2.3. Fault Diagnosis

2.3.1. Basic concepts
2.3.2. Combinational fault diagnosis methods
- **2.3.2.1. Fault tables**
- **2.3.2.2. Fault dictionaries**
- **2.3.2.3. Minimization of diagnostic data**
- **2.3.2.4. Fault location by structural analysis**

2.3.3. Sequential fault diagnosis methods
- **2.3.3.1. Fault location by edge-pin testing**
- **2.3.3.2. Generating tests to distinguish faults**
- **2.3.3.3. Guided-probe testing**
- **2.3.3.4. Fault location by UUT reduction**

2.3.1. Basic Concepts

A unit under test (UUT) *fails* when its observed behavior is different from its expected behavior. Diagnosis consists of locating the physical fault(s) in a structural model of the UUT. The degree of acuracy to which faults can be located is called *diagnostic resolution*. *Functionally equivalent faults* (FEF) cannot be distinguished. The partition of all faults into distinct subsets of FEF defines the *maximal fault resolution*. A test that achieves the maximal fault resolution is said to be a *complete fault-location test*.

Repairing the UUT often consists of substituting one of its *replaceable units* (RU) referred to as a *faulty* RU, rather than in an accurate identification of the real fault inside an RU. We characterize this process by *RU resolution*. Suppose that the results of the test do not allow to distinguish between two suspected RUs U1 and U2. We could replace now one of these Rus, say U1 with a good RU, and rerun the test experiment. If the new results are correct, the faulty RU was the replaced one; otherwise, it is the remaining one U2. This type of procedure we call *sequential diagnosis procedure*.

The diagnosis process is often *hierarchical*, carried out as a top-down process (with a system operating in the field) or bottom-up process (during the fabrication of the system).

In the *top-down approach* (system → boards → ICs) first-level diagnosis may deal with “large” RUs like boards called also *field-replaceble units*. The faulty board is then tested in a maintenance center to locate the faulty component (IC) on the board. Accurate location of faults inside a faulty IC may be also useful for improving its manufacturing process.

In the *bottom-up approach* (ICs → boards → system) a higher level is assembled only from components already tested at a lower level. This is done to minimize the cost of diagnosis and repair, which increases significally with the level at which the faults are detected.

<u>The rule of 10:</u> if it costs $1 to test an IC, the cost of locating the same defective IC when mounted on a board and of repairing the board is about $10; when the defective board is plugged into a system, the cost of finding the fault and repairing the system is $100.

In manufacturing, the most likely faults are fabrication errors affecting the interconnections between components; in the field the most likely faults are physical

failures internal to components (because every UUT has been successfully tested in the past). Knowing the most likely class of faults helps in fault location.

2.3.2. Combinational Fault Diagnosis methods

This approach does most of the work before the testing experiment. It uses fault simulation to determine the possible responses to a given test in the presence of faults. The data base constructed in this step is called a *fault table* or a *fault dictionary*. To locate faults, one tries to match the actual results of test experiments with one of the precomputed expected results strored in the data base. The result of the test experiment represents a *combination* of effects of the fault to each test pattern. That's why we call this approach *combinational fault diagnosis method*. If this look-up process is successful, the fault table (dictionary) indicates the corresponding fault(s).

