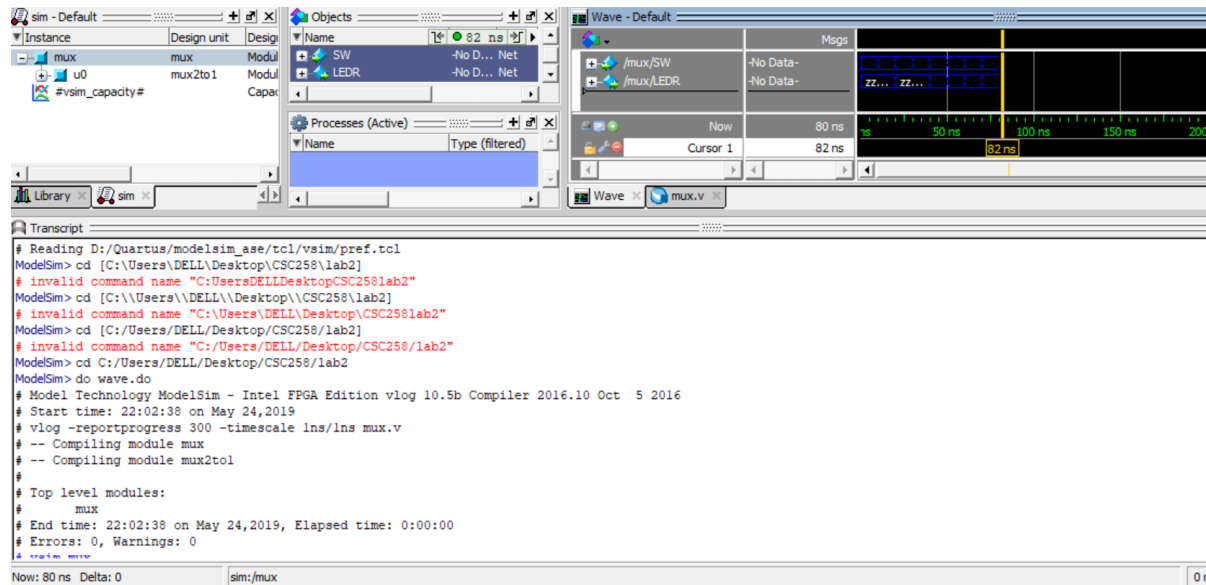


Pre-Lab for Lab 2

Xiaoyu Zhou, 1004081147

Part I

1.

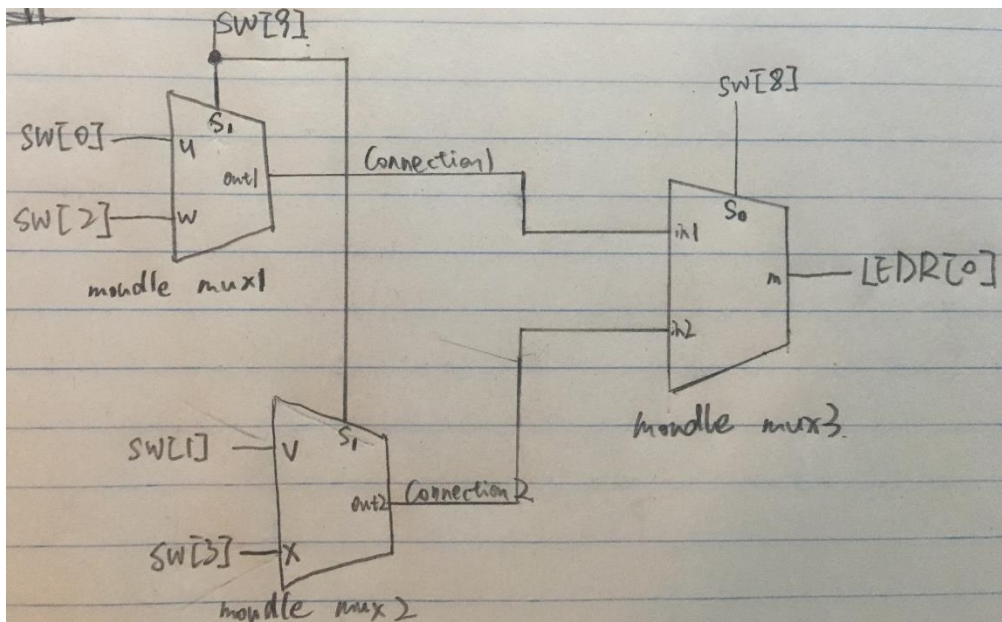


Part II

1. If the truth table in Table 2 was given in full, how many rows would it have?

There would be 2^6 rows in the truth table, which is 64 rows.

