

PRACTICALLY REALIZING RANDOM ACCESS SCAN

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

Anandshankar S. Mudlapur

Certificate of Approval:

Adit D. Singh
James B. Davis Professor
Electrical and Computer Engineering

Vishwani D. Agrawal, Chair
James J. Danaher Professor
Electrical and Computer Engineering

Victor P. Nelson
Professor
Electrical and Computer Engineering

Stephen L. McFarland
Acting Dean
Graduate School

PRACTICALLY REALIZING RANDOM ACCESS SCAN

Anandshankar S. Mudlapur

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama

May 11, 2006

PRACTICALLY REALIZING RANDOM ACCESS SCAN

Anandshankar S. Mudlapur

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

Signature of Author

Date of Graduation

VITA

Anandshankar S. Mudlapur, son of Mrs. Durga Shivakumar and Mr. M. A. Shivakumar, was born in Bangalore, Karnataka, India. He graduated from Kendriya Vidyalaya NAL Bangalore in 1999. He earned the degree Bachelor of Engineering in Electronics and Communications from Bangalore Institute of Technology affiliated to Visvesvaraya Technological University, Bangalore, India in 2003.

THESIS ABSTRACT
PRACTICALLY REALIZING RANDOM ACCESS SCAN

Anandshankar S. Mudlapur

Master of Science, May 11, 2006,
(B.E., Visvesvaraya Technological University, 2003)

81 Typed Pages

Directed by Vishwani D. Agrawal

The number of clock cycles in a serial scan (SS) test is often prohibitive as the number of flip-flops (FF) increases. Besides, scan-in and scan-out sequences result in unwanted circuit activity. This increases the test power enormously. The scan process activates all flip-flops in the scan chain, although very few flip-flops need to be set for a targeted fault and only a subset of all the flip-flops needs to be observed. A technique known as Random Access Scan (RAS) can solve these problems. Here every flip-flop is addressed uniquely.

In RAS, only the required number of flip-flops is set or reset for a given test and this reduces the set up time of flip-flops significantly. Due to the flexibility of setting the required flip-flops randomly, the test power drastically reduces to a bare minimum. Thus two complementary problems are addressed using the single technique of RAS. These advantages come at a cost of increased area overhead and that is often unacceptable.

In this work, we have addressed the problem in such a way that the implementation is practical and the additional area overhead is justified. We have developed a new RAS cell, which minimizes the number of signals otherwise routed to it compared to earlier designs. This improvement saves silicon area.

Another contribution of this work is a new RAS cell without a scan-in signal and an added toggle feature. This flip-flop toggles its state when addressed and hence any desired state can be achieved by just addressing it if the current state is known. The scan out structure is also designed in such a way that when a flip-flop is addressed or toggled, the existing value of the flip-flop is read out. This is done using a hierarchical bus structure that drives the data from the addressed flip-flops to a primary output. Considering the limited drive capability of the flip-flops, the hierarchical bus restricts the load that the addressed flip-flop must drive.

The flip-flops are addressed using a grid structure controlled by row and column decoders. We evaluated different decoding schemes and concluded that the grid scheme requires the least routing overhead. The intersection of selected row and column addresses lines sets a flip-flop in the scan mode of operation. The address inputs to the decoders are provided from primary input pins.

Using this design we have shown that the test cycles can be reduced by 60% compared to a single chain serial scan and the test power saving can be as high as 99% compared to the serial scan. We also provide an algorithm to further decrease the test cycles.

ACKNOWLEDGMENTS

I would like to thank my advisor, Prof. Vishwani Agrawal for his guidance and direction. He has been the major source of inspiration to pursue this work and in life as a whole. I thank Prof. Adit Singh who motivated and encouraged me to pursue work related to electronic testing during my first semester of graduate study. I would also like to thank Prof. Victor Nelson for being on my committee and helping me with the numerous doubts I might have had during the course of my study. My sincere thanks to my parents without whose encouragement and numerous sacrifices I wouldn't be what I am today. The other people whom I would like to thank are my sister Ambika and my friends Srinath, Sunil, Gowri, Bikram, Ajay, Abhilash, Vidyadharan, Harish, Arun, Rohit and all my friends in Auburn University.

My special thanks to Mr. Alok Doshi for having taken a summer off and implement this work in an industrial circuit in Texas Instrument India Pvt. Ltd.

Style manual or journal used L^AT_EX: A Document Preparation System by Leslie Lamport (together with the style known as “aums”).

Computer software used The document preparation package T_EX (specifically L^AT_EX) together with the departmental style-file `aums.sty`. The images were generated using XFig.

TABLE OF CONTENTS

LIST OF TABLES	xi
LIST OF FIGURES	xii
1 INTRODUCTION	1
1.1 Problem Statement	2
1.2 Contribution of this Thesis	2
1.3 Organization of Thesis	2
2 BACKGROUND	3
2.1 Need for efficient testability	4
2.2 Basic Concept of Scan	5
2.3 Serial Scan Architectures	6
2.4 Traditional Serial Scan Design Rules	8
2.5 Limitations of Serial Scan Techniques	9
2.6 Alternate solutions to Serial Scan	10
2.7 Why Random Access Scan?	11
3 PREVIOUS WORK ON RANDOM ACCESS SCAN	14
3.1 Previous and Current Work on Random Access Scan	14
3.1.1 Ando <i>et al.</i> 's Method	14
3.1.2 Wagner <i>et al.</i> 's Method	17
3.1.3 Ito <i>et al.</i> 's Method	19
3.1.4 Baik <i>et al.</i> 's Initial Method	21
3.1.5 Baik <i>et al.</i> 's Modified Methods	26
3.2 Summary	31
4 TOGGLE RANDOM ACCESS SCAN	32
4.1 Toggle flip-flop	32
4.1.1 Design	33
4.1.2 Working	34
4.2 Decoder Design	36
4.3 Routing	37
4.3.1 Gate area overhead of RAS	38
4.4 Testing	42
4.5 Algorithm to compact test vectors	42
4.6 Results	44
4.7 Modifying ATPG to Further Decrease the Number of Vectors	45

4.8	Summary	46
5	SCAN-OUT DESIGN	48
5.1	Macro-cell design	48
6	AN EXPERIMENTAL STUDY	52
6.1	Physical Design	56
6.2	Design and Implementation Issues	56
7	CONCLUSIONS	58
7.1	Future Work	58
7.1.1	Delay Testing	58
7.1.2	Random-Pattern BIST using RAS	59
	BIBLIOGRAPHY	60
	APPENDICES	64
A	DESCRIPTION OF THE PROGRAMS USED TO IMPLEMENT THE VECTOR COMPACTING ALGORITHM	65
B	DESCRIPTION OF THE PROGRAMS USED TO CALCULATE THE POWER DISSIPATION DURING TEST	68

LIST OF TABLES

3.1	Hardware Requirements for ARAS.	19
3.2	Testability circuit size per chip [32].	21
3.3	Example vectors [7].	23
3.4	Peak and average switching activity during scan [7].	26
3.5	Circuit statistics & test data volume and test application time reduction [7].	27
4.1	RAS signals.	33
4.2	Gate overhead of RAS vs Serial Scan.	41
4.3	Results of Vector Compaction for various Benchmark Circuits.	44
4.4	Power estimation based on number of transitions at the inputs for various Benchmark Circuits.	47
A.1	Example vectors	65

LIST OF FIGURES

2.1 Sequential system.	3
2.2 Shift-register modification (from Williams and Angell 1973; ©1973 IEEE).	6
2.3 Standard D flip-flop (DFF).	6
2.4 Typical scan flip-flop (SFF).	7
2.5 A two-clock scan flip-flop.	7
2.6 A scan design schematic.	8
2.7 BIST process.	10
3.1 Input MPX type addressable latch.	14
3.2 Set/Reset type addressable latch.	15
3.3 Random access scan-In/Out network.	16
3.4 Scannable (master) latch as described in [61].	18
3.5 Delay testing between latches as described in [32].	20
3.6 Abstracted structure of RAS.	22
3.7 RAS scan-in operation.	23
3.8 Test application using RAS [7].	23
3.9 Mux based RAS [7].	24
3.10 RAM-based RAS [4].	28
3.11 Test generation procedure [5].	30
4.1 Toggle random access scan flip-flop.	34
4.2 Decoder design.	37

4.3	Design of RAS as described in [7].	39
4.4	Decoder built using pass transistors [65].	40
5.1	Macro level description of scan-out structure.	49
5.2	Scan-out Macro-cell.	50
5.3	Hierarchical scan-out illustration.	51
A.1	Vector compaction program flow.	66

CHAPTER 1

INTRODUCTION

It's a human tendency to seek perfection. But one seldom achieves it in the first attempt. It's a recursive process to attain perfection. It is no different when it comes to integrated circuits (IC) where millions of components have to work together. The guaranteed working of these components each time necessitates testing them effectively. This is a dimension in itself, as the complexity of these ICs are humanly unperceivable. Errors may occur at various steps in the life cycle of an IC. For instance the cause of failure may be due to a faulty fabrication process, or an incorrect design, or the test itself may not have been appropriate, or due to invalid specifications, or any other reason which may not be obvious. Testing can be broadly classified as two types: to verify if the device is faulty, and diagnosis, which describes what exactly went wrong.

As the complexity of a digital circuit increases, the difficulty to test it increases. Some factors that are very crucial limiting the test effectiveness are the increased chip clock rates, increased transistor density and integration of analog and digital devices onto one chip. A good test must ensure that all the parts of the circuit are working correctly. Testing is usually assisted by adding extra logic. This field of engineering is called design for testability. A circuit or a design is modified to incorporate the assistance needed for testing the remaining circuit faster and more accurately.

The extra logic added has to satisfy certain rules. It's a compromise between the ease of testing and the extra area of silicon needed. It is our endeavour here to present a design which can achieve the two complementary objectives simultaneously.

1.1 Problem Statement

The problem solved in this thesis is: *A design and algorithm to practically implement Random Access Scan.*

1.2 Contribution of this Thesis

We have developed a new flip-flop design known as ‘TOGGLE’ random access scan flip-flop to implement random access scan (RAS). This design eliminates the need of two globally routed wires namely, Scan-in and Test Control from the earlier RAS designs. We have developed an algorithm to compact the test vectors which is best suited for the ‘TOGGLE’ flip-flop.

We have shown that our method reduces the area overhead significantly compared to other existing RAS designs. An estimate of the increase in area is provided compared to serial scan. Results of the the vector compaction and test power reduction is illustrated.

Papers describing this work have been accepted at the International Test Conference (ITC) 2005 [41] and VLSI Design and Test Symposium (VDATE) 2005 [40].

1.3 Organization of Thesis

The organization of the thesis is as follows. In Chapter 2 we discuss the general concepts of testing in a broader sense and the need for design for testability. In chapter 3, we review the previous and current work in the area of RAS development in depth. In Chapter 4 we describe the design and operation of the ‘TOGGLE’ flip-flop. In chapter 5, we describe the Scan-out structure. Chapter 6 focuses on the Experimental study performed at Texas Instruments India Pvt. Ltd. and the last Chapter concludes this work and identifies areas for future work.

CHAPTER 2

BACKGROUND

A digital circuit is made up of combinational logic elements called gates and sequential logic elements called flip-flops. Any digital system can be represented as shown in Figure 2.1. The functioning of combinational logic elements are independent of the past inputs applied and depend only on the present inputs. On the other hand, sequential elements store the previous inputs in some form or other along with current inputs.

Hence a digital system has a set of discrete inputs and discrete outputs. Digital testing is the engineering to assert that the outputs obtained are a correct consequence of the input values. So testing can be defined as [63] “A process of evaluating a circuit/system to detect the presence of hardware failure due to faults and also to locate such faults to facilitate repair activities”.

Testing involves application of test stimuli, known as vectors or patterns, at the inputs of a device under test (DUT) and analysis of the corresponding responses to the applied test by collecting the data observed at the outputs of the DUT. The expected responses are

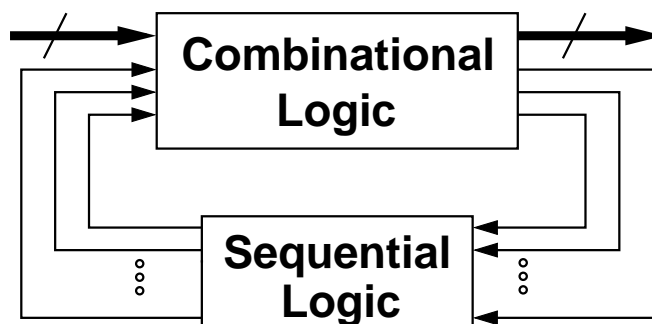


Figure 2.1: Sequential system.

matched against the actual responses from the DUT and once a deviation from the correct value is observed the device is said to have a fault. This change of actual value compared to the expected value is said to have detected the fault in the device.

A fault may be due to a physical failure or defect of one or more components in a digital circuit/system caused by the manufacturing process, extreme operating conditions, or wear out of the physical components [63].

2.1 Need for efficient testability

Testing is usually integrated with the design process. It is one of the dominating costs in an IC design process [10] amounting to 30% or more of the total cost. Hence testing is of utmost importance in the design process. The objective is to minimize time and money associated with testing.

A combinational circuit is easier to test compared to a sequential circuit. All the output states in a combinational circuit are directly controlled by the stimuli at the inputs of the circuit. In the case of a sequential circuit, the output depends on the states of the internal sequential elements too. Controlling all the states of the sequential elements is very intractable. Hence setting the circuit to any given state requires a greater effort compared to a combinational circuit.

Circuits typically are required to have a test coverage of nearly 100% before they are shipped. This guarantee depends on the fault models used and how exactly they represent the various manufacturing defects. To ensure that a circuit passes all tests at an economical cost a designer may utilize design for testability (DFT).

Electronic systems contain three types of components namely, digital logic, memory blocks and analog or mixed signal circuits. There are specific DFT techniques of each type of component such as *scan* and *partial scan* for digital logic, *built-in self-test* (BIST)

for memory and digital logic, and *boundary-scan* and *analog test bus*, to provide access to components embedded in a complex system for system level DFT.

2.2 Basic Concept of Scan

The main idea of scan design is to obtain control and observability for flip-flops. A test mode is added in the scan design when all the flip-flops functionally form one or more shift registers. These are known as scan registers. The inputs to these scan registers are coupled with the primary inputs and the outputs of the scan registers are multiplexed with the primary outputs. This way any flip-flop can be set to a desired value during the test mode by shifting appropriate values from the primary inputs. Similarly the logic states of the flip-flops are observed by shifting out the values from the scan registers.

All flip-flops can be set or observed in a time (in terms of clock periods) that equals the number of flip-flops in the longest scan register. These operations can be performed simultaneously. When one set of values from the scan registers is read, a new set of values is shifted in, which relate to the next test to be applied.

The concept of scan for hardware test was first illustrated by Williams et. al. [66]. The design is shown in Figure 2.2. In the paper the authors indicate the cost feasibility of *Shift-Register Modifications for Synchronous Circuits*.

The procedure for testing such circuits is as follows: Switch to shift-register mode and load the initial state for a test pattern into the flip-flops. Return to the normal-function mode and apply the test input pattern. Switch to the shift-register mode and shift out the final state while shifting in the starting state for the next test [23]. This way, one can design a sequential circuit such that it can be treated as a purely combinational circuit, with the flip-flop inputs and outputs treated as pseudo primary inputs and pseudo primary outputs, respectively. There are several variations of scan flip-flop/Latch designs illustrated in [22].

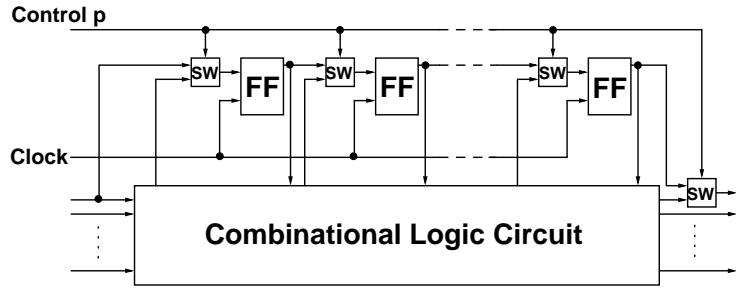


Figure 2.2: Shift-register modification (from Williams and Angell 1973; ©1973 IEEE).

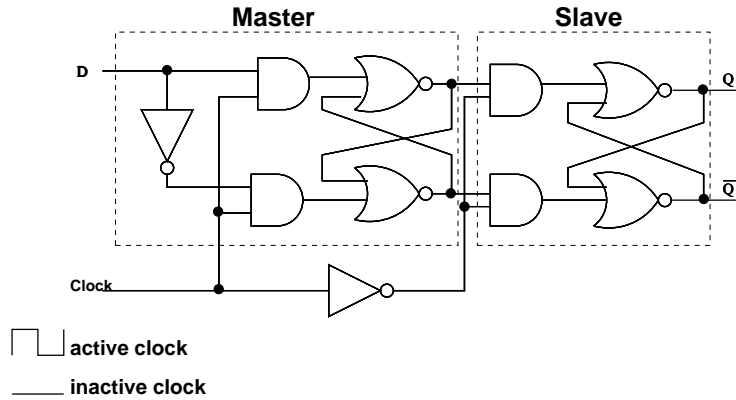


Figure 2.3: Standard D flip-flop (DFF).

2.3 Serial Scan Architectures

For a circuit to have scan capability, first the designer uses only a *D type flip-flop* (DFF) with one or more clock signals, all of which are controlled from primary inputs. A typical DFF is shown in Figure 2.3. Once the circuit is functionally verified, the DFFs are replaced by *scan flip-flops* (SFF). One typical SFF is shown in 2.4. Here a multiplexer and two new signals, scan-data SD and test control TC , are added to the DFF. The original data input D is stored in the flip-flop when TC is 1 and SD is stored when TC is 0.

Another popular design style, called *level-sensitive scan design* LSSD, uses two non-overlapping clock signals. Figure 2.5 shows a scan flip-flop with two function clocks, MCK and SCK . When MCK is high, data D is latched in the master latch. When SCK is high, the state of the master latch is copied to the slave latch. For a proper operation of a general

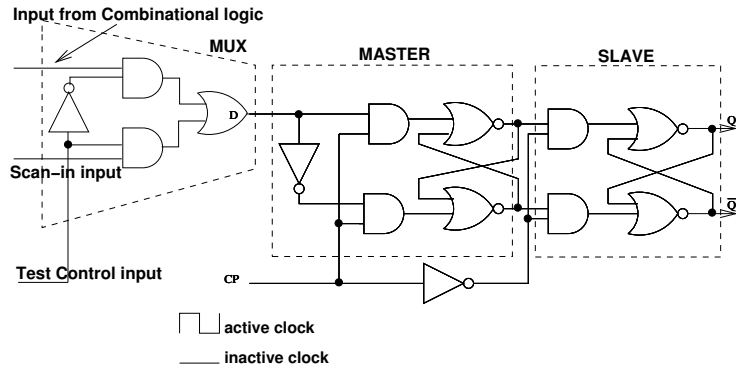


Figure 2.4: Typical scan flip-flop (SFF).

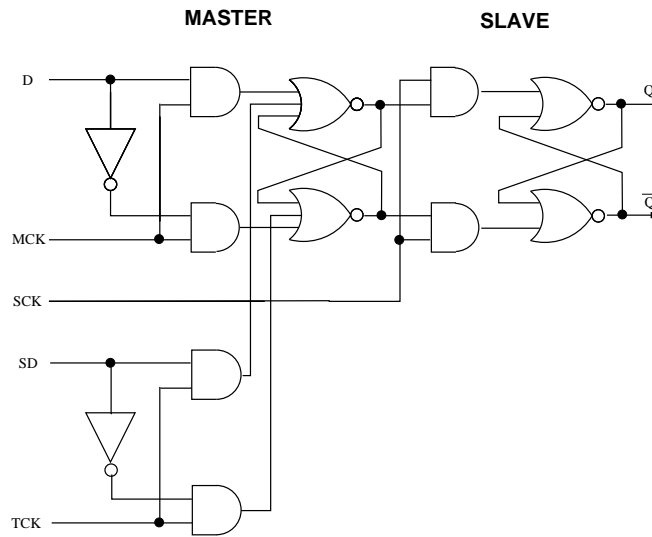


Figure 2.5: A two-clock scan flip-flop.

sequential circuit, MCK and SCK are never turned high simultaneously. In the scan mode, MCK is held low and scan data SD is latched in by using clocks TCK and SCK as master and slave clocks, respectively [22].

