Course Code: CS223

Course Name: Digital Design **Section:** 4

Lab 2

Name: Berk Saltuk Yılmaz **ID:** 21903419

Date: 16.10.2020

Part (b):

Behavioral SystemVerilog module for 2-to-4 decoder:

Project Summary | tb_two_to_four_decoder.sv | two_to_four_decoder.sv

C:/Users/bsy/vivadows/lab3/lab3.srcs/sources_1/imports/Lab03 SystemVerilog Codes/two_to_four_decoder.sv

```
module two_to_four_decoder(input logic a, b,
                           output logic y0, y1, y2, y3);

    assign y0 = ~a & ~b;
    assign y1 = ~a & b;
    assign y2 = a & ~b;
    assign y3 = a & b;

endmodule
```

Testbench for behavioral SystemVerilog module for the full adder:

Project Summary | tb_two_to_four_decoder.sv | two_to_four_decoder.sv

C:/Users/bsy/vivadows/lab3/lab3.srcs/sources_1/imports/Lab03 SystemVerilog Codes/tb_two_to_four_decoder.sv

```
module tb_two_to_four_decoder;

logic a, b;
logic y0, y1, y2, y3;

two_to_four_decoder ttfd1( a, b, y0, y1, y2, y3);

initial begin
    a = 0; b = 0; #10;
    a = 0; b = 1; #10;
    a = 1; b = 0; #10;
    a = 1; b = 1; #10;
end
endmodule
```

Part (c):

Behavioral SystemVerilog module for 4-to-1 multiplexer:

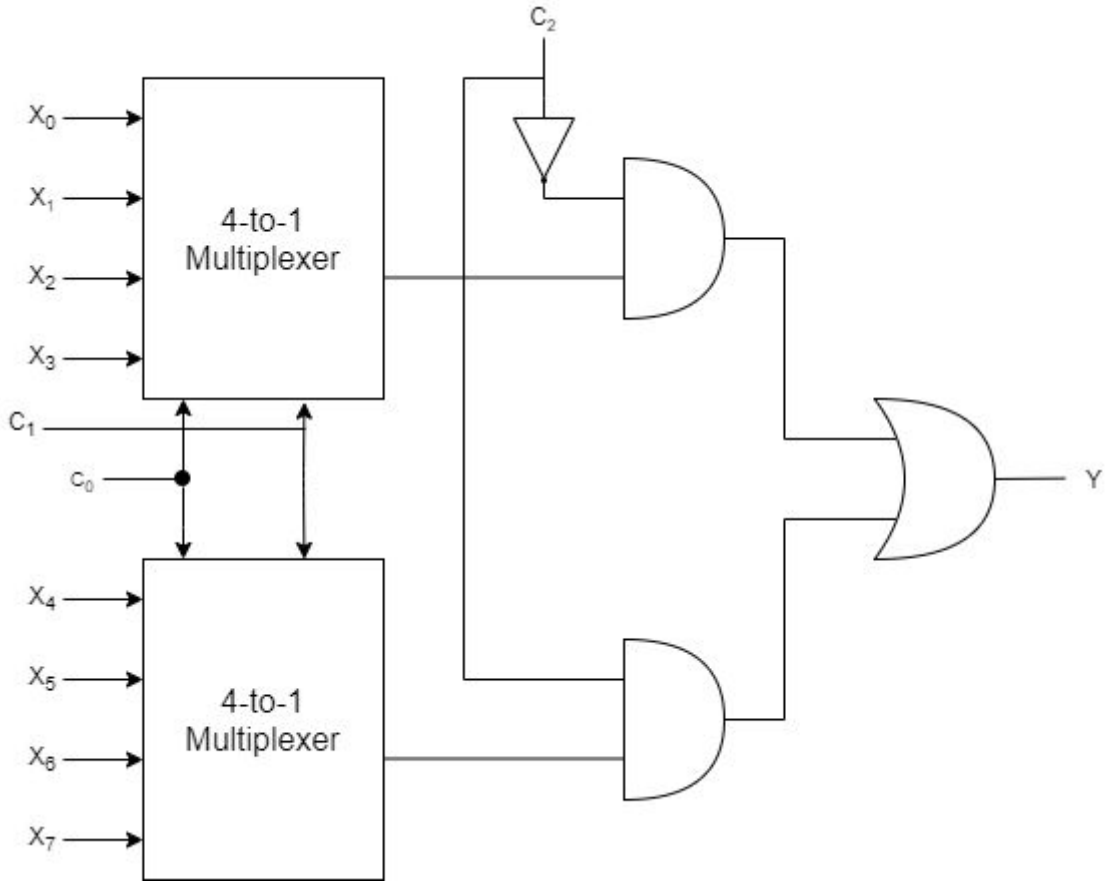
Project Summary | tb_two_to_four_decoder.sv | two_to_four_decoder.sv | four_to_one_mux.sv *

C:/Users/bsy/vivadows/lab3/lab3.srcs/sources_1/imports/Lab03 SystemVerilog Codes/four_to_one_mux.sv

```
module four_to_one_mux( input logic x0, x1, x2, x3, c0, c1,
                        output logic y);

    logic and0, and1, and2, and3;

    assign and0 = x0 & ~c0 & ~c1;
    assign and1 = x1 & ~c0 & c1;
    assign and2 = x2 & c0 & ~c1;
    assign and3 = x3 & c0 & c1;

    assign y = and0 | and1 | and2 | and3;
endmodule
```

