CS223: Digital Design

Section 01

Lab 3

Efe Beydoğan

21901548

23.10.2020

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

Behavioral SystemVerilog module for 2-to-4 decoder:

```
module decoder2_to_4( input logic a[1:0], output logic y[3:0]);

   assign y[3] = a[0] & a[1];

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

   assign y[1] = a[0] & ~a[1];

   assign y[0] = ~a[0] & ~a[1];

endmodule
```

Testbench for the 2-to-4 decoder:

```
module decoder2_to_4_testbench();

   logic a[1:0];

   logic y[3:0];


   decoder2_to_4 dut( a[1:0], y[3:0]);


   initial begin

      a[0] = 0; a[1] = 0; #10;

      if (y[0] != 1 | y[1] != 0 | y[2] != 0 | y[3] != 0) $display("00 failed.");

      a[0] = 1; #10;

      if (y[0] != 0 | y[1] != 1 | y[2] != 0 | y[3] != 0) $display("10 failed.");

      a[0] = 0; a[1] = 1; #10;

      if (y[0] != 0 | y[1] != 0 | y[2] != 1 | y[3] != 0) $display("01 failed.");

      a[0] = 1; #10;

      if (y[0] != 0 | y[1] != 0 | y[2] != 0 | y[3] != 1) $display("11 failed.");

      end

endmodule
```

c) Behavioral SystemVerilog module for 4-to-1 multiplexer:

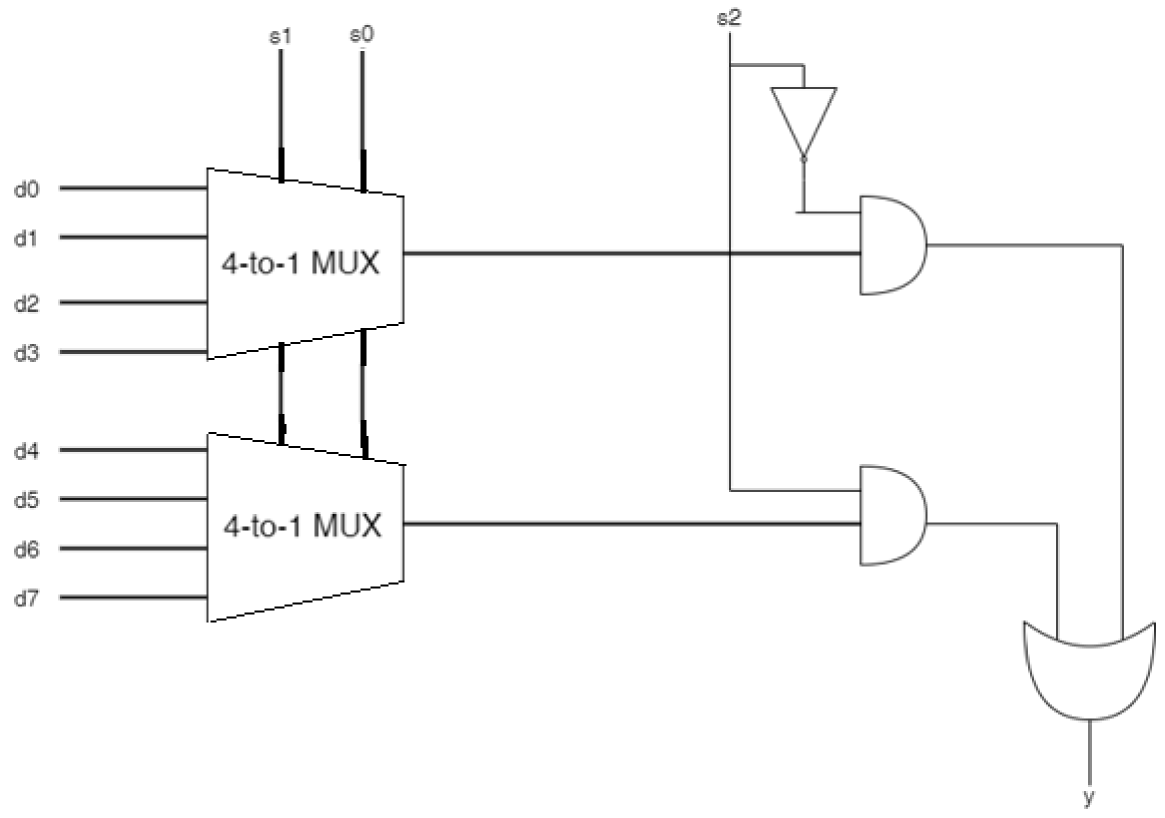
```
module mux4_to_1( input logic [3:0]d, [1:0]s, output logic y);

   assign y = s[1] ? (s[0] ? d[3] : d[2]) : ( s[0] ? d[1] : d[0]);

endmodule
```

Testbench:

```
module mux4_to_1_testbench();

   logic [3:0]d;

   logic [1:0]s;

   logic y;

   mux4_to_1 dut( d[3:0], s[1:0], y);

   initial begin

      for ( int i = 0; i < 64; i++) begin

         {d[3:0], s[1:0]} = i; #0.1;

         if ( y != d[s]) $display("%0d failed", {d,s});

       end

    end

endmodule
```

d) 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:

