CS 223 - Digital Design

Section – 1

Lab – 03

Erdem Ege Eroğlu

21601636

25.03.2020

(b) Write a *Behavioral* SystemVerilog module for the 3:8 decoder and a testbench for it.

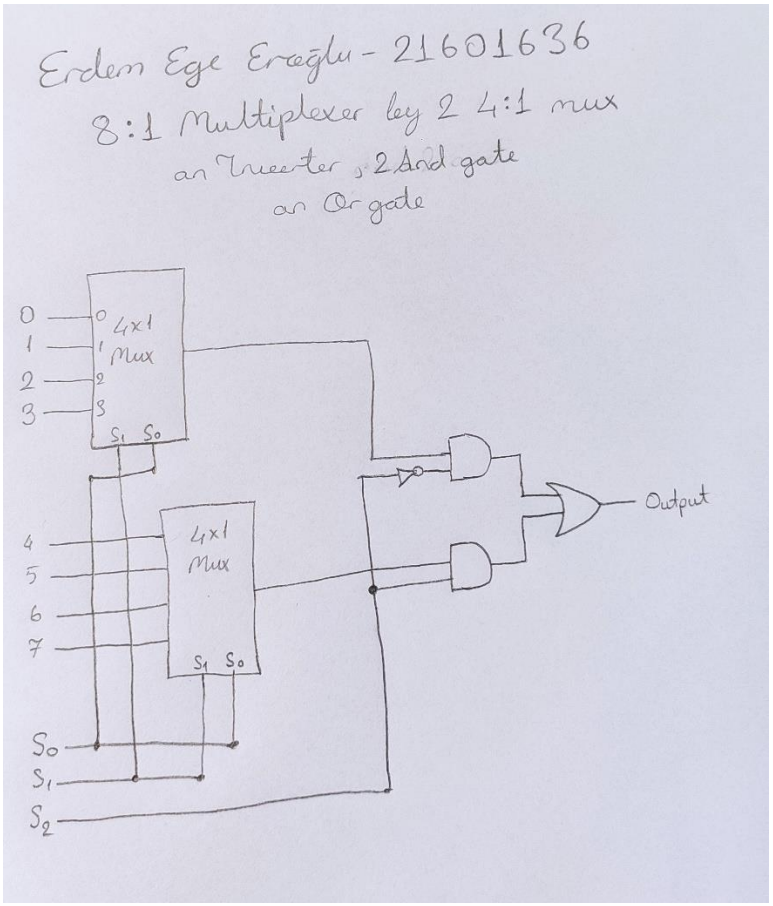
```
module decoder3_8(input logic a, b, c,
                  output logic y0,y1,y2,y3,y4,y5,y6,y7);
        assign y0=(~a&~b&~c);
        assign y1=(~a&~b&c);
        assign y2=(~a&b&~c);
        assign y3=(~a&b&c);
        assign y4=(a&~b&~c);
        assign y5=(a&~b&c);
        assign y6=(a&b&~c);
        assign y7=(a&b&c);

endmodule

module decoderTest();
    logic a,b,c, d0,d1,d2,d3,d4,d5,d6,d7;

    decoder3_8 uut(a,b,c, d0,d1,d2,d3,d4,d5,d6,d7);
    initial begin
        a = 0; b = 0; c = 0; #10;
        a = 0; b = 0; c = 1; #10;
        a = 0; b = 1; c = 0; #10;
        a = 0; b = 1; c = 1; #10;
        a = 1; b = 0; c = 0; #10;
        a = 1; b = 0; c = 1; #10;
        a = 1; b = 1; c = 0; #10;
        a = 1; b = 1; c = 1; #10;
    end
endmodule
```

(c) Write a *Behavioral* SystemVerilog module for the 4:1 multiplexer.

```
module mux4_1( a, b, c, d, s0, s1, out );
   input wire a,b,c,d;
   input wire s0,s1;
   output reg out;

   always @ (a or b or c or d or s0 or s1)
   begin

      case(s0|s1)
      2'b00 : out <= a;
      2'b00 : out <= b;
      2'b00 : out <= c;
      2'b00 : out <= d;
      endcase
   end
endmodule


module mux4_1test;
wire out;
reg a;
reg b;
reg c;
reg d;
reg s0, s1;
```

```
mux4_1 uut(.out(out), .a(a), .b(b), .c(c), .d(d), .s0(s0), .s1(s1));
initial
   begin

   a=1'b0; b=1'b0; c=1'b0; d=1'b0;
   s0=1'b0; s1=1'b0;
   #500 $finish;
   end
always #40 a = ~a;
always #20 b = ~b;
always #10 c = ~c;
always #5 d = ~d;
always #80 s0 = ~s0;
always #160 s1 = ~s1;

always@(a or b or c or d or s0 or s1)
$monitor("At time = %t, Output = %d", $time, out);
Endmodule
```

(d) Draw a circuit schematic (block diagram) for the 8:1 multiplexer by using two 4:1 multiplexers, an INVERTER, two AND gates, and an OR gate.

