

Military Technical College  
Kobry Elkobbah,  
Cairo, Egypt



5<sup>th</sup> International Conference  
on Electrical Engineering  
ICEENG 2006

## Design for Testability of Circuits and Systems; An overview

Emad H. Khalil\*, M. H. El-Mahlawy\*, Fawzy Ibrahim\* and M. H. Abdel-Azeem\*

### ABSTRACT

Integrated circuits (ICs) are reaching complexity that was hard to imagine. ICs incorporating hundreds of millions of transistors, mega-bit memories, complicated pipelined structures, etc., are in high demand. Obviously, designing such complex circuits poses real challenges to engineers. Certainly, no relief comes from the competitive marketplace, with increasing demands for a very narrow window of time (time-to-market) in engineering a ready product. Therefore, a systematic and well-structured approach to designing ICs to be testable is a must. With the growth in complexity of very large scale integration (VLSI) circuits, test generation for circuits is becoming increasingly difficult and time consuming. Even though the computing power and resources have multiplied dramatically over last few decades, an increasing number of memory elements in VLSI circuits require more effective and powerful sequential test generators. This paper is represented to review concepts and techniques for testing electronic circuits and systems as part of a lecture review.

This covers various testing and design-for-test (DFT) techniques starting from (Automatic Test Equipment) ATE basics (definition, construction and types). Exploring testing strategies for digital combinational and sequential circuits, and introduces a comparative study between the common fault models. Finally the paper ends with design for testability guiding rules and possible challenges and difficulties that need development and research in the testing problem.

**Keywords** Design-for-test (DFT), Automatic Test Equipment, Testing of electronic circuits

### 1. INTRODUCTION

In the past, the test process has been characterized as an “over-the-wall” event that occurred when the design team completed the design and threw it to a dedicated team of test or verification professionals [1]. Since the mid-1970s, the structural testing of printed circuit boards has relied very heavily on the use of the so-called in-circuit bed-of-nails technique. This method of testing makes use of a fixture containing a bed-of-nails to access individual devices on the board through test lands laid into the copper interconnects. Then testing proceeds in two phases: power-off tests check the integrity of the physical contact between nail and the on-board access point, followed by power-on tests open and short tests based on impedance measurements.

The conceptual start point of design for test was probably the first Cherry Hill Test Conference in 1970. At that time, the concept was largely unknown. There was no general perception of the problem and industry viewed the functional test approaches (Does the product perform the specification?) as adequate. This situation has been changed [2].

---

\* Egyptian Armed Forces

Creating testable design is the key to develop complex hardware and/or software systems that function reliably throughout their operational life. Without testability, design flaws may escape detection until a product is in the hands of users; equally, operational failures may prove difficult to detect and diagnose. Increased system complexity makes thorough assessment of system integrity by testing external black-box behavior almost impossible. System complexity also complicates test equipment and procedures. Design for testability should increase a system's testability, resulting in improved quality while reducing time to market and test costs. [3]

In the mid-1980s, the basic motivation for design-for-test (DFT) was the miniaturization of device packaging, the development of surface-mounted packaging, and the associated development of the multi-layer board to accommodate the extra interconnects between the increased densities of devices on the board. These factors led to a reduction of the in-circuit tester requires: physical access for the bed-of-nails probes.

A recent technological advance is the art of designing circuits to test themselves, referred to as a Built-In Self-Test (BIST) where it considered moving of the tester into the chip. This idea was first proposed around 1980 and has grown to become one of the most important testing techniques at the current time, as well as for the future.

## **2. TESTING DEVELOPMENT**

### **2.1. Testing Terminology**

Testing of a system is an experiment to get an answer for the question "Does it work properly?". In this experiment the system is stimulated and its resulting response is analyzed to be certain whether it behaved correctly or not. If incorrect behaviour is detected, a second goal of a testing experiment may be to diagnose, or locate, the cause of the misbehaviour, so that those can be eliminated.

Testing types can be classified according to many criteria. Appendix (A) summarizes the most important attributes of the testing methods. Also, Appendix (B) lists some of the basic definitions and the associated terminology used in the test field technology.

There are two main test strategies for electronic testing namely: specification-based test and fault-model based test. Specification-based test in its simplest form is testing the circuit for all its specifications. Whereas, Fault-model based test, is designed to target a specific set of modeled faults.

### **2.2. Automatic Test Equipment**

ATE can be defined as an automatic verification and failure diagnosis of electronic assemblies and sub-assemblies. It is a complete measurement system of test instrumentation and a controller, a simplified block diagram of an ATE that emphasizes the basic components is shown in Figure 1. The switching system may be a scanner or a matrix. The scanner connects one of n inputs to one of m outputs while, the matrix connects a number of inputs to a number of outputs. The fixture is a mechanical interface between the DUT (Device Under Test) and the ATE; it may be Bed-Of-Nails for In-Circuit tester or edge connector for PCB (Printed Circuit Board) functional tester.

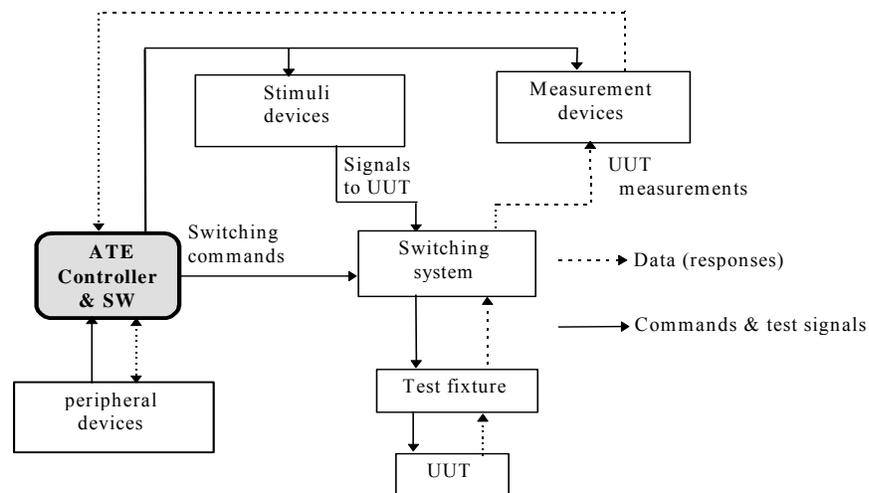


Fig. 1. Basic ATE components

Some of the advantages of ATE systems are: automatic measurements which increase testing speed; greater accuracy due to elimination of human involvement and cost reduction for high repetition rates of DUT's [4]. The disadvantages of ATE are few, but significant. They cost more than manual test sets on the average (but they often pay for themselves in a few years if they are utilized properly). Its reliability also is reduced due to complex mix of hardware and software. ATE is increasingly used by both original equipment manufacturer (OEM) and IC manufacturer (vendor), especially those with medium to high volume production. Two of the real benefits of ATE are seen as increased product reliability and reduced field service [5]. The following sections review the ATE types with some details.

### 2.2.1 ATE Types

Generally, ATE is used to test a product in conjunction with the normal stages of that product's life. Thus, ATE systems may be used in a product's design and development, production, reliability and certification test or service stage. A variety of ATE systems at various levels of cost, complexity and diagnostics ability are available. These include, starting with the most basic, the following types [5]:

#### 1- Component testers (Semiconductor, IC, VLSI, Memory or ASIC testers)

These may range from small bench-top bridges, with associated software control, to quite large parametric testers. The smallest are used for checking discrete components but the later, which may be expensive, may carry out a full parameter check on an IC before it is assembled on to the PCB. Component testers include a wide range of ATE systems that test individual components. Some of them can test digital devices including VLSI circuits, memory chips, and microprocessors. While, others specialize in analog devices such as, transistors, diodes, zener diodes, resistors or capacitors.

#### 2- Bare-board testers

These ATE systems are used to check the tracks on PCB's before assembling the components looking for short and/ or open circuits. The PCB test fixture may be expensive but very high-speed testing is possible and the success rate is much higher than visual inspection.

### 3- Manufacture Defect Analysers (MDA) or pre-screeners

It is an extension of the bare-board ATE to include check of some components. The static test pattern generated checks for continuity, short circuits, breaks, component misplacement and the correct value of some components.

