İstanbul Technical University

CEN 242E – Logical Circuits Lab

LAB 6: Verification

About LAB:

Verification

Stages and scores of LAB :

1- Testbench

Please write a Verilog testbench for the following module.

```
module exampleRTL ( input a, b, c, output reg y);
always @(*)
y = ~b & ~c | a & ~b;
endmodule
```

Your testbench must:

Instantiate exampleRTL with inputs a, b, c and output y.

Drive the registers in exactly this order and timing:

At time 0 ns set a=0, b=0, c=0;

After 10 ns, change only c to 1;

After another 10 ns, change b to 1 and c back to 0;

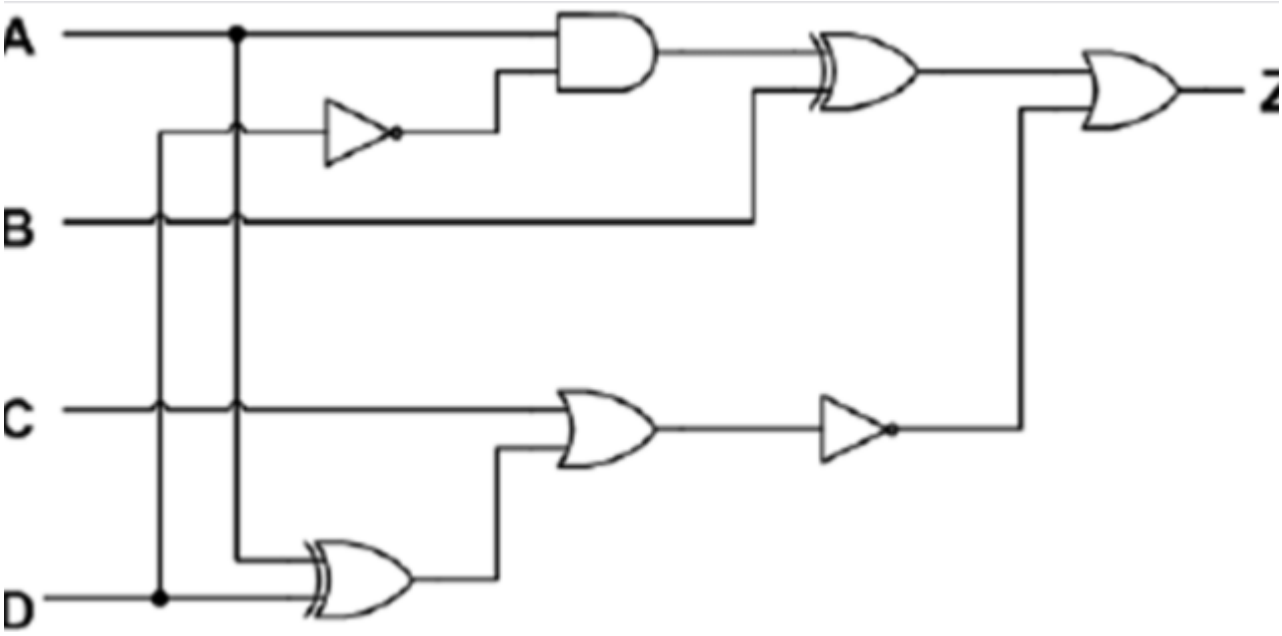
After a final 10 ns, change only c to 1;

End the simulation automatically when the last vector has been applied.

Make sure your stimulus exactly matches these four steps—no extra delays, no additional assignments.

2- Testbench

Design a simple Verilog testbench for the digital circuit shown in the image below. The circuit includes a combination of logic gates such as NOT, AND, OR, and XOR, with inputs labeled as A, B, C, and D, and an output labeled as Z. The testbench should cover at least 4 different input conditions to validate the functionality of the circuit.



3- Buggy Adder Module

You have a module buggyAdder that is supposed to compute the 2-bit sum of its inputs in1 and in2, but contains a defect for one particular input combination. Your job is to write a Verilog testbench called tb_buggyAdder that:

1. Instantiates buggyAdder without modifying it.
2. Drives every possible pair of 2-bit values on in1 and in2.
3. Monitors the DUT's 3-bit output out and compares it to the expected sum (in1 + in2).
4. As soon as the very first mismatch occurs (i.e. out !== in1 + in2), sets a flag register found_bad to 1 and captures the two inputs that caused the failure into registers bad_in1 and bad_in2.
5. Leaves bad_in1 and bad_in2 holding exactly the values of in1 and in2 at the moment that mismatch was detected.