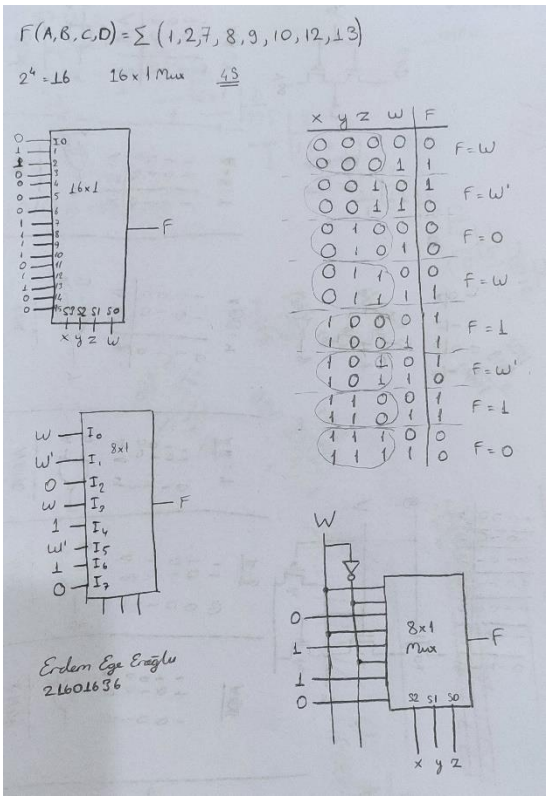


e) Write a *Structural* SystemVerilog module for the 8:1 multiplexer you designed in part (d) and a testbench for it.

```
module and_gate(input a,b, output c);

        assign c = a & b;

endmodule

module or_gate(input d,e, output f);

  assign f = d | e;

endmodule

module not_gate(input g, output h);

  assign h = ~g;

endmodule

module mux8_1(a,b,c,d,e,f,g,h,s2,s1,s0,out);

  input a, b, c, d, e, f, g, h, s2, s1, s0;

  output out;

  wire s2bar, t1, t2, t3, t4;

  not_gate u1(s2,s2bar);

  mux4_1 u2(a,b,c,d,s0,s1,t1);

  mux4_1 u3(e,f,g,h,s0,s1,t2);

  and_gate u4(t1,s2bar,t3);

  and_gate u5(t2,s2,t4);

  or_gate u6(t3,t4,out);

endmodule

module mux8_1test;

  reg a,b,c,d,e,f,g,h,s0,s1,s2;

  wire out;

    mux8_1 mux(.a(a), .b(b), .c(c), .d(d), .e(e), .f(f), .g(g), .h(h), .s0(s0), .s1(s1), .s2(s2), .out(out));

  initial begin

    a = 1'b0;

    b = 1'b0;

    c = 1'b0;

    d = 1'b0;

    e = 1'b0;

    f = 1'b0;
```

```
        g = 1'b0;
        h = 1'b0;
        s0 = 1'b0;
        s1 = 1'b0;
        s2 = 1'b0;
        #45 $finish;
    end
    always #2 a = ~a;
    always #4 b = ~b;
    always #6 c = ~c;
    always #8 d = ~d;
    always #10 e = ~e;
    always #12 f = ~f;
    always #14 g = ~g;
    always #16 h = ~h;
    always #3 s0 = ~s0;
    always #3 s1 = ~s1;
    always #3 s2 = ~s2;
    always @ (out)
    $display("time=%0t INPUT VALUES:\t a=%b b=%b c=%b d=%b e=%b f=%b g=%b h=%b s0=%b
s1=%b s2=%b \t OUTPUT VALUE: out=%b",$time,a,b,c,d,e,f,g,h,s0,s1,s2,out);
endmodule
```

(f) Draw a circuit schematic (block diagram) to implement the Boolean function F(A,B,C,D) = Σ(1, 2, 7, 8, 9, 10, 12, 13) by using only one 8:1 multiplexer and one INVERTER.



(g) Write a *Structural* SystemVerilog module for the circuit you designed in part (f) and a testbench for it.

```
module fun( a, b, c, d, out );
   input a, b, c, d;
   output out;
   wire t1;
   not_gate(d,t1);
   mux8_1(d,t1,0,d,1,t1,1,0,out);
endmodule

module funtest();
   reg a,b,c,d,e,f,g,h,s0,s1,s2;
   wire out;

   mux8_1 mux(.a(a), .b(b), .c(c), .d(d), .e(e), .f(f), .g(g), .h(h), .s0(s0), .s1(s1), .s2(s2), .out(out));
```

```
  initial begin
    a = 1'b0;
    b = 1'b0;
    c = 1'b0;
    d = 1'b0;
    e = 1'b0;
    f = 1'b0;
    g = 1'b0;
    h = 1'b0;
    s0 = 1'b0;
    s1 = 1'b0;
    s2 = 1'b0;

    #45 $finish;
  end

   always #2 a = ~a;
   always #4 b = ~b;
   always #6 c = ~c;
   always #8 d = ~d;
   always #10 e = ~e;
   always #12 f = ~f;
   always #14 g = ~g;
   always #16 h = ~h;
   always #3 s0 = ~s0;
   always #3 s1 = ~s1;
   always #3 s2 = ~s2;
   always @ (out)
   $display("time=%0t INPUT VALUES:\t a=%b b=%b c=%b d=%b e=%b f=%b g=%b h=%b s0=%b
s1=%b s2=%b \t OUTPUT VALUE: out=%b",$time,a,b,c,d,e,f,g,h,s0,s1,s2,out);
Endmodule
```