### 4- In-Circuit Testers (ICT)

The idea behind the ICT is that a board should be good if each track and component is tested and found to be good. However, it is not capable of testing overall performance, in a real-time dynamic situation. Generally, the board is placed onto a "bed-of-nails" fixture that contains hundreds of spring-loaded pins that make contact at desired points on the board shown in Figure 2. The ICT may require a different fixture configuration for each tested board. Bare-board fixtures are similar to those for assembled board, except bare-board fixture requires a test probe to be positioned at each feed through hole on the circuit board. Main groups of bed-of-nails fixtures are:

- Manual; hand placement of boards.
- Mechanical; boards are clamped in position ( up to 400 probes)
- Vacuum; an air pressure reduction on one side of the board holds it in position (up to 20,000 probes)
- Pneumatic; pneumatic liquid at high pressure on one side of the board forces it into position (over 20,000 probes)

The ICT will isolate one device at a time. For passive components a guard method is used and for digital IC's the isolation technique is called node-forcing or backdriving. Each test probe could be a driver for node-forcing the inputs to a particular state, or sensor (receiver) for sensing the state of the outputs. The generated test pattern has a low drive time to avoid damaging the IC under test. The fault responses such as short circuits, misplaced components, wrongly valued components, poor soldered joints and defective tracks may be detected. The sequence of tests is as follows [5]:

- Track continuity/ short circuits
- Measurement of passive components
- Controlled powering up of board
- Test of the operation of each active component

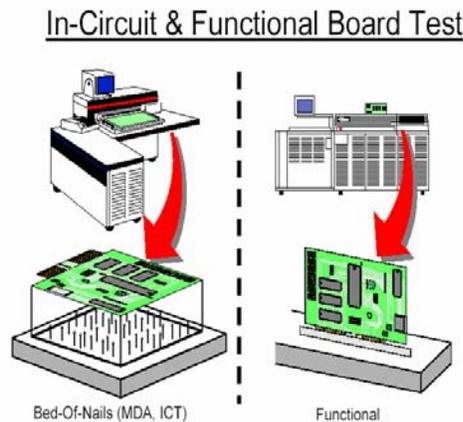


Fig. 2. In-circuit & Functional testing

The ICT disadvantages are as follow, it can not do the following [4]:

- PCB functional testing
- Complex VLSI testing
- High accuracy and full speed testing
- Parametric testing

### 5- Functional testers

A functional tester performs its tests on a DUT (board, assembly, system, etc.) by applying various stimuli at the DUT inputs, then measuring the resulting DUT response, allowing a check of overall performance to be made. Access to test nodes of the assembly is usually through the assembly's edge connectors and/ or guided probing by operator via software instructions at various points of test program. In this respect, functional testing is usually more rapid than in-circuit testing, and is often a simple pass or fail test (sometimes called GO/NO-GO testing) which may benefit in high volume production of DUTs, or it could be a full parametric tester.

The functional tester generally uses one of two methods to check the response of the DUT to the stimuli: stored response or signature analysis. With stored response, an actual bit pattern is stored in memory for direct comparison with the DUT response. In signature analysis, a data compaction technique that requires less memory is used. The stream of data is compacted into a unique "signature", usually four hexadecimal digits. If the same data passes the same point over the same period it will have the same signature every time. Dynamic testing is possible and some systems have built-in heuristic software in that they can be presented with a known 'good' board and themselves generate the test patterns [5].

### 6- Combinational testers

These testers combine the features of the ICT and the functional tester into one system. After bare-board testing, a functional ATE test strategy in addition to in-circuit test may be required on the fully completed boards. A typical flowchart of the test strategy showing combinational testing is given in Figure 3. Such a combination of in-circuit and functional testing is often chosen because of the heavy investment required. A dual test ATE strategy allows for changes without major investment in new equipment; the initial costs may be higher but the flexibility will decrease the long-term costs.

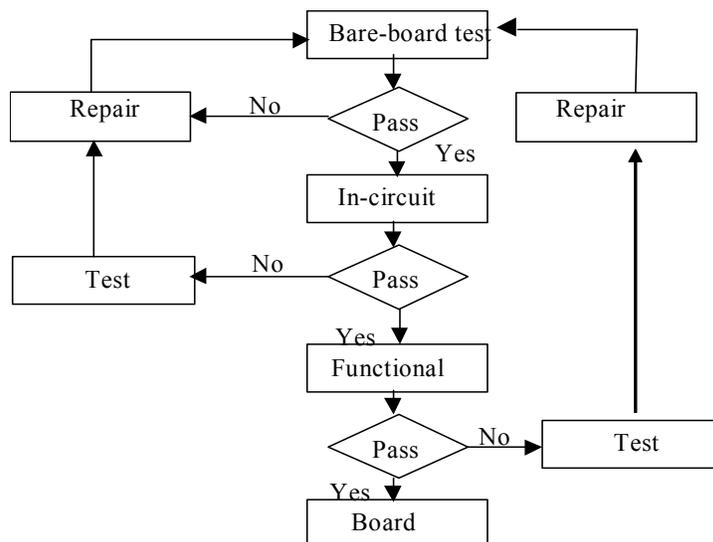


Fig. 3. Combinational test strategies

### 2.2.2 Board and Device ATE

#### (i) Board test strategies

The move to surface-mount packaging had a serious impact on the ability to place a nail accurately onto a target test land. The whole question of access was further compounded by the development of multi-layer boards created to accommodate the increased number of interconnects between all the devices.

All OEM (Original Equipments Manufactured) systems employ some form of PCB for the assembly of ICs and other components. PCB complexity ranges from very simple one- or two-sided boards to extremely complex multilayer boards containing ten or more layers of interconnect. All ATE types described in the previous section, except the component tester could be used to test the PCB. PCBs with hundreds of components may not be cost-effectively tested in one step, because fault diagnosis resolution is required in addition to fault detection. Board functional test from I/O lines is inefficient in isolating a faulty pull-up resistor or pinpointing open surface mount connections. But a test relying on internal access which is good at detecting missing components and open connections cannot determine if a board will perform its function at speed. For these reasons, loaded boards are tested in two steps: assembly test (in-circuit testing) and system functional test. In assembly testing, each component is tested to verify correct operation and connection to the substrate. With the advent of SMT (Surface Mount Technology) and densely packed boards, the access that required by bed-of-nails became difficult to achieve. For That reason, boundary scan was developed.

The common board faults that occur in production, i.e. in the manufacturing process follow approximately this pattern [5]:

- Track faults: 33%
- Faulty components: 25%
- Dry/ unsoldered joints: 12%
- Incorrectly fitted components: 10%
- Incorrect components: 8%
- Broken components: 2%
- Others: 10%

Short circuits are the largest fault category on Plated-Through Hole (PTH) boards, while open circuits are the largest fault category on SMT boards [6].

#### (ii) Device (Chip) test strategy

Fundamentally, the in-circuit bed-of-nails technique relied on physical access to all devices on a board. For plated-through-hole technology, the access is usually gained by adding test lands into the interconnects on the “B” side of the board — that is, the solder side of the board. The advent of inserted devices packaged in surface mount styles meant that system manufacturers began to place components on both sides of the board — the “A” side and the “B” side. The smaller pitch between the leads of surface-mount components caused a corresponding decrease in the physical distance between the interconnects.

The sort of test to be applied to a circuit clearly depends on what function the engineer wants the test to perform. Very broadly speaking these may be categorized into three areas [7]:

- **Development test.** For the verification of a new design and/ or process using suitable simulator
- **Production test.** For localized fabrication faults that will cause failure of the circuit in the form simple go/no-go decision.

- **Field test.** For IC failure mechanisms that are dependent on variables such as temperature, moisture and electrostatic discharge, and so a circuit that may have been functioning correctly when first manufactured will fail at some later time. This may be achieved using built-in-self-test (BIST), in which the IC may be switched to a self-testing mode to check out the functional part of the circuit.

The standard method for testing ICs is full functional test on high speed ATE. To increase the yields of packaged ICs, most manufacturers pre-test their chips in the wafer form. Wafer probe is typically done with the same ATE as final test, but with different fixturing. The biggest technical issue in IC test is the test generation effort. scan design and BIST are used to reduce test efforts [8]. The conventional test of unpackaged chips as performed by most IC suppliers today consists of a simple parametric test and a low speed functional test at the wafer level to verify if a chip is alive [9].