Part (d):

Schematic for structural 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate:



Structural SystemVerilog module for 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate:

two_to_four_decoder.sv | four_to_one_mux.sv | eight_to_one_mux.sv | tb_eight_to_one_mux.sv

C:/Users/bsy/vivadows/lab3/lab3.srcs/sources_1/imports/Lab03 SystemVerilog Codes/eight_to_one_mux.sv

```
module eight_to_one_mux( input logic x0, x1, x2, x3,
                                     x4, x5, x6, x7,
                                     c0, c1, c2,
                         output logic y);

    logic result1, result2;

    four_to_one_mux ftom1(x0, x1, x2, x3, c0, c1, result1);
    four_to_one_mux ftom2(x4, x5, x6, x7, c0, c1, result2);

    assign y = (result1 & c2) | (result2 & c2);

endmodule
```

Testbench for structural SystemVerilog module for 8-to-1 MUX:

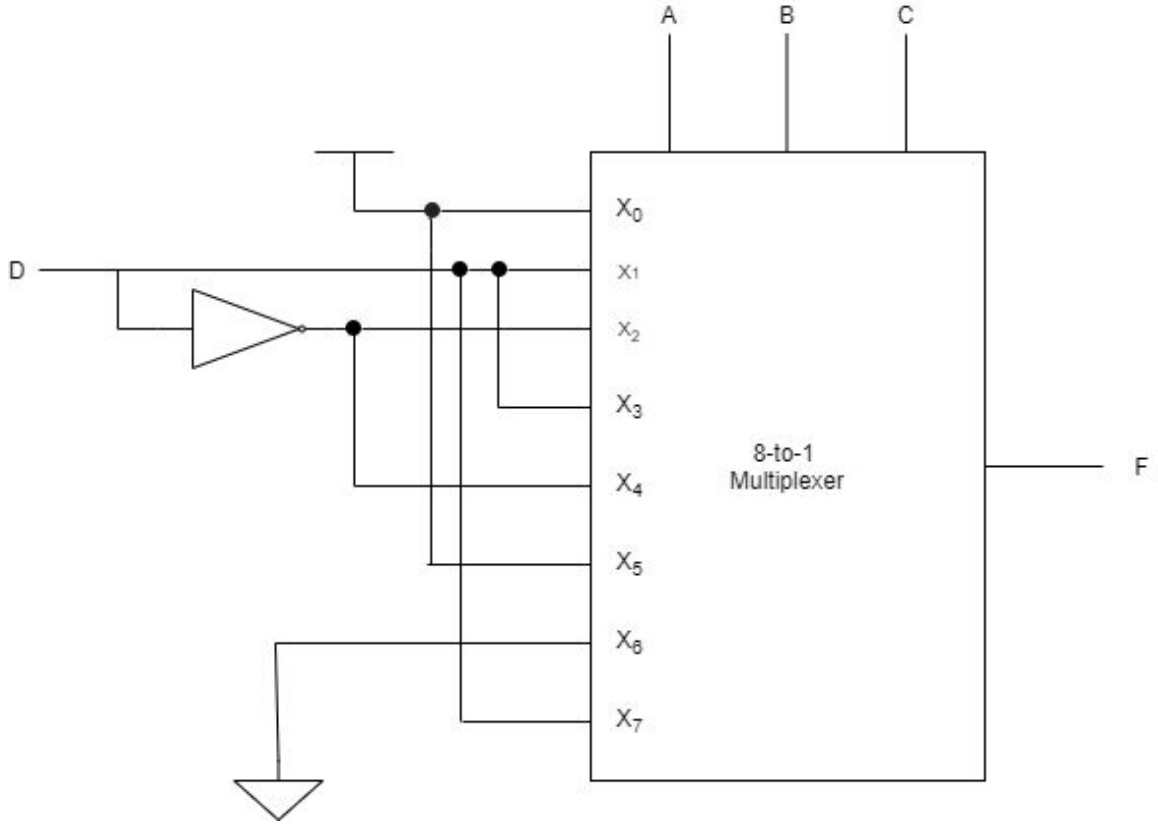
two_to_four_decoder.sv | four_to_one_mux.sv | eight_to_one_mux.sv | tb_eight_to_one_mux.sv

C:/Users/bsy/vivadows/lab3/lab3.srcs/sim_1/imports/Lab03 SystemVerilog Codes/tb_eight_to_one_mux.sv

```
module tb_eight_to_one_mux;
logic x0, x1, x2, x3, x4, x5, x6, x7, c0, c1, c2, y;

eight_to_one_mux etom1(x0, x1, x2, x3, x4, x5, x6, x7, c0, c1, c2, y);

integer i;
initial begin

    for( i = 0; i < 2**11; i++) begin
        {x0, x1, x2, x3, x4, x5, x6, x7, c0, c1, c2} = i; #1;
    end
end

endmodule
```

Part (e):

Schematic for F(A,B,C,D)=∑(0,1,3,4,7,8,10,11,15) function, using one 8-to-1 multiplexer and an inverter:



SystemVerilog module for F(A,B,C,D)=∑(0,1,3,4,7,8,10,11,15) function, using one 8-to-1 multiplexer and an inverter: