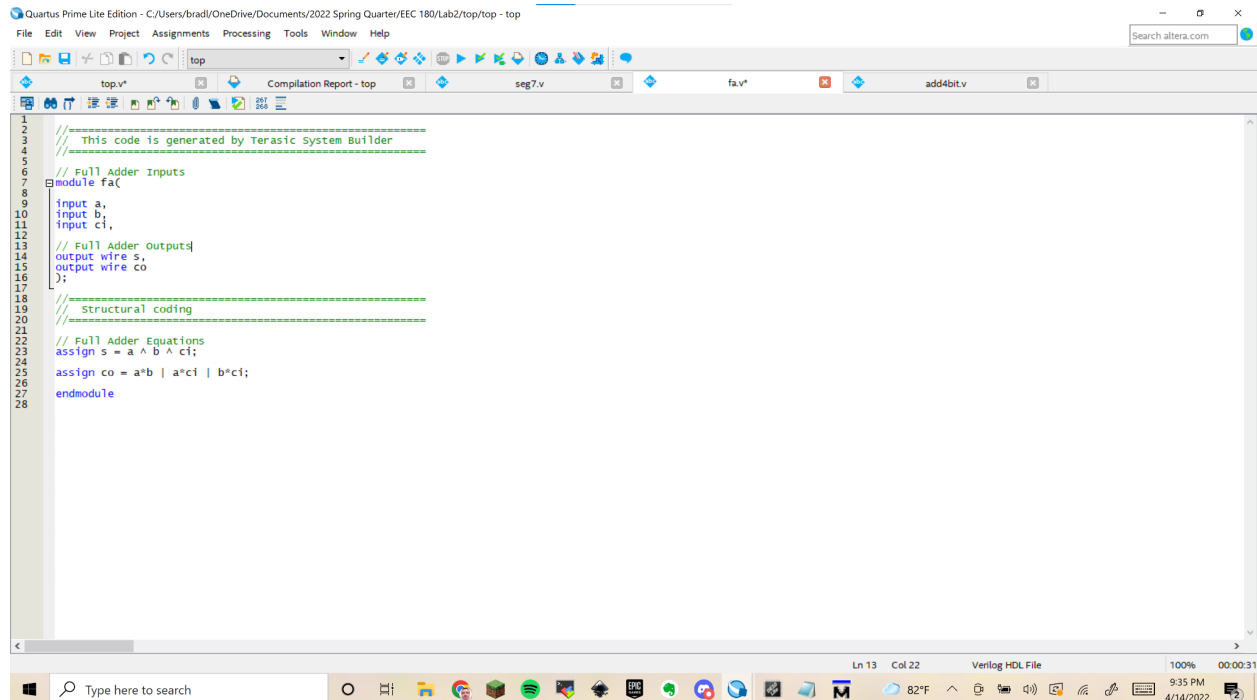


Figure 2: fa.v

