**Course Code: EE461**

**Assignment**

**PREPARED BY**

**Khandoker Samiul Hoque**

**Student Id: 19837**

**1 No Answer:**

module static\_LG(

input a\_i,

input b\_i,

input c\_i,

output f\_o

);

supply1 vdd;

supply0 gnd;

wire dp1, dn12;

pmos p1(dp1, vdd, a\_i);

pmos p2(f\_o, dp1, b\_i);

pmos p3(f\_o, vdd, c\_i);

nmos n1(dn12, gnd, a\_i);

nmos n2(dn12, gnd, b\_i);

nmos n3(f\_o, dn12, c\_i);

endmodule

**TestBench:**

module tb;

reg a\_r;

reg b\_r;

reg c\_r;

wire f\_w;

static\_LG prob\_1 (

.a\_i(a\_r),

.b\_i(b\_r),

.c\_i(c\_r),

.f\_o(f\_w)

);

initial begin

$dumpfile("dump.vcd");

$dumpvars(0, tb);

// All input combinations

a\_r = 0; b\_r = 0; c\_r = 0;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 0; b\_r = 0; c\_r = 1;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 0; b\_r = 1; c\_r = 0;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 0; b\_r = 1; c\_r = 1;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 1; b\_r = 0; c\_r = 0;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 1; b\_r = 0; c\_r = 1;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 1; b\_r = 1; c\_r = 0;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

a\_r = 1; b\_r = 1; c\_r = 1;

#5;

$display("a=%b b=%b c=%b f=%b", a\_r, b\_r, c\_r, f\_w);

$finish;

end

endmodule

**2 No Answer:**

module udp\_mux2to1(input wire S, input wire A, input wire B, output reg Y);

always @ (S or A or B) begin

case (S)

1'b0: Y <= A;

1'b1: Y <= B;

default: Y <= 1'bx;

endcase

end

endmodule

**TestBench;**

module tb;

reg S, A, B;

wire Y;

udp\_mux2to1 prob2 (.S(S), .A(A), .B(B), .Y(Y));

initial begin

$monitor($time, " S=%b A=%b B=%b Y=%b", S, A, B, Y);

// Test case 1: S = 0

S = 0; A = 0; B = 0; #1;

S = 0; A = 0; B = 1; #1;

S = 0; A = 1; B = 0; #1;

S = 0; A = 1; B = 1; #1;

// Test case 2: S = 1

S = 1; A = 0; B = 0; #1;

S = 1; A = 0; B = 1; #1;

S = 1; A = 1; B = 0; #1;

S = 1; A = 1; B = 1; #1;

// Test case 3: S = X/Z

S = 1'bx; A = 0; B = 0; #1;

S = 1'bx; A = 0; B = 1; #1;

S = 1'bx; A = 1; B = 0; #1;

S = 1'bx; A = 1; B = 1; #1;

S = 1'bz; A = 0; B = 0; #1;

S = 1'bz; A = 0; B = 1; #1;

S = 1'bz; A = 1; B = 0; #1;

S = 1'bz; A = 1; B = 1; #1;

end

endmodule

**3 No Answer:**

primitive Mux(f\_o, a, b,

? 0 1 : 0;

? 1 1 : 1;

endtable

endprimitive

primitive udpOR\_gate(out, a, b);

output out;

input a, b;

table

//A B : out

1 ? : 1;

? 1 : 1;

0 0 : 0;

endtable

endprimitive

primitive udpnot\_gate(out, a);

output out;

input a;

table

//A : out

1 : 0;

0 : 1;

endtable

endprimitive

**TestBench:**

module tb(out, a, b, c, d);

output out;

input a, b, d, c;

int i;

wire or1, n1, muxo, muxor, muxon, dn;

#5

udpOR\_gate u1 (or1, a, b);

#3

udpnot\_gate u2(n1, c);

#8

Mux u3 (muxo, or1, n1, d);

#5

udpOR\_gate u4 (muxor, muxo, muxo);

#3

udpnot\_gate u5 (muxon, muxo);

#3

udpnot\_gate u6 (dn, d);

#8

Mux u7 (out, muxor, muxon, dn);

initial begin

$monitor("A=%b, B=%b, C=%b, D=%b, out=%b", a, b, c, d)

#2

for(i=0; i<16; i=i+1)begin

{d,c,b,a} = i;

#2

end

end

endmodule

**4 No Answer:**

primitive example(a, b, c, d, e, f, g, h, i, j, r);

output a;

input b, c, d, e, f, g, h, i, j, r;

reg a;

table

b c d e f g h i j r : a

0 0 0 1 0 1 0 1 0 1 : 0

0 0 0 1 0 1 0 1 0 1 : 1

0 0 0 1 0 1 0 1 0 1 : 0

... ...

endtable

endprimitive

for edu playground we don’t have to call other files so we change the first line, here output a is declared instant of a and r.we don’t have to declare the basic primitive also and it was not in the right line.

**5 No Answer:**

module udp\_dff(

input D, CLK,

output reg Q, Qbar

);

wire notD, nand1, nand2;

not u\_not(

.A(D),

.Y(notD)

);

nand u\_nand1(

.A(notD),

.B(CLK),

.Y(nand1)

);

nand u\_nand2(

.A(D),

.B(nand1),

.Y(nand2)

);

always @(posedge CLK) begin

Q <= nand2;

Qbar <= not(nand2);

end

endmodule

**Testbench:**

module tb();

reg D;

reg CLK;

wire Q;

wire Qbar;

udp\_dff prob\_5(

.D(D),

.CLK(CLK),

.Q(Q),

.Qbar(Qbar)

);

always #5

CLK = ~CLK;

initial begin

$dumpfile("tb.vcd");

$dumpvars();

$display("Test case 1: Set D to 0, toggle CLK");

D = 0;

#10;

CLK = 0;

#10;

CLK = 1;

#10;

if (Q !== 0 || Qbar !== 1) $error("Error: Q and Qbar should be 0 and 1, respectively");

$display("Test case 2: Set D to 1, toggle CLK");

D = 1;

#10;

CLK = 0;

#10;

CLK = 1;

#10;

if (Q !== 1 || Qbar !== 0) $error("Error: Q and Qbar should be 1 and 0, respectively");

$display("Test case 3: Set D to 1, hold CLK high");

D = 1;

#10;

CLK = 1;

#10;

if (Q !== 1 || Qbar !== 0) $error("Error: Q and Qbar should be 1 and 0, respectively");

$display("Test case 4: Set D to 0, hold CLK high");

D = 0;

#10;

CLK = 1;

#10;

if (Q !== 0 || Qbar !== 1) $error("Error: Q and Qbar should be 0 and 1, respectively");

$finish;

end

endmodule