Testing of VLSI devices involves simulating the voltage, current and timing environment that might be expected in a real system, sequencing the device through a series of states and checking its actual response against its expected one. Where complex VLSI components (with clock rate higher than 100 MHz and pin counts over 100) needed to be tested, special testers are available. These systems use a technique called 'tester-per-pin' architecture [10]. The basic architecture of such a system has four subsystems: namely, per-pin resources, tester-control resources, computer and the system peripherals. The per-pin resources are: (i) the test vector memory board, which allocates over one million elements of test vectors for each pin. (ii) The timing generators, which generate timing with picosecond's accuracy. (iii) The pin electronics board which turns the timing and command data from the timing generator into complete waveforms. This technique results in high throughput for most VLSI circuits. These systems are very expensive and used only in very sophisticated production lines.

### 2.2.3 ATE Complementary Strategies

The major problem with in-circuit test is the decreasing physical nodal access resulting from finer board pitches, escalating pin counts and SMT chips. Because of these, manufacturers no longer have to rely on a 100% electrical test to check that their boards work. Instead, if boards have been built properly, there is an extremely high probability that they will work to specification. This new approach has opened the gates to a growing array of alternative and complementary strategies, including flying probers, boundary scan, vectorless test and automated optical inspection techniques.

#### 1-Flying (moving, Fixtureless ICT) probers

Moving probers eliminate the need of simultaneous access to perhaps thousands of test points onto a circuit board. Instead, two or more probes are moved around above the board, thus, no costly mechanical bed-of-nail fixture is required. Moving probers are common in test for bare-boards and assembled products especially analogue components. The problem with flying probers is that they are too slow only for prototype and low volume production. The mechanical positioning, lifting and re-positioning motion of its test probes limit its speed [11].

#### 2- Boundary scan

When a group of concerned test engineers in a number of European electronics systems companies got together to examine the board-test problem of limited access and its possible solutions. The group of people initially called themselves the Joint European Test Action Group (JETAG). Their preferred method of solution was to bring back the access to device pins by means of an internal serial shift register around the boundary of the device - a boundary scan register.

Later, the group was joined by representatives from North American companies and the ‘E’ for “European” was dropped from the title of the organization leaving it Joint Test Action Group, JTAG. Scan test strategies are currently defined standards IEEE 1149.1 for digital scan. This standard basically consists of adding a scan register to the inputs and outputs of ICs, four I/O ports to the chip-test access port (TAP) and TAP controller to control the testing process. Scan test is efficient at the board level to check the board interconnection and isolating a chip for functional testing without internal nodal access. If a board is going to contain some non-scan logic, using scan where possible is still effective because a non-scan chip can be tested via boundary scan if it is surrounded by scan chips [12]. A simplified IC and PCB with digital scan are illustrated in Figure 4.

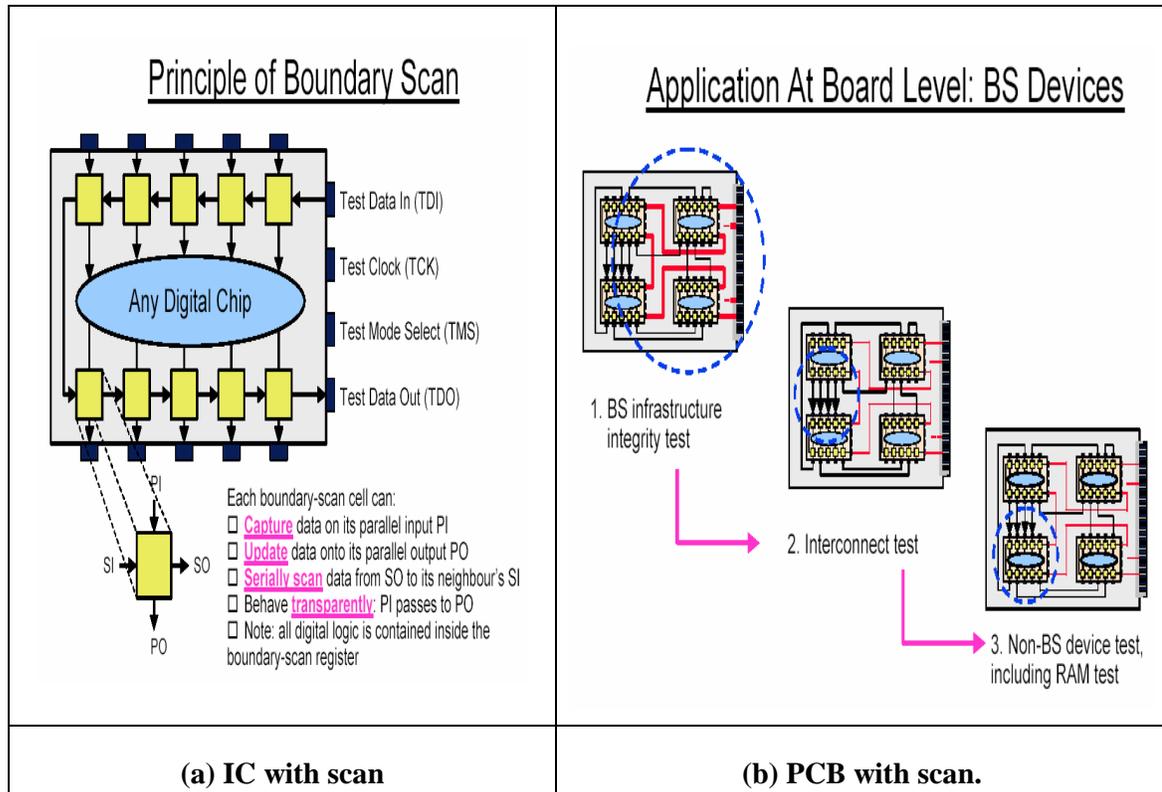


Fig. 4. Basic Scan Architecture

### 3- Vectorless test

Vectorless test methods are well suited for finding pin-related process faults on SMT boards. They use analogue stimulus/measurement approaches in contrast to traditional in-circuit digital back-drive methods, which require the application and measurement of patterns (vectors) to assess pin integrity. These power-off technologies can be considered a breakthrough in reducing test development time while improving fault coverage. The three primary vectorless techniques are analogue junction test, radio frequency (RF) magnetic inductive test, and capacitive coupling test [13].

### 4- Automated Optical Inspection (AOI)

AOI systems are a branch of machine vision systems that use image-detection hardware and image-analysis software to detect assembly defects in PCBs. AOI will detect leads that are physically separated from the solder, while ICT will discover the lifted lead if electrical contact is not made. AOI can detect some solder-paste faults (excess solder, bridging), component and component placement faults [14].

### 2.3. Digital testing methodologies

This section presents techniques that are used to detect defect in digital ICs and PCBs. While other structured digital circuits (arrays and memories) and microprocessors will not be described here. A complete chapter in [15] contains an excellent survey for testing these structured circuits. Optimum test set will detect the greatest number of defects that can be present in a device with the least number of test vectors (high defect coverage). Every digital test involves a situation such as shown in Figure 5 (a). At each step every output test response has to be checked, which requires prior knowledge of what the fault-free responses should be. For every simple circuit, particularly of SSI and MSI complexity, the procedure shown in Figure 5 (b) may be used; for complex circuits the healthy output responses may be held in memory as shown in Figure 5 (c) [15].

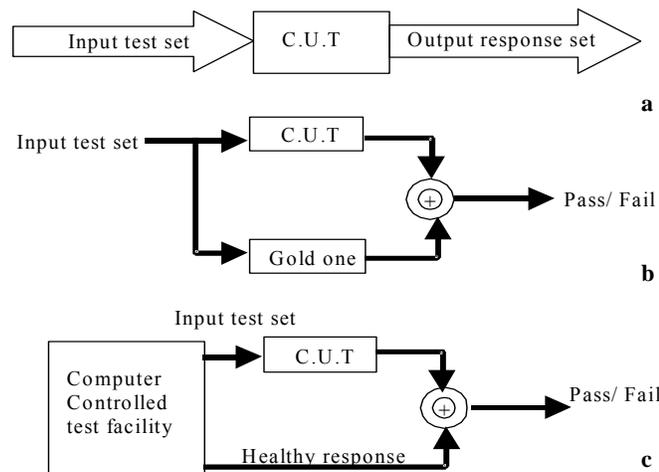


Fig. 5. Digital testing procedures

Due to the complexity of integrated circuits, the process of testing a manufactured integrated circuit has become a difficult problem. Such a problem can only be solved by a methodology approach to the testing process. A software tool called a **fault simulator** helps to meet this requirement. The purpose of a fault simulator is to determine how a circuit's input signals exercise all of the functions of the circuit. The set of input signals which drive the circuit is known as the test pattern. A critical part of the fault simulator is the type of faults which it models in the circuit.