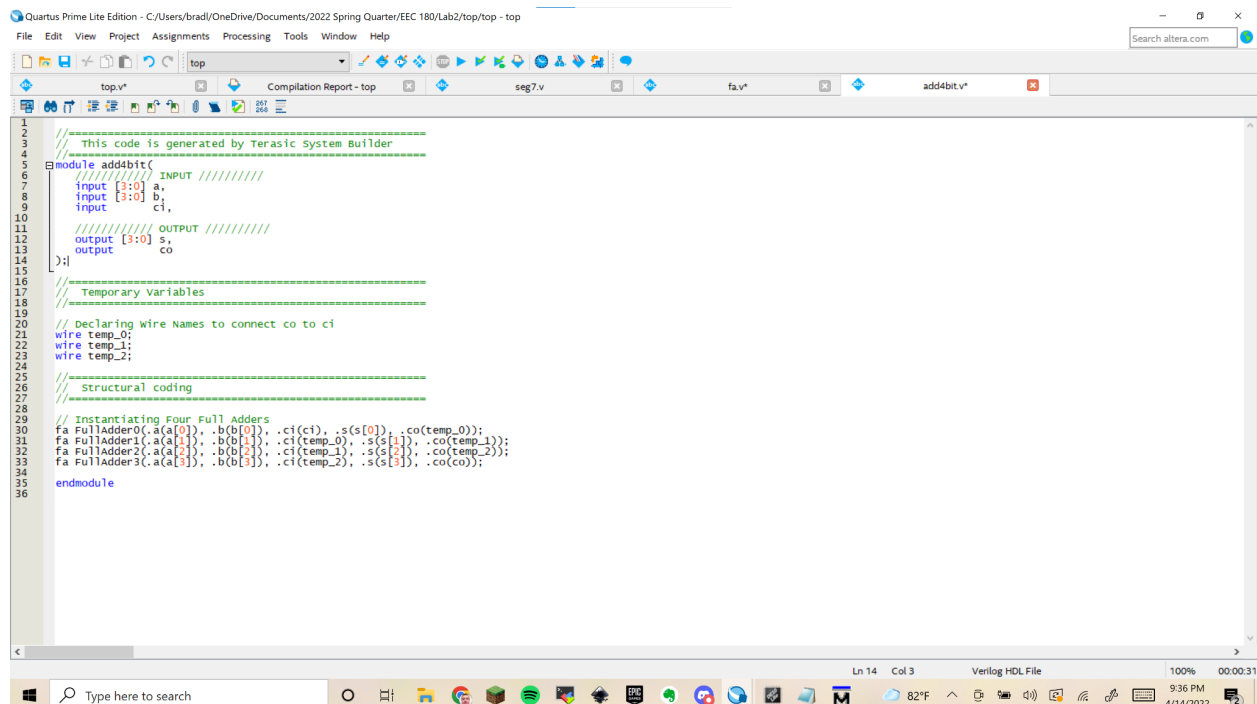
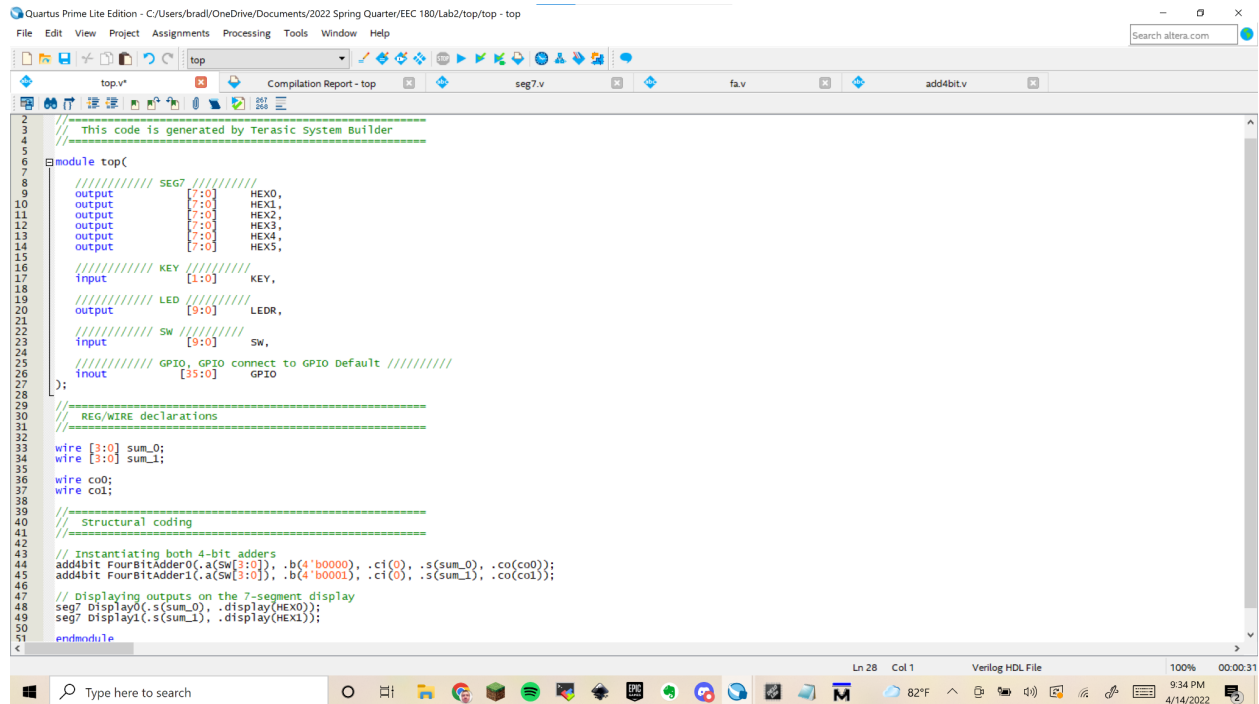


