

CS223 - DIGITAL DESIGN

SECTION 3 - LAB 3

ARDA TAVUSBAY

21902722

31.10.2021

B) Behavioral SystemVerilog module for a 2-to-4 decoder and a Testbench**Behavioral 2-to-4 Decoder:**

```
'timescale 1ns / 1ps
```

```
module BehavioralTwoFourD(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:

```
module TB_BehavioralTwoFourD();
```

```
logic a, b, y0, y1, y2, y3;
BehavioralTwoFourD dut(a, b, y0, y1, y2, y3);

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

endmodule
```

C) Behavioral SystemVerilog module for a 2-to-1 multiplexer

```
module BehavioralTwoOneM(input logic i0, i1, s, output logic y);

assign y = s ? i1 : i0;

endmodule
```

D) Structural SystemVerilog module for a 4-to-1 multiplexer by using three 2-to-1 multiplexers and its Testbench

Structural 4-to-1 multiplexer by using three 2-to-1 multiplexers:

```
'timescale 1ns / 1ps
```

```
module StructuralFourOneM(input logic i0, i1, i2, i3, s0, s1, output logic y);

logic w0, w1;

BehavioralTwoOneM mux1 (i0, i1, s0, w0);
BehavioralTwoOneM mux2 (i2, i3, s0, w1);
BehavioralTwoOneM mux3 (w0, w1, s1, y);

endmodule
```

TestBench:

```
'timescale 1ns / 1ps
```

```
module TB_StructuralFourOneM();

logic i0, i1, i2, i3, s0, s1, y;
StructuralFourOneM dut(i0, i1, i2, i3, s0, s1, y);

initial begin
    s1 = 0; s0 = 0; i3 = 0; i2 = 0; i1 = 0; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 0; i1 = 0; i0 = 1; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 0; i1 = 1; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 0; i1 = 1; i0 = 1; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 1; i1 = 0; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 1; i1 = 0; i0 = 1; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 1; i1 = 1; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 0; i2 = 1; i1 = 1; i0 = 1; #10;
    s1 = 0; s0 = 0; i3 = 1; i2 = 0; i1 = 0; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 1; i2 = 0; i1 = 0; i0 = 1; #10;
    s1 = 0; s0 = 0; i3 = 1; i2 = 0; i1 = 1; i0 = 0; #10;
    s1 = 0; s0 = 0; i3 = 1; i2 = 0; i1 = 1; i0 = 1; #10;
```

s1 = 0; s0 = 0; i3 = 1; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 0; s0 = 0; i3 = 1; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 0; s0 = 0; i3 = 1; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 0; s0 = 0; i3 = 1; i2 = 1; i1 = 1; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 0; i1 = 1; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 0; i1 = 1; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 0; i2 = 1; i1 = 1; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 0; i1 = 1; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 0; i1 = 1; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 0; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 0; i2 = 1; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 0; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 0; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 0; i3 = 1; i2 = 1; i1 = 1; i0 = 1; #10;

