

Lab Assignment 7

ES 204: Digital Systems

Prof. Joycee Mekie

27th Feb, 2024

Indian Institute of Technology, Gandhinagar

Aditya N. Mehta

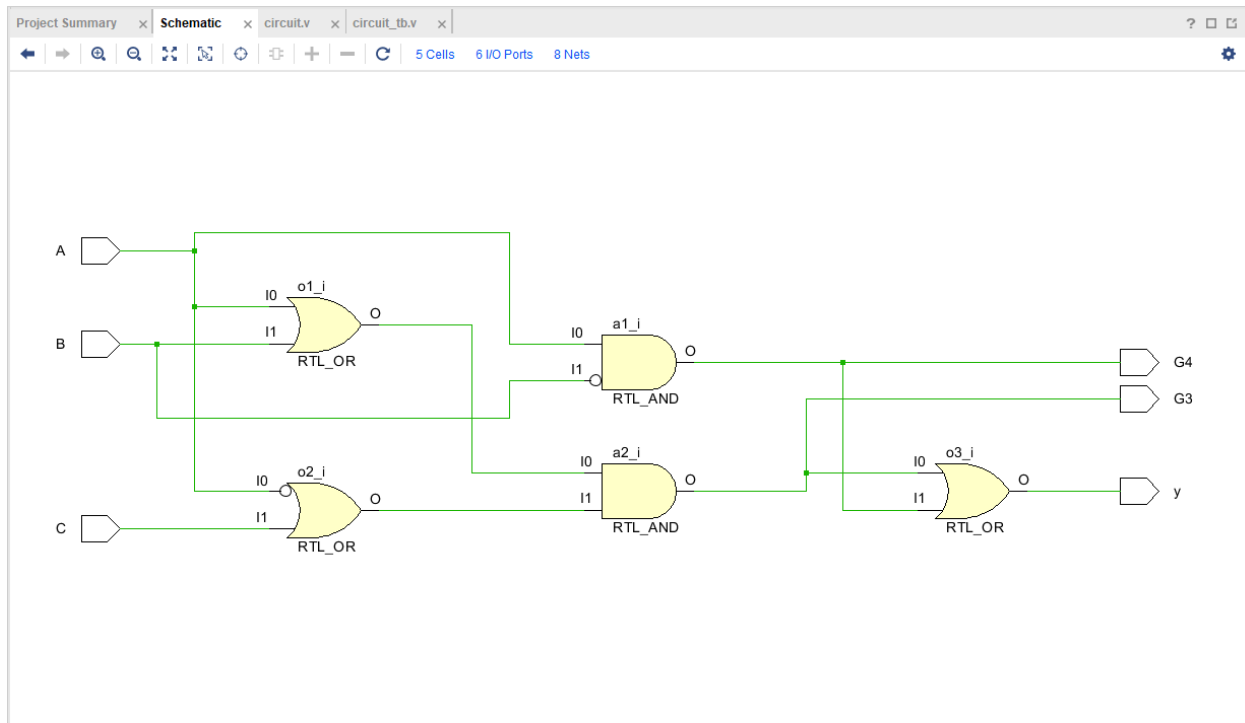
22110017

Hrriday V. Ruparel

22110099

Analyzing Static and Dynamic Hazards

A) Schematic:



B) Static Hazard (Verilog Code):

```
Project Summary x Schematic x circuit.v x circuit_tb.v x
F:/MITGNES 204 Digital Systems/LabAssignment7/LabAssignment7.srcs/sources_1/new/circuit.v

1 `timescale 1ns / 1ps
2
3 module circuit(A,B,C,G3,G4,y);
4   input A,B,C;
5   output G3, G4, y;
6   not #10 n1(A1, A);
7   not #10 n2(B1, B);
8   or #10 o1(G1, A, B);
9   or #10 o2(G2, A1, C);
10  and #10 a1(G4, A, B1);
11  and #10 a2(G3, G1, G2);
12  or #10 o3(y, G3, G4);
13 endmodule
14
```