#### 2.3.1 Fault Models in Digital Circuits

##### Why model faults?

I/O function tests are inadequate for manufacturing test, real defects (often mechanical) too numerous and often not analyzable, a fault model identifies targets for testing and makes analysis possible. Finally, effectiveness becomes measurable by experiments.

##### Common Fault Models

Single stuck-at faults, transistor open and short faults, memory faults, PLA faults (stuck-at, cross-point, bridging), functional faults (processors), delay faults (transition, path) and analogue faults

Two distinct philosophies may be used for digital circuit testing [15]:

(a) **Functional test**: test every function of the device for the correct fault free response.

For example, if a finite state machine has  $n$  primary inputs and  $m$  lines fed back from output to input, the total number of possible states is  $2^{n+m}$ . Say  $n = 24$  and  $m = 20$ , then

this results in  $2^{44}$  possible states. Testing at a rate of  $10^6$  per second means it would take 6 months of continuous test to test the circuit exhaustively.

- (b) **Fault model derived test (structured testing):** test for every modeled fault. The key is to select a fault model that can be applied to the appropriate level of circuit abstraction and that map to the most possible physical defects.

Broadly, faults can be categorized into two distinct groups. The first is catastrophic or hard faults, which cause a complete failure of a particular component, and so include such effects as short and open circuits. The second class is parametric or soft faults, here the device may still operate in its normal way, but the parameter values may be altered [7]. Table 1 shows the most common fault models for digital circuits [7, 15 and 16].

Table 1. Possible digital fault models

Fault model	Assumption	Advantages	Disadvantages
<b>Single stuck-at (SSA)</b>	One fault per circuit, permanent, the faulty node is tied to logic 1 ( <b>s@1</b> ) or to logic 0 ( <b>s@0</b> ), and all components are functioning correctly.	ATPG is well developed, can be applied to logic or module level, covers about 90% of all CMOS defects, and other fault models can be mapped to sequences of SSA faults.	Does not cover all CMOS defects
<b>Multiple stuck-at</b>	Same as SSA except that two or more faults can exist in the system	Covers some defects that SSA cannot model	ATPG is complex and not well developed and small defects could be detected that SSA can not.
<b>Stuck-open for CMOS circuits.</b>	A single physical line is broken and left floating, needs a two vector (reset vector followed by another test vector)	Covers defects not covered by SSA and multiple stuck-at fault models.	Large test sequence, ATPG is complex and not well developed, require transistor level description.
<b>Stuck-on for CMOS circuits.</b>	A transistor is permanently switched on.		Normal test vector approach is impossible
<b>Bridging fault</b>	Two circuit nodes are physically shorted together.	Covers about 30% of physical defects	Results in large test sequences, ATPGs are complex, and require transistor level description.
<b>Delay fault:</b> <b>-Gate (transitional)</b> <b>-Path delay</b>	-Slow gate output response.  - Path take to long to be exercised	-Simple, linear and can be modeled as temporary SSA.  - Delay of faulty gate can be compensated by faster gates in the path.	<b>Complex test generation.</b>

From that table, we concluded that detecting timing defects requires models other than the well known stuck at fault model. The main characteristics of these delay fault models are summarized in Table 2, where the path delay fault model is usually considered to be closest to the ideal model for delay defects so the path delay fault model is the most realistic model.

Testing all the paths in a circuit achieves 100% delay fault coverage according to traditional path delay fault coverage metrics. These metrics result in unrealistically low fault coverage if only a subset of paths is tested, and the real test quality is not reflected. [17]

Table 2. comparison of different delay fault models [17]

Delay fault model	No of faults w. r. to number of gates	Faults that can be tested	Size of detectable fault	Test generation
transition	linear	Lumped at gate	large	Modified stuck at ATPG
gate	linear	Lumped at gate	Larger than threshold	Takes timing into account
line	linear	Lumped at gate or Distributed along paths	Small to large	Requires finding longest path through the line
Path	Exponential ( <b>Worst</b> )	Distributed along paths	Small to large	hard
segment	Linear to exponential	Distributed along paths	Small to large	Depends on segment length

### 2.3.2 Test Pattern Generation

Regardless of how test vectors are developed, it is important to have some measure (Fault Coverage FC) of the quality of the generated test vectors.

**FC** is often defined as the percentage of all possible faults that the test vectors will detect. Faults can be defined on the behavioral level, or on the structural level. Once the appropriate levels chosen, a model to define a fault must be selected.

On the behavioral level, the model of faults helps determine which of the output-versus-input test vectors are redundant (this decision is also architecture-dependent).

On the structural level, most workers use the SSA model (Single Stuck At):

Fault-free structure is one in which all logic gates work properly, and all interconnections assume either logic 1 or logic 0, as necessary. Further, it is assumed that all faults (whether arising from flaws on interconnections or within the gates) manifest themselves as if the interconnection were permanently held at either 1 or 0.

#### 2.3.2.1 Test Pattern Generation for Combinational Logic Circuits

Based on the single stuck-at-fault condition, it is possible to determine the required test pattern to determine any particular fault, provided that the fault is testable. This process can be repeated for each possible fault, resulting in a maximal set of test vectors. The generation of acceptable reduced set of test vectors may be done in the following ways:

##### (a) Algorithmic (automatic) test pattern generation (ATPG)

ATPG programs normally use a gate-level representation of the circuit, with all nodes or paths enumerated. Figure 6 illustrates the concept of ATPG, which will be terminated when fault coverage *FC* has reached acceptable level [15]. There are many test vector-generating algorithms: D-algorithm, Boolean difference and FAN

##### (1) D-algorithm

The D-algorithm is due to Roth et al. (1967). It is valid for small non-redundant combinational logic circuits only, although it may be modified for sequential circuits. The algorithm is based on a structured approach to the test vector search and comprises four steps:

- (1) Fault excitation. Inputs are set up so that the net under test is driven to the logical value that is the opposite to the effect of the fault.
- (2) Fault effect propagation. To make the result of the test observable, the effect has to be propagated to the primary output. This conditioning of inputs to create a path through the circuit along which the test result is propagated is termed 'path sensitisation'.

- (3) Line value justification. The implication of the path sensitisation on the primary inputs must be examined by backtracking in order to check for any contradiction in the conditions of step 1.
- (4) Line value consistency. Once the primary input values have been determined, it is necessary to track forward through the circuit to check for any problems with multiple path sensitisation and reconvergence which can mask the propagation of a test result to an observable output.

A modification of the D-algorithm which works from the primary outputs backwards to determine a sensitive path was the LASER (logic automated stimulus and response) and the PODEM (path oriented decision making) [15].

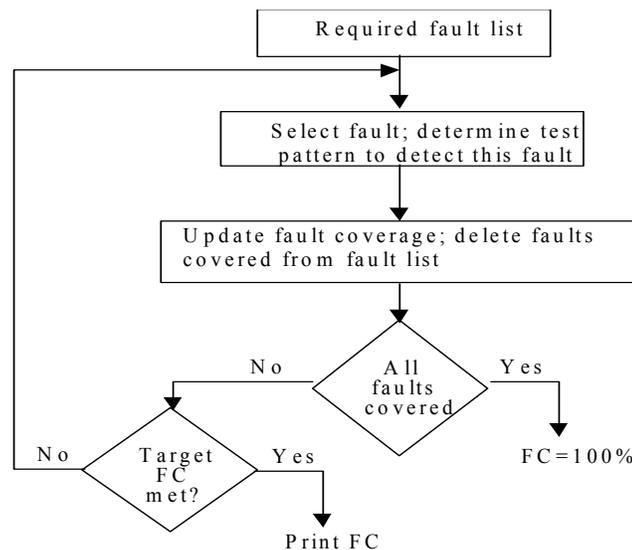


Fig. 6. The ATPG procedure

## (2) Boolean difference

It is an analytical approach that generates all possible tests for a given fault. It is based on a form of differential calculus, which in Boolean algebra is achieved through the exclusive-OR operation. The result of the Boolean difference calculation results in the complete set of tests for the stuck-at faults.

