LAB 03

MELİKE
DEMİRCİ
21702346

12/03/2020

b) Behavioral SystemVerilog module for 3-to-8 decoder and a testbench for it

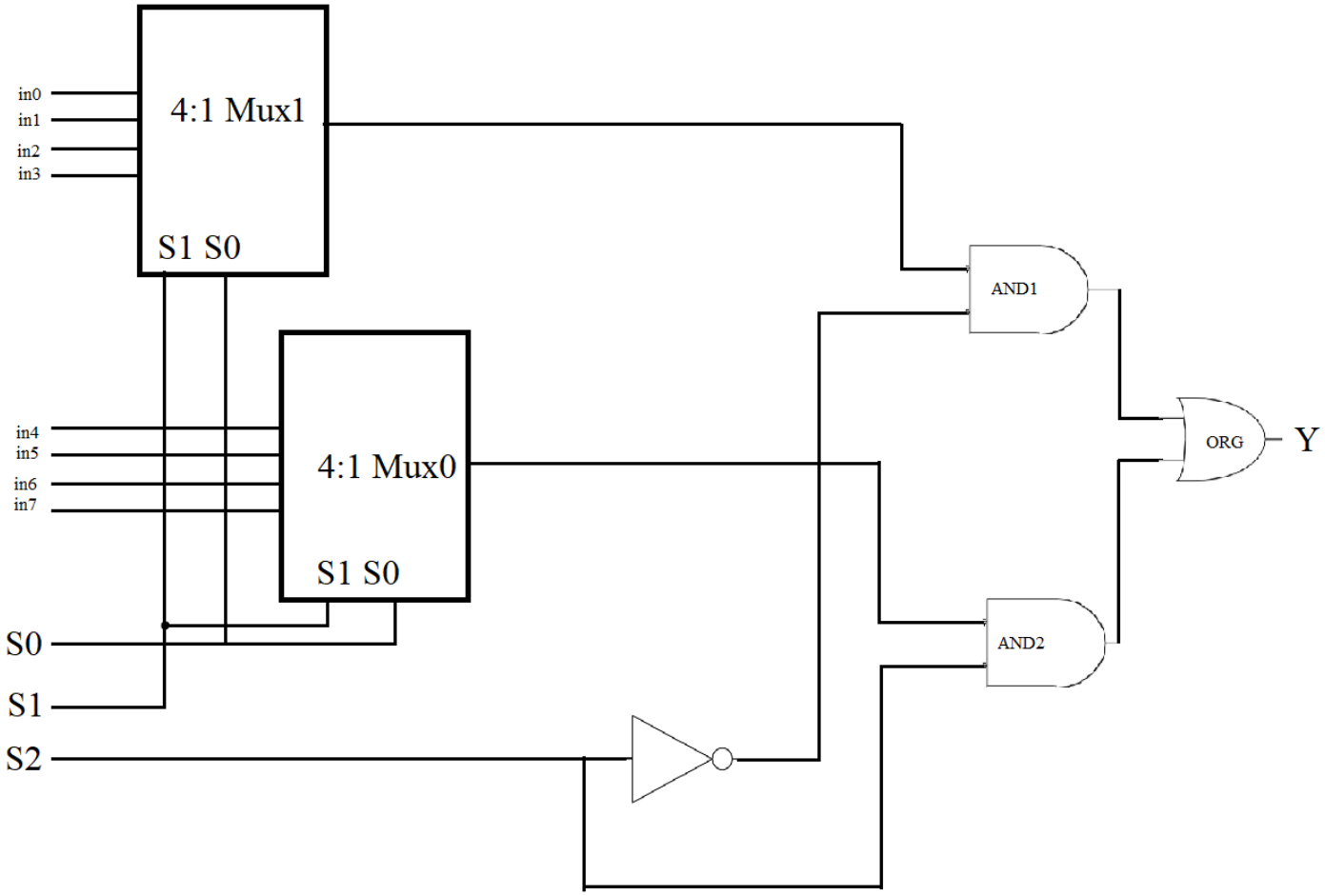
```
module  decoder3to8 ( input logic in2, in1, in0,
                      output logic y7, y6, y5, y4, y3, y2, y1, y0) ;
        assign y7 = in2 & in1 & in0 ;
        assign y6 = in2 & in1 & ~in0
        assign y5 = in2 & ~in1 & in0
        assign y4 = in2 & ~in1 & ~in0
        assign y3 = ~in2 & in1 & in0
        assign y2 = ~in2 & in1 & ~in0
        assign y1 = ~in2 & ~in1 & in0
        assign y0 = ~in2 & ~in1 & ~in0
endmodule
```

```
module decoder3to8_tb();
logic in2, in1, in0;
logic y7, y6, y5, y4, y3, y2, y1, y0;
decoder3to8 decoder(in2, in1, in0, y7, y6, y5, y4, y3, y2, y1, y0);  // instantiate dut
    initial begin
        in2 = 0; in1 = 0; in0 = 0; #100;
        in2 = 1; in1 = 0; in0 = 0; #100;
        in2 = 0; in1 = 1; in0 = 0; #100;
        in2 = 1; in1 = 1; in0 = 0; #100;
        in2 = 0; in1 = 0; in0 = 1; #100;
        in2 = 1; in1 = 0; in0 = 1; #100;
        in2 = 0; in1 = 1; in0 = 1; #100;
        in2 = 1; in1 = 1; in0 = 1; #100;
    end
endmodule
```

c) Behavioral SystemVerilog module for 4-to-1 multiplexer.

```
module mux4to1( input logic in3, in2, in1, in0, s1, s0,
                output logic y);
        assign out = s1 ? (s0 ? in3 : in2) : (s0 ? in1 : in0);
endmodule
```

(d) Circuit Schematic of 8:1 Multiplexer(two 4:1 multiplexers, an INVERTER, two AND, an OR)



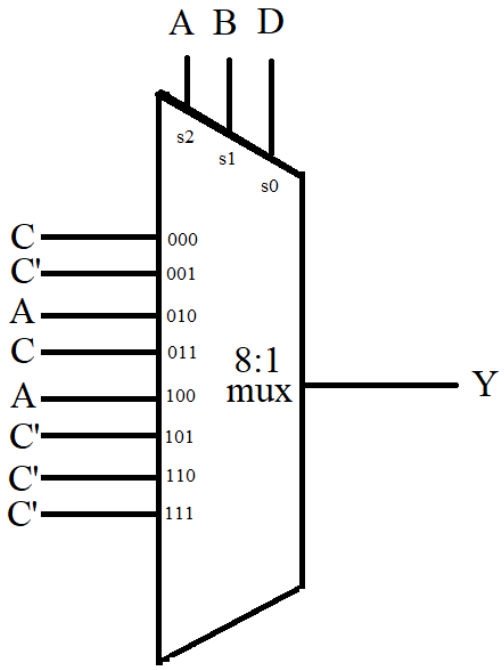
(e) Structural SystemVerilog module for the 8:1 multiplexer and test bench

```
module mux8to1( input logic in0, in1, in2, in3, in4, in5, in6, in7, s2, s1, s0,
                        output logic y);
        logic y1, y2, c, e, d;
        mux4to1 mux1(in3, in2, in1, in0, s1, s0, y1);
        mux4to1 mux2(in7, in6, in5, in4, s1, s0, y2);
        not notgate(c, s2);
        and and1(d, y1, c);
        and and2(e, y2, s2);
        or org(y, e, d);
endmodule
```

```
module mux8to1_tb();
   logic  in0, in1, in2, in3, in4, in5, in6, in7, s0, s1, s2;
   logic y;
   mux8to1 mux(in0, in1, in2, in3, in4, in5, in6, in7,s0, s1, s2, y);  // instantiate dut
   initial begin // apply inputs, check results one at a time
      in0 = 1; in1 = 0; in2 = 1; in3 = 0; in4 = 0; in5 = 1; in6 = 1; in7 = 1; s0 = 0; s1 = 0; s2 = 0; #100;
      s0 = 0; s1 = 0; s2 = 1; #100;
      s0 = 0; s1 = 1; s2 = 0; #100;
      s0 = 0; s1 = 1; s2 = 1; #100;
      s0 = 1; s1 = 0; s2 = 0; #100;
      s0 = 1; s1 = 0; s2 = 1; #100;
      s0 = 1; s1 = 1; s2 = 0; #100;
      s0 = 1; s1 = 1; s2 = 1; #100;
    end
endmodule
```

(f) Circuit schematic for function F(A,B,C,D) = ∑(1, 2, 7, 8, 9, 10, 12, 13)

A	B	C	D	Y
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0



(g) Structural SystemVerilog module for function and testbench

```
module function(input logic A, B, C, D,
                output Y);
  logic c;
  not notgate(c, C);
  mux8to1 mux(C,c,A,C,A,c,c,c, A, B, D, y);
endmodule
```

```
module function_tb();
logic A, B, C, D;
logic Y;
function f(A, B, C, D, Y);  // instantiate dut
    initial begin
      A = 0; B = 0; C = 0; D = 0; #100;
      A = 0; B = 0; C = 0; D = 1; #100;
```

```
        A = 0; B = 0; C = 1; D = 0; #100;
        A = 0; B = 0; C = 1; D = 1; #100;
        A = 0; B = 1; C = 0; D = 0; #100;
        A = 0; B = 1; C = 0; D = 1; #100;
        A = 0; B = 1; C = 1; D = 0; #100;
        A = 0; B = 1; C = 1; D = 1; #100;
        A = 1; B = 0; C = 0; D = 0; #100;
        A = 1; B = 0; C = 0; D = 1; #100;
        A = 1; B = 0; C = 1; D = 0; #100;
        A = 1; B = 0; C = 1; D = 1; #100;
        A = 1; B = 1; C = 0; D = 0; #100;
        A = 1; B = 1; C = 0; D = 1; #100;
        A = 1; B = 1; C = 1; D = 0; #100;
        A = 1; B = 1; C = 1; D = 1; #100;
        end
endmodule
```