CS 223 Digital Design

Bilkent University
Fall 2023-2023

Laboratory Assignment 3
Preliminary Report

Section 1

Mehmet Akif Şahin
22203673

Sun. October 15th, 2023

b) [5] Behavioral SystemVerilog module for 2-to-4 decoder and a testbench for it.

Module

```
module decoder_2to4_behav(
        input a[1:0],
        output y[3:0]
        );
        assign y[0] = ~a[1] & ~a[0];
        assign y[1] = ~a[1] & a[0];
        assign y[2] = a[1] & ~a[0];
        assign y[3] = a[1] & a[0];
endmodule
```

Testbench

```
module tb_decoder_2to4(

        );
        reg a[1:0];
        wire y[3:0];
        decoder_2to4_behav uut (a[1:0], y[3:0]);

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

c) [5] Behavioral SystemVerilog module for 2-to-1 multiplexer.

Module

```
module mux_2to1_behav(
        input a[1:0],
        input s,
        output y
        );

        assign y = a[1] & s || a[0] && ~s;
endmodule
```

Testbench

```
module tb_mux_2to1(

        );
        reg a[1:0], s;
        wire y;

        mux_2to1_behav uut (a[1:0], s, y);

        initial begin
        a[0] = 0;
        a[1] = 0;
        s = 0;
        #10;
        a[0] = 0;
        a[1] = 0;
        s = 1;
        #10;
        a[0] = 1;
        a[1] = 0;
        s = 0;
        #10;
        a[0] = 1;
        a[1] = 0;
        s = 1;
        #10;
        a[0] = 0;
        a[1] = 1;
        s = 0;
        #10;
        a[0] = 0;
        a[1] = 1;
        s = 1;
        #10;
        a[0] = 1;
        a[1] = 1;
        s = 0;
        #10;
        a[0] = 1;
        a[1] = 1;
        s = 1;
        #10;
        end
endmodule
```

d) [5] Behavioral SystemVerilog module for 4-to-1 multiplexer by using three 2-to-1 multiplexer.

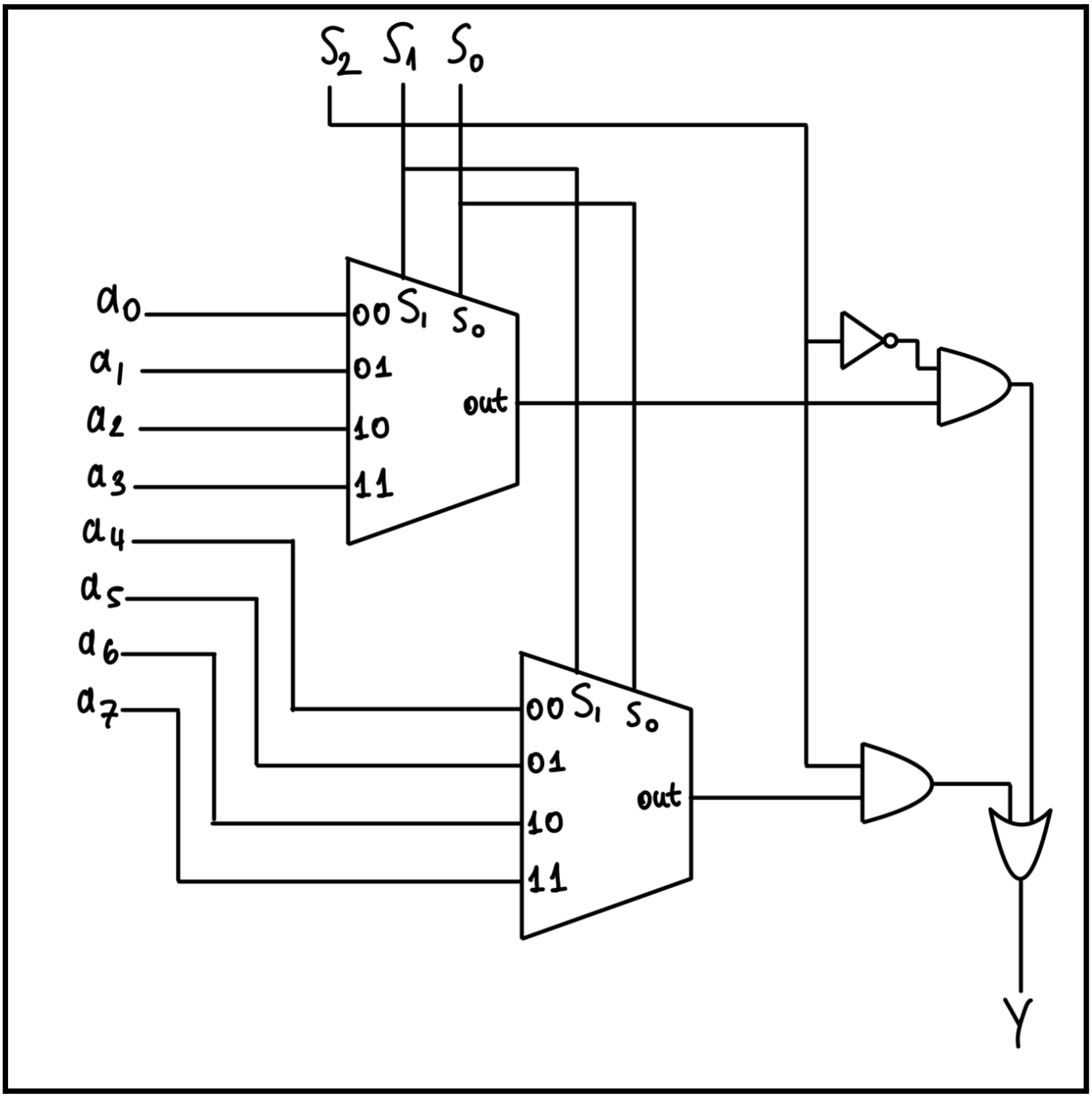
Module

