

LAB NO 10



Fall 2024

CSE-304L Computer Organization and Architecture Lab

Submitted by:

Name : **Hassan Zaib Jadoon**

Reg no. : **22PWCSE2144**

Class Section : **A**

Signature: _____

Submitted to:

Dr. Amaad Khalil

Department of Computer Systems Engineering
University of Engineering and Technology, Peshawar

MULTIPLEXER

TASK 01:

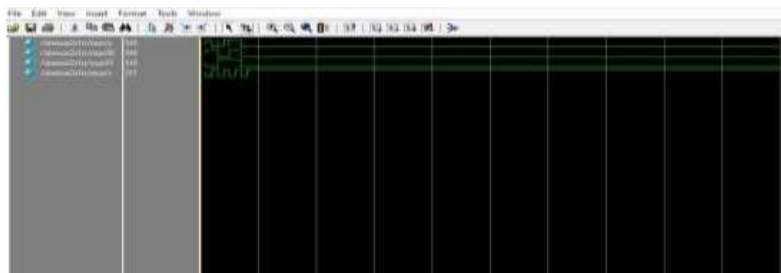
Write a Verilog code for 2x1 Mux using Dataflow Level modeling.

```
1 module mux_2to1(a,b,sel,f);
2 output f;
3 input a,b,sel;
4 and g1(f1,a,nsel);
5 and g2(f2,b,sel);
6 or g3(f,f1,f2);
7 not g4(nsel,sel);
8 endmodule
9
```

```
1 module stimN();
2 reg A,B,SEL;
3 wire F;
4 mux_2to1 M(A,B,SEL,F);
5 initial
6 begin
7   $display("A B SEL F");
8   A=0; B=0;
9   SEL=0;
10  #1 $display("%b %b %b %b",A,B,SEL,F);
11  #5 A=0; B=0;
12  SEL=1;
13  $display("%b %b %b %b",A,B,SEL,F);
14  #10 A=0; B=1;
15  SEL=0;
16  $display("%b %b %b %b",A,B,SEL,F);
17  #15 A=0; B=1;
18  SEL=1;
19  $display("%b %b %b %b",A,B,SEL,F);
20  #20 A=1; B=0;
21  SEL=0;
22  $display("%b %b %b %b",A,B,SEL,F);
23  # 25 A=1; B=0;
24  SEL=1;
25  $display("%b %b %b %b",A,B,SEL,F);
26  #30 A=1; B=1;
27  SEL=0;
28  #35 $display("%b %b %b %b",A,B,SEL,F);
29  A=1; B=1;
30  SEL=1;
31  $display("%b %b %b %b",A,B,SEL,F);
32 end
33 endmodule
```

Output:

```
vsim work.stimN
# vsim work.stimN
# Loading work.stimN
# Loading work.mux_2to1
run -all
# A B SEL F
# 0 0 0 0
# 0 0 1 0
# 0 1 0 0
# 0 1 1 0
# 1 0 0 1
# 1 0 1 1
# 1 1 0 1
# 1 1 1 1
VSIM 10>
```



Task 02:

Write a Verilog code for 4x1 Mux using Dataflow Level modeling.

```

1 module mux_4to1(a, b, c, d, sel, f);
2     input a, b, c, d;
3     input [1:0] sel;
4     output f;
5
6     wire nsel1, nsel0;
7     wire t1, t2, t3, t4;
8
9     not g1(nsel1, sel[1]);
10    not g2(nsel0, sel[0]);
11
12    and g3(t1, nsel1, nsel0, a);
13    and g4(t2, nsel1, nsel0, b);
14    and g5(t3, sel[1], nsel0, c);
15    and g6(t4, sel[1], sel[0], d);
16
17    or g7(f, t1, t2, t3, t4);
18 endmodule