The TCK (or TC for the single-clock flip-flop of Figure 2.4) inputs of all scan flip-flops are supplied by a new primary input. The SD input of one SFF is supplied by another new primary input $SCANIN$. All Scan flip-flops are chained by connecting the Q output of one SFF to the SD input of the next SFF. The Q output of the last SFF in the chain is a new primary output $SCANOUT$. The complete design is given in Figure 2.6, with the wiring added for scan design shown in broken lines. This design has the advantage of reducing the

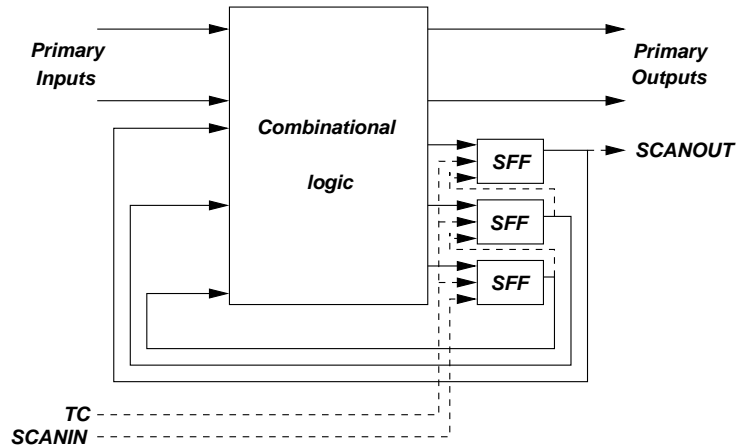


Figure 2.6: A scan design schematic.

effort of test generation, especially for the case of *full-scan*, where all flip-flops are scanned. A combinational ATPG program (much simpler than sequential ATPG) can produce tests for all stuck-at faults in the circuit. There are several results presented to enhance and reduce the test patterns in LSSD [50].

2.4 Traditional Serial Scan Design Rules

A circuit is designed to meet its functional requirements. After the functional correctness of the design is verified, it is modified to include the scan function. In order to be able to make it scan-testable, the designer must adhere to certain rules during the functional design. In general, these rules depend upon the specific design environment, which may dictate choices such as single versus multiple clocks, etc. The following four rules, however, are found to be useful:

R-1: *Only D-type master-slave flip-flops should be used.* This rule prohibits the use of other types of flip-flops (JK, toggle, etc.) or other forms of asynchronous logic (unlocked RS latches, combinational feedback elements.)

R-2: *At least one primary input pin must be available for test.* In general, flip-flops can be connected as multiple scan registers, each of which will require a scan-in and a scan-out terminal. If extra pins are not available, then any normal primary input can be used as scan-in and any primary output pin can be multiplexed as scan-out. This is illustrated in Figure 2.2 where *Control P* is the only pin added. One ordinary primary input pin serves as *SCANIN* and a primary output pin is multiplexed with *SCANOUT*.

R-3: *All flip-flop clocks must be controllable from primary inputs.* This rule is necessary for flip-flops to function as a scan register. Some violations of this rule, if they exist, can be removed by a simple work-around.

R-4: *Clocks must not feed data inputs of flip-flops.* A violation of this rule can potentially lead to a race condition in the normal mode. Thus the value captured in the flip-flop cannot be guaranteed to be the state of the signal produced by the combinational logic. In scan design, flip-flops play a dual role. They capture combinational data in the normal mode and then carry the data out for observation in the scan mode. The test procedure relies on the flip-flop correctly capturing data in the normal mode and hence no race condition is permitted.

2.5 Limitations of Serial Scan Techniques

As the number of flip-flops increase in a circuit, the setup time increases proportionally in serial scan. The values need to be serially scanned in and this process takes a very large time. This problem can be overcome if multiple scan chains are used but there are other constraints that come into picture; for example, the number of tester pins need to be minimized as much as possible. Hence a trade off has to be made with the usable number

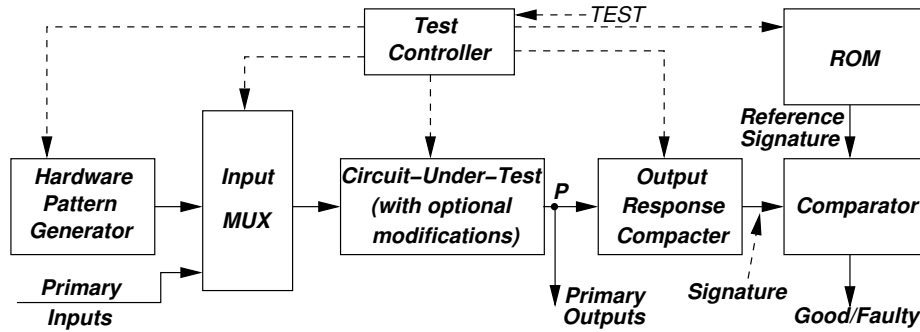


Figure 2.7: BIST process.

of tester pins. The test data volume that has to be stored in the tester is a general point of concern.

Another disadvantage of scan testing, in general, is that the scan flip-flops increase the delay through the circuit because of the addition of a multiplexer. The test also has to be performed at a slow speed in order to have full control over the circuit operation.

Performing delay testing in serial scan circuits is not very straight forward and requires other modifications. Serial scan induces unnecessary circuit activity during scan-in/scan-out. The continuous change of states causes constant power dissipation in the combinational logic. This may cause serious problems to the circuit during test. A functionally working circuit may be subjected to extreme heat dissipation, resulting in defects in the circuit.

2.6 Alternate solutions to Serial Scan

There are several alternate testing methods apart from serial scan. These include built-in self-test (BIST), random access scan (RAS), boundary scan and variations of the kind. In BIST, a *linear feedback shift register* (LFSR) is built into the circuit and is used to generate pseudo exhaustive patterns to test the circuit. The responses are collected in *multiple input signature register* (MISR) to verify the correctness of the circuit. The block diagram of the BIST process is shown in Figure 2.7.

In random access scan (RAS), every flip-flop is addressed individually. An address decoder is used to select a particular flip-flop and the required value is stored in it for testing. This scheme reduces the test time and test data volume significantly compared to serial scan. The test power, which is a serious matter of concern, is reduced drastically. But these advantages are marred since the implementation of RAS has not been practically feasible. This work is aimed at realizing the objective of RAS with a minimal increase in hardware overhead. A detailed description of RAS follows in chapter 3.

In order to invoke the BIST procedures and facilitate their correct execution at the board, module or system level, certain design rules must be applied. In 1990, a new testing standard was adopted by the Institute of Electrical and Electronics Engineers, Inc., and it is now defined as the IEEE Standard 1149.1, IEEE Standard Test Access Port and Boundary-Scan Architecture [47]. The details of it can be found in [37].

2.7 Why Random Access Scan?

Although serial scan enables the application of combinational test generation algorithm, alternative methods are sought after because of some inherent drawbacks like increased test time and test power consumption. Several methods have been suggested and implemented to circumvent this problem. A widely successful method is partial scan [1], but it provides a trade off between the ease of testing and the costs associated with scan design. The problem of efficiently selecting the scan registers is still widely open to research. Cross check methodology [14] provides a comprehensive solution to test sequential circuits and almost solves all the problems related to test application time and provides massive controllability and observability.

Power consumption during testing is much higher than during normal circuit operation. It is important and vital to maintain low power dissipation during testing, since excessive heat can damage the circuit under test. The long scan-in/scan-out sequences trigger random

circuit activity resulting in high power consumption. Test scheduling is a common approach to avoid the damage of complex devices, such as SOC [16, 17]. As a result, test parallelism is reduced and testing time eventually increases. It is a well known fact that serial scan operation may create unacceptably high activity due to frequent transitions in the scan chain. To circumvent this problem the scan clock is slowed down [12]. This increases the test application time, which is undesirable.

ATPG based methods have also been used to target the power issue [64]. However, this method often results in longer test sequences. Compaction of test vectors can reduce the length of tests, but the compacted vector set generally induces more activity, resulting in higher power consumption [51]. To overcome this problem, modification of test vectors for power saving has also been addressed [33]. Another method studied to reduce test power and/or test application time is modifying the order of scan cells or inserting inversion logic between scan cells after the test generation [19]. Seth et al., in [9], describe a double-tree scan architecture to reduce test power. Although the power saving is quite significant, the test time and test data volume either remains the same or more. A modified scan-architecture to reduce test time in full-scan circuits has been addressed in [28]. They illustrate a reduction of test time by 50%; nevertheless test power still remains a matter of concern.

Testing for path delay faults in non-scan sequential circuits is complicated by the limited state transitions during normal operation. An accepted method for overcoming this difficulty is to use a scan chain consisting of enhanced scan flip-flops which makes the application of arbitrary vector pairs possible. However this technique requires a hold-latch connected to each flip-flop in addition to a “HOLD” signal that must be routed to every hold latch. This increases the area overhead and also adds some delay in the scan path [10]. Normal-scan sequential circuits can be tested for delay faults, but the vector-pairs must be specially generated [15]. Here, the first vector V1 is scanned in (usually with a slow scan clock) and is then replaced in the scan register by either (a) applying V2 which is

obtained by a one-bit shift to the scan register also known as *scan-shift delay test* [52, 53], or (b) propagating V1 through the combinational logic in the normal mode, where the state portion of V2 may be justified by V1, known as *Functional broad-side delay test* [54]. However, high fault coverages are dependent on the circuit and cannot be guaranteed due to the correlation between the two vectors.

All the problems stated above are due to the underlying architecture used, which is serial scan! Random Access Scan (RAS) [2] is a single concurrent solution to all of them. As the name implies, each scan-cell is randomly and uniquely addressable. The architecture described in [7] targets reduction of both test application time as well as power consumption simultaneously, which are otherwise complementary objectives. A modified scheme of RAS has been described in [3], although with a different name. Here, the captured response of the previous pattern in the flip-flops is used as a template and modified by a circular shift for the subsequent pattern.

CHAPTER 3

PREVIOUS WORK ON RANDOM ACCESS SCAN

The concept of Random Access Scan (RAS) was first proposed by H. Ando [2] in 1980. Although it was a novel idea, it failed to impress researchers and industry because of the hardware area overhead associated with it. In this chapter, the literature related to previous RAS architectures are discussed starting with the earliest reference.

3.1 Previous and Current Work on Random Access Scan

3.1.1 Ando *et al.*'s Method

Ando [2] proposed an addressable latch shown in Figure 3.1. An addressable latch or flip-flop is a basic storage element in a random access scan-in/out network, which serves as a test input holding latch, test output holding latch and part of an output multiplexer while testing in addition to information storage function in normal operation.

