

ENEE 244: Build and Test a Verilog Circuit

10/20/26

A guided, three-file programming activity | 45 minutes | Work individually or in pairs

Your goal

Create two gate modules, connect them into one circuit, and write a testbench that confirms the circuit behaves correctly. At the end, you should have three working files: module2.v, design.v, and tb.v.

Circuit specification

```
z ← x OR y
out ← z AND w
```

The OR gate takes x and y as inputs. Its result, z, is combined with w using an AND gate.

1. Predict before coding | 5 min

Write the Boolean expression for out using x, y, and w:

out = _____

Complete the output column. Use the input order shown; you will test these exact cases later.

x	y	w	predicted out
0	0	0	_____
0	1	0	_____
1	0	0	_____
1	1	0	_____
0	0	1	_____
0	1	1	_____
1	0	1	_____
1	1	1	_____

2. Write the two gate modules | 8 min

Open module2.v. Complete the assignments using the Verilog logical operators || (OR) and && (AND). Each gate has two input ports and one output port.

```
module or_gate(input x, y, output z);
  // TODO: assign z using x and y
endmodule

module and_gate(input x, y, output z);
  // TODO: assign z using x and y
endmodule
```

Checkpoint: With inputs (1, 0), what does each gate output? OR: _____ AND: _____

3. Connect the modules in top | 9 min

Open design.v. The top module has three inputs (x, y, w) and one output (out). Declare an intermediate wire z, then instantiate the two gates with the connections described below.

```
module top(
    input x,
    input y,
    input w,
    output out
);

    // TODO: declare wire z

    // TODO: instantiate or_gate as u1 with ports (x, y, z)

    // TODO: instantiate and_gate as u2 with ports (z, w, out)

endmodule
```

Use positional port connections, e.g., gate_name instance_name(a, b, c);. The order must match the module definition.

Quick check: Why do we need z? Why is z declared wire rather than reg?

4. Create the testbench structure | 8 min

Open tb.v. Your testbench is named tb, and it has no input or output ports. It will provide input values to your circuit and observe the output.

```
module tb;
    // TODO: declare x, y, w as reg and out as wire
    // TODO: instantiate top as uut with ports (x, y, w, out)

    initial begin
        $dumpfile("dump.vcd");
        // TODO: add $dumpvars(0, tb.x, tb.y, tb.w, tb.out);
        $display("x y w | out");

        // Test cases go here in Step 5.

    end
endmodule
```

Checkpoint: Why are the testbench inputs declared reg while the output is declared wire?

5. Test all eight combinations | 10 min

Inside the initial block in tb.v, assign x, y, w each of the eight combinations from Step 1, in the same order. After each assignment:

- Use #20; to give the circuit time to update.
- Print x, y, w, and out with \$display and %b for binary values.

Here is the first case; write the other seven in your code editor:

```
x=0;y=0;w=0;#20;$display("%b %b %b | %b",x,y,w,out);
```

At the end of the initial block, print the simulation time and then end the simulation:

```
$display("Simulation finished at time %0t", $time);
$finish;
```

6. Run, compare, and debug | 5 min

Compile and simulate all three Verilog files together (in the class simulator). You should see the header, one line for each test case, and a final message reporting the simulation time.

Your verification checklist

- The simulation prints eight output rows.
- Every output matches the prediction you made in Step 1.
- The simulation ends after eight #20 delays (time 160 in the simulator's default time units).
- The generated dump.vcd file can be opened in a waveform viewer, if available.

If something is wrong, check these first

- Are the ports in your gate instances connected in the correct order?
- Is z connected to the OR output and the AND input?
- Did you assign all three testbench inputs before each #20 delay?
- Are all three Verilog files included in the simulation?

Exit ticket — answer in one or two sentences

1. Under what conditions is out = 1?

2. What is the role of the top module, and what is the role of tb?

What to submit / show the instructor

A working module2.v, design.v, and tb.v, plus the eight-line simulation output. Your finished code should implement the same three-file circuit and testbench described in this activity.