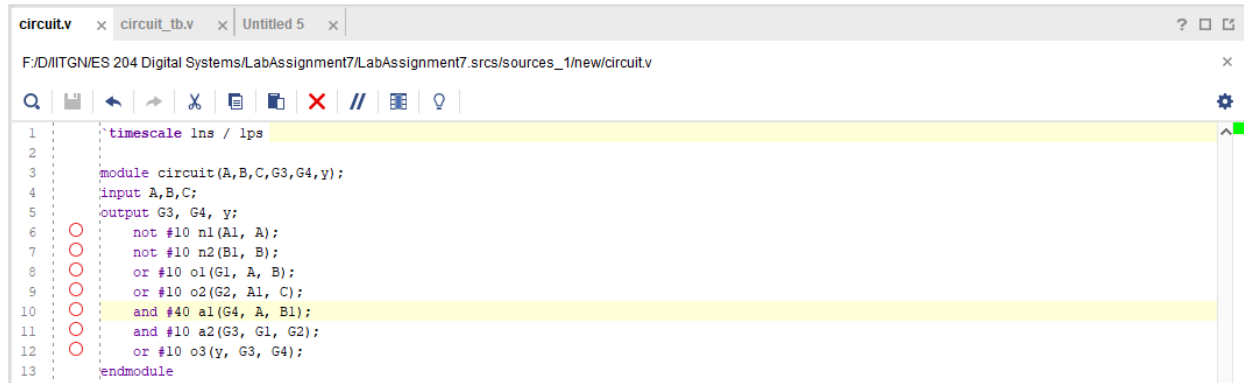
C) Static Hazard (Testbench Code):

```
circuit.v x circuit_tb.v x Untitled 7 x
F:/D/ITGN/ES 204 Digital Systems/LabAssignment7/LabAssignment7.srscs/sim_1/new/circuit_tb.v
1 timescale 1ns / 1ps
2
3 module circuit_tb();
4 reg A, B, C;
5 wire G3, G4, y;
6
7 circuit uut (A, B, C, G3, G4, y);
8
9 initial
10 begin
11     A = 0; B = 0; C = 0;
12     #50;
13     // Transition
14     A = 1; B = 0; C = 0;
15     #100;
16
17 $finish();
18 end
19 endmodule
```

D) Static Hazard (Simulation Waveform):



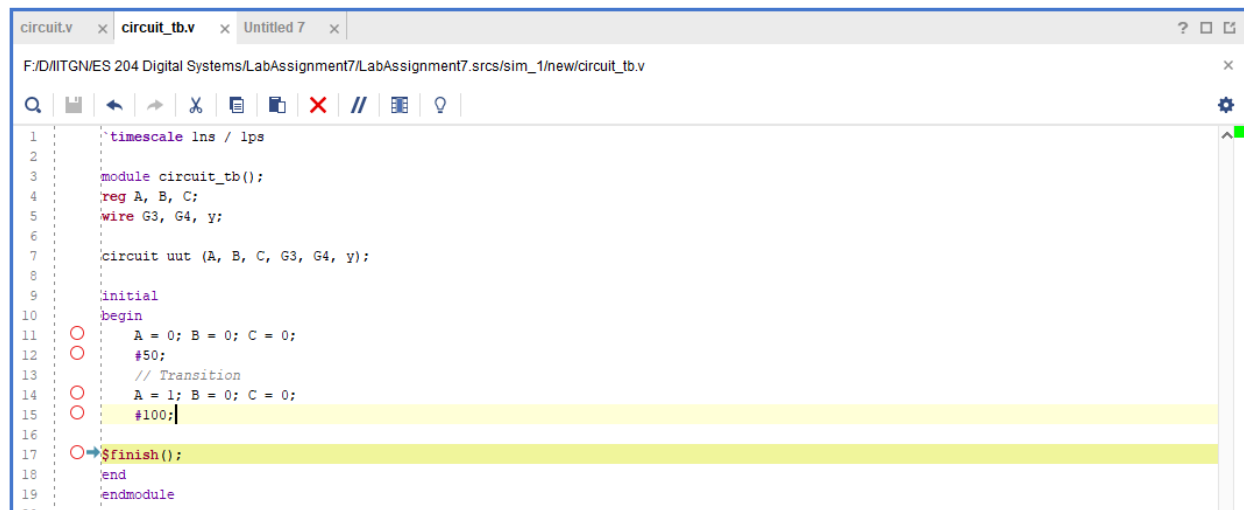
E) Dynamic Hazard (Verilog Code):



The screenshot shows a Verilog code editor with a single file named 'circuit.v'. The code defines a module 'circuit' with inputs A, B, C and outputs G3, G4, y. It includes a timescale of 1ns / 1ps. The logic consists of several gates: a NOT gate for n1 (A), a NOT gate for n2 (B), an OR gate for o1 (G1, A, B), an OR gate for o2 (G2, A1, C), an AND gate for a1 (G4, A, B1), an AND gate for a2 (G3, G1, G2), and an OR gate for o3 (y, G3, G4). The module ends with 'endmodule'.

```
1  `timescale 1ns / 1ps
2
3  module circuit(A,B,C,G3,G4,y);
4  input A,B,C;
5  output G3, G4, y;
6  not #10 n1(A1, A);
7  not #10 n2(B1, B);
8  or #10 o1(G1, A, B);
9  or #10 o2(G2, A1, C);
10 and #40 a1(G4, A, B1);
11 and #10 a2(G3, G1, G2);
12 or #10 o3(y, G3, G4);
13 endmodule
```