```
module mux_4to1(
        input a[3:0],
        input s[1:0],
        output y
        );

        wire y_intermediate[1:0];

        mux_2to1_behav mux1 (a[1:0], s[0], y_intermediate[0]);
        mux_2to1_behav mux2 (a[3:2], s[0], y_intermediate[1]);

        mux_2to1_behav mux3 (y_intermediate[1:0], s[1], y);

endmodule
```

Testbench

```
module tb_mux_4to1(

        );

        reg a[3:0], s[1:0];
        wire y;

        mux_4to1 uut (a[3:0], s[1:0], y);

        initial begin
        for (int i = 0; i < 2 ** 6; i = i + 1) begin
        {a[3], a[2], a[1], a[0], s[1], s[0]} = i;
        #10;
        end
        end
endmodule
```

e) [5] Schematic (block diagram) and 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. Prepare a test bench for it.



Module

```
module mux_8to1(
        input a[7:0],
        input s[2:0],
        output y
        );

        reg muxouts[1:0], s2_complement, andouts[1:0];
        mux_4to1 mux1 (a[3:0], s[1:0], muxouts[0]);
        mux_4to1 mux2 (a[7:4], s[1:0], muxouts[1]);

        not (s2_complement, s[2]);
        and (andouts[0], s2_complement, muxouts[0]);
        and (andouts[1], s[2], muxouts[1]);
        or ( y, andouts[0], andouts[1]);
endmodule
```

Testbench

```
module tb_mux_8to1(

        );

        reg a[7:0], s[2:0];
        wire y;

        mux_8to1 uut (a[7:0], s[2:0], y);

        initial begin
        for (int i = 0; i < 2 ** 11; i = i + 1) begin
                {a[7], a[6], a[5], a[4], a[3], a[2], a[1], a[0], s[2], s[1], s[0]} =
i;
                #10;
        end
        end
endmodule
```

f) [10] Schematic (block diagram) and SystemVerilog module for F(A,B,C,D,E)=∑(2,4,5,9,10,11,13,18,23,24,26,27) function (note that there are 12 terms), using one (not two) 8-to-1 multiplexer and a 2-to-4 decoder. Do not use any other gates (AND, OR, XOR, etc.). Your design must contain only one 8-to-1 multiplexer and one 2-to-4 decoder. Complements of signals are available.

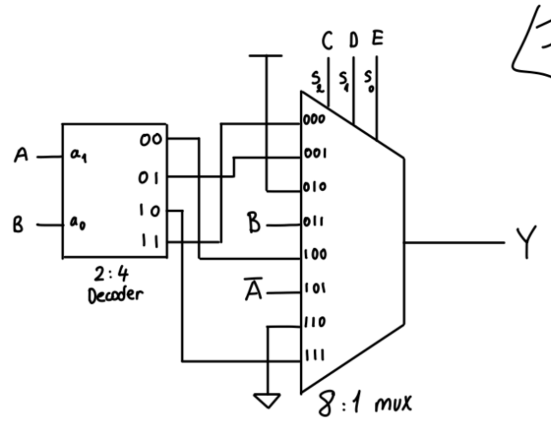
	A	B	C	D	E	Y
24	1	1	0	0	0	1
9	0	1	0	0	1	1
2	0	0	0	1	0	1
10	0	1	0	1	0	1
18	1	0	0	1	0	1
26	1	1	0	1	0	1
11	0	1	0	1	1	1
27	1	1	0	1	1	1
4	0	0	1	0	0	1
5	0	0	1	0	1	1
13	0	1	1	0	1	1
23	1	0	1	1	1	1

2:4 Decoder with A and B

A	B	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Truth Table of F(A,B,C,D,E)

C	D	E	Y
0	0	0	D_3
0	0	1	D_1
0	1	0	1
0	1	1	B
1	0	0	D_0
1	0	1	$\bar{A}$
1	1	0	0
1	1	1	D_2



Module

```
module my_function(
        input a, b, c, d, e,
        output y
        );
```

```
	wire d_out[3:0];
	reg mux_in[7:0];

	decoder_2to4_behav decoder (.a({a, b}), .y(d_out));

	assign mux_in = {d_out[2], 1'b0, ~a, d_out[0], b, 1'b1, d_out[1],
d_out[3]};

	mux_8to1 multiplexer (.a(mux_in), .s({c, d, e}), .y(y));
endmodule
```