
Title: Design for Testability

Course:
Year: 1993
Author(s): J. A. Kuzmack

Report No: P93044

Abstract: This paper describes why DFT (Design for Testability) is important and some of the methods by which DFT is being implemented in the electronics industry. Also, some of the future aspects of DFT are described.

# DESIGN FOR TESTABILITY

John Kuzmack

EMP9344

# DESIGN FOR TESTABILITY

By

John A. Kuzmack

for

Dr. Richard Deckro

Professor of Engineering Management 510

Department of Engineering Management

Portland State University

June 4, 1993

## INTRODUCTION

Design for Testability (DFT) is an important and intricate part of concurrent engineering. This paper describes why DFT is important and some of the methods by which DFT is being implemented in the electronics industry. Also, some of the future aspects of DFT will be described.

## WHY DFT?

In recent years, the devices (components, boards, systems) in the electronics industry have become very complex. As electronic devices become more complex, testing of those devices becomes exponentially more complex. This is in part due to the number of Input/Output (I/O) pins per device. For example, in order to test an electronic device in an exhaustive test, all possible inputs must be applied to the device while observing all possible outputs. The observed outputs are then compared to the known correct (designed) outputs to confirm the correct operation of the device. If the device is a system containing many boards, sub-boards, and components, the complexity, and hence the cost, of the test is very difficult to justify.

The issue of design for testing has not been of critical importance until recently. This is due to the complexity, size, speed of operation, accessibility, and other test factors of the components, boards, and systems. The evolution of test has reached a point where all aspects of testing a device must be addressed in the design cycle of the product. This leads to concurrent

engineering design for testability.

## WHAT IS TESTABILITY?

According to Simpson and Sheppard, testability is

> "a design characteristic which allows the status (operable, inoperable, or degraded) of an item to be determined and the isolation of faults within the item to be performed in a timely and efficient manner." [1]

This definition applies to just about anything that can be tested, regardless of the type of test. A test is "a signal, indication, or other observable event that may be a normal output of a system or be caused to happen."[2] Tests take place during most aspects of a product life cycle. There are tests performed during the design and prototyping of a product. There are production tests, and operational tests. Tests are done to assure quality and are applied under extreme environmental conditions if necessary. There are unit tests and system tests. The maintenance group will perform tests as will the repair group. And, of course, the customer will perform, in some cases the most difficult, tests.

The main goal of DFT is to understand all the tests that will be required throughout the products life and to design the product such that all aspects of testability are optimized. The primary aspect of testability is cost but the cost of tests come in many

---

[1]William R. Simpson and John W. Sheppard, "System Complexity and Integrated Diagnostics", IEEE Design & Test of Computers, December, 1991, p. 18.

[2]Ibid.

different forms such as time-to-market, production, operation, quality, maintenance, and others.

## WHAT TYPE OF DFT?

There are many different types of design for testability techniques and implementation methods. Some of the methods and reasons for implementing a DFT plan depend upon what type of product is being tested and where. In a concurrent project, the testability goals must be defined at the beginning of the project. This definition will be the main determining factor for answering testability questions concerning how much, who, where, what, etc. Ad Hoc, Structured, Built-In Self-Test,[3] and Integrated Diagnostics[4] are the types of DFT techniques that will be described here. They are presented primarily in a manner which is similar to their development. The exception to this is Integrated Diagnostics which was developed over the last 10 years at Arinc and will be described separately.

## AD HOC DFT

Ad Hoc DFT is the earliest and simplest to implement DFT technique. Ad Hoc does not necessarily have to be an integral part of the development project. The four Ad Hoc techniques are

---

[3]Kenneth Parker, "Design for Testability-A Survey", Proceedings of the IEEE, Vol 71, No 1, January, 1983, p. 98.

[4]Simpson and Sheppard, p. 16.

to the DUT becomes more difficult to create. Also, the time spent testing and the speed of the test become more of an issue for more complex components and systems. There has been some concentration of development for the test pattern generators. Traditionally the test pattern was generated manually after the design was complete. Some of the CAD software manufacturer's are beginning to include automatic test pattern generators (ATPG) in their product.[7] This will help reduce the cost and time of testing.

