Lab report 07

DECODERS 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**

**DECODERS IN VERILOG**

# Objective:

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

# Questions:

#### 1: Implement 2X4 Decoder in Verilog.

module decoder2x4(A, B, e, I);

input A, B, e;

output [3:0]I;

wire nA, nB;

not n1(nA, A);

not n2(nB, B);

and i1(I[0], nA, nB, e);

and i2(I[1], nA, B, e);

and i3(I[2], A, nB, e);

and i4(I[3], A, B, e);

endmodule

Verilog code for 2x4 Decoder

module decoder2x4TB();

reg A, B, e;

wire [3:0]I;

decoder2x4 d2x4(A, B, e, I);

initial begin

$display("A B e I0 I1 I2 I3");

e = 0; A = 0; B = 0; #100

$display("%b %b %b %b %b %b %b", A, B, e, I[0], I[1], I[2], I[3]);

e = 1; A = 0; B = 0; #100

$display("%b %b %b %b %b %b %b", A, B, e, I[0], I[1], I[2], I[3]);

e = 1; A = 0; B = 1; #100

$display("%b %b %b %b %b %b %b", A, B, e, I[0], I[1], I[2], I[3]);

e = 1; A = 1; B = 0; #100

$display("%b %b %b %b %b %b %b", A, B, e, I[0], I[1], I[2], I[3]);

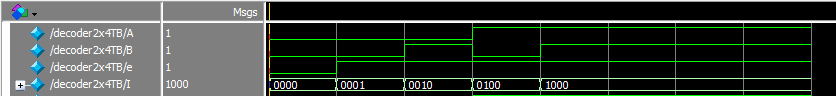
e = 1; A = 1; B = 1; #100

$display("%b %b %b %b %b %b %b", A, B, e, I[0], I[1], I[2], I[3]);

**end**

endmodule

TestBench for 2x1 Multiplexer

Fig 01: Output

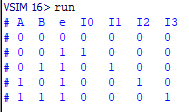


Fig 02: Output

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

#### 2: Implement 3X8 Decoder.

module d3x8(A, B, C, e, I);

input A, B, C, e;

output [0:7] I;

wire nA, nB, nC;

assign nA = ~A;

assign nB = ~B;

assign nC = ~C;

assign I[0] = nA & nB & nC & e;

assign I[1] = nA & nB & C & e;

assign I[2] = nA & B & nC & e;

assign I[3] = nA & B & C & e;

assign I[4] = A & nB & nC & e;

assign I[5] = A & nB & C & e;

assign I[6] = A & B & nC & e;

assign I[7] = A & B & C & e;

endmodule

3x8 Decoder Using Gates

module d3x8TB();

reg A, B, e, C;

wire [0:7]I;

d3x8 d3(A, B, C, e, I);

initial begin

$display("A B C e I0 I1 I2 I3 I4 I5 I6 I7");

e = 0;A = 0;B = 0;C = 0;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 0;B = 0;C = 0;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 0;B = 0;C = 1;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 0;B = 1;C = 0;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 0;B = 1;C = 1;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 1;B = 0;C = 0;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 1;B = 0;C = 1;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7])

e = 1;A = 1;B = 1;C = 0;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

e = 1;A = 1;B = 1;C = 1;#100

$display("%b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7]);

**end**

endmodule

Test bench for 3x8 Decoder

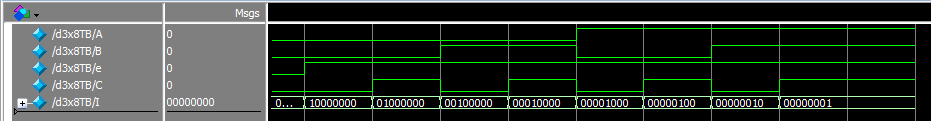


Fig 03: Output

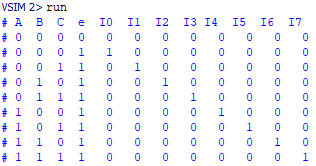


Fig 04: Output

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

#### 3: Design 3x8 Decoder using 2x4 Decoders.

module decoder3x8(A, B, e, I1, I2);