2.3.2.1. Fault Table

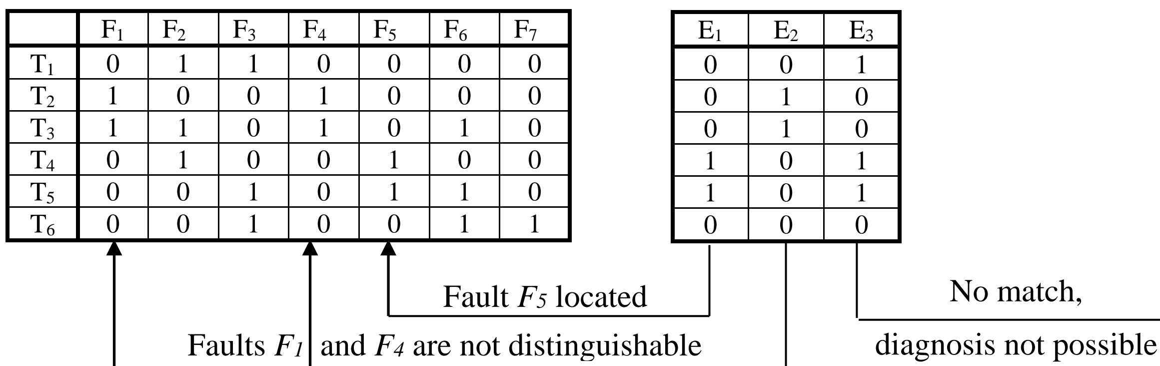
In general, a *fault table* is a matrix FT = $||a_{ij}||$ where columns F_j represent faults, rows T_i represent test patterns, and $a_{ij} = 1$ if the test pattern T_i detects the fault F_j, otherwise if the test pattern T_i does not detect the fault F_j, $a_{ij} = 0$.

Denote the actual result of a given test pattern by 1 if it differs from the precomputed expected one, otherwise denote it by 0. The result of a test experiment is represented by a vector E = $|e_i|$ where $e_i = 1$ if the actual result of the test patterns does not match with the expected result, otherwise $e_i = 0$. Each column vector f_j corresponding to a fault F_j represents a possible result of the test experiment in the case of the fault F_j.

Three cases are now possible depending on the quality of the test patterns used for carrying out the test experiment:

1. The test result E matches with a single column vector f_j in FT. This result corresponds to the case where a single fault F_j has been located. In other words, the maximum diagnostic resolution has been obtained.
2. The test result E matches with a subset of column vectors $\{f_i, f_j \dots f_k\}$ in FT. This result corresponds to the case where a subset of indistinguishable faults $\{F_i, F_j \dots F_k\}$ has been located.
3. No match for E with column vectors in FT is obtained. This result corresponds to the case where the given set of vectors does not allow to carry out fault diagnosis. The set of faults described in the fault table must be incomplete (in other words, the real existing fault is missing in the fault list considered in FT).

Example:



	F_1	F_2	F_3	F_4	F_5	F_6	F_7
T_1	0	1	1	0	0	0	0
T_2	1	0	0	1	0	0	0
T_3	1	1	0	1	0	1	0
T_4	0	1	0	0	1	0	0
T_5	0	0	1	0	1	1	0
T_6	0	0	1	0	0	1	1

E_1	E_2	E_3
0	0	1
0	1	0
0	1	0
1	0	1
1	0	1
0	0	0

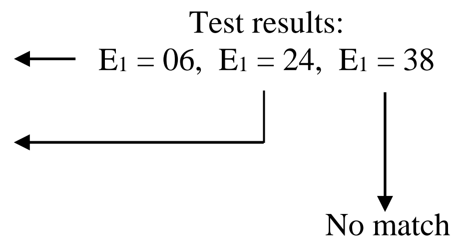
In the example the results of three test experiments E_1, E_2, E_3 are demonstrated. E_1 corresponds to the first case where a single fault is located, E_2 corresponds to the second case where a two indistinguishable faults are located, and E_3 corresponds to the third case where no fault can be located because of the mismatch of E_3 with the column vectors in the fault table.

2.3.2.2. Fault Dictionary

Fault dictionaries (FD) contain the sama data as the fault tables with the difference that the data is reorganised. In FD a mapping between the potential results of test experiments and the faults is represented in a more compressed and ordered form. For example, the column bit vectors can be represented by ordered decimal codes (see the example) or by some kind of compressed signature.

Example:

No	Bit vectors	Decimal numbers	Faults
1	000001	01	F_7
2	000110	06	F_5
3	001011	11	F_6
4	011000	24	F_1, F_4
5	100011	35	F_3
6	101100	44	F_2



2.3.2.3. Minimization of Diagnostic Data

To reduce large computational effort involved in building a fault dictionary, in fault simulation the detected faults are *dropped* from the set of simulated faults. Hence, all the faults detected for the first time by the same vector will produce the same column vector (signature) in the fault table, and will included in the same equivalence class of faults. In this case the testing experiment can stop after the first failing test, because the information provided by the following tests is not used. Such a testing experiment achieves a lower diagnostic resolution. A tradeoff between computing time and diagnostic resolution can be achieved by dropping faults after k>1 detections.

