Course Name: CS

Code Number: 223

Number of the lab: 3

Name: Ahmet Faruk Ulutaş

Student ID: 21803717

Date: 07.03.2021

B) Behavioral Two to Four Decoder

```
module behavioralTwoToFourDecoder( input logic w0, w1, output logic d0, d1, d2, d3);

logic inv0, inv1;


assign inv0 = ~w0;

assign inv1 = ~w1;


assign d0 = inv0 & inv1;

assign d1 = inv0 & w1;

assign d2 = w0 & inv1;

assign d3 = w0 & w1;

endmodule
```

Testbench

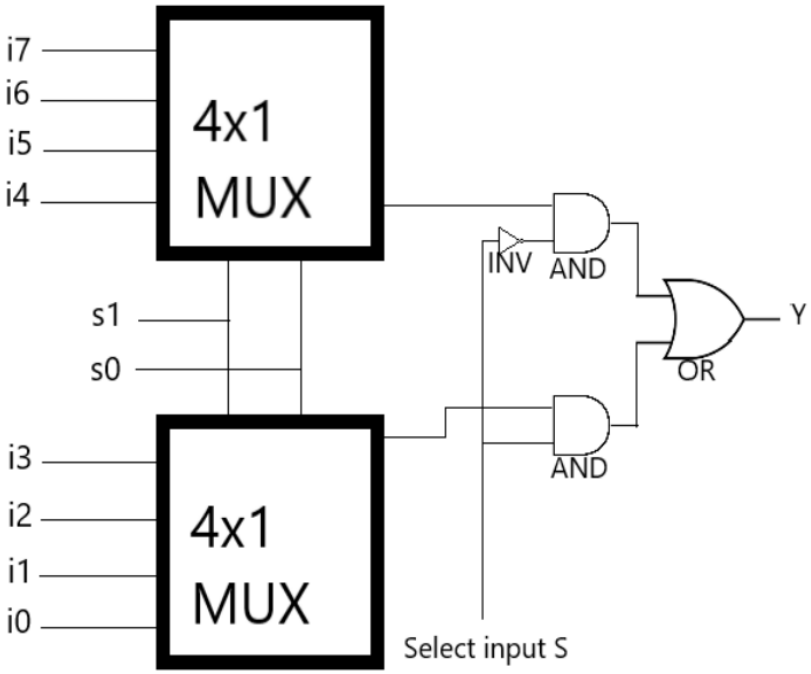
```
module testbenchOfTwoToFourDecoder();

logic w0, w1, d0, d1, d2, d3;


behavioralTwoToFourDecoder dut( w0, w1, d0, d1, d2, d3);


initial begin

    w0 = 0; w1 = 0; #10;

    w1 = 1; #10;

    w0 = 1; w1 = 0; #10;

    w1 = 1; #10;
```

```
end

endmodule
```

C) Behavioral 4to1 mux

```
module behavioralFourToOneMux( input logic S1, S0, D0, D1, D2, D3,
output logic Y);

logic inv0, inv1, and0, and1, and2, and3;

assign inv0 = ~S0;

assign inv1 = ~S1;

assign and0 = inv1 & inv0 & D0;

assign and1 = inv1 & S0 & D1;

assign and2 = S1 & inv0 & D2;

assign and3 = S1 & S0 & D3;

assign or0 = and0 | and1 | and2 | and3;

assign Y = or0;

endmodule
```

D)



8to1 mux system verilog

```
module eightToOneMux( input logic i0, i1, i2, i3, i4, i5, i6, i7, s0, s1, S, output
Y);
logic mux4to1_0, mux4to1_1, and0, and1, inv0;

mux4to1 mux4to1__0( i7, i6, i5, i4, s1, s0, mux4to1_0);
mux4to1 mux4to1__1( i3, i2, i1, i0, s1, s0, mux4to1_1);

inv0 inv_0( S, inv0);
and0 and_0( mux4to1_0, inv0, and0);
and0 and_1( mux4to1_1, S, and1);
or0 or_0( and0, and1, Y);

endmodule

module mux4to1( input logic a, b, c, d, s1, s0, output logic out);
```

```
assign out = (~s1 & ~s0 & a) | (~s1 & s0 & b) | (s1 & ~s0 & c) | (s1 & s0 & d);
endmodule

module inv0( input logic a, output logic out);
assign out = ~a;
endmodule

module and0( input logic a, b, output logic out);
assign out = a & b;
endmodule

module or0( input logic a, b, output logic out);
assign out = a | b;
endmodule
```

