Faults Detection and Location Methods

Classification:

- Single faults:
  - Fault table method (Fixed schedule)
  - Adaptive schedule (using Diagnosing tree)
  - Path sensitizing method
  - Boolean difference method
- Multiple faults:
  - Kohavi algorithm method

Adaptive Schedule Method:

- Choice of test schedules is dependent on the outcomes of the experiment (length of test schedule may vary depending upon the fault)
- Example: If test set = {2,3,4,5}
- Then length = 4 (fixed) for fixed-schedule or fault table method
- But for adaptive, length may be 1 or 2 or 3 or 4 depending upon which fault needs to be identified.
- Uses Diagnosing tree (directed graph whose nodes are tests)
- 3 var= 8 tests possible = 8 nodes
- Levels 1 2 3 4 5 6 7 8 length=8
- Adaptive Fault table, length=4 (detection) and <=5(location)

Diagnosing Tree:

- Directed graph whose nodes are tests
- Outgoing branches from a node represent the different outcomes of the particular test.

Diagnosing Tree Preparation:

- Test set = {2,3,5,6} for fault detection

  = {2,3,6 + 1,4 or 1,5 or 4,5} for fault location

Test	x_1 x_2 x_3	f_0	f_1	f_2	f_3	f_4	f_5	f_6
0	000	0	0	0	0	0	0	1
1	001	0	0	0	0	0	1	1
2	010	0	0	1	0	0	0	1
3	011	1	1	1	1	0	1	1
4	100	0	0	0	1	0	0	1
5	101	0	0	0	1	0	1	1
6	110	1	0	1	1	1	1	1
7	111	1	1	1	1	1	1	1

Fault Detection & Location Diagnosing Tree:

- **Test set = {2,3,5,6} for fault detection**
- Length of test set= 4 (whether for fixed-schedule or adaptive-schedule method)
- Fault-free output (f0) needs to be separated using diagnosing tree.

- **Test set = {2,3,6 + 1,4 or 1,5 or 4,5} for fault location**
- Lets assume Test set = {2,3,6,4,5} , so length = 5 (fault location)
- **Minimum Length of test set = 5 {tests in any order} for fixed-schedule**
- **Length of test set = 4 {5, 3, 6, 2} or 5 {5,4,6,3,2} depending upon order of tests for adaptive-schedule method.**

Adaptive-Schedule Using Matrix Form Method:

- **Test set = {2,3,5,6} for fault detection**
- Length of test set= 4 (whether for fixed-schedule or adaptive-schedule method)
- Fault-free output (f0) needs to be separated using diagnosing tree.
- **Test set = {2,3,6 + 1,4 or 1,5 or 4,5} for fault location**
- Lets assume Test set = {2,3,6,4,5} , so length = 5 (fault location)
- **Minimum Length of test set = 5 {tests in any order} for fixed-schedule**
- **Length of test set = 4 {5, 3, 6, 2} or 5 {5,4,6,3,2} depending upon order of tests for adaptive-schedule method.**

Path Sensitizing Method:

- Fault table method requires construction of big tables if there are many lines within the circuit.
- Need to have an alternative method.
- **Principle**:

  Examine the path of transmission from the location of an assumed fault to one of its primary outputs.

Definitions:

- **Primary input:** A line that is not fed by any other line in the circuit.
- **Primary output:** A line whose signal output is accessible to the exterior of the circuit.
- **Transmission path:** Path of a combinational circuit is a connected directed graph containing no loops from a primary input or internal line to one of its primary outputs.

Steps for Path Sensitizing Method:

1. Choose a path from the faulty line to one of its primary outputs.

2. Assign a faulty line a value of ‘0’ or ‘1’ if the fault is a s-a-1 or s-a-0.

3. Along the chosen path, except the lines of path,

   Assign a value ‘0’ to the OR and NOR gates in the path.

   Assign a value ‘1’ to the AND and NAND gates in the path.

4. Trace back along the sensitized path towards the circuit inputs.

Tree-line Circuits:

- Tree-line circuit is defined as a circuit in which
  - each input is an independent input line to the circuit
  - Fan-out of every gate is 1.

Fan-out: defines number of devices/gates which can be connected at output of that particular gate/device.

The complete test set for tree-like circuits by using path sensitizing method.

Here, every path of the circuit is sensitizable.

But if the fan-out of a gate is >1, then some paths may not be sensitizable.

Boolean Difference Method:

- Algebraic method to determine set of test vectors for fault detection and location by using properties of Boolean Algebra.

