LAB #06

MULTIPLEXER IN VERILOG

Fall 2023

CSE-304L Computer Organization & Architecture

Submitted by: MUHAMMAD SADEEQ

Registration No.: 21PWCSE2028

Section: C

“On my honor, as a student of the University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work”

Submitted to:

Dr. Bilal Habib

(9 Dec 2023)

Department of Computer systems engineering
University of Engineering and Technology,
Peshawar

Tasks 1

Code:

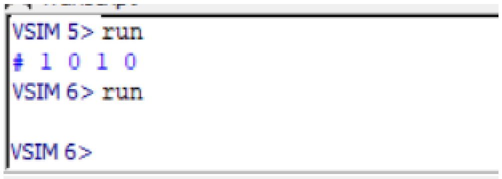
2X1 MUX:

```
module mux2x1(I,sel,out);

input[1:0]I; //I[0].I[1]
input sel;
output out;
reg out;
always @(I,sel)
case(sel)
            1'b0 : out = I[0];
            1'b1 : out = I[1];
endcase
endmodule
```

Test Bench:

```
module Tb();
  reg [1:0] I;
  reg sel;
  wire out;

  mux2x1 mux (I, sel, out);

  initial begin
    I[0] = 0;
    I[1] = 1;
    sel = 0;
    #10;

    I[0] = 1;
    I[1] = 0;
    sel = 1;
    #10;

    $monitor("%b %b %b %b ", I[0], I[1], sel, out);
  end
endmodule
```

Output:



```
VSIM 5> run
# 1 0 1 0
VSIM 6> run

VSIM 6>
```

Tasks 2

Code:

<u>4X1 MUX:</u>

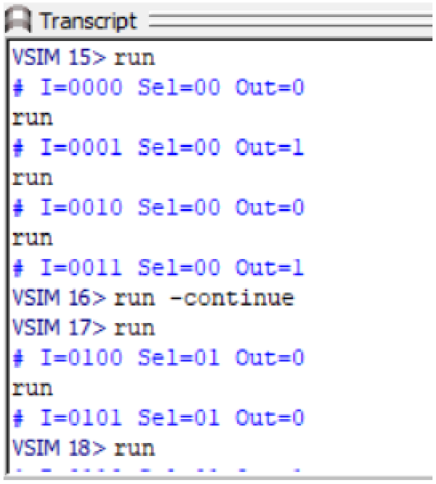
```
module mux4x1(I, Sel, Out);
input [3:0] I;
input [1:0] Sel;
output Out;
wire [1:0] s1, s2;
wire w1, w2;
mux2x1 m1(I[0],I[1], Sel[0], w1);
mux2x1 m2(I[2], I[3], Sel[0], w2);
mux2x1 m3(w1, w2, Sel[1], Out);
endmodule
```

<u>Test Bench:</u>

```
module test_mux4x1;
reg [3:0] I;
reg [1:0] Sel;
wire Out;

mux4x1 dut(.I(I), .Sel(Sel), .Out(Out));

initial begin
   $monitor("I=%b Sel=%b Out=%b", I, Sel, Out);
   I = 4'b0000; Sel = 2'b00; #100;
   I = 4'b0001; Sel = 2'b00; #100;
   I = 4'b0010; Sel = 2'b00; #100;
   I = 4'b0011; Sel = 2'b00; #100;
   I = 4'b0100; Sel = 2'b01; #100;
   I = 4'b0101; Sel = 2'b01; #100;
   I = 4'b0110; Sel = 2'b01; #100;
   I = 4'b0111; Sel = 2'b01; #100;
   I = 4'b1000; Sel = 2'b10; #100;
   I = 4'b1001; Sel = 2'b10; #100;
   I = 4'b1010; Sel = 2'b10; #100;
```

```
    I = 4'b1011; Sel = 2'b10; #100;
    I = 4'b1100; Sel = 2'b11; #100;
    I = 4'b1101; Sel = 2'b11; #100;
    I = 4'b1110; Sel = 2'b11; #100;
    I = 4'b1111; Sel = 2'b11; #100;
    $finish;
end

endmodule
```

Output:



```
VSIM 15> run
# I=0000 Sel=00 Out=0
run
# I=0001 Sel=00 Out=1
run
# I=0010 Sel=00 Out=0
run
# I=0011 Sel=00 Out=1
VSIM 16> run -continue
VSIM 17> run
# I=0100 Sel=01 Out=0
run
# I=0101 Sel=01 Out=0
VSIM 18> run
```

Tasks 3

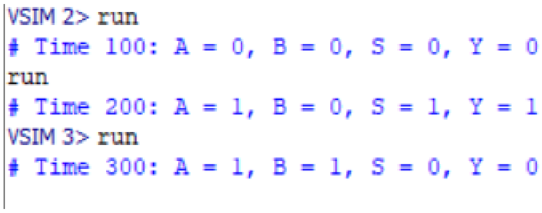
Code:

```
module mux_4to1 (A, B, S, out);
        input A, B;
        input [0:1]S;
        output reg out;
        always @(A, B, S)
        case (S)
           2'b00:out = A;
           2'b01:out = B;
           2'b10:out = A + B;
           default:out = A - B;
        endcase
endmodule
```

Test Bench:

```
module mux_4to1_tb;
  // Declare the inputs and outputs
  reg A, B, S;
  wire Y;
  // Instantiate the MUX module
  mux_4to1 uut(A,B,S,Y);
  // Initialize the inputs
initial begin
    A = 'b0; B = 'b0; S = 'b00; #100;
    $display("Time %0t: A = %b, B = %b, S = %b, Y = %b", $time, A, B, S, Y);
    A = 'b1; B = 'b0; S = 'b01; #100;
    $display("Time %0t: A = %b, B = %b, S = %b, Y = %b", $time, A, B, S, Y);
    A = 'b1; B = 'b1; S = 'b10; #100;
    $display("Time %0t: A = %b, B = %b, S = %b, Y = %b", $time, A, B, S, Y);
    A = 'b1; B = 'b0; S = 'b11; #100;
    $display("Time %0t: A = %b, B = %b, S = %b, Y = %b", $time, A, B, S, Y);
  end
endmodule
```

Output:



```
VSIM 2> run
# Time 100: A = 0, B = 0, S = 0, Y = 0
run
# Time 200: A = 1, B = 0, S = 1, Y = 1
VSIM 3> run
# Time 300: A = 1, B = 1, S = 0, Y = 0
```