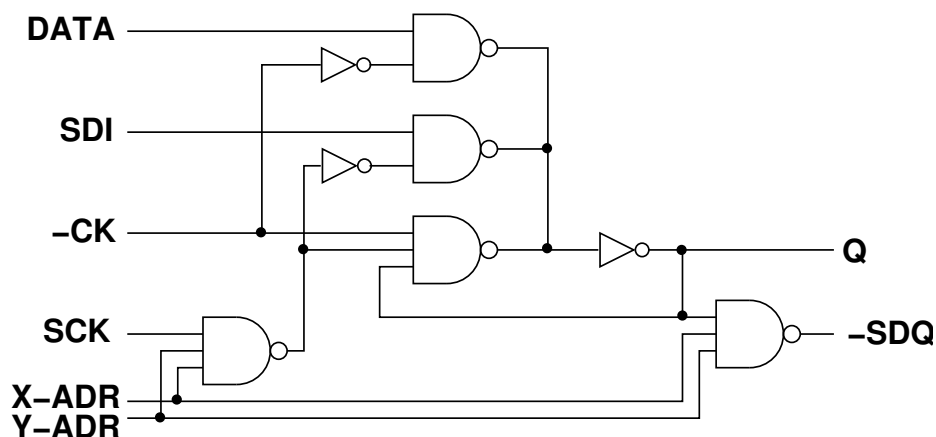


Figure 3.1: Input MPX type addressable latch.

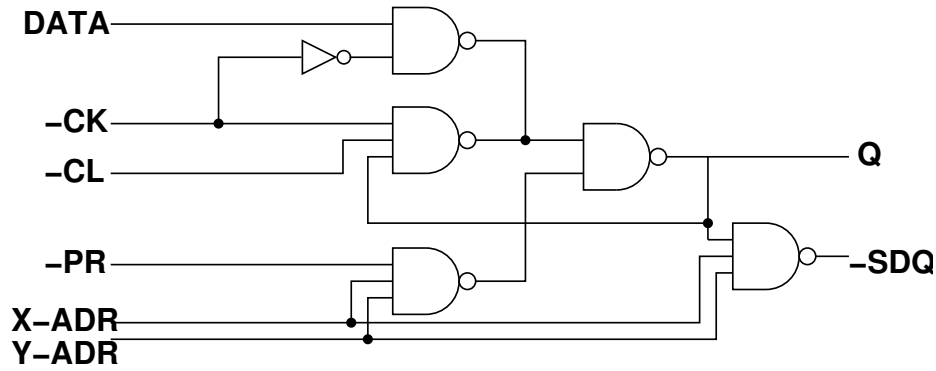


Figure 3.2: Set/Reset type addressable latch.

An addressable latch is a latch whose state can be controlled and observed through scan-in/out lines only when it is selected by some address means. Figure 3.1 is an example of an addressable latch which selects one of the two inputs and holds it depending upon which clock is applied. When the latch is selected, the test input on the SDI line is sampled and held in response to the SCK clock. The state of the latch is observable through the other address gate. Latch state $-SDQ$ signals from all latches are ANDed together to produce a chip scan out signal. Another type of addressable latch is shown in Figure 3.2. A Set/Reset type addressable latch is made of a latch and two address gates, one for gating a preset signal and the other for gating the output. Any latch or flip-flop with asynchronous preset and clear can be used. Prior to scan-in operation, all latches have to be cleared with a common $-CL$ signal. Then the latch is selected with address lines and the PR pulse is applied to flip latch state.

To complete the scan-in/out network, a pair of address decoder and AND gate trees to combine all latch state signals is necessary. One address decoder is called the X address decoder and drives X-ADR lines. The other one is the Y address decoder. An address latch which locates at the intersection of selected X and Y address lines is accessed like a memory cell in an array. Any point in combinational circuits can be observed with one

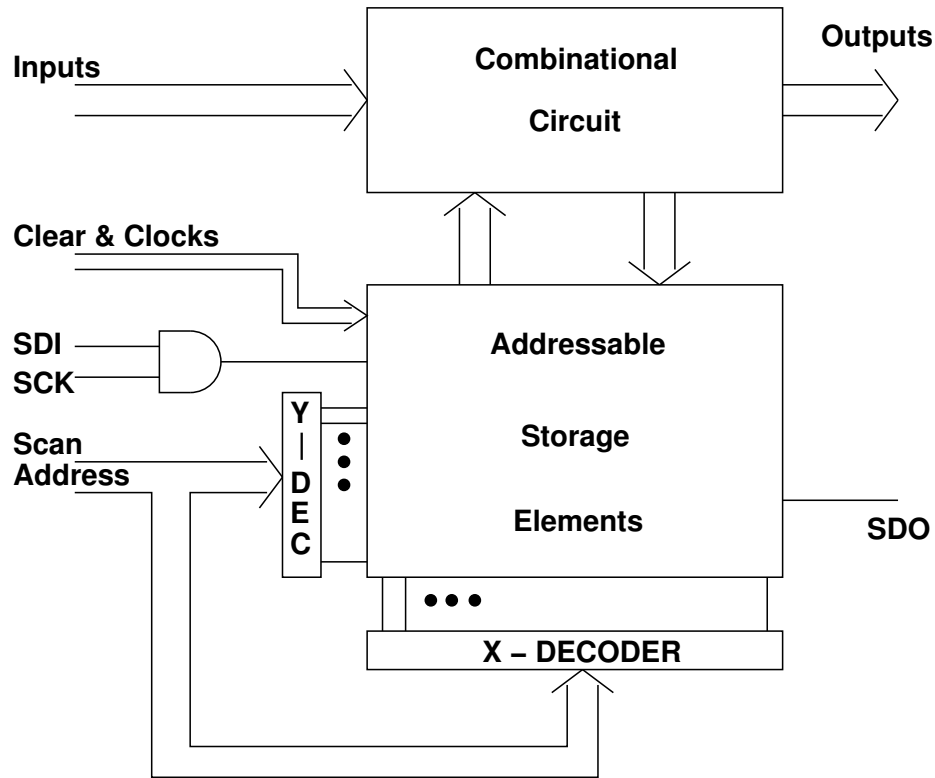


Figure 3.3: Random access scan-In/Out network.

additional gate and one address. The general description of a logic circuit with a scan-in/out network is shown in Figure 3.3. Ando also suggested that the input pin requirement could be reduced by adding a scan address register which has serial shift input capability and count-up capability. The shortcomings or investments for the use of random access scan network were:

1. Two address gates for each storage element, address decoders and output AND tree. Those total about 3 to 4 gates overhead per storage element.
2. Scan control, data and address pins are required. This 10 to 20 pins requirement can be cut down to around 6 pins with a serially loadable address counter.
3. Some logic design limitations are imposed, such as the exclusion of asynchronous latch operation.

3.1.2 Wagner *et al.*'s Method

Wagner [61] extended the idea of Ando [2], and described the implementation of Amdahl Random Access Scan (ARAS) at the (1) latch; (2) chip; (3) board, and (4) system levels. A study of the (1) cost; (2) functional value, and (3) testability benefits was performed.

The basic latch used in Amdahl was built using 8 transistors and wired-collector logic. Two gates were added to the basic latch to perform scan-in and scan-out functions shown in Figure 3.4.

Within a chip each scannable element is assigned a unique scan address consisting of a plane (PL), column (COL) and row (ROW) number. Element addresses range from PLane 0, COLumn 0, ROW 0 through PLane 3, COLumn 4, ROW 3. An address space of size 62 results since 2 addresses are not used by real elements but are instead dedicated to control. In order to scan-out this latch, $\overline{COL} = \overline{ROW} = 0$ and $\overline{DATAOUT}$ appears at *SNOP* of the Scan-Out Gate. To scan in, clocks must be inactive ($\overline{CS} = 1, \overline{CH} = 0$), *SIPH* is pulsed to 1 to set the latch, inputs $\overline{SIPL} = \overline{COL} = \overline{ROW} = 0$ must be applied to the Scan-In Gate.

Every scannable chip in the system has an on-chip scan machine. The machine receives 64 $\overline{SCANCLK}$ s accompanied by 64 SCAN DATA bits during a scan operation. The first 2 bits are used for control and the last 62 bits for serial data transfer. The scan-out operation can be performed concurrent with system operation. All \overline{SIPL} and *SIPH* element inputs remain inactive. The *SNOP* outputs for all elements in each plane are wire-ORed. At any address, the $\overline{DATAOUT}$ for the selected element is placed on its Scan-Out PLane with the guarantee that no other elements are active on this PLane. A final 4-to-1 selection between the 4 Scan-Out PLanes is performed to extract the correct SCAN DATA bit for output off chip. The scan-in operation is more complex and cannot be performed with active system clocks. Initially, all latches with scan-in receive the *SIPH* pulse, setting them. The ship

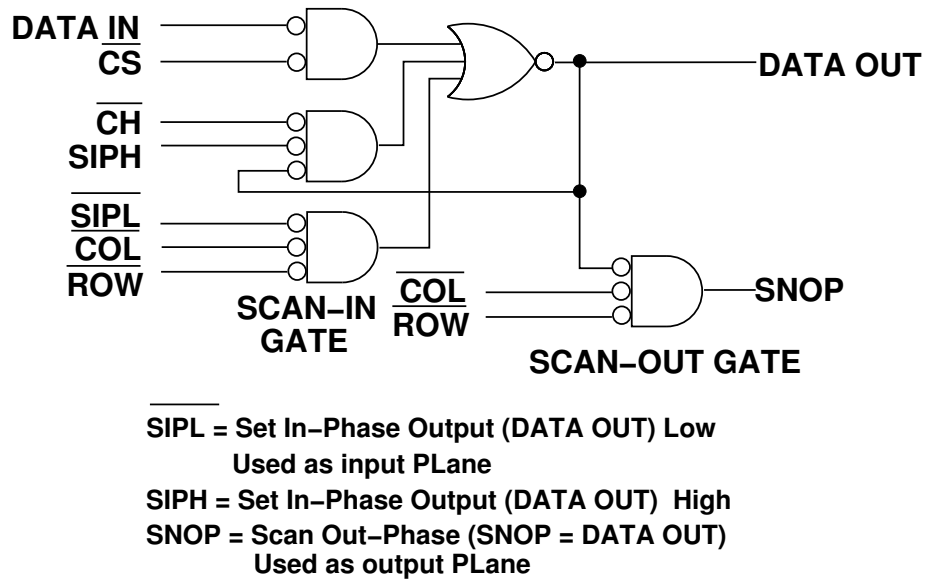


Figure 3.4: Scannable (master) latch as described in [61].

scan machine then distributes the remaining 62-bit serial stream that it receives, activating the \overline{SIPL} input to a Scan-In inactive. It thus resets only selected latches as directed by the SCAN DATA input bit pattern.

Scan-in of a latch must be preceded by a full scan-out of its chip, followed by a full scan-in with the same data, altered only in its one bit location. Each chip is assigned a unique address within the Multi-Chip Carrier (MCC) where it is located. The scan address hierarchy consists of

- MCC Number
- Chip Number (16×8)
- On-Chip Element Address (4 PLanes × 4 COLs × 4 ROWs)

Cost of ARAS

The cost of Amdahl scan is summarized in Table 3.1. From these results, the author estimated the ARAS overhead per uniprocessor system to be 19.7% in chip space and 3.4%

Table 3.1: Hardware Requirements for ARAS.