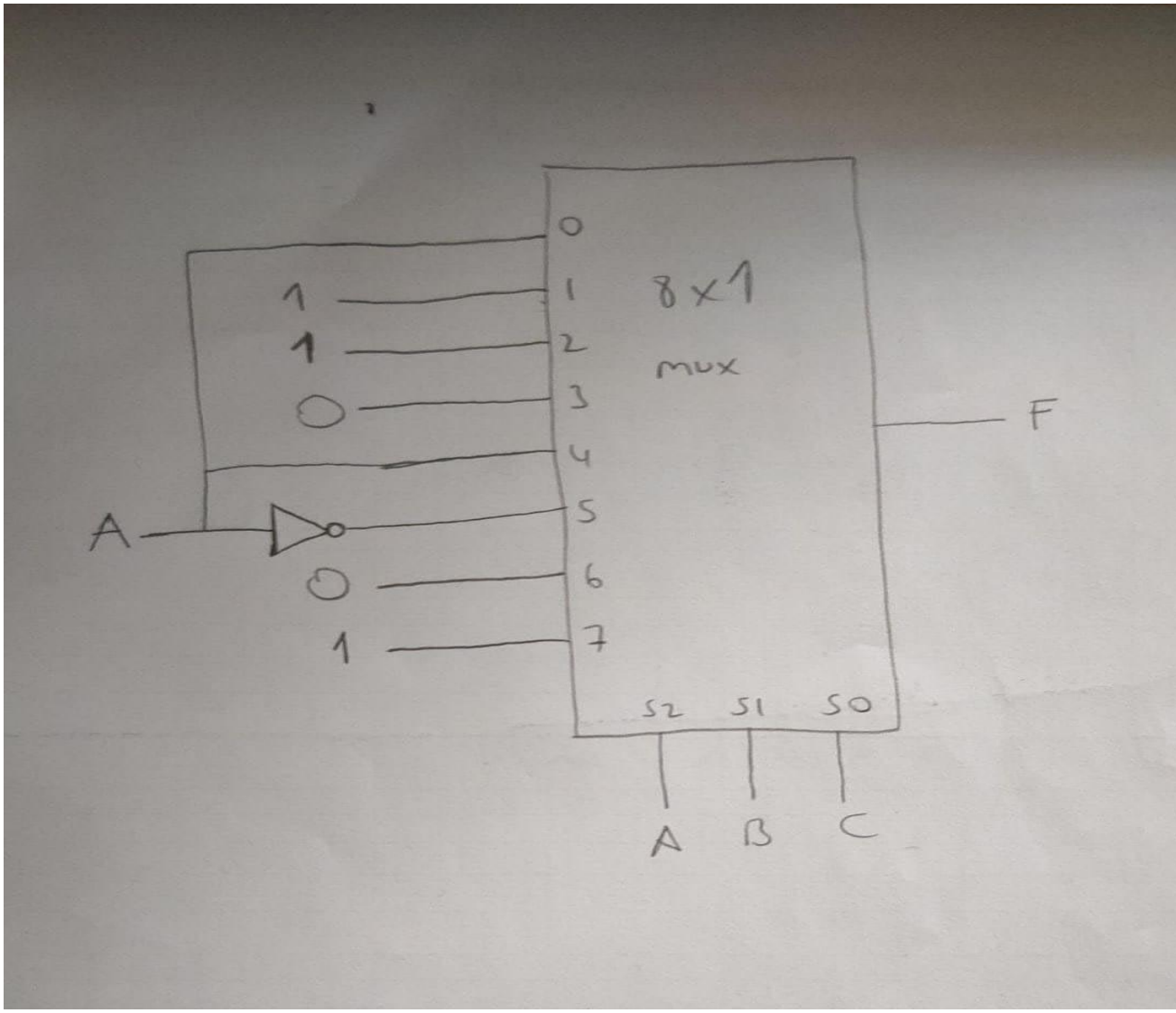
Testbench

```
module testbenchEightToOneMux();
logic i0, i1, i2, i3, i4, i5, i6, i7, s0, s1, S, Y;

eightToOneMux dut( i0, i1, i2, i3, i4, i5, i6, i7, s0, s1, S, Y);

initial begin
i0 = 0; i1 = 1; i2 = 0; i3 = 1; i4 = 0; i5 = 1; i6 = 0; i7 = 1;
S = 0; s1 = 0; s0 = 0; #10;
s0 = 1; #10;
s1 = 1; s0 = 0; #10;
s0 = 1; #10;
S = 1; s1 = 0; s0 = 0; #10;
s0 = 1; #10;
s1 = 1; s0 = 0; #10;
```

```
s0 = 1; #10;
end


endmodule
```

E) Schematic (block diagram) and SystemVerilog module for F(A,B,C,D)=∑(1,2,5,7,8,9,10,12,15) function, using one8-to-1 multiplexer and an inverter

Schematic (block diagram)



System verilog module

```
module in4f( input logic A, B, C, D, output logic F);

logic inv0;

inv1 inv2( A, inv0);
eight_to_one_mux eight_to_one_mux0( A, 1, 1, 0, A, inv0, 0, 1, B, C, D, F);

endmodule

module inv1( input logic a, output logic out);
assign out = ~a;
```

```
endmodule


module eight_to_one_mux( input logic in0, in1, in2, in3, in4, in5, in6, in7, s2, s1, s0, output
logic out);
assign out = (in0 & ~s2 & ~s1 & ~s0) | (in1 & ~s2 & ~s1 & s0) | (in2 & ~s2 & s1 & ~s0) |
(in3 & ~s2 & s1 & s0) | (in4 & s2 & ~s1 & ~s0) | (in5 & s2 & ~s1 & s0) | (in6 & s2 & s1 &
~s0) | (in7 & s2 & s1 & s0);
endmodule
```