LAB #08

DECODERS 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

(17 Dec 2023)

Department of Computer systems engineering

University of Engineering and Technology, Peshawar

**Tasks 1**

**Code:**

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

input A;

input B;

input e;

output [3:0]I;

wire nA;

wire nB;

not n1(nA,A);

not n2(nB,B);

and A1(I[0],nA,nB,e);

and A2(I[1],nA,B,e);

and A3(I[2],A,nB,e);

and A4(I[3],A,B,e);

endmodule;

**Test Code:**

module tb();

reg A;

reg B;

reg e;

wire [0:3]I;

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

initial begin

A=0;

B=0;

e=0;

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

A=0;B=0;e=1;

#10 $display("%b %b %b %b %b %b %b",A,B,e,I[0],I[1],I[2],I[3]);

A=0;B=1;e=1;

#10 $display("%b %b %b %b %b %b %b",A,B,e,I[0],I[1],I[2],I[3]);

A=1;B=0;e=1;

#10 $display("%b %b %b %b %b %b %b",A,B,e,I[0],I[1],I[2],I[3]);

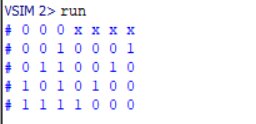
A=1;B=1;e=1;

#10 $display("%b %b %b %b %b %b %b",A,B,e,I[0],I[1],I[2],I[3]);

end

endmodule

**Output:**

****

**Tasks 2**

**Code:**

module Decoder3x8(

input [2:0] in,

output [7:0] out

);

assign out[0] = ~(in[2] & in[1] & in[0]);

assign out[1] = ~(in[2] & in[1] & ~in[0]);

assign out[2] = ~(in[2] & ~in[1] & in[0]);

assign out[3] = ~(in[2] & ~in[1] & ~in[0]);

assign out[4] = ~(~in[2] & in[1] & in[0]);

assign out[5] = ~(~in[2] & in[1] & ~in[0]);

assign out[6] = ~(~in[2] & ~in[1] & in[0]);

assign out[7] = ~(~in[2] & ~in[1] & ~in[0]);

endmodule

**Tasks 3**

**Code:**

module Decoder2x4(

input [1:0] in,

output [3:0] out

);

assign out[0] = ~(in[1] & in[0]);

assign out[1] = ~(in[1] & ~in[0]);

assign out[2] = ~(~in[1] & in[0]);

assign out[3] = ~(~in[1] & ~in[0]);

endmodule

module Decoder3x8(

input [2:0] in,

output [7:0] out

);

wire [3:0] intermediate\_out1;

wire [3:0] intermediate\_out2;

// First stage 2x4 Decoders

Decoder2x4 uut1 (

.in(in[1:0]),

.out(intermediate\_out1)

);

Decoder2x4 uut2 (

.in(in[2:1]),

.out(intermediate\_out2)

);

// Second stage OR gates

assign out[0] = intermediate\_out1[0];

assign out[1] = intermediate\_out1[1];

assign out[2] = intermediate\_out1[2];

assign out[3] = intermediate\_out1[3];

assign out[4] = intermediate\_out2[0];

assign out[5] = intermediate\_out2[1];

assign out[6] = intermediate\_out2[2];

assign out[7] = intermediate\_out2[3];

endmodule

**Tasks 4**

**Code:**

module Decoder4x16(

input [3:0] in,

output [15:0] out

);

wire [7:0] intermediate\_out1;

wire [7:0] intermediate\_out2;

// First stage 3x8 Decoders

Decoder3x8 uut1 (

.in(in[2:0]),

.out(intermediate\_out1)

);

Decoder3x8 uut2 (

.in(in[3:1]),

.out(intermediate\_out2)

);

// Second stage OR gates

assign out[0] = intermediate\_out1[0];

assign out[1] = intermediate\_out1[1];

assign out[2] = intermediate\_out1[2];

assign out[3] = intermediate\_out1[3];

assign out[4] = intermediate\_out1[4];

assign out[5] = intermediate\_out1[5];

assign out[6] = intermediate\_out1[6];

assign out[7] = intermediate\_out1[7];

assign out[8] = intermediate\_out2[0];

assign out[9] = intermediate\_out2[1];

assign out[10] = intermediate\_out2[2];

assign out[11] = intermediate\_out2[3];

assign out[12] = intermediate\_out2[4];

assign out[13] = intermediate\_out2[5];

assign out[14] = intermediate\_out2[6];

assign out[15] = intermediate\_out2[7];

endmodule

module Decoder3x8(

input [2:0] in,

output [7:0] out

);

assign out[0] = ~(in[2] | in[1] | in[0]);

assign out[1] = ~(in[2] | in[1] | ~in[0]);

assign out[2] = ~(in[2] | ~in[1] | in[0]);

assign out[3] = ~(in[2] | ~in[1] | ~in[0]);

assign out[4] = ~(~in[2] | in[1] | in[0]);

assign out[5] = ~(~in[2] | in[1] | ~in[0]);

assign out[6] = ~(~in[2] | ~in[1] | in[0]);

assign out[7] = ~(~in[2] | ~in[1] | ~in[0]);

endmodule