Schematic:



Structural SystemVerilog module of 8-to-1 MUX:

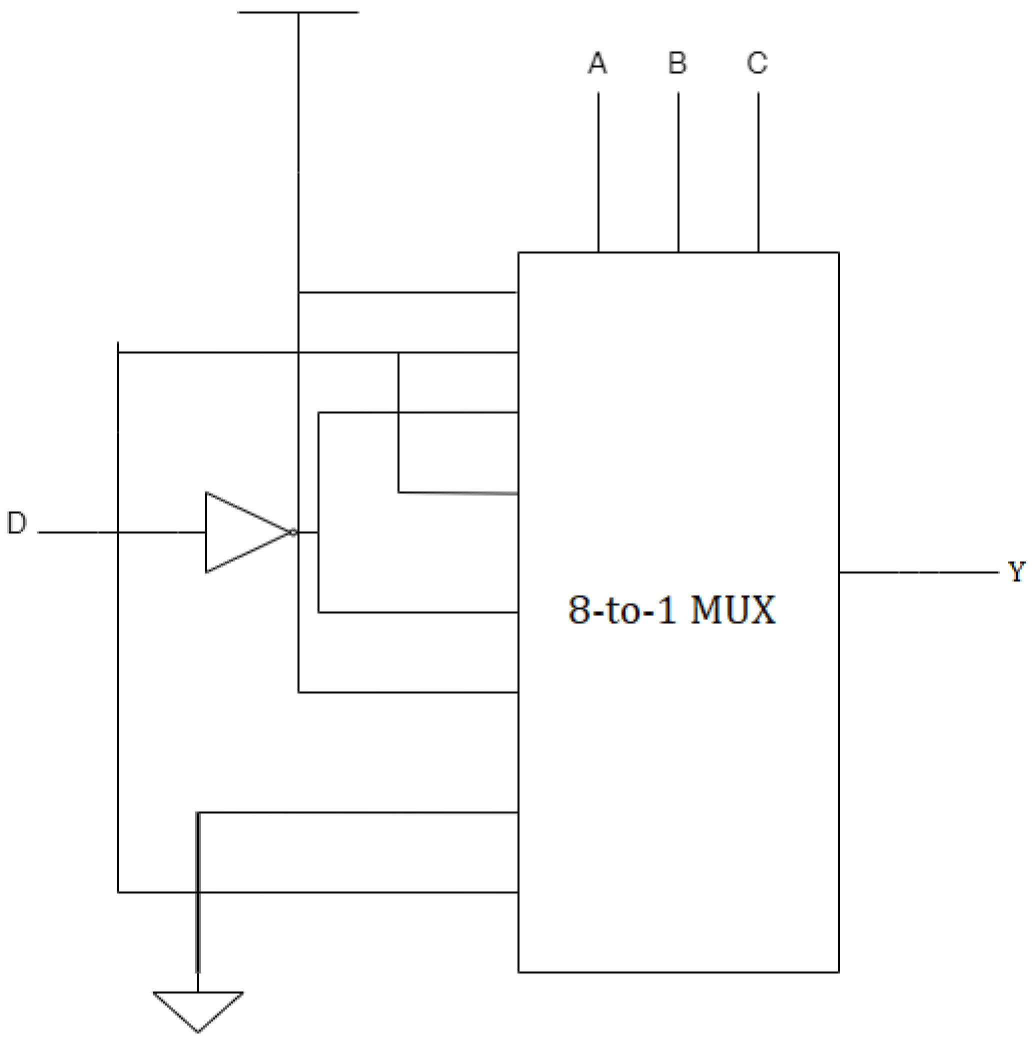
```
module mux8_to_1( input logic [7:0]d, [2:0]s, output logic y);
   logic output1, output2;
   logic n1, n2;


   mux4_to_1 mux1( d[3:0], s[1:0], output1);
   mux4_to_1 mux2( d[7:4], s[1:0], output2);


   and( n1, output1, ~s[2]);
   and( n2, output2, s[2]);
   or( y, n1, n2);
endmodule
```

Testbench:

```
module mux8_to_1_testbench();

   logic [7:0]d;

   logic [2:0]s;

   logic y;


   mux8_to_1 dut( d[7:0], s[2:0], y);

   initial begin

      for ( int i = 0; i < 2048; i++) begin

         {d[7:0], s[2:0]} = i; #0.1;

         if ( y != d[s]) $display("%0d failed", {d,s});

       end

    end

endmodule
```

e) Schematic (block diagram) and SystemVerilog module for F(A,B,C,D)=∑(0,1,3,4,7,8,10,11,15) function, using one (not two) 8-to-1 multiplexer and an inverter:

Schematic:



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

```
module partE( input logic a, b, c, d, output logic y);
   mux8_to_1 mux1( {d, 1'b0, 1'b1, ~d, d, ~d, d, 1'b1}, {a, b, c}, y);
endmodule
```