input A, B, e;

output [3:0]I1, I2;

wire ne;

not n1(ne, e);

decoder2x4 d1(A, B, ne, I1);

decoder2x4 d2(A, B, e, I2);

endmodule

3x8 using 2x4 Decoders Verilog code

module decoder3x8TB();

reg A, B, e;

wire [3:0]I1, I2;

decoder3x8 d4(A, B, e, I1, I2);

initial begin

$display("A B e I0 I1 I2 I3 I4 I5 I6 I7");

e = 0;A = 0;B = 0;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 0;A = 0;B = 1;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 0;A = 1;B = 0;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 0;A = 1;B = 1;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 1;A = 0;B = 0;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 1;A = 0;B = 1;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 1;A = 1;B = 0;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

e = 1;A = 1;B = 1;#40

$display("%b %b %b %b %b %b %b %b %b %b %b", A, B, e, I1[0], I1[1], I1[2], I1[3], I2[0], I2[1], I2[2], I2[3]);

**end**

endmodule

Test Bench

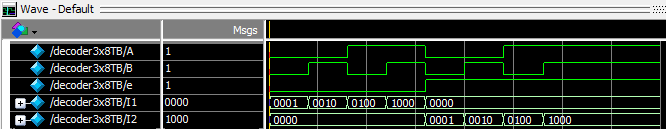


Fig 05: Output

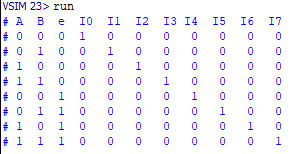


Fig 06: Output

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

#### 4: Design 4x16 Decoder using 3x8 Decoders.

module decoder4x16(A, B, C, e, I);

input A, B, C, e;

output [0:15]I;

wire ne;

not n(ne, e);

d3x8 d1(A, B, C, ne, I[0:7]);

d3x8 d2(A, B, C, e, I[8:15]);

endmodule

4x16 using 3x8 Decoders Verilog code

module decoder4x16TB();

reg A, B, e, C;

wire [15:0]I;

decoder4x16 d3(A, B, C, e, I);

initial begin

$monitor("e=%b, A=%b, B=%b, C=%b, I=%b %b %b %b %b %b %b %b %b %b %b %b %b %b %b %b", A, B, C, e, I[0], I[1], I[2], I[3], I[4], I[5], I[6], I[7], I[8], I[9], I[10], I[11], I[12], I[13], I[14], I[15]);

e = 0; A = 0; B = 0; C = 0; #20;

e = 0; A = 0; B = 0; C = 1; #20;

e = 0; A = 0; B = 1; C = 0; #20;

e = 0; A = 0; B = 1; C = 1; #20;

e = 0; A = 1; B = 0; C = 0; #20;

e = 0; A = 1; B = 0; C = 1; #20;

e = 0; A = 1; B = 1; C = 0; #20;

e = 0; A = 1; B = 1; C = 1; #20;

e = 1; A = 0; B = 0; C = 0; #20;

e = 1; A = 0; B = 0; C = 1; #20;

e = 1; A = 0; B = 1; C = 0; #20;

e = 1; A = 0; B = 1; C = 1; #20;

e = 1; A = 1; B = 0; C = 0; #20;

e = 1; A = 1; B = 0; C = 1; #20;

e = 1; A = 1; B = 1; C = 0; #20;

e = 1; A = 1; B = 1; C = 1;

**end**

endmodule

Test Bench

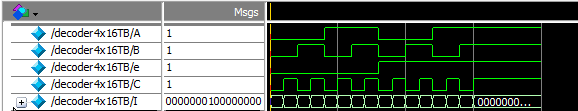


Fig 07: Output

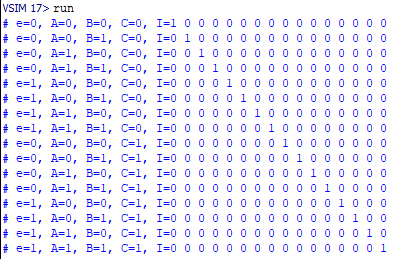


Fig 08: Output

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

# Conclusion:

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

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