Bus architecture is a method by which a board (or component) is tested through some interface.[8] The interface accesses the board bus and can write test input signals and also read test output data. This eliminates the need for complex test fixture test points but does not allow for the same controllability as test points. Not all bus architecture allow for high controllability and high observability. Similarly, a test software program must be generated.

Signature analysis actually falls somewhere between Ad Hoc DFT and Structured DFT.[9] Signature analysis can be used in the design, the manufacturing or the repair area and more closely resembles a concurrent engineering testability method. The key to signature analysis is to design a network which can stimulate

---

[7]Barbara Tuck, "EDA Leaders Getting Serious About Automatic Test Generation", Computer Design, September, 1992, p. 35.

[8]Parker, p. 101.

[9]Parker, p. 102.

itself (one that has a microprocessor).[10] Signature analysis depends upon a linear feedback shift register. By probing the shift registers on the board, the tester can observe a signature of the board which is compared to the signature of a known good board. This technique was developed by Hewlett Packard.

STRUCTURED DFT

Structured DFT is more of a progression towards concurrent engineering than Ad Hoc but the methods described here do not meet the requirements of being fully concurrent. The four Structured techniques are level sensitive scan design (LSSD), scan path, scan/set logic, and random access scan.[11] These can generally be referred to as boundary scan. Scan refers to the ability to shift data into or out of any state of the network. The goal is to have complete controllability and observability of all internal latches. The four boundary scan methods were developed at IBM, NEC, Sperry Univac, and Fujitsu respectively.

---

[10]Parker, p. 104.

[11]Parker, p. 105.

BUILT-IN SELF-TEST (BIST)

The key feature of BIST is that the hardware, through some hierarchical structure, has the capability of performing self tests. A simplified visual of BIST can be seen in Figure 1.[12]