LATCH CHIP		MCC	SYSTEM
Master Latch	Scan Machine		
1. 2 Gates 2. Added inter-connect	Serial: 2 pins & 50 gates OR Parallel: 9 pins & 18 gates	1 MCC Scan Chip	1. Console Software 2. 1/8 Console Processor 3. Interface Chips (2)

in chip pins. In conclusion the ARAS method accomplished savings at all levels of system development.

3.1.3 Ito *et al.*'s Method

In this work [32], the design described in [2] and two other types of testability circuits are implemented in the Fujitsu VP-2000 series supercomputer and a detailed study was made. The method of performing delay testing using RAS is described in this work.

A delay fault can be detected by transmitting a signal transition between latches in a clock cycle of normal operation [29, 34]. In Figure 3.5 the path from a data-out line of latch L1 to a data-in line of latch L2 is sensitized, and the paths from a clock primary input to clock lines of L1 and L2 are sensitized. Then, '0' is scanned into L1 and L2. Next, required values are scanned into the related latches and placed on the related primary inputs so that '1' is set on the data-in line of L1. Under these preparations, a clock is issued to L1 in the first cycle, and to L2 in the second cycle. Then L2 is scanned out. If its value is '0', an over-delay fault exists between L1 and L2.

Boundary scan was used to enable the chips to be controlled and observed without direct probing. Scan points with respect to this work were categorized into three types according to the testing purposes. The first type is a set of scan points which make internal

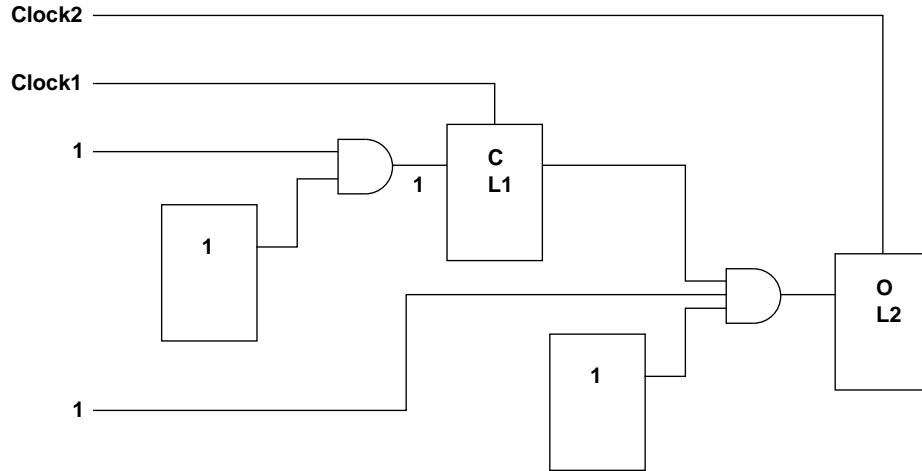


Figure 3.5: Delay testing between latches as described in [32].

latches controllable and observable mainly for the static functional testing. These are determined based on the result of the adequate testability analysis. The second type is a set of scan points which suppress clocks to latches for the delay fault testing. These are determined by adding scan only latches to control the clock enable lines of clock choppers and latches, which is done after the completion of system logic placement. The third type is a set of scan points which make chip I/O pins observable for the board testing. Pin scan-out circuits are added to all the primary inputs and outputs of the chip except scan address pins and a scan-out pin.

Table 3.2 shows the gate overhead, where S_{ORG} is the average number of gates used per chip before testability circuits were incorporated, S_{SCANFF} is the average number of scannable latches or flip-flops per chip before testability circuits were incorporated, S_{DELAY} is the average gate count of the incorporated testability circuit for clock suppression and S_{SCAN} is the average gate count of the incorporated scan control circuit including the pin scan-out circuit and reset distribution circuit. The percentage overhead due to the testability circuits was calculated as per the expression given below:

Table 3.2: Testability circuit size per chip [32].

Chip Type	S _{ORG}	S _{SCANFF}	S _{DELAY}	S _{SCAN}
L _{15K}	9487	621	636	1392
LOGIC + RAM _{64K} bit	1871	25	147	1016
Chip type 3	5341	394	471	1307
Chip type 4	4894	266	380	1177

$$\frac{S_{\text{DELAY}} + S_{\text{SCANFF}} + S_{\text{SCAN}}}{S_{\text{ORG}} + S_{\text{DELAY}} + S_{\text{SCAN}}} \quad (3.1)$$

3.1.4 Baik *et al.*'s Initial Method

The concept of RAS was shelved for a very long time until it was reinvestigated in 2004 by Baik *et al.* [7]. In this work the authors investigated the potential of RAS architecture and the feasibility of the same in today's technology. They have discussed the advantages of RAS over traditional serial scan. They prove beyond a shadow of doubt that the test power, test data volume and test application time can be reduced simultaneously in RAS. These problems have been studied independently and together, in various contexts such as IC test, microprocessor test, and system-on-a-chip (SOC) test [12, 19, 42, 56]. Due to overheating of devices such as a SOC during testing, test scheduling methods are used [16, 68]. Serial scan operations create unacceptably high activity due to frequent transitions in scan chains. This problem could be solved by either minimizing the flip-flop transitions or slowing down the scan clock [12, 13, 19, 68] which would increase the test time.

The basic architecture of RAS architecture is illustrated in Figure 3.6. The RAS structure allows reading or writing of any flip-flop using $\log_2 n$ address bits where n is the number of scanned flip-flops. The address can be applied in either a parallel manner using multiplexed PIs or in a serial manner using an *address shift register* (ASR). In this work an ASR or an Address Register is used in which the address of the flip-flop could be scanned-in a serial mode. When the address is applied, the address decoder generates a scan enable

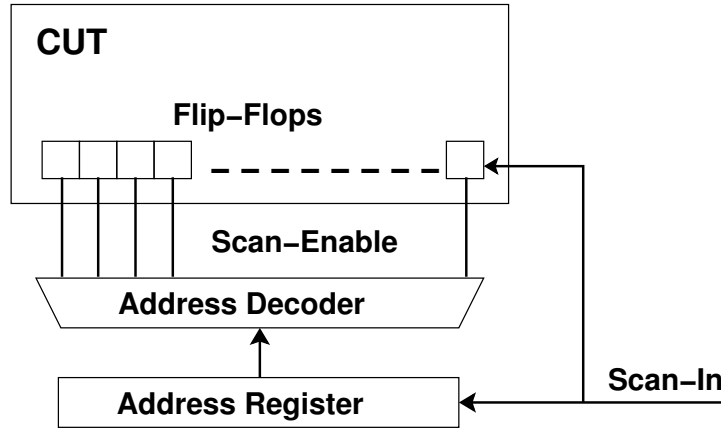


Figure 3.6: Abstracted structure of RAS.

signal to the corresponding flip-flop and the addressed flip-flop is written a new value from the scan-in signal.

A test application example is illustrated in Figure 3.7. An example circuit *CUT* with five flip-flops and test set T shown in Table 3.3 is considered. Since only flip-flops are scanned, the table gives values of the pseudo primary input (PPI) part of input vectors and the pseudo primary output (PPO) part of its fault-free response. An application of T is considered in a sequence $t_1 \rightarrow t_2 \rightarrow t_3 \rightarrow t_4$. After t_1 has been applied, the response is captured as $o1$. If a fault which can be detected by $t1$ is present in the circuit, it will be detected via a PO or the MISR. Otherwise, the value of $o1$ will be equal to the fault free output. The RAS can directly update the fourth bit of $o1$ for application of t_2 as illustrated in Figure 3.7. Scan operations are required for every flip-flop whose value is different between o_i and i_{i+1} . Figure 3.8 represents the entire test application sequence for T drawn as a directed graph. Each vertex represents i_i/o_i pair for t_i and weights on edges are equal to the number of scan operations for the RAS environment. The sum of all the weights on the edges in the graph equals n_T . In this work [7] the authors have developed a scheme to 1) minimize n_T and 2) reduced the cost of address scan operations.

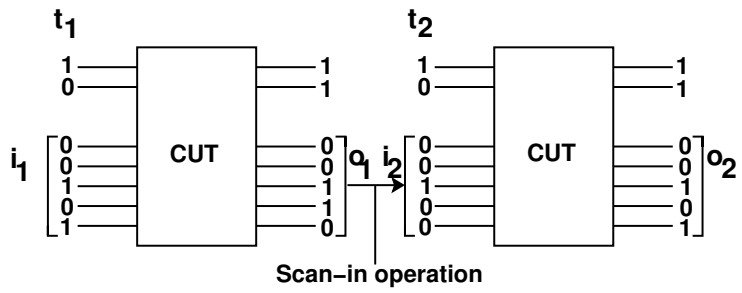


Figure 3.7: RAS scan-in operation.

Table 3.3: Example vectors [7].

Test set	PPI (i_i)	PPO (o_i)
t_1	00101	00110
t_2	00100	00101
t_3	11010	11010
t_4	00111	01011

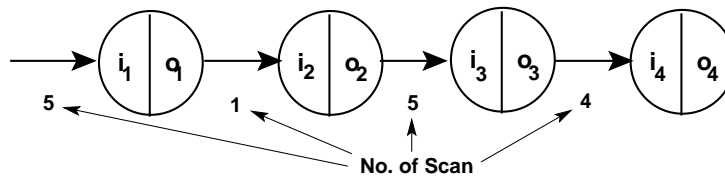


Figure 3.8: Test application using RAS [7].

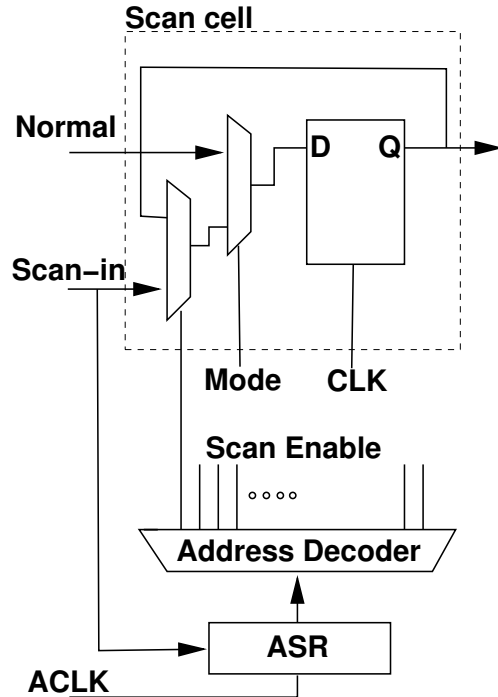


Figure 3.9: Mux based RAS [7].

