Course:	CS 223 Digital Design
Name:	Yahya Mahmoud Ahmed Elnouby Mohamed
Section:	02
ID:	21801332
Date:	5/11/2019
Trainer No.:	

Behavioral SystemVerilog module for 2-to-4 decoder

```
module decoder2to4( input logic i0,i1, output logic z0,z1,z2,z3);
reg z0,z1,z2,z3;
always@(i0,i1)
begin
   if(i0 == 0 && i1==0)
   begin
      z0<=1; z1<=0;
      z2<=0; z3<=0;
   end
   else if (i0==0 && i1==1)
   begin
      z0<=0; z1<=1;
      z2<=0; z3<=0;
   end
   else if (i0==1 && i1==0)
   begin
      z0<=0; z1<=0;
      z2<=1; z3<=0;
   end
   else if (i0==1 && i1==1)
   begin
      z0<=0; z1<=0;
      z2<=0; z3<=1;
   end
end
endmodule
```

TestBench for the 2-to-4 decoder

```
module decoder2to4_tb;
reg i0,i1;
wire z0,z1,z2,z3;

decoder2to4 uut (.i0(i0),.i1(i1),.z0(z0),.z1(z1),.z2(z2),.z3(z3));
initial begin
#100

#10; i0=0; i1=0;
#10; i0=0; i1=1;
#10; i0=1; i1=0;
#10; i0=1; i1=1;
end
endmodule
```

Behavioral SystemVerilog module for 2-to-1 multiplexer

```
module mux2(i1, i0, s, d);
 input i1, i0, s;
output d;
 reg d;
always @(i1, i0, s) begin
if(s==0) d <= i0;
else d <= i1;
 end
 endmodule
```

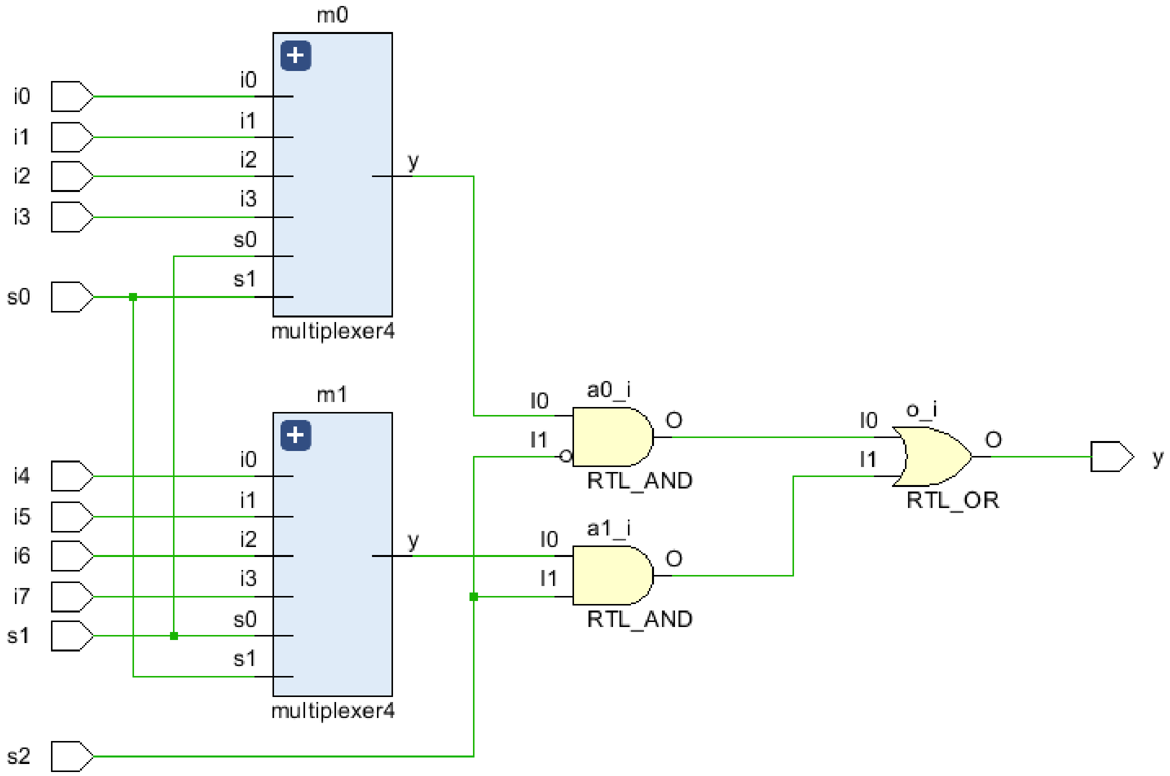
Structural SystemVerilog module for 4-to-1 multiplexer by using three 2-to-1 multiplexer

```
module multiplexer4(input logic i0,i1,i2,i3,s0,s1, output logic y);

reg y;
wire w0,w1;

multiplexer2 m0 (i0,i1,s1,w0);
multiplexer2 m1 (i2,i3,s1,w1);
multiplexer2 m2 (w0,w1,s0,y);
endmodule
```