```

```

1 module mux_4to1;
2     reg A, B, C, D;
3     reg [1:0] SEL;
4     wire F;
5     mux_4to1 H(A, B, C, D, SEL, F);
6
7     initial
8     begin
9         $display("A B C D SEL[1] SEL[0] F");
10
11        A = 0; B = 0; C = 0; D = 0; SEL = 2'b00; F = 0;
12        $display("00 00 00 00 00", A, B, C, D, SEL[1], SEL[0], F);
13
14        A = 0; B = 0; C = 0; D = 1; SEL = 2'b00; F = 0;
15        $display("00 00 00 00 01", A, B, C, D, SEL[1], SEL[0], F);
16
17        A = 0; B = 0; C = 1; D = 0; SEL = 2'b00; F = 0;
18        $display("00 00 01 00 00", A, B, C, D, SEL[1], SEL[0], F);
19
20        A = 0; B = 0; C = 1; D = 1; SEL = 2'b00; F = 0;
21        $display("00 00 01 01 00", A, B, C, D, SEL[1], SEL[0], F);
22
23        A = 0; B = 1; C = 0; D = 0; SEL = 2'b01; F = 0;
24        $display("00 01 00 00 00", A, B, C, D, SEL[1], SEL[0], F);
25
26        A = 0; B = 1; C = 0; D = 1; SEL = 2'b01; F = 0;
27        $display("00 01 00 01 00", A, B, C, D, SEL[1], SEL[0], F);
28
29        A = 0; B = 1; C = 1; D = 0; SEL = 2'b01; F = 0;
30        $display("00 01 01 00 00", A, B, C, D, SEL[1], SEL[0], F);
31
32        A = 0; B = 1; C = 1; D = 1; SEL = 2'b01; F = 0;
33        $display("00 01 01 01 00", A, B, C, D, SEL[1], SEL[0], F);
34
35        A = 1; B = 0; C = 0; D = 0; SEL = 2'b10; F = 0;
36        $display("01 00 00 00 00", A, B, C, D, SEL[1], SEL[0], F);
37
38        A = 1; B = 0; C = 0; D = 1; SEL = 2'b10; F = 0;
39        $display("01 00 00 01 00", A, B, C, D, SEL[1], SEL[0], F);
40
41        A = 1; B = 0; C = 1; D = 0; SEL = 2'b10; F = 0;
42        $display("01 00 01 00 00", A, B, C, D, SEL[1], SEL[0], F);
43
44        A = 1; B = 0; C = 1; D = 1; SEL = 2'b10; F = 0;
45        $display("01 00 01 01 00", A, B, C, D, SEL[1], SEL[0], F);
46
47        A = 1; B = 1; C = 0; D = 0; SEL = 2'b11; F = 0;
48        $display("01 01 00 00 00", A, B, C, D, SEL[1], SEL[0], F);
49
50        A = 1; B = 1; C = 0; D = 1; SEL = 2'b11; F = 0;
51        $display("01 01 00 01 00", A, B, C, D, SEL[1], SEL[0], F);
52
53        A = 1; B = 1; C = 1; D = 0; SEL = 2'b11; F = 0;
54        $display("01 01 01 00 00", A, B, C, D, SEL[1], SEL[0], F);
55
56        A = 1; B = 1; C = 1; D = 1; SEL = 2'b11; F = 0;
57        $display("01 01 01 01 00", A, B, C, D, SEL[1], SEL[0], F);
58
59        $stop;
60    end

```

Output:

Table:

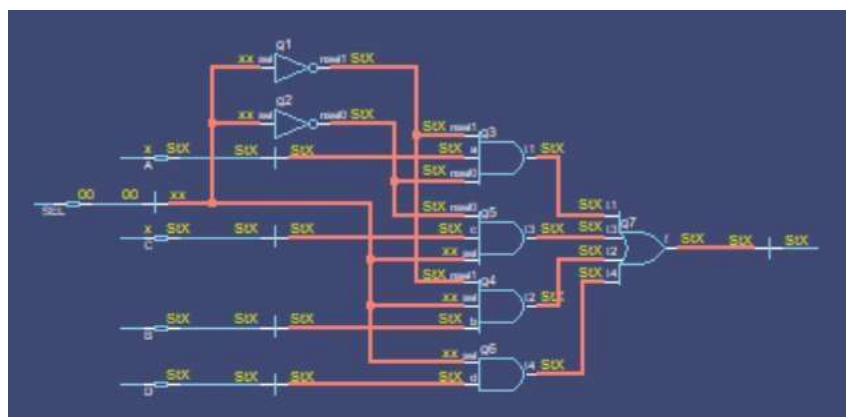
```

run -all
#A B C D SEL[1] SEL[0] F
#0000000
#0001010
#0100100
#0110110
#1000001
#1010010
#1100100
#1111111

VSIM 7>

```

Logical diagram:



Task3:

Write a Verilog code for 8x1 Mux using Dataflow Level modeling.

```

1 module stim1();
2   reg [7:0] A;           // 8-bit data input
3   reg [2:0] SEL;         // 3-bit select input
4   wire F;               // Output of the MUX
5
6   mux_8to1 M(A, SEL, F); // Instantiate the 8x1 MUX
7
8   initial begin
9     $display("A[7:0] SEL[2] SEL[1] SEL[0] F");
10
11     // Test case 1: Select d[0]
12     A = 8'b00000001; SEL = 3'b000; #1
13     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
14
15     // Test case 2: Select d[1]
16     A = 8'b00000010; SEL = 3'b001; #5
17     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
18
19     // Test case 3: Select d[2]
20     A = 8'b00000100; SEL = 3'b010; #10
21     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
22
23     // Test case 4: Select d[3]
24     A = 8'b00001000; SEL = 3'b011; #15
25     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
26
27     // Test case 5: Select d[4]
28     A = 8'b00010000; SEL = 3'b100; #20
29     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
30
31     // Test case 6: Select d[5]
32     A = 8'b00100000; SEL = 3'b101; #25
33     $display("%b %b %b %b %b", A, SEL[2], SEL[1], SEL[0], F);
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

1 module mux_8to1 (
2   input [7:0] A,           // 8 inputs
3   input [2:0] SEL,         // 3-bit select line
4   output F                // Output of the MUX
5 );
6   assign F = (SEL == 3'b000) ? A[0] :
7             (SEL == 3'b001) ? A[1] :
8             (SEL == 3'b010) ? A[2] :
9             (SEL == 3'b011) ? A[3] :
10            (SEL == 3'b100) ? A[4] :
11            (SEL == 3'b101) ? A[5] :
12            (SEL == 3'b110) ? A[6] :
13            (SEL == 3'b111) ? A[7] : 1'b0;
14 endmodule
15

```

Output:

Table:

```

run -all
# A[7:0] SEL[2] SEL[1] SEL[0] F
# 00000001 0 0 0 1
# 00000010 0 0 1 1
# 00000100 0 1 0 1
# 00001000 0 1 1 1
# 00010000 1 0 0 1
# 00100000 1 0 1 1
# 01000000 1 1 0 1
# 10000000 1 1 1 1

VSIM 8>

```