



San Francisco Bay University
EE461L - Digital Design and HDL Lab
Quiz #1

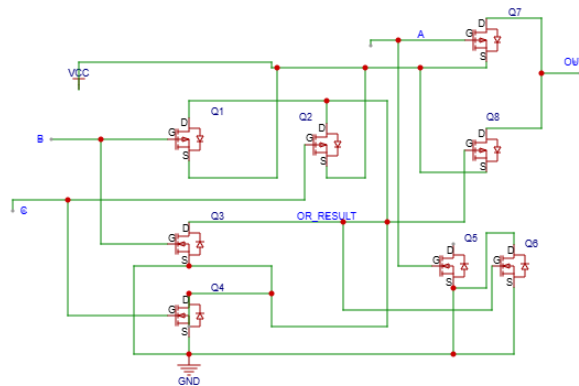
Student Name: Anika Haque

Student ID: 20283

- 1. Design the module for a combinational gate by using *MOS* primitives based on the following logic expression, and then write the testbench to verify it.**

$$out = a(b + c)$$

- a. Draw the circuit schematic in *cmos* and *pmos* first.**



- b. Write the Verilog module according to the above circuit.**

```
module full_circuit(output out, input a, b, c);

    // OR Gate: Implements b + c
    wire or_result;          // Intermediate signal for the OR gate
    output
    wire pmos_pullup_temp;   // Intermediate wire for the PMOS
    network in the OR gate

    // NMOS Network: Pulls or_result low (0) when b = 1 or c = 1
    nmos n1(or_result, 1'b1, b);    // Pull down if b = 1
    nmos n2(or_result, 1'b1, c);    // Pull down if c = 1

    // PMOS Network: Pulls or_result high (1) when b = 0 or c = 0
    pmos p1(pmos_pullup_temp, 1'b0, b); // Pull up temp if b = 0
    pmos p2(or_result, pmos_pullup_temp, c); // Pull up output if
    temp = 1 and c = 0

    // AND Gate: Implements a · (b + c)
```

```

    wire temp; // Intermediate signal for the NMOS network in the
AND gate

    // PMOS Network: Pulls out high (1) when a = 0 or or_result = 0
    pmos n3(out, 1'b0, a);          // Pull up if a = 0
    pmos n4(out, 1'b0, or_result);  // Pull up if or_result = 0

    // NMOS Network: Pulls out low (0) when both a = 1 and or_result
= 1
    nmos p3(temp, 1'b1, a);          // Pull down temp if a = 1
    nmos p4(out, temp, or_result);  // Pull down output if temp = 0
and or_result = 1

endmodule

```

c. Create the testbench to verify it and show the waveforms.

Testbench

```

module full_circuit_tb;

    reg a, b, c;          // Input signals
    wire or_result;       // Intermediate result from the OR gate
    wire out;             // Final output from the AND gate

    // Instantiate the full circuit
    full_circuit uut (
        .out(out),
        .a(a),
        .b(b),
        .c(c)
    );

    // Initial block for stimuli and waveform
    initial begin
        $dumpfile("waveform.vcd");
        $dumpvars(0, full_circuit_tb);

        // Monitor the intermediate OR gate result and final AND
gate result
        $monitor("Time = %0t | a = %b, b = %b, c = %b | or_result
= %b | out = %b",
            $time, a, b, c, uut.or_result, out);

        // Test cases
        a = 0; b = 0; c = 0; #10; // Expect or_result = 0, out = 0
        a = 0; b = 0; c = 1; #10; // Expect or_result = 1, out = 0
        a = 0; b = 1; c = 0; #10; // Expect or_result = 1, out = 0
        a = 0; b = 1; c = 1; #10; // Expect or_result = 1, out = 0
        a = 1; b = 0; c = 0; #10; // Expect or_result = 0, out = 0
        a = 1; b = 0; c = 1; #10; // Expect or_result = 1, out = 1
        a = 1; b = 1; c = 0; #10; // Expect or_result = 1, out = 1
        a = 1; b = 1; c = 1; #10; // Expect or_result = 1, out = 1
    end
endmodule