Figure 3: add4bit.v

Part IV. Top-level design



```
// This code is generated by Terasic System Builder
module top(
    output [7:0] SEG7,
    output [7:0] HEX0,
    output [7:0] HEX1,
    output [7:0] HEX2,
    output [7:0] HEX3,
    output [7:0] HEX4,
    output [7:0] HEX5,
    input [1:0] KEY,
    output [9:0] LEDR,
    input [9:0] SW,
    input [35:0] GPIO,
    output [35:0] GPIO connect to GPIO Default
);
// REG/WIRE declarations
wire [3:0] sum_0;
wire [3:0] sum_1;
wire co0;
wire col;
// Structural coding
// Instantiating both 4-bit adders
add4bit FourBitAdder0(.a{sw[3:0]}, .b{4'b0000}, .ci{0}, .s(sum_0), .co(co0));
add4bit FourBitAdder1(.a{sw[3:0]}, .b{4'b0001}, .ci{0}, .s(sum_1), .co(col));
// Displaying outputs on the 7-segment display
seg7 Display0(.s(sum_0), .display(HEX0));
seg7 Display1(.s(sum_1), .display(HEX1));
endmodule
```

Figure 4: top.v

Part V. Implementation and Verification on the DE10 - Lite

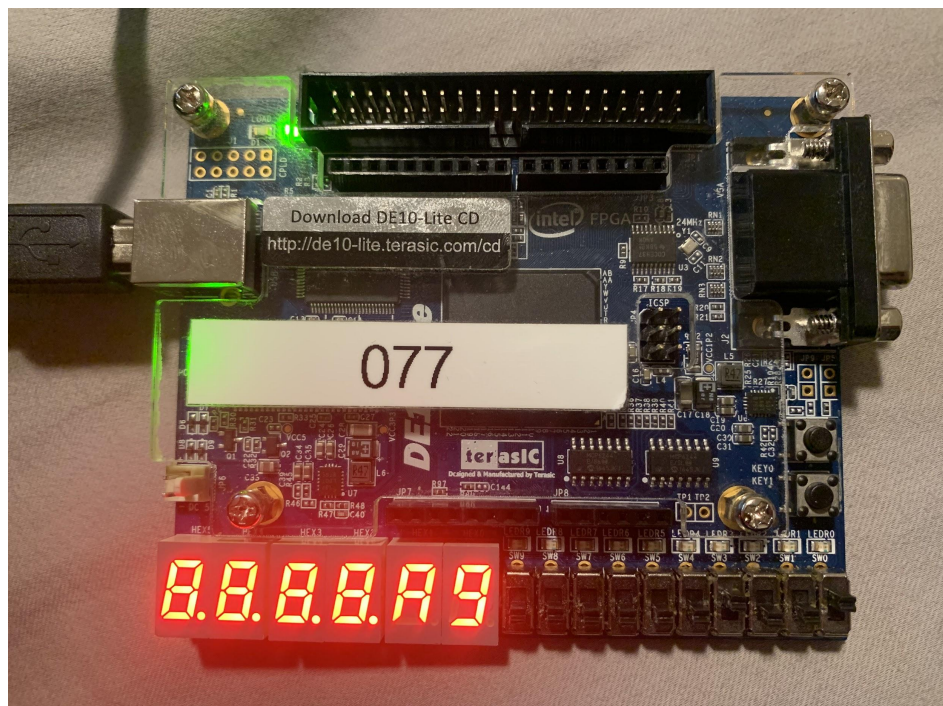


Figure 5: Implementation on DE-10 Lite Board

Part VI. Submitted Work

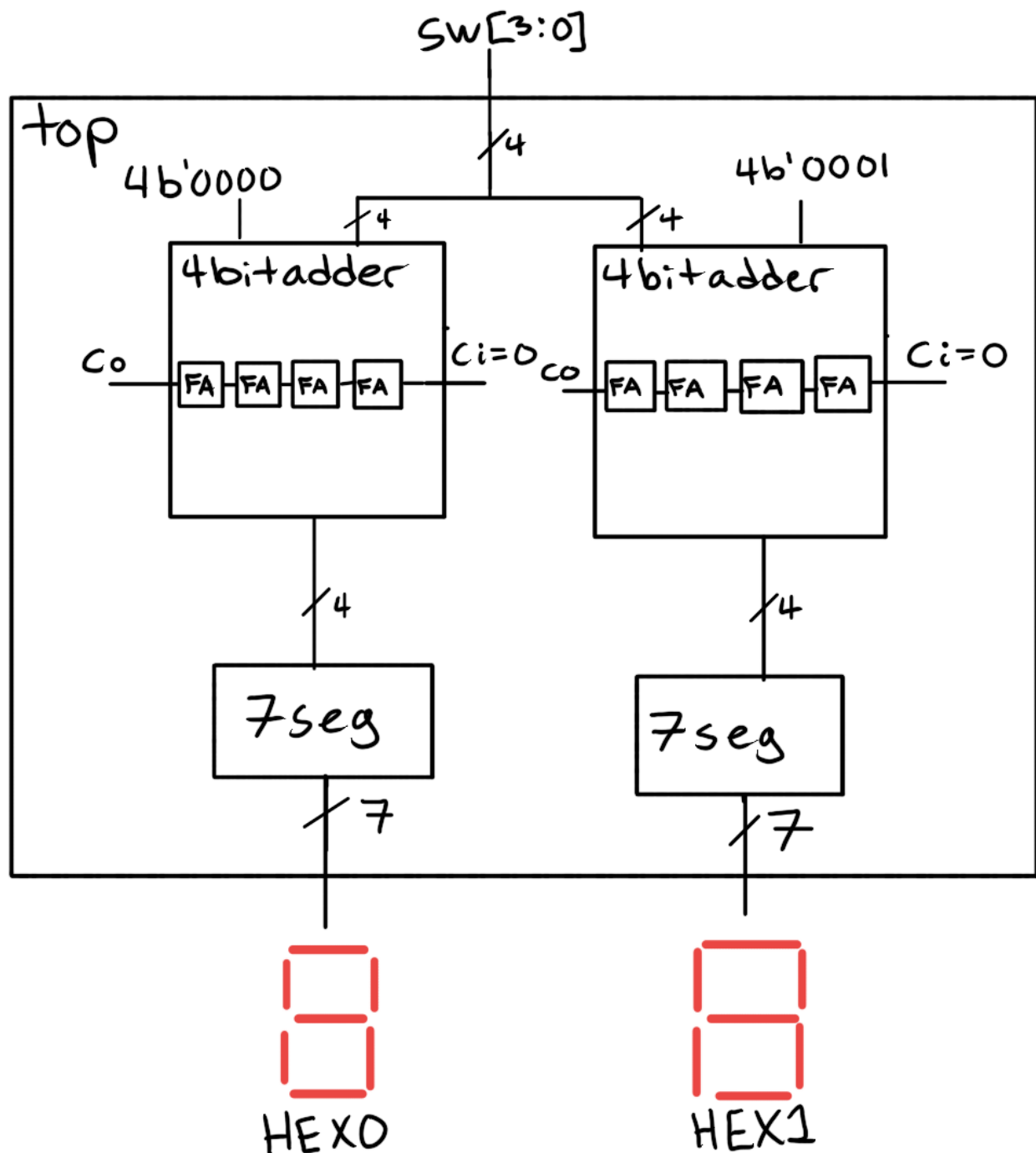


Figure 6: Circuit Diagram

Conclusion: After completing this lab I feel I have gained a greater understanding of how top level designs are created within quartus and programmed in Verilog. I hope to put this knowledge of instantiating and defining modules to use in future labs.