Lab report 07

MULTIPLEXER IN VERILOG

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**304L-Computer Organization and Architecture Lab**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

Submitted to: **Dr. Bilal Habib**

Submitted by: **Muhammad Saad**

Reg ID: **21PWCSE1997**

**DCSE, Batch 23, Section “B”**

Date: **Sat, Dec 9, 2023**

**ASSESSMENT RUBRICS COA LABS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **LAB REPORT ASSESSMENT** | | | | |
| **Criteria** | **Excellent** | **Average** | **Nill** | **Marks Obtained** |
| 1. **Objectives of Lab** | All objectives of lab are properly covered  [Marks 10] | Objectives of lab are partially covered  [Marks 5] | Objectives of lab are not shown  [Marks 0] |  |
| 1. **MIPS instructions with**   **Comments and proper indentations.** | All the instructions are well written with comments explaining the code and properly indented  [Marks 20] | Some instructions are missing are poorly commented code  [Marks 10] | The instructions are not properly written  [Marks 0] |  |
| 1. **Simulation run without error and warnings** | The code is running in the simulator without any error and warnings  [Marks 10] | The code is running but with some warnings or errors.  [Marks 5] | The code is written but not running due to errors  [Marks 0] |  |
| 1. **Procedure** | All the instructions are written with proper procedure  [Marks 20] | Some steps are missing  [Marks 10] | steps are totally missing  [Marks 0] |  |
| 1. **OUTPUT** | Proper output of the code written in assembly  [Marks 20] | Some of the outputs are missing  [Marks 10] | No or wrong output  [Marks 0] |  |
| 1. **Conclusion** | Conclusion about the lab is shown and written  [Marks 20] | Conclusion about the lab is partially shown  [Marks 10] | Conclusion about the lab is not shown[Marks0]  [Marks 0] |  |
| 1. **Cheating** |  |  | Any kind of cheating will lead to 0 Marks |  |
| Total Marks Obtained:\_\_\_\_\_\_\_\_\_\_  Instructor Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | | | |

**Lab 07**

**MULTIPLEXER IN VERILOG**

# Objective:

* In this lab we will be implementing multiplexer using verilog.

# Questions:

#### 1: Implement 2X1 MUX in Verilog.

**module** mux2x1(I0, I1, Sel, Out);

**input** I0, I1;*//I1, I2 and Sel are inputs*

**input** Sel;

**output** Out;/Output

reg Out;

**always** @(I0, I1, Sel)

**case**(Sel)

1'b0:Out=I0;*//If Sel==0->Out=I0 else Out=I1*

1'b1:Out=I1;

**endcase**

**endmodule**

Verilog code for 2x1 Multiplexer

**module** stm\_mux2x1();

reg I1, I2;

reg Sel;

wire Out;

mux2x1 mux(I1, I2, Sel, Out);*//Intanciating the mux2x module*

**initial**

**begin**

#10

I1 = 1;

I2 = 0;

Sel = 0;

*//Testing for each case*

#10

I1 = 0;

I2 = 0;

Sel = 1;

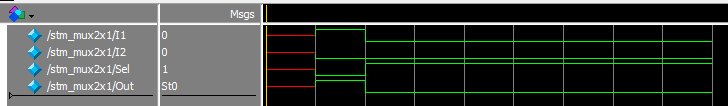
**end**

**initial**

$monitor("%d %b %b %b %b", $time, I1, I2, Sel, Out);

**endmodule**

TestBench for 2x1 Multiplexer

Fig 01: Output

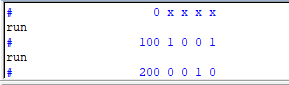


Fig 02: Output

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 2: Implement 4X1 MUX using 2X1 MUX.

**module** mux4x1(I, Sel, Out);

**input** [3:0] I;

**input** [1:0] Sel;

**output** Out;

wire [1:0] s1, s2;

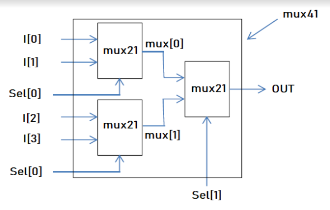
wire mux[0], mux[1];

mux2x1 m1(I[0],I[1], Sel[0], mux[0]);

mux2x1 m2(I[2], I[3], Sel[0], mux[1]);

mux2x1 m3(w1, w2, Sel[1], Out);

**endmodule**



4x1 Multiplexer using 2x1 Multiplexer Verilog code

**module** test\_mux4x1;

reg [3:0] I;

reg [1:0] Sel;

wire Out;

mux4x1 mux(.I(I), .Sel(Sel), .Out(Out));*//Instanciating module mux4x1*

**initial** **begin**

$monitor("I=%b Sel=%b Out=%b", I, Sel, Out);

I = 4'b0000; Sel = 2'b00; #100;*//Testing each case with 100ps delay*

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**

Test bench for 4x1 Multiplexer

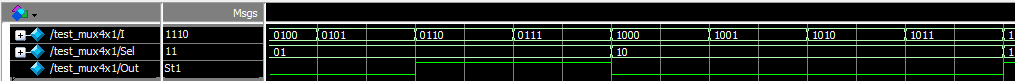


Fig 03: Output

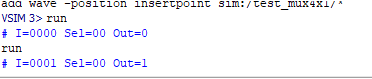


Fig 04: Output

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#### 3: Design MUX where if the select to a MUX is 00 it will output A , 01 the output will be B , for 10 the output will be A+B and for 11 the output will be A-B.

**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**

4 to 1 Multiplexer Verilog code

**module** mux\_4to1\_tb;

reg A, B, S;

wire Y;

mux\_4to1 mux(A,B,S,Y);

**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**

Test Bench

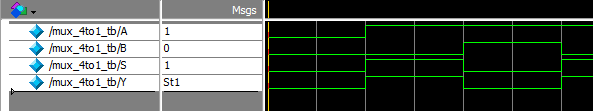


Fig 05: Output

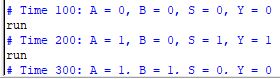


Fig 06: Output

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Conclusion:

In this lab we created Multiplexers and tested the output with the help of test bench.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**