## (3) FAN Algorithm

The FAN (Fan-out Oriented) test algorithm is similar to PODEM but employs additional speedup strategies. Fujiwara and Shimono [18] proposed the FAN algorithm, which is a refinement of PODEM with the aim to reduce the number of backtracks. FAN performs special processing of fanout points and has been found to be more efficient and faster than PODEM.

## (b) Pseudorandom test pattern generation

The ATPG algorithms of the previous section are deterministic, being based upon the choice and detection of a SSA fault by an appropriate input test vector. The disadvantage is the complexity and cost of generating this minimum test set. Therefore, for a circuit with  $n$  primary inputs, it is appropriate to take very small subset of the  $2^n - 1$  pseudorandom sequence to use as the random test set. The number of faults that are covered by this test is determined by normal simulation, leaving the small percentage of faults that have not been detected to be covered by using PODEM or some other deterministic ATPG procedure.

### (c) Pseudo exhaustive test pattern generation

The pseudoexhaustive test retains almost all benefits of an exhaustive test. The choice of pseudoexhaustive test technique depends on whether or not any combinational circuit outputs depend on all of the circuit inputs. If any circuit output depends on all of its inputs, a partitioning (or segmentation) test technique must be used to test these circuits [19]. For circuits with restricted output dependency, the pseudoexhaustive test techniques provide an alternative test method. The combinational circuit with  $n$  inputs and  $m$  outputs is modelled as a direct acyclic graph. The nodes represent gates and the interconnection signals are represented by edges. Each output cone of the circuit forms a subgraph need not be disjoint. The dependency set,  $D_i$ , of the output cone  $i$  is considered the set of the primary inputs and the pseudo-primary inputs that feed it directly or affect it through another node. The dependency,  $|D_i|$ , of the output cone  $i$  is the cardinality of its dependency set. Let  $k$  be the maximum value among the dependencies of the  $m$  output cones. The circuit can be characterized as an  $(n, m, k)$  circuit. The circuit is segmented into  $m$  output cones, and each cone is tested exhaustively. The test ensures detection of all irredundant combinational faults with a single pattern within individual cones of the circuit without fault simulation. The time required for pseudoexhaustive testing depends on the sizes of the output cones. So pseudoexhaustive testing reduces the testing time to a feasible workable value while retaining many of the advantages of exhaustive testing. Many test pattern generators have been proposed for pseudoexhaustive testing, where the circuit is segmented into  $m$  output cones, and each output cone is tested exhaustively.[20] The test ensures detection of all detectable combinational faults within the individual output cones of the circuit without the need for the fault simulation[30].

#### 2.3.2.2 Test Pattern Generation for Sequential Circuits

In a typical ATPG process a test or test set is evaluated by the length of the test sequence or the number of test vectors in it, and by the number of faults covered by the test.

Ibarra and Sahni [21] showed that the problem of generating a test for a combinational circuit is NP-complete. While the test generation for purely combinational circuits is challenging due to the high circuit complexity of VLSI circuits, additional memory elements in sequential circuits add even more to the complications. This additional complication calls for the use of clever heuristics in order to yield acceptable fault coverage.

Sequential circuits may be classified as synchronous or asynchronous. In the synchronous sequential circuits, inputs are allowed to change only during periods when output changes are disabled by the synchronizing signals, called clocks. Asynchronous circuits are designed to operate without the synchronizing signals. Test generation of asynchronous circuits is not addressed in this survey. Several different approaches have been proposed for sequential circuit test generation. These can be divided into the following different classes. The first approach, which is structure-based, uses a deterministic algorithm by breaking feedback loops in the sequential circuit based on the circuit structure. An iterative combinational array containing copies of the original sequential circuit is used for the structure-based approach. The second approach is a state transition graph-based (STG-based) algorithm, and it uses an abstract model, called finite-state machine, describing the behavior of the circuit. The third approach is simulation-based and it uses the simulation under guidance of cost functions. Other approaches to sequential test generations are random, weighted random, functional, and expert-system methods. The functional methods and the expert-system methods require intensive user interaction, higher-level modeling libraries, and often restricted architectures. For these reasons, it has been difficult to effectively integrate them with existing design methodology. Thus, *the structure-based, state-based and the simulation-based approaches are more commonly used in sequential test generation.*

Some efforts have been made to partition the sequential circuit into an interactive cascade of one-state circuits, effectively spreading out the synchronous machine linearly in time instead of going around the one circuit model on each clock pulse, but unfortunately this introduces the equally difficult problem of having to model multiple stuck-at combinational faults. The only practical approach is partitioning and re-configuration techniques [22]. The sequential ATPG process is usually modeled as a combination of 4 processes. The first is *combinational ATPG* for the target fault on the combinational logic block. The second, *state justification*, is the process of finding a sequence of inputs that will drive the state machine from the reset (or unknown) state to the present state required by the test above. The third, *state differentiation*, is the process of finding an input sequence which will cause a different output sequence for the machine in the good state and faulty state as a result of the test found in the first step. And lastly, because the state justification and differentiation processes are typically done using the information about the fault-free machine, the fourth process, *sequential fault simulation*, is required to determine if the resulting input sequence is in fact a test for the target fault as well as other faults [23].

### 2.3.2.3 Supply Current Testing

The stuck-on fault is extremely difficult to test using the standard voltage test vector approach. The presence of that fault means that there will be a low resistance connected between the power supply and the ground. Therefore a relatively large supply current will be drawn, by monitoring the amount of quiescent supply current drawn by the IC ( $IDD_q$ ) the stuck-on fault should be detected. So, in general  $IDD_q$  testing must be as supplement to the normal vector measurement and can be considered more diagnostic than functional testing since it points to a particular gate in the circuit being faulty. Some commercial ATE hardware can directly monitor the current flows in its external leads. However, the line being monitored must be protected from voltage spikes from the ATE meter when it changes ranges. So a capacitor must be included to suppress these spikes, which means that there is a settling time associated with the measurement slowing the rate of measurement. The alternative is to have a current monitoring circuit that converts the current flow into a voltage which may be more accurately monitored. There are two approaches to this conversion on-chip and off-chip testing. The on-chip is faster but costly due to chip overhead, while the off-chip is slower but without overhead.

### 2.3.2.4 Delay Fault Testing (AC)

Delay testing is a way to confirm that the digital circuit will work at their intended maximum speed. Failures causing logic circuits to malfunction at the required clock speed referred to as delay faults. Also, called AC faults this comes from the use by certain manufacturers in data sheets. Where, DC data tables are the logical relationship and the AC data tables are the timing and other parametric specs. These delays are the cumulative delay of all gates in a path from primary inputs to primary outputs.

Test application strategy is an integral part of delay test generation. This is especially true for testing sequential circuit designs for which several strategies exist. Enhanced and standard scan schemes allow trade-offs between overheads (area and test application time) and fault coverage. Enhanced scan requires high area and test time overheads but results in higher fault coverage than standard scan techniques. [17].

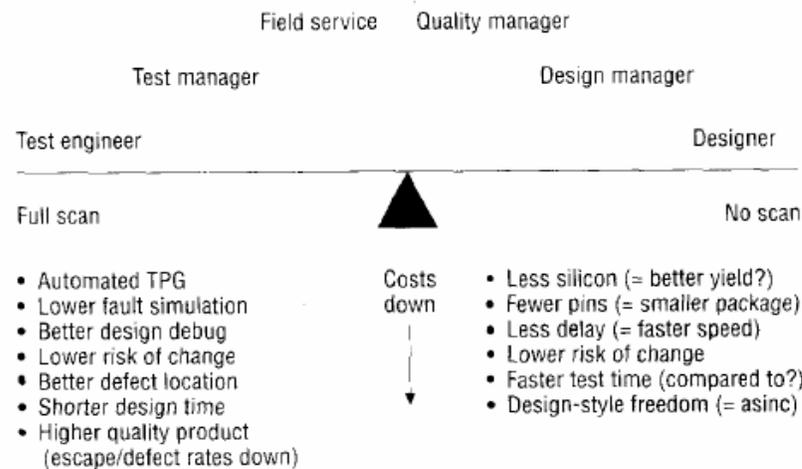
An important factor in delay test generation is the tester's speed. The speed of the testers lags behind the speed of the speed of the new design so these are important for VLSI circuit working at the leading edge of the performance of available technologies.