Example:

In the fault table produced by fault simulation with fault dropping, only 19 faults need to be simulated compared to the case of 42 faults when simulation without fault dropping is carried out (the simulated faults in the fault table are shown in shadowed boxes). As the result of the fault dropping, however, the following faults remain not distinguishable: {F_2, F_3},{F_1, F_4},{F_2, F_6}.

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
T_1	**0**	**1**	**1**	**0**	**0**	**0**	**0**
T_2	**1**	0	0	**1**	**0**	**0**	**0**
T_3	0	0	0	0	**0**	**1**	**0**
T_4	0	0	0	0	**1**	0	**0**
T_5	0	0	0	0	0	0	**0**
T_6	0	0	0	0	0	0	**1**

2.3.2.4. Fault Location by Structural Analysis

Assume a single fault in the circuit. Then there should exist a path from the site of the fault to each of the outputs where errors have been detected. Hence the fault site should belong to the intersection of cones of all failing outputs. A simple structural analysis can help to find faults that can explain all the observed errors.

2.3.3. Sequential Fault Diagnosis methods

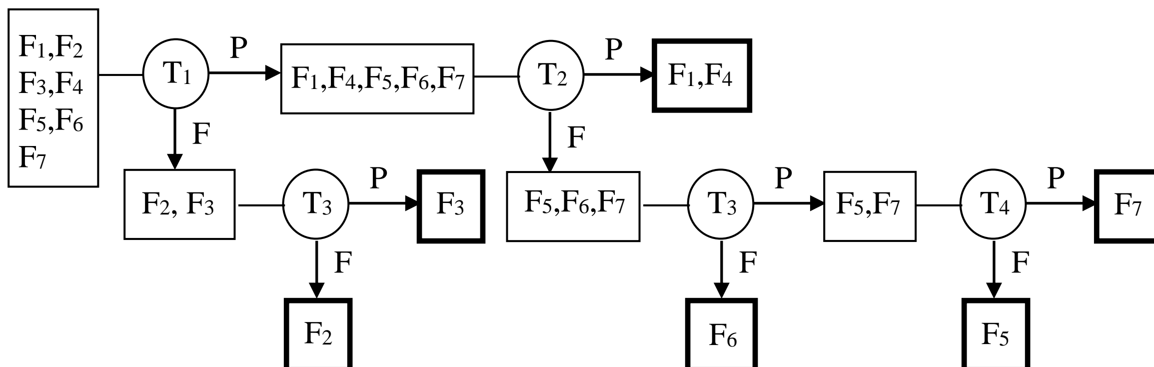
In sequential fault diagnosis the process of fault location is carried out step by step, where each step depends on the result of the diagnostic experiment at the previous step. Such a test experiment is called *adaptive testing*. Sequential experiments can be carried out either by observing only output responses of the UUT or by pinpointing by a special probe also internal control points of the UUT (*guided probing*). Sequential diagnosis procedure can be graphically represented as *diagnostic tree.*

2.3.3.1. Fault Location by Edge-Pin Testing

In fault diagnosis test patterns are applied to the UUT step by step. In each step, only output signals at edge-pins of the UUT are observed and their values are compared to the expected ones. The next test pattern to be applied in adaptive testing depends on the result of the previous step. The diagnostic tree of this process consists of the fault nodes FN (rectangles) and test nodes TN (circles). A FN is labeled by a set of not yet distinguished faults. The starting fault node is labelled by the set of all faults. To each FN k a TN is linked labelled by a test pattern T_k to be applied as the next. Every test pattern distinguishes between the faults it detects and the ones it does not. The task of the test pattern T_k is to divide the faults in FN k into two groups – detected and not detected by T_k faults. Each test node has two outgoing edges corresponding to the results of the experiment of this test pattern. The results are indicated as *passed* (P) or *failed* (F). The set of faults shown in a current fault node (rectangle) are equivalent (not distinguished) under the currently applied test set.