The authors proposed two techniques, namely test vector ordering and Hamming distance reduction to minimize the total number of RAS operations (n_T). To illustrate this they have used an example. For the test set T shown in Table 3.3 and for the sequence of Figure 3.8, n_T is 15, including the initialization. However, using test vector ordering, the vectors can be reordered as $t_2 \rightarrow t_1 \rightarrow t_4 \rightarrow t_3$ and the weights on the edges become 5, 0, 1 and 2, which results in $n_T = 8$.

Hamming distance reduction was achieved by modifying the test vectors to minimize n_T . In the above example if the first two bits of i_4 can be switched to 1 without losing fault coverage, then the weight of edge e_{34} in Figure 3.8 becomes 2 instead of 4 and results in $n_T = 12$. A method called *Don't-care identification* [39] has been proposed to identify x 's on specific bits in the test set. However this algorithm works best when the outputs are free of x 's. This requires a modification of *Don't-care identification* and iteration of the *Don't-care identification* procedure before and after the test vector ordering.

A multiplexor based RAS architecture was used in this work. Figure 3.9 illustrates the architecture. The *Mode input* is set to 1 and the address bits are scanned into ASR via the *scan-in* port and *Address shift Clock (ACLK)* to update a value of a flip-flop. After the address is scanned-in the *ACLK* is deactivated and the *system clock (CLK)* is applied to write to the selected flip-flop. All the unaddressed flip-flops hold their values. The *scan-in* port can be shared by both value scan and address scan. This operation can be explained as follows.

Assume a set of addresses $A_{ij} = \{a_1, a_2, \dots, a_3\}$ to be accessed to apply t_j after t_i . The scan operations will be repeated for all addresses in A_{ij} . Once all flip-flops are set to desired values, the *Mode* is set to 0 and *CLK* is applied to capture the test response. If there are n_{ff} flip-flops, the total number of test data bits required to apply t_j is $\lceil \log_2 n_{ff} \rceil \times c_{ij} + c_{ij}$, where c_{ij} is the number of RAS operations to change o_i to i_j . This formula holds assuming that all $\lceil \log_2 n_{ff} \rceil$ bits of the ASR are scanned for each address. However, if the address scan operation is modified using the *scan address ordering* method, the number of data bits required for the RAS can be reduced. The *scan address ordering* method uses an asymmetric traveling salesman problem (ATSP) [18] that finds the optimum solution. The results for the total number of clock cycles and test volume reduction for some *ISCAS89* and *ITC99* benchmark circuits compared to serial scan is given in Table 3.5.

The Table 3.5 is divided into four blocks. The first block contains the circuit statistics. The second block lists the ASR width and the number of the RAS operations when this method was used on the initial test set. The third and the fourth blocks compare the test data volume and the test application time of this method against conventional serial-scan method. The peak and average switching activity is compared between conventional serial scan and RAS in Table 3.4. This work mainly outlined the significant advantages and potential of RAS compared to serial scan.

Table 3.4: Peak and average switching activity during scan [7].

	Peak switching activity			Average switching activity		
Circuit name	Serial (%)	RAS (%)	Ratio (%)	Serial (%)	RAS (%)	Ratio (%)
s5378	39.76	5.00	12.58	22.79	0.218	0.957
s9234	42.27	10.81	25.57	25.72	0.220	0.857
s13207	38.80	4.15	10.70	24.93	0.052	0.207
s15850	40.75	8.51	20.89	24.55	0.092	0.374
s35932	21.50	0.21	0.96	6.30	0.032	0.506
s38417	34.58	1.46	4.22	23.62	0.001	0.002
s38584	31.31	18.86	60.23	24.23	.040	0.165
b17s	30.65	5.01	16.34	13.50	0.004	0.033
b20s	37.87	12.37	32.67	24.39	0.006	0.027
b22s	36.52	8.16	22.34	22.67	0.003	0.015

3.1.5 Baik *et al.*'s Modified Methods

In the subsequent works related to RAS by Baik *et al.*, they have extended the idea that they evolved in their previous work [7] and given a practical dimension to implement their design. They point out in this work that although multiple scan chains can be used to reduce the length of scan chains, and hence the test application time, the number of scan chains is limited by the number of test channels/pins on an automatic test equipment (ATE) whose cost may be prohibitive [10]. There are several techniques that have been researched to reduce the test application time for the limited scan I/O pins [3, 49] but the test power has not been considered in these works. A very popular and one of the most significant works in data compression has been by J. Rajski *et al.* [48] on *Embedded Deterministic Test*. However they fail to address the test power related issues. The technique developed by the authors in their work is called *Progressive Random Access Scan* (PRAS) and test application methods for PRAS were proposed with the goal of simultaneous reduction of test application time, test data volume and test power with relatively small hardware overhead.

The PRAS structure [4] is similar to static random access memory (SRAM) or grid addressable latch [58]. In PRAS architecture, scan-cells are configured as an $m \times n$ SRAM

Table 3.5: Circuit statistics & test data volume and test application time reduction [7].

Circuit statistics			RAS properties		Test application time			Test application time		
Circuit name	No. FF	No. vector	ASR width	No. RAS operation	Serial (bits)	RAS (bits)	Reduction (%)	Serial (cycles)	RAS (cycles)	Speed up (ratio)
s5378	179	100	8	2089	17900	7532	57.92	18179	9721	1.87
s9234	228	111	8	3407	25308	11055	56.32	25647	14573	1.76
s13207	669	235	10	5043	157215	27891	82.26	158119	33169	4.77
s15850	597	97	10	4881	57909	23163	60.00	58603	28141	2.08
s35932	1728	12	11	5668	20763	15495	25.27	22476	21175	1.06
s38417	1636	87	11	15203	142332	58905	58.61	144055	74195	1.94
s38584	1452	114	11	13940	165528	57471	65.28	167094	71525	2.34
b17s	1415	617	11	24467	873055	145430	83.34	975087	170514	5.13
b20s	490	438	9	17680	214620	73867	65.58	215548	91985	2.34
b22s	735	481	10	27245	353535	132355	62.56	354751	160081	2.22

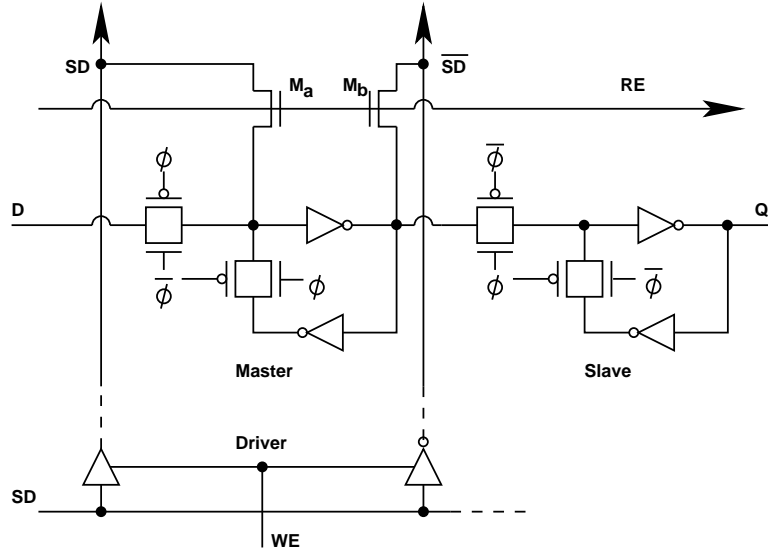


Figure 3.10: RAM-based RAS [4].

like grid structure, and some additional peripheral and test control logic was added. The number of rows and columns are decided by the geometry of the circuit or the number of available test pins. During test mode, scan-cells in one of the m -rows is enabled, allowing it to be read or written by the horizontal row enable signal available from the row enable shift register.

The RAM-based PRAS cell is shown in Figure 3.10. A read operation is performed when the contents of cells in the enabled row are placed on the vertical bidirectional scan-data lines and passed to the sense amplifier. The data read from the scan-cells in a row are passed to a multiple input signature register (MISR) to calculate the signature of the test responses. The write operation is performed one cell at a time. A read cycle is followed by the write cycle progressively for every row. Test cost reduction is done in a similar fashion as in the previous work. The authors have also estimated the routing and area overhead of this architecture. The PRAS architecture needed only marginal extra routing compared to the Multiple serial scan (MSS) implementation, and the transistor overhead was negligible compared to MSS in most cases among the benchmark circuits.

In another recent work by the authors, they have developed a test generation technique for PRAS [5]. Here the goal is to minimize both n_w (total number of PRAS write operations to apply test sequence T) as well as N (Number of test patterns in T). The traditional test generation measures such as SCOAP [27] are used to estimate the difficulty of controlling or observing each line in the circuit. SCOAP measures the controllability and observability by approximating the minimum number of lines to be set to control or observe a specific line. On similar lines, the authors have defined a testability measure that approximates the minimum number of scan-cells to be set for controlling or observing a specific line in the circuit. A static component and a dynamic component are used in this testability measure. The static component is calculated without considering the present state of the circuit and the dynamic component is calculated taking into account the present state of the circuit. Hence every time the state of the circuit changes the dynamic component is recalculated.

The algorithm to compact test vectors is given in Figure 3.11. The static testability is calculated first and the initial states of all the scan-cells are assigned with random values. Based on the initial state of the scan-cells, the dynamic testability including DP (Detection Progress, represents the approximate percentage of scan-cells that are already set by the current state of the circuit to detect a certain fault), is assigned to all faults. Then, the fault with maximum DP is targeted. A test is generated for a targeted fault and is fixed. The circuit is fault simulated and additional detected faults are dropped. After the fault simulation, the dynamic testability (DT) and DP are recalculated. Then the faults with maximum DP are iteratively targeted until all faults are targeted or the maximum DP is above a dynamically changing threshold value. Once all the faults above the current threshold DP are targeted, one test vector is generated and fault simulated to permanently drop the detected faults. All the scan-cells are updated to the next state.