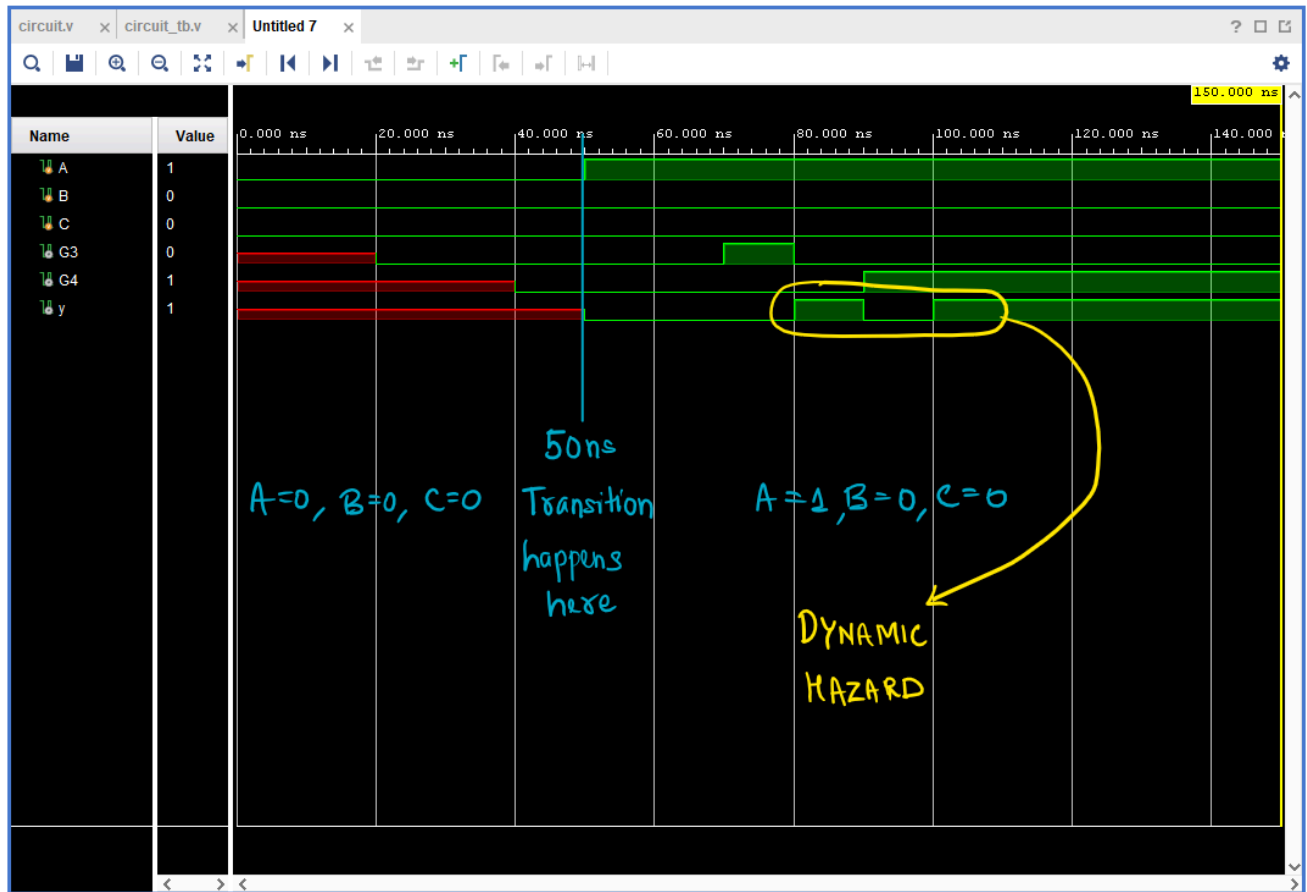
F) Dynamic Hazard (Testbench Code):



The screenshot shows a Verilog testbench code editor with a single file named 'circuit_tb.v'. The code defines a module 'circuit_tb()' with registers A, B, C and wires G3, G4, y. It instantiates the circuit module 'circuit' as ' uut'. The testbench includes an initial block with a begin statement. Inside the begin block, it sets A = 0, B = 0, C = 0; waits for 50ns; then sets A = 1, B = 0, C = 0; and waits for 100ns. The testbench ends with '\$finish();', 'end', and 'endmodule'.

```
1  `timescale 1ns / 1ps
2
3  module circuit_tb();
4  reg A, B, C;
5  wire G3, G4, y;
6
7  circuit uut (A, B, C, G3, G4, y);
8
9  initial
10 begin
11     A = 0; B = 0; C = 0;
12     #50;
13     // Transition
14     A = 1; B = 0; C = 0;
15     #100;
16
17     $finish();
18 end
19 endmodule
```

G) Dynamic Hazard (Stimulation Waveform):



H) Important Points:

- We won't observe a Static Hazard on Y as an OR gate caps it, and thus, a single input change won't lead to a Static Hazard. However, Static Hazard can be observed on the output of G3 Gate (uncapped in KMap)
- Dynamic Hazard is observed by adding an additional delay (of 30 ns) on the G4 AND gate. This is due to the delayed transition of G4 from 0 to 1.