2. Draw a schematic.



3. Create a new Quartus Prime project for your circuit and write the Verilog code.

```
module mux(LED[0], SW);
    input [9:0] SW;
    output [9:0] LED[0];

    mux1 u1(
        .u(SW[0]),
        .w(SW[2]),
        .s1(SW[9]),
        .out1(Connection1)
    );

    mux2 u2(
        .v(SW[1]),
        .x(SW[3]),
        .s1(SW[9]),
        .out2(Connection2)
    );

    mux3 u3(
        .in1(Connection1),
        .in2(Connection2),
        .s0(SW[8]),
        .m(LED[0])
    );
endmodule

module mux1(u, w, s1, out1);
    input u; //selected when s is 0
    input w; //selected when s is 1
    input s1; //select signal
    output out1; //output

    assign out1 = s1 & w | ~s1 & u;
endmodule

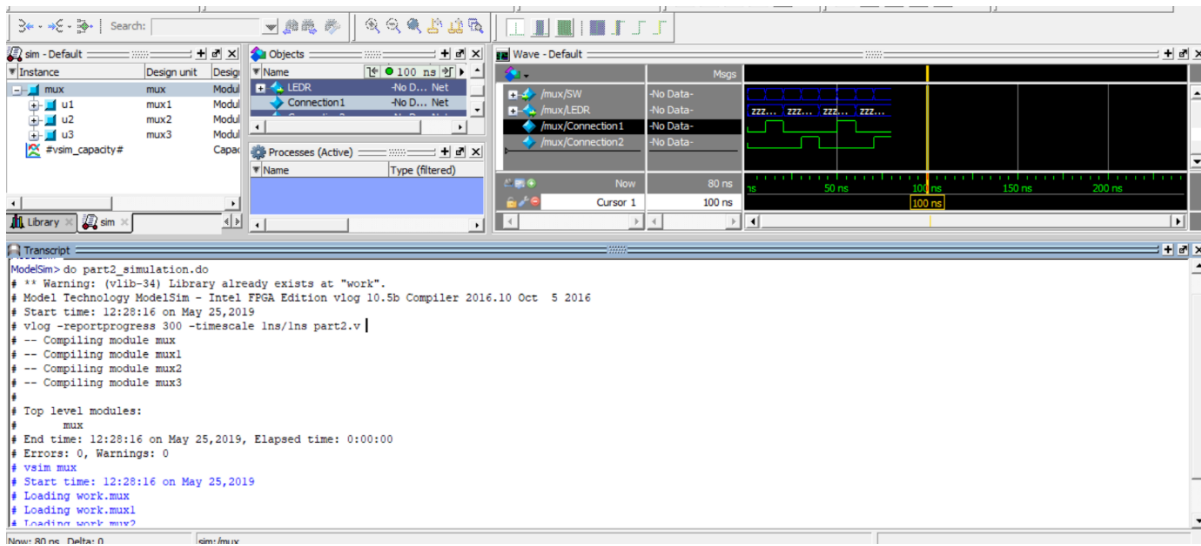
module mux2(v, x, s1, out2);
    input v; //selected when s is 0
    input x; //selected when s is 1
    input s1; //select signal
    output out2; //output

    assign out2 = s1 & x | ~s1 & v;
endmodule

module mux3(in1, in2, s0, m);
    input in1; //selected when s is 0
    input in2; //selected when s is 1
    input s0; //select signal
    output m; //output

    assign m = s0 & in1 | ~s0 & in2;
endmodule
```

5. Simulate your circuit with ModelSim for different values of s, u, v, w and x.



Part III

1. Write the expressions for seven Boolean functions, one for each segment of the 7-segment decoder.

	HEX ₀	HEX ₁	HEX ₂	HEX ₃	HEX ₄	HEX ₅	HEX ₆	c ₃	c ₂	c ₁	c ₀
0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	1	1	1	1	0	0	0	1
2	0	0	1	0	0	1	0	0	0	1	0
3	0	0	0	0	1	1	0	0	0	1	1
4	1	0	0	1	1	0	0	0	1	0	0
5	0	1	0	0	1	0	0	0	1	0	1
6	0	1	0	0	0	0	0	0	1	1	0
7	0	0	0	1	1	1	1	0	1	1	1
8	0	0	0	0	0	0	0	1	0	0	0
9	0	0	0	1	1	0	0	1	0	0	1
A10	0	0	0	1	0	0	0	1	0	1	0
B11	1	1	0	0	0	0	0	1	0	1	1
C12	0	1	1	0	0	0	1	1	1	0	0

D13	1	0	0	0	0	1	0	1	1	0	1
E14	0	1	1	0	0	0	0	1	1	1	0
F15	0	1	1	1	0	0	0	1	1	1	1

HEX ₀	00	01	11	10
00	0	1	0	0
01	1	0	0	0
11	0	1	0	0
10	0	0	1	0

HEX ₁	00	01	11	10
00	0	0	0	0
01	0	1	0	1
11	1	0	1	1
10	0	0	1	0

HEX ₂	00	01	11	10
00	0	0	0	1
01	0	0	0	0
11	1	0	1	1
10	0	0	0	0

HEX ₃	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	1	0
10	0	1	0	1

HEX ₄	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	0	0	0	0
10	0	1	0	0

HEX ₅	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	0	0
10	0	0	0	

HEX ₆	00	01	11	10
00	1	1	0	0
01	0	0	1	0
11	1	0	0	0
10	0	0	0	0

So, we can have:

