Course: CS 223 Digital Design

Section: 6

Lab: 3

Name: Halil AVCI

Student ID: 22003476

Date: 16/10/2023

Trainer Pack: 31

**Behavioral 2-to-4 Decoder:**

`timescale 1ns / 1ps

module behavioral\_2to4\_decoder( input logic A0, A1,

output logic Y0, Y1,

Y2, Y3);

assign Y0 = ~(A0 | A1);

assign Y1 = ~(A0 | ~A1);

assign Y2 = ~(~A0 | A1);

assign Y3 = A0 & A1;

endmodule

**Testbench:**

`timescale 1ns / 1ps

module sim\_decoder();

logic A0, A1, Y0, Y1, Y2, Y3;

behavioral\_2to4\_decoder dut( A0, A1, Y0, Y1, Y2, Y3);

initial begin

#10 A0 = 0; A1 = 0;

#10 A0 = 0; A1 = 1;

#10 A0 = 1; A1 = 0;

#10 A0 = 1; A1 = 1;

end

initial $monitor($time,,"A0=%b, A1=%b, Y0=%b, Y1=%b, Y2=%b, Y3=%b", A0, A1, Y0, Y1, Y2, Y3);

endmodule

**Behavioral 2-to-1 Multiplexer:**

`timescale 1ns / 1ps

module behavioral\_2to1\_mux( input logic A0, A1, S,

output logic Y);

assign Y = S ? A1 : A0;

endmodule

**Behavioral 4-to-1 Multiplexer:**

`timescale 1ns / 1ps

module behavioral\_4to1\_mux( input logic A0, A1, A2, A3, S0, S1,

output logic Y);

logic mux0\_output, mux2\_output;

assign mux0\_output = S0 ? A2 : A0;

assign mux1\_output = S0 ? A3 : A1;

assign Y = S1 ? mux1\_output : mux0\_output;

endmodule

**Structural 8-to-1 Multiplexer**:

`timescale 1ns / 1ps

module structural\_8to1\_mux( input logic A0, A1, A2, A3, A4, A5, A6, A7, S0, S1, S2,

output logic Y);

wire mux0\_Y, mux1\_Y, SS, Y0, Y1;

behavioral\_4to1\_mux mux0(A0, A1, A2, A3, S1, S2, mux0\_Y);

behavioral\_4to1\_mux mux1(A4, A5, A6, A7, S1, S2, mux1\_Y);

not not1(SS, S0);

and and1(Y0, mux0\_Y, SS);

and and2(Y1, S0, mux1\_Y);

or or1(Y, Y0, Y1);

endmodule

**Testbench:**

`timescale 1ns / 1ps

module sim\_8to1\_mux();

logic A0, A1, A2, A3, A4, A5, A6, A7, S0, S1, S2, Y;

structural\_8to1\_mux dut( A0, A1, A2, A3, A4, A5, A6, A7, S0, S1, S2, Y);

initial begin

#10 A0 = 0; A1 = 0; A2 = 0; A3 = 0; A4 = 0; A5 = 0; A6 = 0; A7 = 0; S0 = 0; S1 = 0; S2 = 0;

for (int i=0; i<4096; i=i+1) begin

{A0, A1, A2, A3, A4, A5, A6, A7, S0, S1, S2} = i;

#1;

end

end

initial $monitor($time,,"A0=%b, A1=%b, A2=%b, A3=%b, A4=%b, A5=%b, A6=%b, A7=%b, S0=%b, S1=%b, S2=%b, Y=%b, ",

A0, A1, A2, A3, A4, A5, A6, A7, S0, S1, S2, Y);

endmodule

**Structural Function:**

`timescale 1ns / 1ps

module structural\_Function( input logic A, B, C, D, E,

output logic Y);

wire decoder1, decoder2, decoder3, decoder4;

behavioral\_2to4\_decoder decoder(A, B, decoder1, decoder2, decoder3, decoder4);

structural\_8to1\_mux mux(decoder4, decoder2, 1, B, decoder1, ~A, 0, decoder3, C, D, E, Y);

endmodule