- Any circuit with ‘n’ variables $F(x_1, x_2, \ldots x_n)$ has:

  F(X) representing output of fault-free circuit

  F’(X) representing output in presence of a fault

- Complete test set of test vectors for any input vector $X = (x_1, x_2, \ldots x_n)$ is $\{X | F(X) \oplus F'(X)\} = 1$

- Boolean difference of a logic function F(X) wrt to an input variable x_i is defined as:

$$\frac{d\,F(X)}{dxi} = F(x_1, x_2, \ldots\, x_i, \ldots x_n) \oplus F(x_1, x_2, \ldots\, x'_i, \ldots x_n)$$

Boolean Difference Theorems:

- Boolean difference of a logic function F(X) wrt to an input variable x_i is defined as:

$$\frac{d\,F(X)}{dxi} = \mathbf{F(x_1, x_2, \ldots x_i, \ldots x_n) \oplus F(x_1, x_2, \ldots x'_i, \ldots x_n)}$$

- F(X) when x_i assumes values ‘0’ and ‘1’ as $F_i(0)$ and $F_i(1)$

- $F_i(0) = F(x_1, x_2, \ldots 0, \ldots x_n)$

- $F_i(1) = F(x_1, x_2, \ldots 1_i, \ldots x_n)$

$$\frac{d\,F(X)}{dxi} = F_i(0) \oplus F_i(1)$$

- Test set for stuck-at-0 fault on input line x_i is $F(X) \oplus F_i(0) = 1$

- Test set for stuck-at-1 fault on input line x_i is $F(X) \oplus F_i(1) = 1$

Let us denote the values of the function $F(X)$ when x_i assumes the values 0 and 1, respectively, as $F_i(0)$ and $F_i(1)$, that is,

$$F_i(0) = F(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$
$$F_i(1) = F(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$$

Then it is easy to see that

$$\frac{dF(X)}{dx_i} = F_i(0) \oplus F_i(1) \tag{1}$$

First, we describe the method of test generation assuming a fault in one of the input lines. Later this result is generalized to a fault in any line. Consider a stuck-at-0 fault on an input line, say x_i. Then the function realized in the presence of the fault is $F'(X) = F_i(0)$. Hence, by Theorem 6.7.1, the test set for the fault can be obtained by solving the equation

$$F(X) \oplus F_i(0) = 1 \tag{2}$$

Similarly, it can be seen that the test set for a stuck-at-1 fault on input line x_i can be obtained by solving

$$F(X) \oplus F_i(1) = 1 \tag{3}$$

Now we state a theorem that provides a method for solving the equations above using Boolean differences.

Theorem 6.7.2. Equations (2) and (3) can be expressed, respectively, as

$$x_i \frac{dF(X)}{dx_i} = 1 \tag{4}$$

and

$$x_i' \frac{dF(X)}{dx_i} = 1 \tag{5}$$

Hence, the test sets for the faults SA0 and SA1 on input x_i are, respectively,

$$\left\{ X \mid x_i \frac{dF(X)}{dx_i} = 1 \right\}$$

and

$$\left\{ X \mid x_i' \frac{dF(X)}{dx_i} = 1 \right\}$$

Kohavi Algorithm Method:

- Used for multiple faults (multiple faults at the same time) in two-level networks.
- Determine two sets of tests: a-tests and b-tests.
- Three Conditions are:

1. The network must be a 2-level AND-OR or OR-AND network.
2. Each AND gate must realize a prime cube.
3. AND-OR network must implement a Boolean function, which is a *sum of irredundant prime implicants* (sum does not contain either a redundant PI or a redundant literal i.e. it is not in minimal SOP form).

Example:

Example: If a function, f= ∑m (0,1,3,5,7,8,12,12,13) has two irredundant sum forms.

f_1= 0XX1 + X000 + 110X = 0$\underline{00}$1, 0$\underline{01}$1, 0$\underline{10}$1, 0$\underline{11}$1, $\underline{0}$000, $\underline{1}$000, 110$\underline{0}$, 110$\underline{1}$

f_2= 0XX1 + 000X + X101 + 1X00 = 0$\underline{00}$1, 0$\underline{01}$1, 0$\underline{10}$1, 0$\underline{11}$1, 000$\underline{0}$, 000$\underline{1}$, $\underline{0}$101, $\underline{1}$101,

1$\underline{0}$00, 1$\underline{1}$00 (Irredundant sum term)