Example:

The diagnostic tree in Figure corresponds to the example considered in 2.3.2.1. We can see that the most faults are uniquely identified, two faults F_1,F_4 remain indistinguishable. Not all test patterns used in the fault table are needed. Different faults need for identifying test sequences with different lengths. The shortest test contains two patterns the longest four patterns



Rather than applying the entire test sequence in a fixed order as in combinational fault diagnosis, adaptive testing determines the next vector to be applied based on the results obtained by the preceding vectors. In our example, if T_1 failes, the possible faults are $\{F_2, F_3\}$. At this point applying T_2 would be wasteful, because T_2 does not distinguish among these faults. The use of adaptive testing may substantially decrease the average number of tests required to locate a fault.

2.3.3.2. Generating Tests to Distinguish Faults

To improve the fault resolution of a given test set *T*, it is necessary to generate tests to distinguish among faults equivalent under *T*.

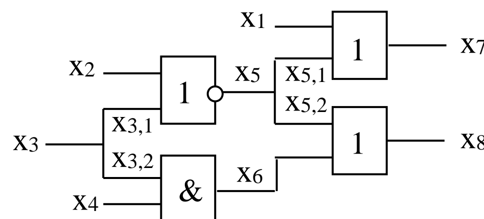
Consider the problem of generating a test to distinguish between faults *F1* and *F2*. Such a test must detect one of these faults but not the other, or vice versa. The following cases are possible.

1. *F1* and *F2* do not influence the same set of outputs. Let *OUT*(*Fk*) be the set of outputs influenced by the fault *Fk*. A test should be generated for *F1* using only the circuit feeding the outputs *OUT*(*F1*), or for *F2* using only the circuit feeding the outputs *OUT*(*F2*).
2. *F1* and *F2* influence the same set of outputs. A test should be generated for *F1* without activating *F2,* or vice versa, for *F2* without activating *F1*.

Three possibilities can be mentioned to keep a fault *F2*: $x_k \equiv e$ not activated, where x_k denotes a line in the circuit, and $e \in \{0,1\}$:

a) The value $\neg e$ should be assigned to the line x_k.
b) If this is not possible then the activated path from *F2* should be blocked, so that the fault *F2* could not propagate and influence the activated path from *F1*.
c) If the case b) is also not possible then the values propagated from the sites *F1* and *F2* and reaching the same gate G should be oposite on the inputs of G.

Example:



1. There are two faults in the circuit: *F1*: $x_{3,1} \equiv 0$, and *F2*: $x_4 \equiv 1$. The fault *F1* may influence both outputs, the fault *F2* may influence only the output x_8. A test pattern 0010 activates *F1* up to the both outputs, and *F2* only to x_8. If both outputs will be wrong, *F1* is present, and if only the output x_8 will be wrong, *F2* is present.

2. There are two faults in the circuit: *F1*: $x_{3,2} \equiv 0$, and *F2*: $x_{5,2} \equiv 1$. Both of them influence the same output of the circuit. A test pattern 0100 activates the fault *F2*. The fault *F1* is not activated, because the line $x_{3,2}$ has the same value as it would have if *F1* were present.

3. There are the same two faults in the circuit: *F1*: $x_{3,2} \equiv 0$, and *F2*: $x_{5,2} \equiv 1$. Both of them influence the same output of the circuit. A test pattern 0110 activates the fault *F2*. The fault *F1* is activated at his site but not propagated through the AND gate, because of the value $x_4 = 0$ at his input.

4. There are the same two faults in the circuit: *F1*: $x_{3,1} \equiv 1$, and *F2*: $x_{3,2} \equiv 1$. A test pattern 1001 consists the value $x_1 \equiv 1$ which creates the condition where both of the faults may influence only the same output x_8. On the other hand, the test pattern 1001 activates both of the faults to the same OR gate (i.e. none of them is blocked). However, the faults produce different values at the inputs of the gate, hence they are distinguished. If the output value on x_8 will be 0, *F1* is present. Otherwise, if the output value on x_8 will be 1, either *F2* is present or none of the faults *F1* and *F2* are present.