A further modification to PRAS has been proposed in [6]. While the PRAS configuration uses an $m \times n$ grid structure by the distribution of scan-cells to minimize the routing

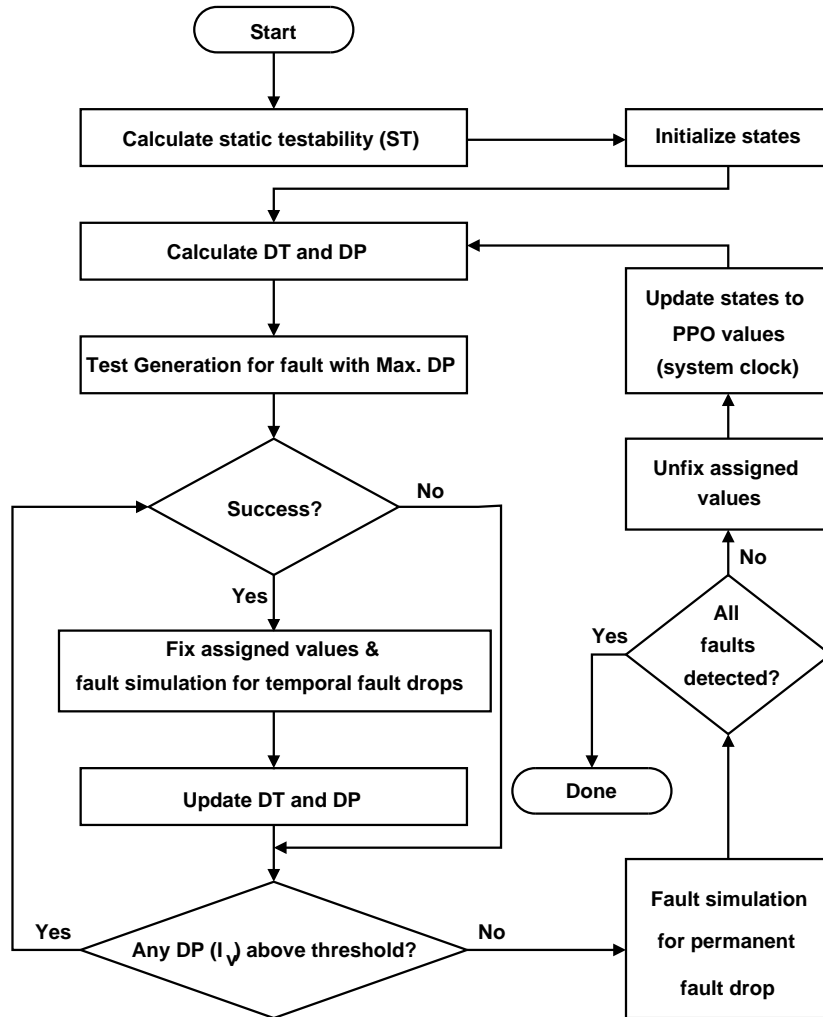


Figure 3.11: Test generation procedure [5].

overhead, where the number of columns(n) and the number of address pins ($\log_2 n$) are predetermined by the grid configuration, regardless of the number of available test pins or test channels, the partitioned grid approach takes into account the number of pins available and structures the grid accordingly. This reconfiguration is done such that it minimizes the routing overhead and reduces the test application time.

3.2 Summary

In this chapter we have explained most of the previous RAS architectures starting from the very first reference in the literature. Our work is based on the work Baik *et. al.* in [7]. As we can see from the progress of work on RAS by the research community, RAS may emerge as a very powerful DFT technique in the near future.

CHAPTER 4

TOGGLE RANDOM ACCESS SCAN

In this chapter we introduce the new concept of *Toggle Random Access Scan*. The motivation and the working of the design is explained in detail. The fundamental objective while starting the work was to minimize the area and routing overhead associated with earlier RAS designs and practically implement the design on an industrial circuit. Also the idea was to motivate the paradigm shift from conventional serial scan toward RAS.

4.1 Toggle flip-flop

In Serial-Scan (SS), flip-flops form a seamless chain from the scan-in pin to the scan-out pin in the test mode, forming a shift register structure. During normal mode of operation the inputs to the flip-flops are from the combinational logic. During scan-in/scan-out, every flip-flop is subject to change in state. This leads to continuous activity in the flip-flops, as well as the combinational circuits, dissipating a lot of power, which is very undesirable. In RAS, a decoder is used to address every flip-flop. Hence at any given point of time only one flip-flop is accessed while the other flip-flops retain their states. This way no activity takes place in the circuit during the scan mode or the test mode. The architectures described in the literature [2, 7, 45] mainly consist of a scan-in signal that is broadcast to all the flip-flops, a test control signal that is also broadcast to all flip-flops and a unique decoder signal from the decoder to every flip-flop. The output from the flip-flop is either fed into a MISR or the outputs are ORed to a primary output justifying the logic.

Table 4.1: RAS signals.

Function	Clock	Address decoder outputs	
		Row (x)	Column (y)
Normal data	active	0	0
Toggle data	inactive	1	active clock
	inactive	active clock	1
Hold data	inactive	1	0
	inactive	0	1
	inactive	0	0

4.1.1 Design

The design could become cumbersome if a unique decoder signal is routed to every flip-flop and the scan-in signal is broadcast. In the design that we have developed, we use a unique toggling scheme wherein the addressed flip-flop toggles its present state in the test mode, thereby eliminating a separate globally routed scan-in signal. The output from the flip-flop is fed into a bus. Thus the addressed flip-flop places its value on the bus in the test mode providing the necessary observability.

The design of our RAS flip-flop can be described by three operations that are essential to satisfy the test requirements, which are, to capture the response of the circuit in the normal mode, to toggle the current state of the flip-flop being addressed and retrieve the contents simultaneously, and finally to make sure that all unaddressed flip-flops hold their previous states while one flip-flop is being accessed during test mode. The operations are summarized in the first column of Table 4.1. The inherent redundancy in the clock signal [38] is coupled with the signal from the decoder to trigger the latching in the flip-flop. We have assumed the flip-flop to be made up of a master and a slave latch, as shown in Figure 2.3.

Every flip-flop gets two inputs, one from the row (x) and one from the column (y) decoder. The other inputs are clock and data from the combinational logic. The combinations

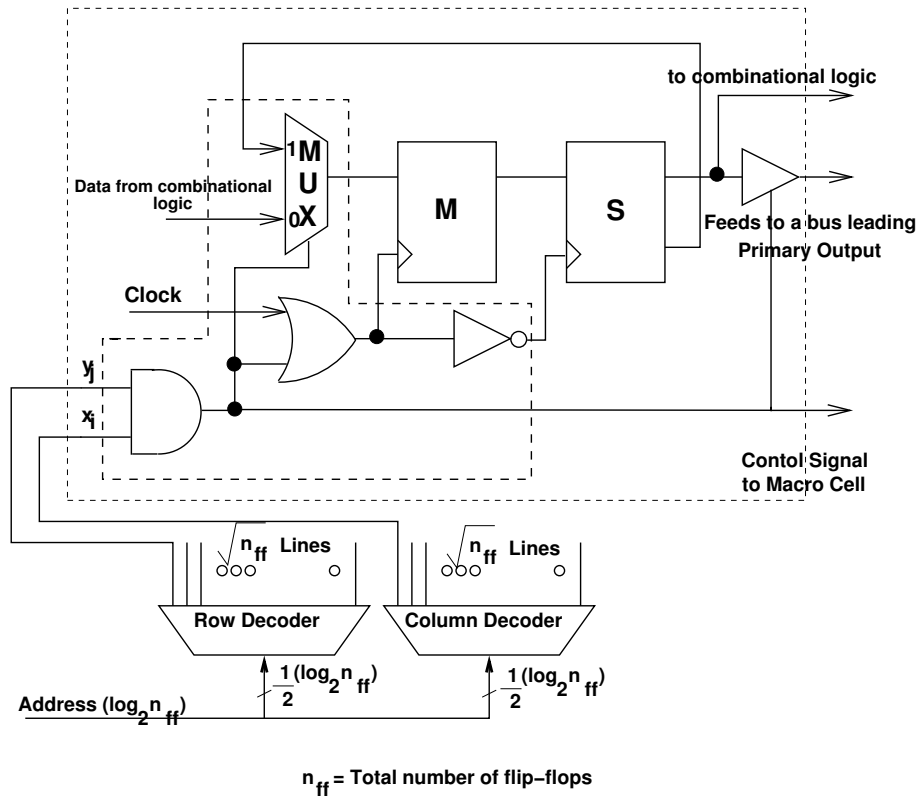


Figure 4.1: Toggle random access scan flip-flop.

used for the three defined functions are listed in Table 4.1. The operation of the modified scan-FF can be described using Figure 4.1.

4.1.2 Working

In the normal mode of operation, the x and y lines are '0's and the decoders are disabled. The output of the AND gate inside the flip-flop is logic '0' enabling the OR gate and routing the data from the combinational logic through the multiplexer to be captured in the flip-flop. The master is latched at the high pulse of the clock and the slave is latched subsequently in the low pulse. In the test mode, the clock is stopped and the row and column decoders select one line each to address a flip-flop at its intersection. Hence only one flip-flop which is addressed, receives a logic '1' at both x and y lines. The multiplexer now routes the inverted contents of the flip-flop to the master; we refer to this as the *toggle*

mode. The signal on x or y is then switched to logic '0', performing the function of a clock to load the slave latch. This operation can happen at any desired frequency (may be slower than the functional clock). Hence the addressed flip-flop toggles its current state and at the same time the tristate buffer is enabled to route the data previously stored in the flip-flop to a common bus. Meanwhile, the other flip-flops have to hold their previous states while the toggle operation is being performed on one flip-flop. Since the output from the AND gate is a logic '0', the master latch never gets activated, since the clock is turned off. Consequently the slave latch holds its previous state. One must note that addressing a flip-flop reads the contents of the flip-flop as well as toggling its contents. Hence the contents of the flip-flop after a read operation would be opposite to the value that was read out. Care is to be taken to avoid a race condition in the flip-flop. This can be achieved by inserting appropriate delays.

All the flip-flops can be cleared initially by using a built-in circuit, which in the clear mode would read each flip-flop and, based on its current contents, determine if another read operation is to be performed to clear it. For example, during the clear mode, if a flip-flop is read and is found to contain a logic '0', the contents of that flip-flop would have toggled to logic '1' and the same flip-flop is addressed again to toggle its state to clear the flip-flop (logic '0' state). This operation requires two clock cycles. In the case when the first read is a logic '1', the next cycle is a dummy cycle and the flip-flop is left unaddressed, since it would have toggled to state '0'. Hence, the number of clock cycles to clear all flip-flops would be twice the number of flip-flops in the circuit. The working of the *toggle* flip-flop is explained in detail in our paper presented at the VLSI Design and Test symposium '05 (VDAT '05) [40]

4.2 Decoder Design

The row and column decoders are built in such a way that the row and column lines intersect to address a flip-flop. This design has the least area and routing overhead compared to other decoding schemes. One may think of it as a Random Access Memory structure where the combinational logic is built around the memory element. The total number of rows and columns depends on the number of flip-flops and the actual layout of the circuit. The least number of horizontal and vertical lines would be the case when both are equal in number and numerically equal to the square root of the number of flip-flops in the circuit. Let us assume that the row decoder decodes one among the ‘m’ lines and the column decoder decodes one among the ‘n’ lines, where the total number of flip-flops are $m \times n$. It is assumed that the inputs to the decoder fan-out from the primary inputs of the circuit, since during test mode there is no activity in the combinational logic. Therefore the number of inputs to the circuit must be greater than $\log_2 m + \log_2 n$.