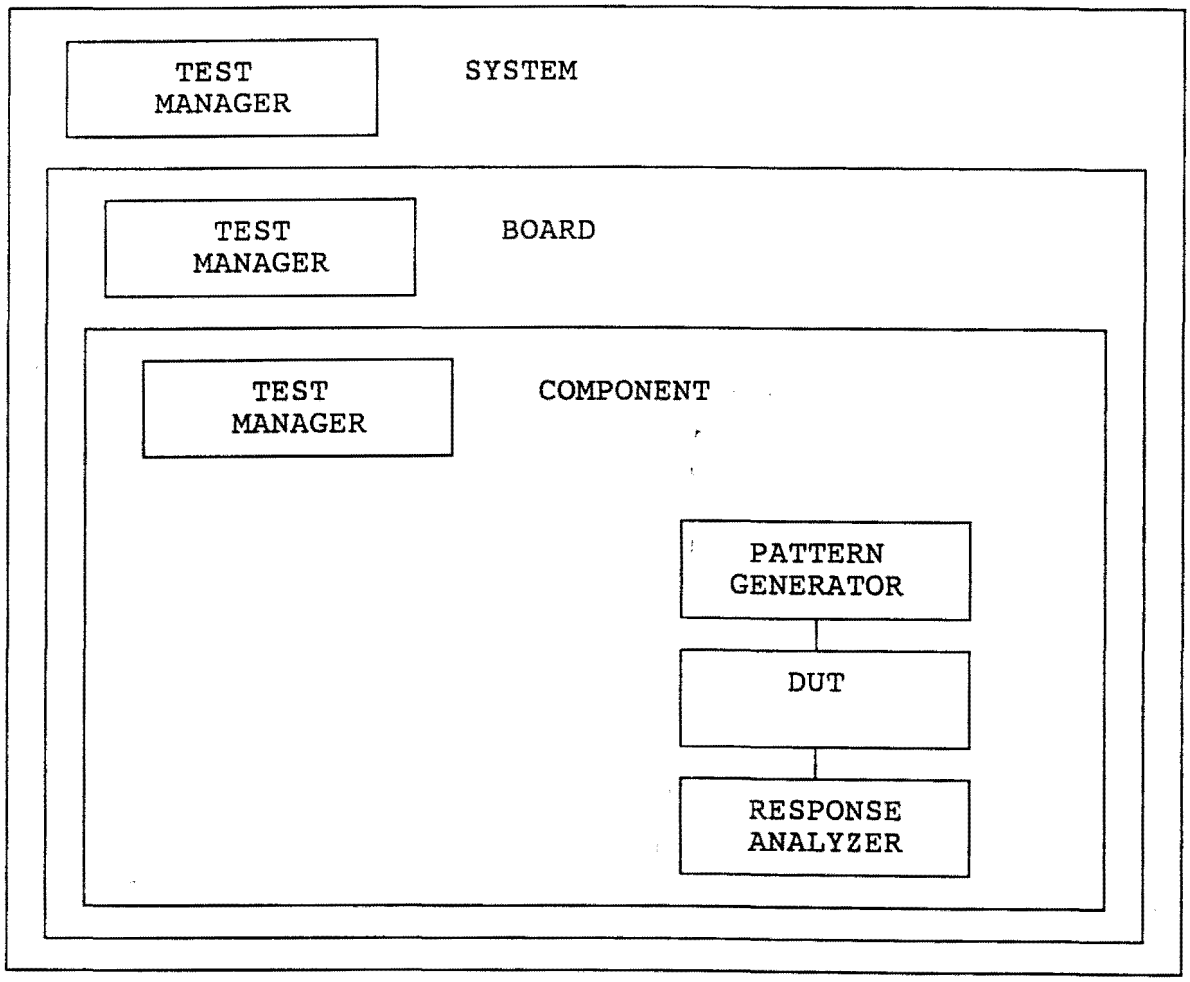


Figure 1

[12]Vishwani D. Agrawal, Charles R. Kime, and Kewal K. Salvja, "A Tutorial on Built-In Self-Test Part 1:Principles", IEEE Design & Test of Computers, March, 1993, p. 79.

The component level has the capability of generating a test pattern for that component. The component level also has the capability to determine the correct response to the input stimulus as shown by the response analyzer. The hierarchical design allows the components to report to the component test manager. The component test manager reports to the board test manager and the board manager reports to the system manager. In this manner, exact faults can be located. Also, since tests can occur simultaneously, the test time is significantly reduced. This is the ideal BIST sometimes referred to as intelligent built in test. Some of the issues of concern with BIST are fault coverage, area overhead, impact on chip yield, additional pins, and performance penalty.

Other BISTs include Built-In Logic Block Observer (BILBO) (which is a combination of LSSD, scan path and signature analysis), syndrome testing, testing by verifying Walsh coefficients, autonomous testing, and tri-state isolation.[13]

## INTEGRATED DIAGNOSTICS

Integrated Diagnostics was developed at Arinc over the last 10 years. and was extensively covered in a series of articles written for Design & Test of Computers.[14] The level of detail is beyond the scope of this paper but a general summary is described.

By definition, integrated diagnostics is "a structured process which maximizes the effectiveness of diagnostics by integrating the

---

[13]Parker, pp. 108-111.

[14]Simpson and Sheppard, p. 16.

individual diagnostic elements of testability, automatic testing, manual testing, training, maintenance aiding, and technical information."[15] The main goal is to optimize field maintenance resources within the systems operational environment. There are four strategies to meet this goal. First is to minimize the mean time to isolate system faults. The second is to minimize the mean time to repair. A third strategy is to minimize sparing requirements associated with systems. The final strategy is to reduce the required training.[16]

The structured approach of integrated diagnostics covers the following points:

- hierarchical details
- applicable at different maintenance levels
- includes information relevant to different technologies and interfaces
- represents system details to enable either on-line or off-line testing
- provides a mechanism for efficient analysis[17]

WHAT IS THE FUTURE OF DFT?

There are two main areas in the future of DFT that shall be focused on. The first is Mil Std - 2165 Testability Program for Electronic Systems and Equipment. The second is IEEE Std 1149.1

---

[15]Simpson and Sheppard, p. 17.

[16]Ibid

[17]Ibid