CS 223 Digital Design

Section 5 Lab 3

Ömer Oktay Gültekin

21901413

01.11.2021

[illegible]

* Behavioral SystemVerilog Module for a 2-to-4 decoder and a testbench for it

```
'timescale 1ns/1ps
module  decoder2to4(
     input logic [1:0] d,
     input logic e,
     output logic [3:0] y
     );

     assign y[0] = ~d[0] & ~d[1] & e;
     assign y[1] = d[0] &~d[1] & e;
     assign y[2] =~d[0] & d[1] & e;
     assign y[3] = d[0] & d[1] & e;
endmodule
```

• Testbench:

```
  module decoder_test_bench();
     logic [1:0] d;
     logic e;
     logic [3:0] y;
     decoder2to4 dec2(d,e, y);
     initial begin
        d=0; e=1; #10;
        d[0]=1; #10;
        d[1]=1; d[0]=0; #10;
        d[0] = 1; #10;
        e = 0; #10;
        d[1]= 0; #10;
        d[0]=0; #10;
        d[1]= 1; #10;
```

* Behavioral SystemVerilog module for a 2-to-1 multiplexer

```
`timescale 1ns/1ps
module mux2to1( input logic [1:0] d,
                input logic s,
                output logic y);
    assign y = s ? d[1] : d[0];
endmodule
```

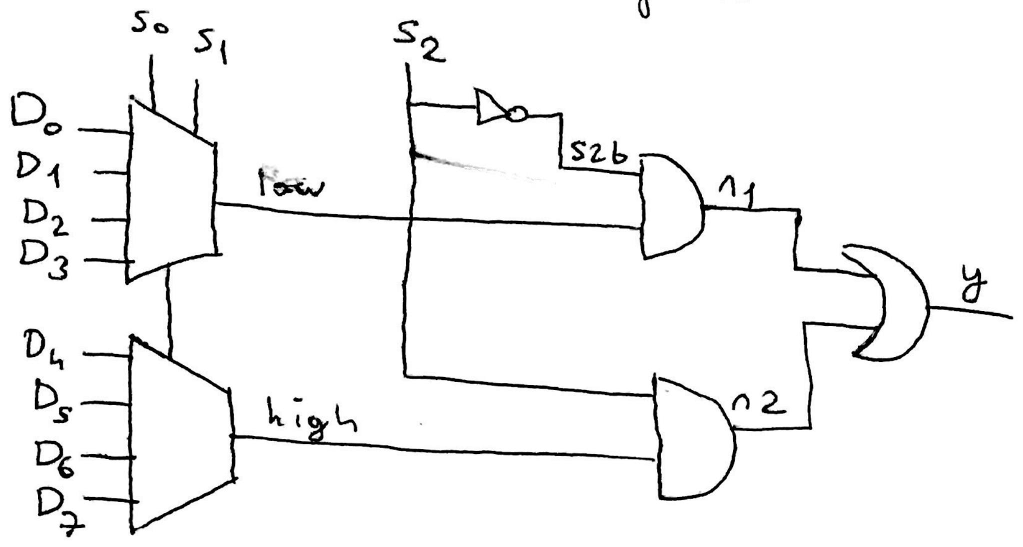
* Structural SystemVerilog Module for a 4-to-1 multiplexer by using three 2-to-1 multiplexers, and Testbench for it.

```
`timescale 1ns/1ps
module mux4to1( input logic [3:0] d,
                input logic [1:0] s,
                output logic y):
    logic [1:0] middle;
    mux2to1 lowmux2 ( d[1;0], s[0], middle[0]);
    mux 2to1 highmux2 ( d[3:2], s[0], middle[1]);
    mux 2to1 finalmux2(middle[1:0], s[1], y);
endmodule
```

• Testbench

```
`timescale 1ns / 1ps
module mux4to1_test_bench();
    logic [3:0] d;
    logic [1:0] s;
    logic y;
    mux4to1 mux4(d, s, y);
    initial begin
        s=0; d= 1; #10;
        d[0] =0; #10;
        s[0]=1; d[0]= 1; #10;
        d[1] =0; #10;
        s[1]=1; s[0]=0; d[1] =1; #10;
        d[2]= 0; # 10;
        s[0]= 1; d[2] = 1; #10;
        d[3] = 0; #10;
    end
endmodule
```

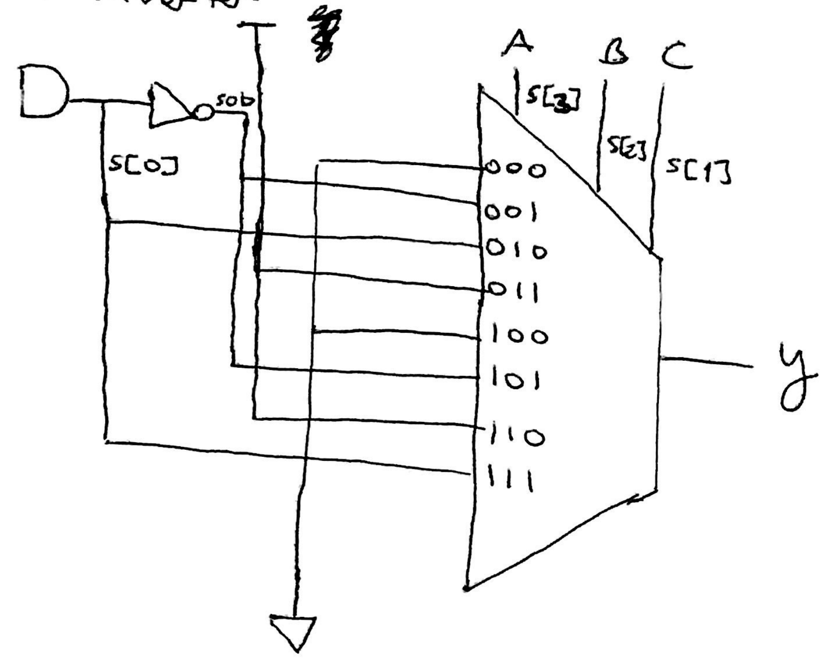
* Block diagram and structural System Verilog Module of 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an Inverter, and an OR gate.



• SystemVerilog Module

```
Module  Mux8to1(
        input logic [7:0] d,
        input logic [2:0] s,
        output logic y);
        logic low, high, n1, n2, s2b;
       Mux4to1  lowMux4 (d[3:0], s[1:0], low);
      Mux4to1  highMux4 (d[7:4], s[1:0], high);
      inv  inv1(s[2], s2b);
     and2  andfirst(low, s2b, n1);
     and2  ~~andsecond~~andsecond(high s[2], n2);
    or2   or1(n1, n2, y);
endmodule
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  inv(input logic a, output logic b);
   assign  b = ~a;
endmodule
```

* Block diagram and 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



• SystemVerilog Module

```
module custom_module(
        input logic [3:0] s,
        output logic y);
      logic [7:0] d;
      logic s0b;
      inv inv1(s[0], s0b);
      assign d[0] = 0;
      assign d[1] = s0b;
      assign d[2] = s[0]
      assign d[3] = 1;
      assign d[4] = 0;
      assign d[5] = s0b;
      assign d[6] = 1;
      assign d[7] = s[0];
      Mux8to1 Mux8(d, s[3:1], y);
endmodule

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