Lab N°1
Fault Simulation

<u>Lab Objectives:</u>
- List of equivalent faults
- Fault simulation
- Minimization of the test sequence

Exercise 1:

1. Lab preparation

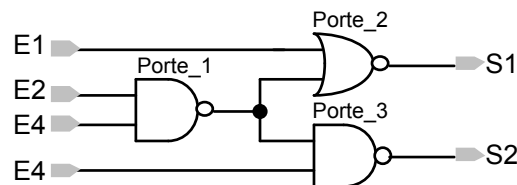Consider as first example the circuit depicted in Figure 1.

Figure 1: Exercise 1

From the schematic view, the VERILOG description is the following one :

Module name

```
module exo1(E1, E2, E3, E4, S1, S2) ;
input E1, E2, E3, E4 ;
output S1, S2 ;
nand gate_1(.Q(L1), .A(E2), .B(E3));
nor gate_2(.Q(S1), .A(E1), .B(L1));
nand gate_3(.Q(S2), .A(L1), .B(E4));
endmodule
```
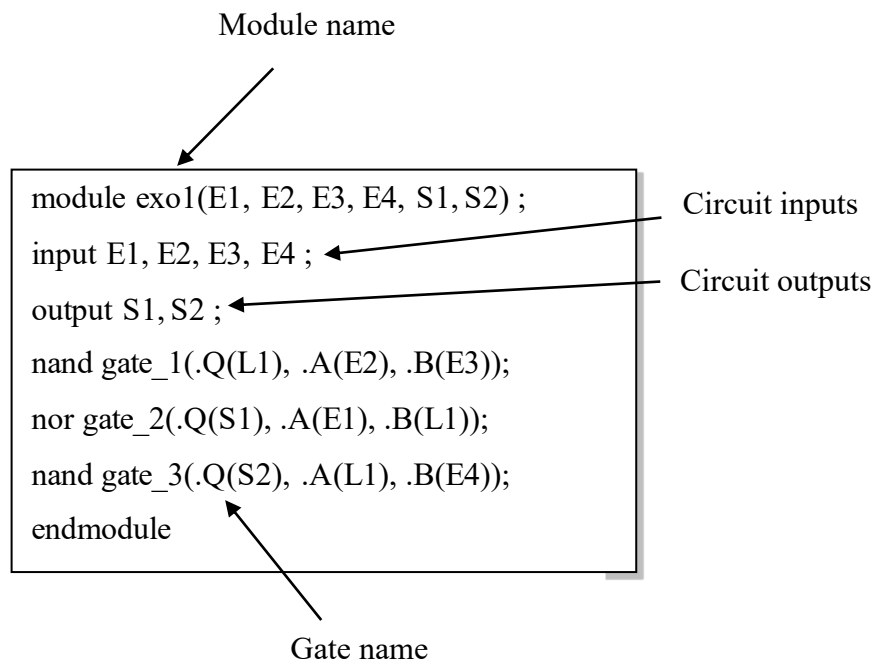
Circuit inputs

Circuit outputs

Gate name

Figure 2: Verilog description

The term *module* determines the beginning of the description. It is followed by the name of the module, in our example *exo1*, as well as all the inputs and outputs. The next two lines indicate separately the inputs (*input*) and the outputs (*output*). From this point, the structural description of the circuit begins. The terms *nand*, *nor*… are related to the gate library used. In the parentheses, the output is given first, followed by the inputs. The description ends with the term *endmodule*.

QUESTIONS:
From the example of Figure 1:
1. What is the total number of stuck-at faults without minimization?
2. Minimize this number using equivalences and implications.
3. Find the test vectors to apply to the circuit inputs in order to test all the faults of the minimal set.

2. Lab practice

Using TESSENT, check if the previously generated test sequence tests for all stuck-at faults.
Some tips:
- Work in the directory that contains the files.
- Source the tool: source /prog/Configs/Config_Mentor/Tessent_conf_2021_3.csh
- Edit the files to learn, especially tp1_command_example.txt and example_exo1.stil

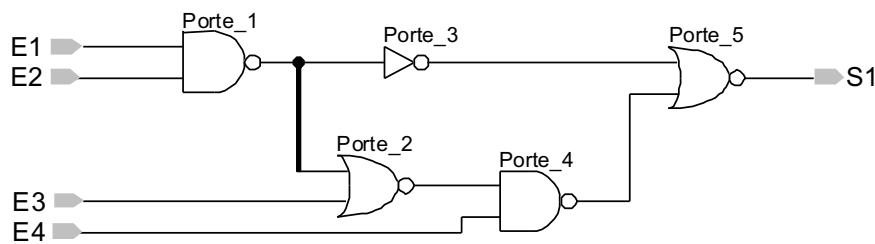# Exercise 2:

Consider the second example depicted in Figure 3.

Figure 3: Exercise 2

QUESTIONS:
- Theory:
    1. What is the total number of stuck-at faults without minimization?
    2. Minimize this number using equivalences and implications
- Practice:
    1. Prepare and apply an exhaustive test sequence.
    2. Using TESSENT, analyse the obtained results.
    3. Propose some solutions to detect all stuck-at faults.

Exercise 3:
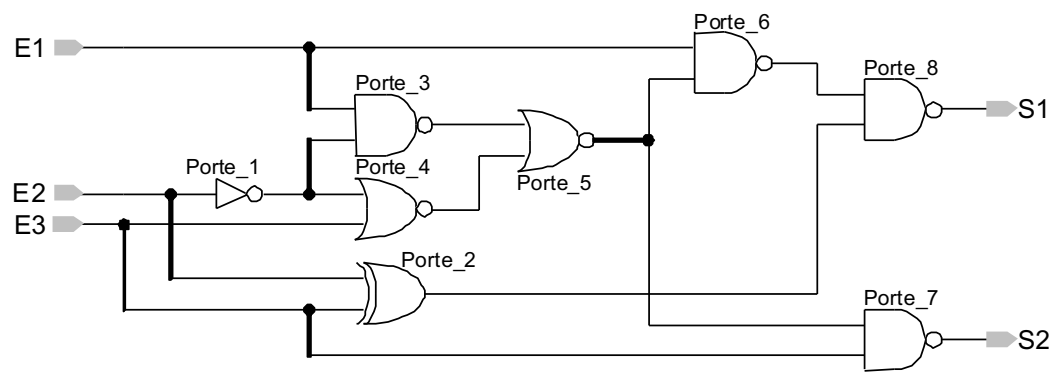
Consider the third example depicted in Figure 4.

Figure 4: Exercise 3