Schematic of 8-to-1 MUX by using two 4-to-1 MUX modules, two AND gates, an INVERTER, and an OR gate



Structural SystemVerilog

```
module multiplexer8(
input logic i0,i1,i2,i3,i4,i5,i6,i7,s0,s1,s2,
output logic y );
reg y;
wire w0,w1,w2,w3,w4;
multiplexer4 m0 (i0,i1,i2,i3,s1,s0,w0);
multiplexer4 m1 (i4,i5,i6,i7,s1,s0,w1);
not         n  (w2,s2);
and         a0 (w3,w0,w2);
and         a1 (w4,w1,s2);
or          o  (y,w3,w4);
endmodule
```

TestBench

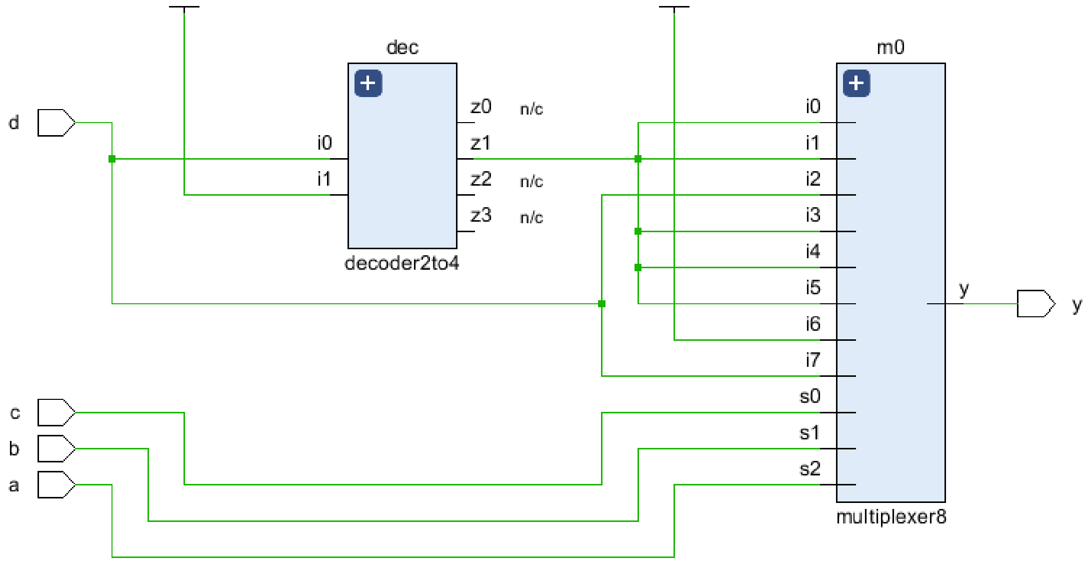
```
module multiplixer8_tb;
reg i0,i1,i2,i3,i4,i5,i6,i7,s0,s1,s2;
wire y;

//instantiate
multiplexer8
uut(.i0(i0),.i1(i1),.i2(i2),.i3(i3),.i4(i4),.i5(i5),.i6(i6),.i7(i7),.s0(s0),.s1(s1),.s2(s2),.y(y));
initial begin
#100
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=0; s1=0; s2=0;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=0; s1=0; s2=1;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=0; s1=1; s2=0;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=0; s1=1; s2=1;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=1; s1=0; s2=0;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=1; s1=0; s2=1;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=1; s1=1; s2=0;
#10; i0=0; i1=1; i2=0; i3=1; i4=1; i5=0; i6=1; i7=0; s0=1; s1=1; s2=1;

end

endmodule
```

Schematic of the function F(A,B,C,D)=∑(0,2,5,6,8,10,12,13,15)



SystemVerilog module for F(A,B,C,D)=∑(0,2,5,6,8,10,12,13,15) function

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

wire dnot, w1, w2, w3;
decoder2to4 decoder(d, 1, w1, dnot,w2,w3);
multiplexer8 m0 (dnot, dnot, d, dnot, dnot,dnot,1 ,d, c, b, a, y);

endmodule
```

Testbench for the function

```
module functionMux_tb;
reg a, b, c, d;
wire y;

//instantiate
functionMux uut (.a(a),.b(b),.c(c),.d(d),.y(y));
initial begin
#100
#10; a=0; b=0; c=0; d=0;
#10; a=0; b=0; c=0; d=1;
#10; a=0; b=0; c=1; d=0;
#10; a=0; b=0; c=1; d=1;
#10; a=0; b=1; c=0; d=0;
#10; a=0; b=1; c=0; d=1;
#10; a=0; b=1; c=1; d=0;
#10; a=0; b=1; c=1; d=1;
#10; a=1; b=0; c=0; d=0;
#10; a=1; b=0; c=0; d=1;
#10; a=1; b=0; c=1; d=0;
#10; a=1; b=0; c=1; d=1;
#10; a=1; b=1; c=0; d=0;
#10; a=1; b=1; c=0; d=1;
#10; a=1; b=1; c=1; d=0;
#10; a=1; b=1; c=1; d=1;
end
endmodule
```