```

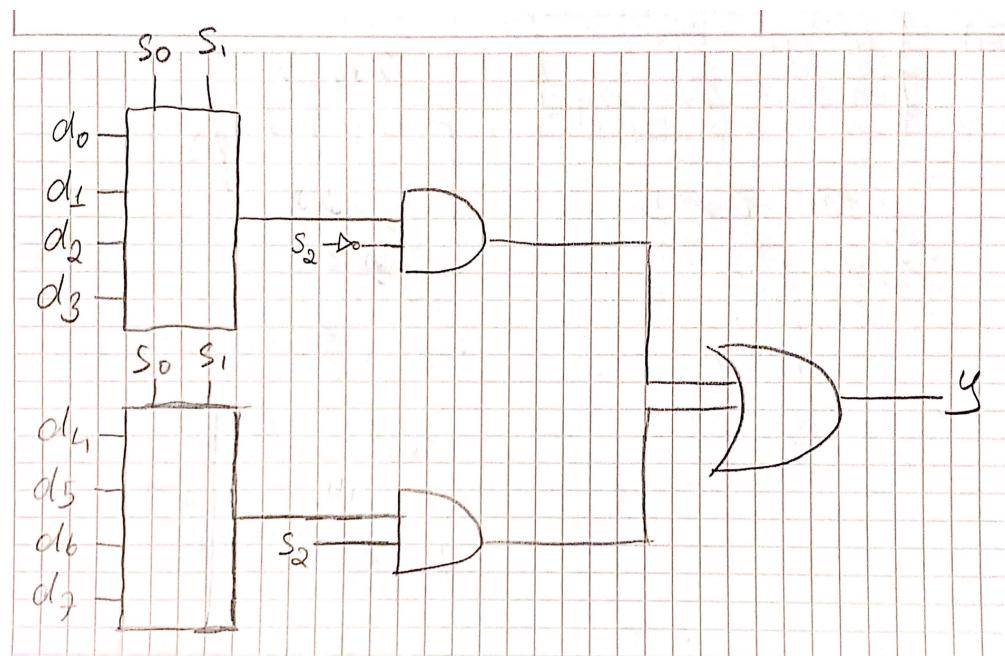
s1 = 1; s0 = 1; i3 = 0; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 0; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 0; i2 = 0; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 0; i2 = 0; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 0; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 0; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 0; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 0; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 0; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 0; i1 = 1; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 0; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 0; i0 = 1; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 0; #10;
s1 = 1; s0 = 1; i3 = 1; i2 = 1; i1 = 1; i0 = 1; #10;
end

```

endmodule

E) Block diagram and structural SystemVerilog module of 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate.

Block diagram of 8-to-1 MUX:



SystemVerilog module of 8-to-1 MUX:

```
module StructuralEightOneM(input logic d0, d1, d2, d3, d4, d5, d6, d7, s0, s1, s2, output logic y);

logic w0, w1, w2, w3, w4;

StructuralFourOneM mux1(d0, d1, d2, d3, s0, s1, w0);
StructuralFourOneM mux2(d4, d5, d6, d7, s0, s1, w1);
Inverter inverter(s2, w3);
AndGate and1(w0, w3, w4);
AndGate and2(w1, s2, w5);
OrGate or1(w4, w5, y);

endmodule
```

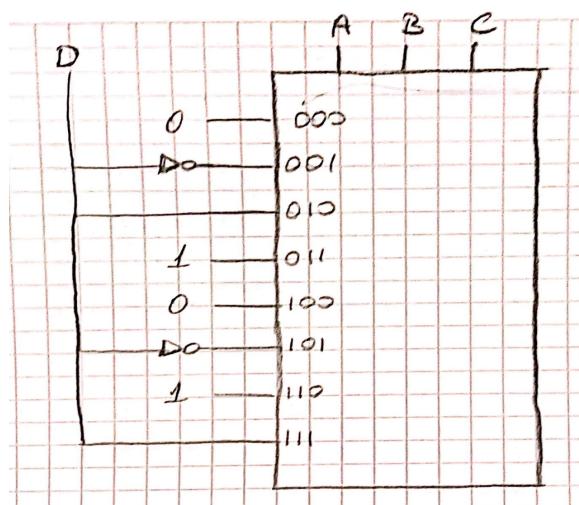
```
module AndGate(input logic a, b, output logic y);
assign y = a & b;
endmodule
```

```
module OrGate(input logic a, b, output logic y);
assign y = a | b;
endmodule
```

```
module Inverter(input logic a, output logic y);
assign y = ~a;
endmodule
```

F) Block diagram and SystemVerilog module for $F(A,B,C,D)=\sum(2,5,6,7,10,12,13,15)$ function, using one (not two) 8-to-1 multiplexer and an Inverter.

Block diagram for module F(A,B,C,D):



SystemVerilog module for F(A,B,C,D):

```
module FunctionABCD(input logic a, b, c, d, output logic y);

logic notD;

Inverter inv(d, notD);
StructuralEightOneM mux3(0, notD, d, 1, 0, notD, 1, d, c, b, a, y);

endmodule
```