### 2.3.3 DFT Techniques

With the high level of possible integration today, it is feasible to consider placing all or part of the test circuitry directly on the same die as the desired circuit. The following is a brief description of some of the proposed and prototyped Design for Test techniques for making the circuits more easily tested.

DFT techniques normally fall into three general categories, namely: Ad hoc design methods; structured design methods (Scan path design and Boundary scan) and self test.

The next Figure compares the benefits of full scan with the benefits of no scan. We can perform similar comparisons between full scan and partial scan, and between boundary scan and no boundary scan. Some of the factors in the Figure are objective (the benefit of automatic and algorithmic test-pattern generation) whereas others are subjective (the benefit of better design debugging). Clearly, the test engineer and test manager will align themselves on the side of DFT, whereas the designer and design manager may fall on the side of no DFT. The quality engineer owns the bottom-line problems associated with a zero -escape and zero-defect program. [2].



#### 2.3.3.1 Ad Hoc Methods

These include partitioning of the system, degating and test points insertion. Partitioning means to partition the overall circuit or system into functional blocks, each of which may be independently tested. The penalty for the partitioning and separate test for partitions is that additional I/O pins may be required to give access to the partition boundaries. This in turn may require the addition of multiplexers within the circuit to switch lines from their normal mode to a test mode. Degating is an on-chip method of partitioning. It allows internal lines on a chip to be externally controlled in a test mode as in Figure 7. It does suffer from the fact that it adds two gate delays to each line that is degated. The test points approach is similar to degating except that the external lines are used to both observe (in normal operation) and drive the test nodes. In this case, the degating signal must be able to tri-state the internal nodes so that, they can be driven by the external test lines.

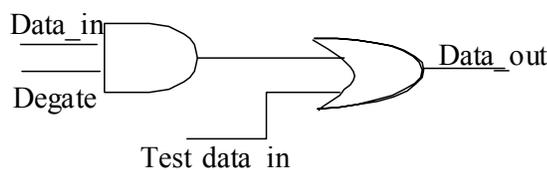


Fig. 7. Degating method

### 2.3.3.2 Structured Design Methods

Scan design is the most common structured DFT technique. In it, all of the latches in the design are made externally controllable and observable. It transforms the testing problem from sequential to combinational testing, which is a much more tractable problem. Scan testing involves switching all the storage elements of the circuit from their normal mode to a test mode shift register configuration. A scan-in I/O allow data to be read into and read out from this reconfiguration for test purposes, thus providing controllability and observability of internal nodes which would not otherwise be readily accessible. The implementation of the basic scan path is straightforward and makes use of multiplexers as illustrated in Figure 8. **The major disadvantage** of scan design is probably the speed overhead because it adds several gate delays to the critical path of the design. The testing overhead can be a big problem too because some ASIC vendors charge by the clock cycle for test application. There are a number of variations of the path approach. One, originated by IBM and termed level sensitive scan design (LSSD), involves the use of a shift register (SRL) which performs the dual function of multiplexer and flip-flop. Another variation, known as scan-set, does not make use of the system flip-flops, but has a separate scan path through which internal nodes of a block of logic can be controlled and observed. Problem: How to test an arbitrary logic which consists of both sequential and combinational circuitry. For VLSI, the injection-oriented approach is promising. Injection of test vectors is enabled by *SCAN PATH* techniques. They make the test pattern problem much easier, and most BIST structures are based on SCAN PATH techniques.

One important development of the scan path system is that of ‘boundary scan’. This is the adaptation of scan path techniques in board testing to improve testability [24].

### 2.3.3.3 Self Test BIST: Built-In Self-Test or Design for Testability

While scan path techniques enhance the testability, they still rely on the test vector input and observing the response externally. Another technique of structured DFT is to include other elements of the test process directly into the IC. This may include any or all of the elements of test vector generation, control, observation and verification. The concept of a digital circuit or network testing itself without the need to provide a separate external test vector generator and output monitoring circuit is attractive, particularly when the size and complexity of the circuit under test grows and therefore requires increasingly complex and expensive test facilities [15].

The full hierarchy of possible self-test techniques is shown in Figure 9.

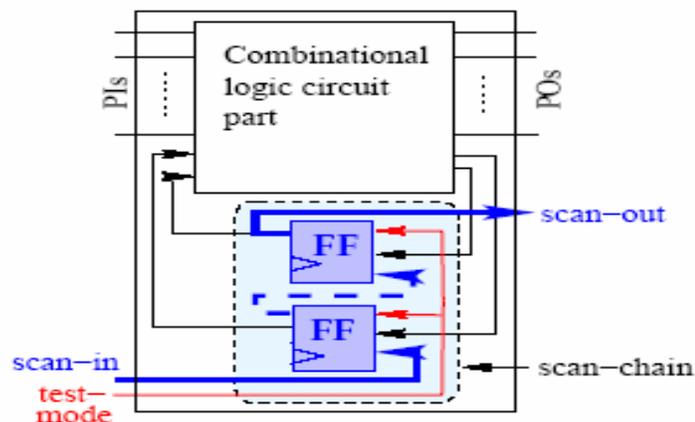


Fig. 8. Scan path considerations

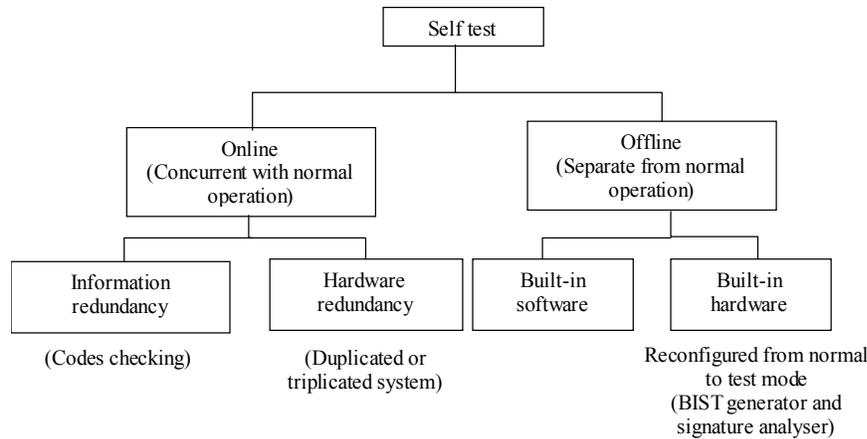


Fig. 9. Self test hierarchy

## 2.4. Design for Testability Guidelines

The following DFT guidelines are suggestions for the improvement of testability of printed circuit boards which are to be tested with the XJTAG development system [25].

- Specify and use 1149.1 compliant devices:
- Check the BSDL files for 1149.1 compliant devices.
- If designing or specifying ASICs include 1149.1 compliance.
- Ensure Low Skew between TAP signals.
- Use Correct termination for all TAP signals.
- Buffer the TAP signals.
- Make Programmable parts accessible from the 1149.1 chain.
- Design I/O interfaces with loopback capability.
- Surround logic clusters with 1149.1 compliant devices.
- Ensure that important board control signals are accessible.
- Connect driver direction control and output enable pins.
- Make full use of bidirectional pins on 1149.1 devices.
- Use spare pins on 1149.1 compliant devices.
- Consider Watchdog operation.
- Bypass optionally fitted 1149.1 compliant devices.
- Include add-on and option boards in the 1149.1 chain.
- Allow control of the clock to synchronous devices.
- Consider Testing Multi-Board Panels as one unit.
- Use non-volatile storage for configuration information.
- Make Full Use of on-board intelligent devices and advanced XJTAG features.

You can find more detailed information at [25].

## 3. TESTING PROBLEMS

### 3.1 Digital Testing Problems

The main problem of digital testing is the optimization of test data volume with time to test and decreasing number of accessible I/O terminals. Some of the specific digital testing problems are:

- Coupling of a VLSI DUT to the tester specially at high frequencies (>500 MHz) at which digital packages can act as antennas and resonators, where open lines may appear as short circuits and shorted lines as opens [3].

