UNIVERSITY OF TEHRAN
Electrical and Computer Engineering Department
Object-Oriented Modeling of Electronic Circuits, Spring 1400
Computer Assignment 2, Week 4-5
Fault Simulation Gate Classes (Preliminary)

Name:
Date:

In this homework, you will be developing gate classes for performing fault collapsing and fault simulation using a fault models called stuck-at fault model. The gate-list input is to be taken from a Verilog file with the format discussed in Computer Assignment 1. For the purpose of this assignment, timing details are not considered.

1. In this part, you are to develop classes which corresponds to the different parts of a circuit and verify their functionality.

    a. Develop a wire class of character type with a fault value, every instance of which can be uniquely identified. A wire class has two variables for *faultyWire* and *faultValue*. These values are externally set, and are read by gate classes that use the wires.
    b. Develop a NOT, NAND and NOR classes with two inputs. Each of these classes have a unique identifier, an *eval()* function, and a function that lists all wires connected to its inputs and output. Before writing a value to the output wire, the *eval()* function checks see if the output is a *faultyWire*. If so, the *falutValue* will be written, otherwise the correct value will be put on the output wire.

2. In the second part of this assignment, you should perform fault collapsing for a combinational circuit. Figures shown below illustrate the concept of fault collapsing in combinational circuits. The purpose of fault collapsing is to reduce the number of distinguishable faults. This process simplifies circuit fault analysis and test generation. To do so, Line Oriented Fault Collapsing method is used. The result of fault collapsing is a list of circuit lines using their unique identifiers and associated faults on the lines, i.e., SA0 and SA1.

    a. Generate fault collapsing gate classes and interconnecting wire classes. Gates should be NOT, NAND and NOR gates. When evaluated, the circuit using these classes will list all its line faults in an external data file. Perform fault collapsing based on table shown below.

TABLE 1 Line Oriented Fault Collapsing Table

Type of target gate	Put this (these) fault(s) on the gate's input line(s)
AND, NAND	SA1
OR, NOR	SA0
INV, BUF	None
FANOUT	SA0, SA1
XOR	SA0, SA1
Primary Output	SA0, SA1
Primary Input	None

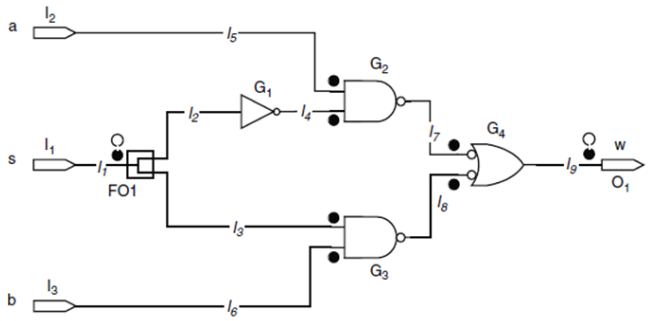

Figure 1 fault collapsing example in a circuit

3. In the third part of this assignment, you form a circuit class (Fault Collapsing, FC, class) consisting of a circuit's gate classes for fault collapsing. Another class is to be formed (Fault Simulation, FS, class) that consists of fault simulation classes of the circuit's gates. In a C++ main, use the FC class to list circuit's fault. Then use the FS class of the gate-level circuit to perform fault simulation of the circuit using the collapsed fault list and a given test vector set.
    a. In a loop, apply all tests to the circuit inputs and record good circuit outputs.
    b. Based on the fault collapsing list, Stuck-at 0 and stuck-at 1 fault should be applied to wires one by one, so a nested loop should be written. The outer loop iterates over the faults of the fault list, and the inner loop iterates over test vectors.
    c. When a test is applied to a faulty circuit, the circuit's response is compared with the good circuit's response and if a discrepancy is found, the fault is said to have been detected.

4. Write a C++ program to extract circuits used in Part 1 and Part 2 from a Verilog file with the format discussed in Computer Assignment 1.

Deliverables:

Generate a report that includes items discussed below for each of the four parts of this CA.

A. Show C++ classes and projects developed for running them.
B. When simulation is complete, or even if you have a partial simulation, include an image of the output waveform showing signal names being displayed.

Make a PDF file of your report and name it with the format shown below:

FirstinitialLastnameStudentnumber-CAn-ECEmmm

Where *nn* is a two-digit number for the Computer Assignment, *mmm* is the three-digit course number under which you are registered, and hopefully you know the rest. For the *Firstinitial* use only one character. For *Lastname* and for the multi-part last names use the part you are most identified with. Use the last five digits of your student id (exclude 8101) for the *Studentnumber* field of the report file name.