```

```

    $finish;
end

```

```
endmodule
```

Simulation Output

testbench.v

```

1 module full_circuit_tb;
2
3   reg a, b, c; // Input signals
4   wire or_result; // Intermediate result from the OR gate
5   wire out; // Final output from the AND gate
6
7   // Instantiate the full circuit
8   full_circuit uut (
9     .out(out),
10    .a(a),
11    .b(b),
12    .c(c)
13  );
14
15  // Initial block for stimuli and waveform
16  initial begin
17    $dumpfile("waveform.vcd");
18    $dumpvars(0, full_circuit_tb);
19
20    // Monitor the intermediate OR gate result and final AND gate result
21    $monitor("Time = %0t | a = %b, b = %b, c = %b | or_result = %b | out = %b",
22            $time, a, b, c, uut.or_result, out);
23
24    // Test cases
25    a = 0; b = 0; c = 0; #10; // Expect or_result = 0, out = 0
26    a = 0; b = 0; c = 1; #10; // Expect or_result = 1, out = 0
27    a = 0; b = 1; c = 0; #10; // Expect or_result = 1, out = 0
28    a = 0; b = 1; c = 1; #10; // Expect or_result = 1, out = 0
29    a = 1; b = 0; c = 0; #10; // Expect or_result = 0, out = 0
30    a = 1; b = 0; c = 1; #10; // Expect or_result = 1, out = 1
31    a = 1; b = 1; c = 0; #10; // Expect or_result = 1, out = 1
32    a = 1; b = 1; c = 1; #10; // Expect or_result = 1, out = 1
33
34    $finish;
35  end
36 endmodule

```

design.v

```

1 module full_circuit(output out, input a, b, c);
2
3   // OR Gate: Implements b + c
4   wire or_result; // Intermediate signal for the OR gate output
5   wire pmos_pullup_temp; // Intermediate wire for the PMOS network in the OR gate
6
7   // NMOS Network: Pulls or_result low (0) when b = 1 or c = 1
8   nmos n1(or_result, 1'b1, b); // Pull down if b = 1
9   nmos n2(or_result, 1'b1, c); // Pull down if c = 1
10
11  // PMOS Network: Pulls or_result high (1) when b = 0 or c = 0
12  pmos p1(pmos_pullup_temp, 1'b0, b); // Pull up temp if b = 0
13  pmos p2(or_result, pmos_pullup_temp, c); // Pull up output if temp = 1 and c = 0
14
15  // AND Gate: Implements a · (b + c)
16  wire temp; // Intermediate signal for the NMOS network in the AND gate
17
18  // PMOS Network: Pulls out high (1) when a = 0 or or_result = 0
19  pmos n3(out, 1'b0, a); // Pull up if a = 0
20  pmos n4(out, 1'b0, or_result); // Pull up if or_result = 0
21
22  // NMOS Network: Pulls out low (0) when both a = 1 and or_result = 1
23  nmos p3(temp, 1'b1, a); // Pull down temp if a = 1
24  nmos p4(out, temp, or_result); // Pull down output if temp = 0 and or_result = 1
25
26 endmodule

```

Log

Share

2024-12-12 15:07:47 UTC

iverilog -wall -g2012 design.v testbench.v && unbuffer vvp a.out

VCD info: dumpfile waveform.vcd opened for output.

Time = 0 | a = 0, b = 0, c = 0 | or_result = 0 | out = 0

Time = 10 | a = 0, b = 0, c = 1 | or_result = 1 | out = 0

Time = 20 | a = 0, b = 1, c = 0 | or_result = 1 | out = 0

Time = 30 | a = 0, b = 1, c = 1 | or_result = 1 | out = 0

Time = 40 | a = 1, b = 0, c = 0 | or_result = 0 | out = 0

Time = 50 | a = 1, b = 0, c = 1 | or_result = 1 | out = 1

Time = 60 | a = 1, b = 1, c = 0 | or_result = 1 | out = 1

Time = 70 | a = 1, b = 1, c = 1 | or_result = 1 | out = 1

testbench.v:34: \$finish called at 80 (1s)

Waveform

EDA Playground

<https://www.edaplayground.com/x/ZnpE>

From:

Get Signals

Radix

100%

◀

▶

⏮

⏭

⏪

⏩

⏴

⏵

⏶

⏷

⏸

Brought to you by DOULOS

EDA Playgro

From:

0s

To:

80s

Get Signals

F

Brought to you by DOULOS

3