- With the increasing amount of circuitry on a chip there will be increasing interest in the development of MCM (Multi-Chip-Module) and SOC.
- Automatic generation of  $IDD_q$  data for test purposes.
- Delay testing is more critical as digital circuit speed and density increase.

### 3.2 Specification Based Testing Problems

Specification based testing in its simplest form consists of testing the circuit for all its specifications both during wafer-probe and final test. In practice, certain simplifications are made during wafer-probe. So, this leads to the general problem of optimal specifications-based testing. Where, the optimum test set and tolerance limit the max yield and cost. Some of the specific testing problems are [15]:

- Failure mechanisms and defect oriented testing for 3 dimensional fault modelling.
- Fault simulation in a VHDL environment.
- Automatic test data generation for circuits.
- Automatic incorporation of the IEEE standard 1149.1 scan requirements and other BIST strategies.
- Fuzzy logic expert system tools to handle test predictions and test data.

### 3.3 System-On-Chip Testing Problems

#### MCM (Multi-Chip-Module) test and diagnosis

Contrary to the conventional packaging technology (i.e., boards populated with packaged chips), an MCM typically consists of a single package containing multiple bare dies (unpacked chips) and/or discrete components with different configuration/sizes and connected to a single substrate. MCMs are primary used in military and advanced high speed consumer products [16]. It allow the use of multivendor components to mix different process technologies (bipolar, GaAs, CMOS, digital and analogue) to reduce the overall development time and cost.

In general, if a bare die is found defective after its assembly onto the MCM substrate, either the whole substrate is scrapped or the bad chip is removed. Both alternatives are often expensive and undesirable. Hence, test strategies, which result in providing bare dies with high quality prior to mounting them onto the substrate are necessary. Another problem is the difficulty to diagnose and to apply performance test at the MCM level since ATE has many limitation of speed and access of internal nodes in a MCM.

#### SOC testing

In recent years, reusable embedded modules have been used in building on-chip systems similar to using ICs on a PCB. Designers formed a rich library of pre-designed, pre-verified building blocks, the so-called embedded cores to import technology to a new system and differentiate the corresponding product by taking the intellectual property advantages. The main challenges of testing system chips are core level test, test access and system chip level test. For core level test the core provider must develops the core test (i.e., DFT structure, internal test requirement, which test method, and test pattern) and deliver it with the core. But in system on a chip the provider often does not have enough information about the target application of the component. Hence, the provided quality level might or might not be adequate. For this reason a standard format being developed by IEEE P1500 and referred to as standardization of a core test description language. Another key challenge facing the test of SOC is an electronic access mechanism is required to connect the core peripheries to the test sources and sinks. In addition to the above mentioned challenges they also have the typical

testing challenges of the very deep-submicron chips, such as defect/fault coverage, overall test cost, and time-to-market [26].

### 3.4 International Technology Roadmap General Testing Problems

The International Technology Roadmap for Semiconductors 1999 report contains one section on test and test equipment [27], which lists the five most difficult challenges through and beyond the year 2005. The most challenge will be to develop low-cost ATE for testing ICs with DFT. Summary of these five challenges is:

1. BIST and DFT for delay faults, at-speed and slow-fast-slow.
2. DUT to ATE interface for high frequency and high pin-counts.
3. Instruments for fault modelling and simulation to support BIST and DFT.
4. Failure analysis specially 3 dimensional fault analysis
5. Test development; ATPGs, test standards, reuse of core test for SOC and virtual testing.

The National Science Foundation Workshop on Future Research Directions in Testing of Electronic Circuits and Systems 1998 [28] identified emerging and mature research areas within the VLSI testing field. The emerging areas are the new and challenging areas where innovative solutions are needed. These areas include failure mechanisms, D.F.T, BIST testing.

## 4. CONCLUSIONS

The majority of industrial, commercial and scientific electronic applications require testing circuits. The testing of circuits and systems is an emerging and challenging area of research. Lists of concluded issues are as follow:

- A circuit needs to be tested after fabrication and during its life time.
- Digital circuits: combinational and sequential [29].
  - \* For a combinational circuit with N inputs, 1 output: using exhaustive test, i.e. All possible combinations ( $2^N$ ). For example a 32-bit adder, 100 MHz, 2 clock cycles per operation leads to 12,000 years. So design for testability of VLSI circuit, full exhaustive testing is not realistic because it consumes very long time.
  - \* For sequential circuits: the problem is the initial state. For Motorola M6800 (8-bits): two million years for a full functional test.
- Fault models are analyzable approximations of defects and are essential for a test methodology.
- For digital logic single stuck-at fault model offers best advantage of tools and experience where many other faults (bridging, stuck-open and multiple stuck-at) are largely covered by stuck-at fault tests.
- Stuck-short and delay faults and technology-dependent faults require special tests.
- It is suggested that path delay fault modeling should be considered in modeling of delay fault circuits.
- Since VLSI technology is not a defect-free fabrication process, Built-In-Self-Test for VLSI circuits and Boundary Scan Architecture are proven techniques for DFT.
- Structured test set is a test which should be efficient if it achieves high fault coverage and less testing time.
- There are two basic problems with the structure-based and a state transition graph-based (STG-based) algorithm approaches. First, the expansion of a sequential circuit into a combinational circuit or finding STG may result in increased complexity and memory requirement. The second problem arises due to the inability of test generation algorithms

to consider the timing behavior of gates. The structure-based and STG-based ATPGs can only be used for fully synchronous sequential

## REFERENCES

- [1] Alfred L. Crouch, Design-for-Test for Digital IC's and Embedded Core Systems, Prentice Hall PTR, (1999).
- [2] R.G. (BEN) BENNETS, Progress in Design for Test: A Personal View, SPRING Synopsys, Inc, (1994).
- [3] HARALD P.E. VRANKEN, Design for Testability in Hardware-Software Systems, Eindhoven University of Technology MARC F. WIITEMAN , RONALD C., V.WUIJTSWINKEL,KPN Research, (1996).
- [4] Brindley, K., Automatic Test Equipment, Newnes, (1991).
- [5] Loveday, G. C., Electronic testing & Fault diagnosis, Longman Scientific& technical, third Ed, (1995).
- [6] Buckroyd, A., In-Circuit Testing, Butterworth-Heinemann Ltd, (1994).
- [7] Shepherd, P.R., Integrated Circuit Design, Fabrication and Test, Macmillan press, (1996).
- [8] Flint, A., MCM Test Strategy Synthesis from Chip Test and Board Test Approaches, JETTA, Vol.10, pp 65-76, (1997).
- [9] Zorian, Y., Fundamentals of MCM Testing and Design-for-Testability, JETTA, Vol. 10, pp 7-14, (1997).
- [10] Kularatna, N., Modern electronic test and measuring instruments, The Institution of Electrical Engineers, London, (1996).
- [11] Wools-king, A., Fixtureless testing, Test, pp 9-10.Nov. (1999).
- [12] IEEE Computer society, Standard test access port and boundary scan architecture, (1993).
- [13] Pynn, C.T., Vectorless Test Boosts Fault Coverage and Cuts Cycle Time, Evaluation Engineering, Vol.34, No. 8, pp. 37-41, Aug. (1995).
- [14] Forbes, J., How to select an AOI system, Evaluation Engineering, Vol. 37, No. 9, pp. 46 (4 pages), Sept. (1998).
- [15] Hurst, S.L., VLSI Testing Digital and mixed analogue/digital techniques, The Institution of Electrical Engineers, London, (1997).
- [16] Klenke, B., Test technology overview: Module 43, tutorial, (1998), [http://www.cedcc.psu.edu/ee497i/rassp\\_43/](http://www.cedcc.psu.edu/ee497i/rassp_43/)
- [17] Angela Krstic and Kwang-Ting (Tim) Cheng, Delay Fault Testing For VLSI Circuits, Kluwer Academic Publishers, (1998).
- [18] H. Fujiwara, T. Shimono, On the acceleration of test generation algorithms, IEEE Trans. Computer. 32 (12) 1137—1144, (1983).
- [19] Mohamed H. El-Mahlawy, Pseudo-Exhaustive Built-In Self-Test for Boundary Scan, Ph.D. thesis, Kent University, U.K., (2000).
- [20] Dimitrios Kagaris and Spyros Tragoudas, "Cost-effective LFSR synthesis for optimal pseudoexhaustive BIST test sets," IEEE transactions on very large scale integration systems, Vol. 1, NO. 4, pp. 526-536, Dec. (1993).
- [21] P.H. Ibarra, S.K. Sahni, Polynomially complete fault detection problems, IEEE Trans. Computer. 24 (3) 242—249, (1975).
- [22] E. Auth, M.H. Schulz, A test-pattern-generation algorithm for sequential circuits, IEEE Des. Test Computer. 8 (2) 72—86, (1991).