2.3.3.3. Guided-Probe Testing

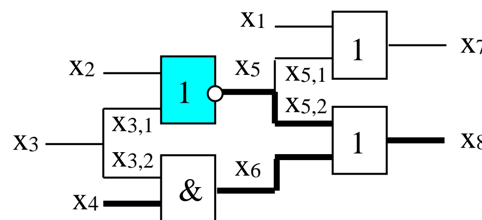
Guided-probe testing extends edge-pin testing process by monitoring internal signals in the UUT via a probe which is moved (usually by an operator) following the quidance provided by the test equipment. The principle of guided-probe testing is to backtrace an error from the primary output where it has been observed during edge-pin testing to its physical location in the UUT. Probing is carried out step-by-step. In each step an internal signal is probed and compared to the expected value. The next probing depends on the result of the previous step.

A diagnostic tree can be created for the given test pattern to control the process of probing. The tree consists of internal nodes (circles) to mark the internal lines to be probed, and of terminal nodes (rectangles) to show the possible result of diagnosis. The results of probing are indicated as *passed* (P) or *failed* (F).

Typical faults located are opens and defective components. An open between two points A and B in a connection line is identified by a mismatch between the error observed at B and the correct value measured at A. A faulty device is identified by detecting an error at one of its outputs, while only correct values are measured at its inputs.

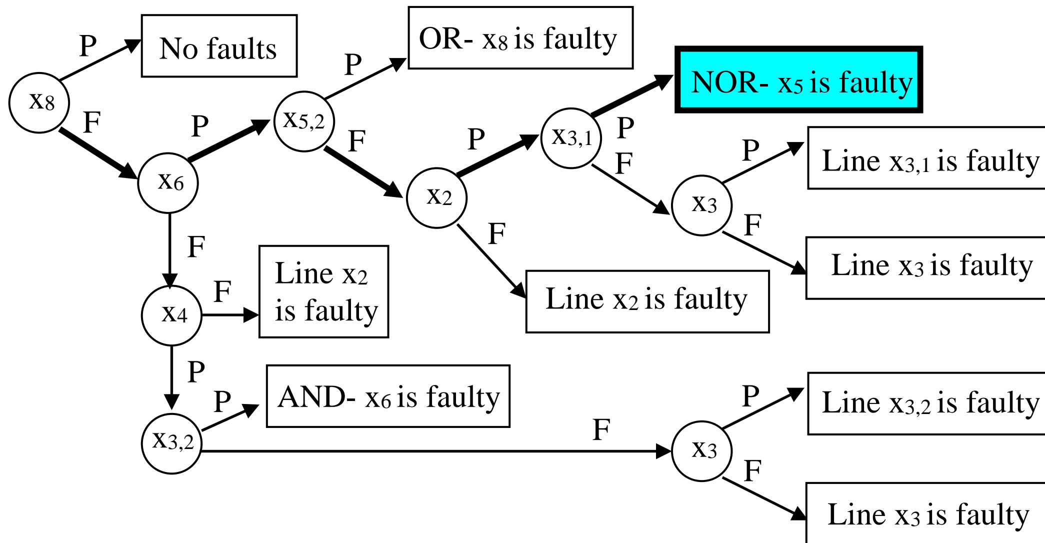
The most time-consuming part of quided-probe testing is moving the probe. To speed-up the fault location process, we need to reduce the number of probed lines. A lot of methods to minimize the number of probings are available.

Example:



Let have a test pattern 1010 applied to the inputs of the circuit. The diagnostic tree is shown created for this particular test pattern. On the output x_8 , instead of the expected

value 0, an erroneous signal 1 is detected. By backtracing (indicated by bold arrows in the diagnostic tree) a the faulty component NOR- x_5 is located.



Diagnostic tree allows to carry out optimization of the fault location procedure, for example to generate a procedure with minimum average number of probes.

2.3.3.4. Fault Location by UUT Reduction

Initially the UUT is the entire circuit and the process starts when its test fails. While the failing UUT can be partitioned, half of the UUT is disabled and the remaining half is tested. If the test passes, the fault must be in the disabled part, which then becomes the UUT. If the test fails, the tested part becomes the UUT.