2021–2022 FALL SEMESTER

CS223 – LAB 03

Modeling Decoders and MUXs in System Verilog

NAME: ABDULLAH RIAZ
ID : 22001296
COURSE:CS223 – DIGTAL DESGIN
SECTION: 01
DATE: 30/10/2021

b) Behavioral SystemVerilog module for a 2-to-4 decoder and a testbench for it.

```
module decoder24(
    input logic [1:0] a,
    input logic e,
    output logic [3:0] y
    );

    assign y[3] = e & a[1]  & a[0];
    assign y[2] = e & a[1]  & ~a[0];
    assign y[1] = e & ~a[1] & a[0];
    assign y[0] = e & ~a[1] & ~a[0];

endmodule
```

#Testbench

```
module decodertest();
    logic [1:0] a;
    logic e;
    logic [3:0] y;

    decoder24 dut(a[1:0], e, y[3:0]);
        initial begin
        e = 0; a[1] = 0; a[0] = 0; #10;
        e = 0; a[1] = 0; a[0] = 1; #10;
        e = 0; a[1] = 1; a[0] = 0; #10;
        e = 0; a[1] = 1; a[0] = 1; #10;
        e = 1; a[1] = 0; a[0] = 0; #10;
        e = 1; a[1] = 0; a[0] = 1; #10;
        e = 1; a[1] = 1; a[0] = 0; #10;
        e = 1; a[1] = 1; a[0] = 1; #10;
        end

endmodule
```

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

```
module mux21(
    input logic s,
    input logic [1:0] d,
    output logic y
    );

    assign y = d[0] & ~s | s & d[1];

endmodule
```

d) Structural SystemVerilog module for a 4-to-1 multiplexer by using three 2-to-1 multiplexers

```
module mux41(
input logic [1:0]s, [3:0]d,
output logic y
);

    logic [1:0]o;

    mux21 mux1(s[1], d[1:0], o[0]);
    mux21 mux2(s[1], d[3:2], o[1]);
    mux21 mux3(s[0], o[1:0], y);


endmodule
```

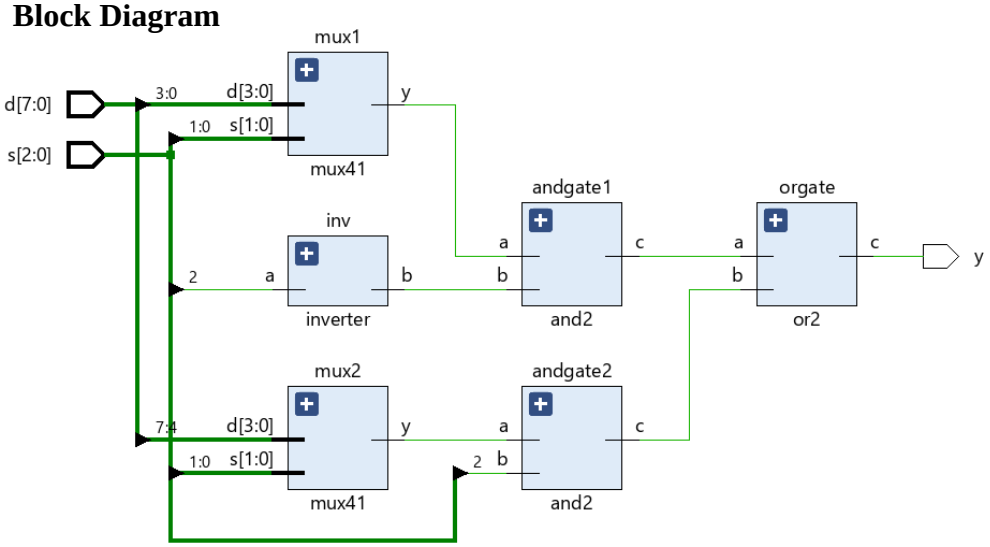
Testbench

```
module mux41tb();
 logic [1:0]s;
 logic [3:0]d;
 logic y;

    mux41 dut(s[1:0], d[3:0], y);
        initial begin
            s[0] = 0; s[1] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 0;#10;
            s[0] = 0; s[1] = 0; d[3] = 0; d[2] = 0; d[1] = 0; d[0] = 1;#10;
            s[0] = 0; s[1] = 0; d[3] = 0; d[2] = 0; d[1] = 1; d[0] = 0;#10;
            s[0] = 0; s[1] = 0; d[3] = 0; d[2] = 0; d[1] = 1; d[0] = 1;#10;
            s[0] = 0; s[1] = 1; d[3] = 0; d[2] = 1; d[1] = 0; d[0] = 0;#10;
            s[0] = 0; s[1] = 1; d[3] = 0; d[2] = 1; d[1] = 0; d[0] = 1;#10;
            s[0] = 0; s[1] = 1; d[3] = 0; d[2] = 1; d[1] = 1; d[0] = 0;#10;
            s[0] = 0; s[1] = 1; d[3] = 0; d[2] = 1; d[1] = 1; d[0] = 1;#10;
            s[0] = 1; s[1] = 0; d[3] = 1; d[2] = 0; d[1] = 0; d[0] = 0;#10;
            s[0] = 1; s[1] = 0; d[3] = 1; d[2] = 0; d[1] = 0; d[0] = 1;#10;
            s[0] = 1; s[1] = 0; d[3] = 1; d[2] = 0; d[1] = 1; d[0] = 0;#10;
            s[0] = 1; s[1] = 0; d[3] = 1; d[2] = 0; d[1] = 1; d[0] = 1;#10;
            s[0] = 1; s[1] = 1; d[3] = 1; d[2] = 1; d[1] = 0; d[0] = 0;#10;
            s[0] = 1; s[1] = 1; d[3] = 1; d[2] = 1; d[1] = 0; d[0] = 1;#10;
            s[0] = 1; s[1] = 1; d[3] = 1; d[2] = 1; d[1] = 1; d[0] = 0;#10;
            s[0] = 1; s[1] = 1; d[3] = 1; d[2] = 1; d[1] = 1; d[0] = 1;#10;
        end

endmodule
```

e) Structural System Verilog module of 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate.

```
module and2(input logic a,b, output logic c);
    assign c = a & b;
endmodule

module or2(input logic a,b, output logic c);
    assign c = a | b;
endmodule

module inverter(input logic a, output logic b);
   assign b = ~a;
 endmodule

module mux81(
    input logic [7:0]d, [2:0]s,
    output logic y
    );

    logic o1,o2,o3,o4,o5;

    mux41 mux1(s[1:0], d[3:0],o1);
    mux41 mux2(s[1:0], d[7:4],o2);
    inverter inv(s[2],o3);
    and2 andgate1(o1,o3, o4);
    and2 andgate2(o2,s[2],o5);
    or2 orgate(o4,o5,y);

endmodule
```

Block Diagram



f) SystemVerilog module for F(A,B,C,D)=Σ(2,5,6,7,10,12,13,15) function, using one 8-to-1 multiplexer and an Inverter.

```
module inverter_gate(input logic a, output logic b);
   assign b = ~a;
 endmodule


module abcdmux(
    input logic a, b, c, d0,
    output logic y
    );
    logic d;
    inverter_gate inv(d0,d);
    mux81 mux({0,d,d0,1,0,d,1,d},{a,b,c},y);

endmodule
```

Block Diagram