- [23] M.H. Schulz, E. Auth, ESSENTIAL: an efficient self-learning test pattern generation algorithm for sequential circuits, Proc. ITC 28—37, (1989).
- [24] Parker, K.P., The Boundary Scan Handbook: Analog and Digital, Kluwer academic, 2nd edition, (1998).
- [25] Design For Testability Guidelines, [www.xjtag.com](http://www.xjtag.com) ,Version 2.0 21st January 2004, XJTAG, The Irwin Centre ,Scotland Road, Dry Drayton , Cambridge,
- [26] Zorian, Y., Marinissen, E.J., and Dey, S., Testing Embedded-Core Based System Chips, IEEE International Test Conference, (1998).
- [27] International Technology Roadmap for Semiconductors, 1999, [http://www.itrs.net/1999\\_SIA\\_Roadmap/Test.pdf](http://www.itrs.net/1999_SIA_Roadmap/Test.pdf)
- [28] National Science Foundation Workshop on Future Research Directions in Testing of Electronic Circuits and Systems, (1998),
- [29] Hawkins et al, The VLSI Circuit Test Problem -A Tutorial, IEEE Trans. Industrial Electronics, Vol. 36, No 2, May (1989).
- [30] Mohamed H. El-Mahlawy, and Winston Waller, " A New Single Test Pattern Generator for Pseudoexhaustive Testing.", 11th International Conference on Aerospace Sciences & Aviation Technology, Military Technical College, Egypt, pp. 989-1002, 15-17 May (2005).

### Appendix (A)

#### Types of tests [25]

CRITERION	ATTRIBUTE OF TESTING METHOD	TERMINOLOGY
When is testing performed?	With normal system operation.	On-line testing.
	As a separate activity.	Off-line testing.
Where is the source of the stimuli?	Within the system itself.	Self-testing.
	Applied by an external device.	External testing.
What do we test for?	Design errors.	Design verification testing.
	Fabrication errors.	Acceptance testing.
	Fabrication defects.	Burn-in.
	Infancy physical failures.	Quality-assurance testing.
What is the physical object being tested?	Physical failures.	Field testing.
	IC.	Component-level testing.
	Board.	Board-level testing.
	System.	System-level testing.
How are the stimuli and/or the expected response produced?	Retrieved from storage.	Stored-pattern testing.
	Generated during testing.	Algorithmic testing. Comparison testing.
How are the stimuli applied?	In a fixed order.	
	Depending on the obtained results.	Adaptive testing.
How fast are the stimuli applied?	Slower than normal speed.	DC (static) testing.
	At normal speed.	AC testing (at-speed testing)
What are the observed results?	The entire output patterns.	
	Some functions of the output patterns.	Compact testing.
What lines are accessible for testing?	Only the I/O lines.	Edge-pin testing.
	I/O and internal lines.	Guided-probe testing. Bed-of-nails testing. Electron-beam testing. In-circuit testing. In-circuit emulation.

## Appendix (B)

### Basic testing definitions

Terminology	Description
A BIST	1) A form of memory BIST (Built in Self Test) for embedded circuits. 2) A form of BIST targeted at testing analog circuits
AC Scan	Form of scan test application, where only the sample interval is required at the specified operating frequency in order to verify timing performance, as well as structural content. The scan data may be shifted at a different frequency (typically slower). AC scan allows slower testers to be utilized and does not place unnecessary constraints on the design to be able to shift at-speed.
APG	Tester resource that generates vectors on-the-fly, programmatically (algorithmically) , usually for memory test.
Asynchronous	An action that takes place at an arbitrary time, without synchronization to a reference timer or clock.
ATE	Automatic Test Equipment. An automated, usually computer-driven, approach to testing semiconductors, electronic circuits, and printed circuit board assemblies.
ATPG	Tool-based approach to test pattern or program development that relies heavily on the design database. DFT and ATPG go hand-in-hand.
At-Speed Scan	Form of scan where both the data shift and sample occur at the rated frequency of operation. Structure and timing performance can both be verified with this kind of scan test.
BILBO	A BILBO is a multitasking logic circuit that can be a state register, a scan register, an LFSR, or a MISR depending on the state of its mode pins. BILBOs are sometimes used to cascade large combinational logic blocks in a BIST engine.
BIST	(Built in Self Test) BIST essentially builds tiny tester models onto the integrated circuit so that it can test itself.
Boundary Scan	Generic term for IEEE 1149.1. It is a methodology allowing complete controllability and observability of the boundary (I/O) pins via a standard interface. (JTAG)
Catastrophic fault	These are faults such as open and short circuits that cause sudden and large variation in component values.
DC Scan	Form of scan where shifting and sampling occurs well below the devices normal operating frequency. This type of scan is effective for a 'pure' structural approach (i.e. for stuck-at faults) and, in general, timing performance cannot necessarily be verified with this type of scan.
Cluster	Part within an isolated partition of the tested circuit
Controllability	Is a testability metric that measures the difficulty in driving a node to a specific value
Detected fault	Fault for which a valid test vectors has been generated.
DFT	Design for testability, sometimes called design for test and almost always abbreviated to DFT, is the philosophy of considering at the design stage how the circuit or system shall be tested, rather than leaving it as an exercise at the end of the design phase
EDA	Electronic design automation. EDA refers to the design tools and environment utilized to render the logic, schematics, insert scan, insert BIST, etc. for a new chip design.

Fault collapsing	The process of test vector reduction. It is a pre-process stage implemented before test pattern generation.
Fault coverage	Is the ratio of detected faults to the total number of faults
Fault modeling	The translation of the processing fault into an equivalent electrical circuit for the resulting faulty component is termed
Hidden and subtle faults	Such as the tendency of an amplifier to oscillate at some frequency or for high levels of crosstalk to be present in a digital system, may not prevent the system from passing its specification but they are symptom of future trouble when the system is in use.
Input test vector	(input vector or test vector) is a combination of logic 0 and 1 signals applied in parallel to the accessible (primary) inputs of the circuit under test. It is the same as a word, but the latter term is not commonly used in connection with testing
Justification	Is the process determining the input combination necessary to drive an internal circuit node to a specified value (consistency)
Observability	Is a testability metric that measures the difficulty in propagating the value on node to a primary output
Propagation	Is the process of driving the circuit under test to a state where the error becomes observable at the primary outputs
Quality	The ability of an item to meet its specification
Redundant fault	Faults for which no test pattern exist (because of redundant logic in the circuit)
Reliability	The probability of quality being maintained for a stated period of time.
Sensitization	Is the process of driving the circuit under test to a state where the fault causes an actual erroneous value in the device at the point of fault E.g., for SSA faults, driving the node to the value opposite to the stuck-at value
Test pattern	Is an input test vector plus the addition of the fault-free output response of the circuit under test to the given test vector
Test set	Is a sequence of test patterns, which ideally should determine whether the circuit under test is fault free or not
Test strategies	The plan showing how the circuit is to be tested. It is basically a flowchart or algorithm of the actions required for effective testing. There are two main approaches. <ul style="list-style-type: none"> <li>• Manual test techniques</li> <li>• Using ATE</li> </ul>
Testability	Concept that the circuit or system under test must have a layout that provides for straightforward simple testing
Testable design	A circuit or system that has built-in facilities that allows simple, efficient and affective testing to be carried out. It has the following features: <ul style="list-style-type: none"> <li>• Easy access to the input and outputs</li> <li>• Test points at all critical nodes</li> <li>• Easy access to power rails and ground</li> <li>• Facilities for clock disabling and single step operation</li> <li>• Feedback disconnection (PLL)</li> <li>• External gating or control signals</li> <li>• Self-checking facilities</li> </ul>
Yield	The proportion of good devices in a lot or run