$$\begin{aligned}
 \text{HEX}_0 &= \bar{C}_3 \cdot \bar{C}_2 \cdot \bar{C}_1 \cdot \bar{C}_0 + C_3 \cdot C_2 \cdot \bar{C}_1 \cdot C_0 + C_3 \cdot \bar{C}_2 \cdot C_1 \cdot C_0 + \bar{C}_3 \cdot C_2 \cdot \bar{C}_1 \cdot C_0 \\
 \text{HEX}_1 &= C_3 \cdot C_1 \cdot \bar{C}_0 + C_3 \cdot C_1 \cdot C_0 + \bar{C}_3 \cdot C_2 \cdot \bar{C}_1 \cdot C_0 + C_3 \cdot C_2 \cdot \bar{C}_0 \\
 \text{HEX}_2 &= \bar{C}_3 \cdot \bar{C}_2 \cdot C_1 \cdot \bar{C}_0 + C_3 \cdot C_2 \cdot C_1 + C_3 \cdot C_2 \cdot \bar{C}_0 \\
 \text{HEX}_3 &= \bar{C}_3 \cdot C_2 \cdot \bar{C}_1 \cdot \bar{C}_0 + \bar{C}_3 \cdot \bar{C}_1 \cdot C_0 + C_2 \cdot C_1 \cdot C_0 + C_3 \cdot \bar{C}_2 \cdot C_1 \cdot \bar{C}_0 \\
 \text{HEX}_4 &= \bar{C}_3 \cdot C_2 \cdot \bar{C}_1 + \bar{C}_3 \cdot \bar{C}_1 \cdot C_0 + \bar{C}_3 \cdot C_0 \\
 \text{HEX}_5 &= \bar{C}_3 \cdot \bar{C}_2 \cdot C_0 + \bar{C}_3 \cdot \bar{C}_2 \cdot C_1 + \bar{C}_3 \cdot C_1 \cdot C_0 + C_3 \cdot C_2 \cdot \bar{C}_1 \cdot C_0 \\
 \text{HEX}_6 &= \bar{C}_3 \cdot \bar{C}_2 \cdot \bar{C}_1 + C_3 \cdot C_2 \cdot \bar{C}_1 \cdot \bar{C}_0 + \bar{C}_3 \cdot C_2 \cdot C_1 \cdot C_0
 \end{aligned}$$

2. Write a Verilog module.

```

module mux(HEX0, SW);
    input [9:0] SW;
    output [6:0] HEX0;

    hex0 u1(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out1(HEX0[0])
    );

    hex1 u2(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out2(HEX0[1])
    );

    hex2 u3(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out3(HEX0[2])
    );

    hex3 u4(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out4(HEX0[3])
    );

    hex4 u5(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out5(HEX0[4])
    );

    hex5 u6(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out6(HEX0[5])
    );

    hex6 u7(
        .c3(SW[0]),
        .c2(SW[1]),
        .c1(SW[2]),
        .c0(SW[3]),
        .out7(HEX0[6])
    );
endmodule

module hex0(c3, c2, c1, c0, out1);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out1; //output

    assign out1 = ~c3 & c2 & ~c1 & ~c0 | c3 & c2 & ~c1 & c0 | c3 & ~c2 & c1 & c0 | ~c3 & ~c2 & ~c1 & c0;
endmodule

```

```

module hex1(c3, c2, c1, c0, out2);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out2; //output

    assign out2 = c2 & c1 & ~c0 | c3 & c1 & c0 | ~c3 & c2 & ~c1 & c0 | c3 & c2 & ~c0;

endmodule

module hex2(c3, c2, c1, c0, out3);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out3; //output

    assign out3 = ~c3 & ~c2 & c1 & ~c0 | c3 & c2 & c1 | c3 & c2 & ~c0;

endmodule

module hex3(c3, c2, c1, c0, out4);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out4; //output

    assign out4 = ~c3 & c2 & ~c1 & ~c0 | ~c2 & ~c1 & c0 | c2 & c1 & c0 | c3 & ~c2 & c1 & ~c0;

endmodule

module hex4(c3, c2, c1, c0, out5);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out5; //output

    assign out5 = ~c3 & c2 & ~c1 | ~c2 & ~c1 & c0 | ~c3 & c0;

endmodule

module hex5(c3, c2, c1, c0, out6);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out6; //output

    assign out6 = ~c3 & ~c2 & c0 | ~c3 & ~c2 & c1 | ~c3 & c1 & c0 | c3 & c2 & ~c1 & c0;

endmodule

module hex6(c3, c2, c1, c0, out7);
    input c3; //selected when s is 0
    input c2; //selected when s is 1
    input c1; //select signal
    input c0;
    output out7; //output

    assign out7 = ~c3 & ~c2 & ~c1 | c3 & c2 & ~c1 & ~c0 | ~c3 & c2 & c1 & c0;

endmodule

```

3. Simulation.