In comparison with cross check [14], where an entire row needs to be addressed and a single flip-flop can be set only if the contents of all other flip-flops in that row are known, our method can be used to set or observe any flip-flop dynamically. This scheme would not work if a MISR is used to capture the outputs. In our architecture we can address any flip-flop without any constraint and read its value. Also, cross-check requires an extra signal, namely scan-in, to set the desired value of the flip-flop. The decoder logic is purely combinational. A control signal may be used to enable and disable the decoder during the test mode and normal mode of operation. The macro level of the row and column decoder implementation is shown in Figure 4.2

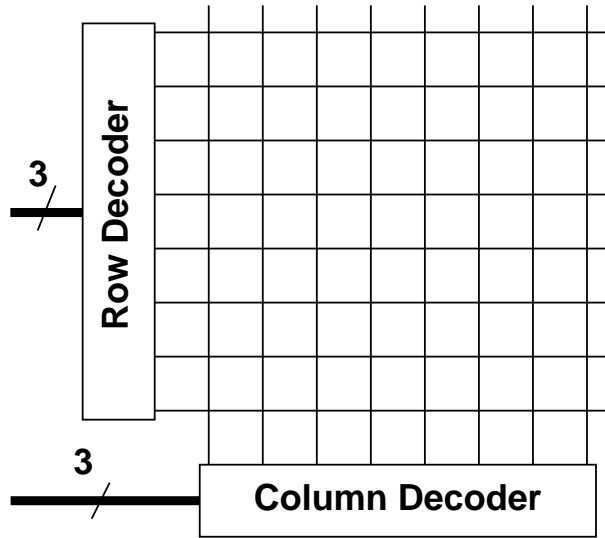


Figure 4.2: Decoder design.

4.3 Routing

The architecture described in [7] used three separate signals to control any given flip-flop, apart from the signal feeding-in from the combinational logic. This design is illustrated in Figure 4.3. Our design performs the equivalent function using only a decoder signal, thereby eliminating two globally routed signals to the flip-flop. The output from every flip-flop is connected to a bus that leads to a primary output pin. This is analogous to the “Test-control” signal being routed in the serial scan, except that the Test-control signal is connected to every flip-flop from a primary input pin. The scan-in signal, which forms a seamless chain from a primary input to a primary output through all the flip-flops in serial scan, is eliminated and a signal from the decoder to each flip-flop is added. The conventional decoder scheme used in [7] becomes very complex and cumbersome to implement since a single wire would have to be routed to every flip-flop. Also the decoder complexity will grow proportionally. For 65536 (64K) flip-flops, 65536 unique wires will have to be routed across the circuit and would require 64K 16-input AND gates to decode 16 address lines.

The outputs of the flip-flops are fed to a MISR, i.e. every flip-flop feeds to a MISR in the previous RAS design by Baik *et. al.*

The grid architecture shown in Figure 4.2 was found to be the most efficient way to lay out the decoders. The total number of extra routes added is $m + n$, where ‘m’ and ‘n’ are the number of row lines and the column lines, respectively. With a minimum of two layers of metal routing, the row wires can be accommodated within the channel in between the cell rows and the column wires can be routed over the cell in the next metal layer. Hence there will be an increase of one track per channel (assuming ‘m’ channels) and ‘n’ tracks that are routed on the next metal layer. Let us assume a circuit with 65536 (64K) flip-flops like before. Let us also assume a square layout that has 256 routing channels. Hence every row will contain 256 flip-flops, i.e. $m = 256$ and $n = 256$. The total number of additional tracks will be $256 + 256 = 512$. Let the length of every channel be ‘l’ μm and assuming the vertical dimension to be a linear multiple of the channel length, i.e. $(q \times l) \mu m$, then the increase in length of routes is $(q + 1) \times l \mu m$. Hence 65536 wires have been reduced to 512 wires.

4.3.1 Gate area overhead of RAS

Assume a circuit with ‘ n_g ’ gates and ‘ n_{ff} ’ flip-flops, each consisting of 10 gates. Assume the scan flip-flop is designed as shown in Figure 2.4, then the gate overhead of serial scan [10] and RAS is given by equations (4.1) and (4.2), respectively

$$Gate\ overhead\ of\ scan = \frac{4 \times n_{ff}}{n_g + 10 \times n_{ff}} \times 100\% \quad (4.1)$$

The RAS flip-flop has 4 gates of the multiplexer similar to scan-flip-flop and the gates in Figure 2.4, the additional gates that are added are one AND-OR-INVERT (AOI) and a tri-state buffer as shown in Figure 4.1, i.e., the logic can be minimized by using one

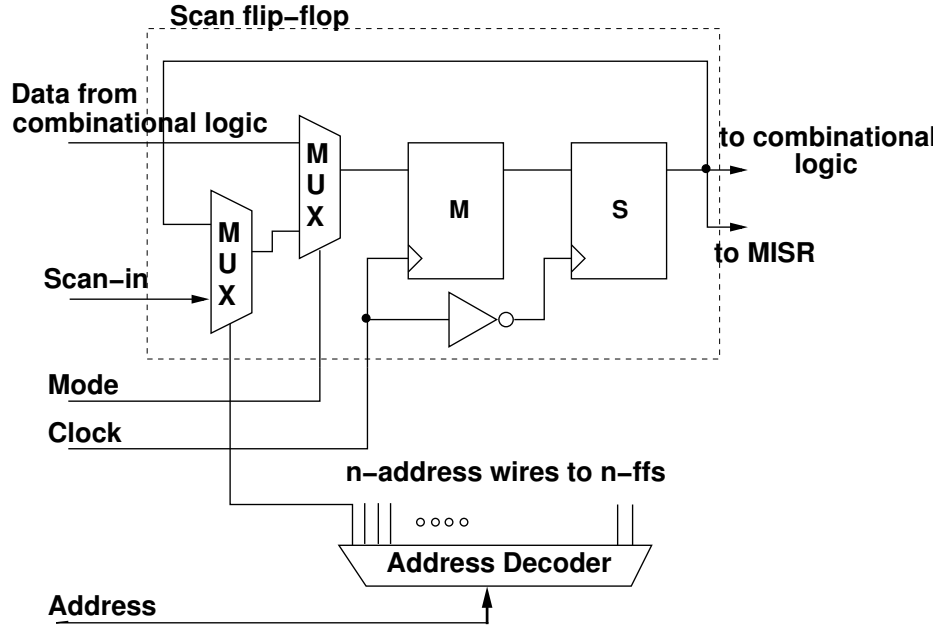


Figure 4.3: Design of RAS as described in [7].

complex gate (AOI) and using the same inverter that is used to invert the clock in a flip-flop. The logic shown within the dotted box in Figure 4.1 can be further minimized. For the number of gates increased by the decoder, let us assume a decoder structure built using pass transistors shown in Figure 4.4. The number of transistors required to decode ‘ $\log_2 c$ ’ lines to ‘ c ’ lines approximately equals $2 \times c$. Let us assume that a gate is made up of 4 transistors and $n_{ff} = c$ (horizontal lines) \times d (vertical lines). The gate overhead of RAS can be approximated by the following equation:

$$\text{Gate overhead of RAS} = \frac{6 \times n_{ff} + \sqrt{n_{ff}}}{n_g + 10 \times n_{ff}} \times 100\% \quad (4.2)$$

Let us consider a circuit with 5,120 gates and assume that there are 512 flip-flops in the circuit. The gate overhead of serial scan is 20% from Equation 1 and the gate overhead of RAS is 30.2% from Equation 2. Hence there is an increase of 10% in the x dimension of the layout.

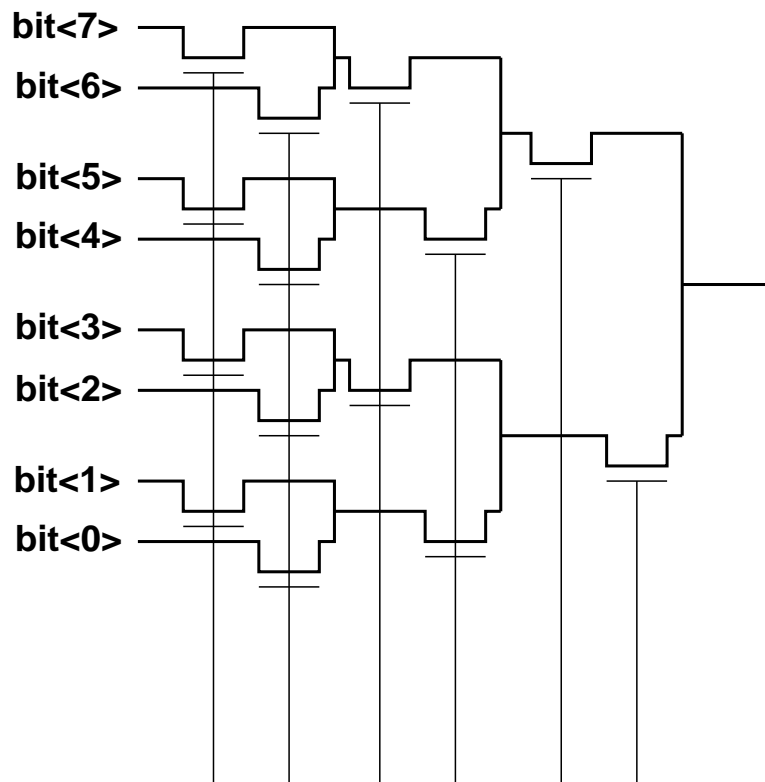


Figure 4.4: Decoder built using pass transistors [65].

Table 4.2: Gate overhead of RAS vs Serial Scan.

Circuit	No. of combi. gates	No. of Flip-Flops	Gate overhead Serial Scan	Gate overhead RAS	(%) Increase in gate area over Serial Scan
s208	96	8	18.18	28.88	10.7
s349	161	11	19.29	30.18	10.89
s386	159	6	10.96	17.56	6.6
s420	196	16	17.98	28.09	10.11
s510	211	6	8.86	14.19	5.33
s641	379	19	13.36	20.80	7.44
s838	390	32	18.03	27.84	9.81
s1196	529	18	10.16	15.83	5.67
s1269	569	37	15.76	24.29	8.53
s3271	1572	116	16.98	25.87	8.89
s3384	1685	183	20.83	31.62	10.79
s5378	2779	179	15.67	23.80	8.13
s13207	7951	638	17.80	26.89	9.09

Comparing the transistor level implementations of serial scan and RAS from the synthesized schematics obtained from the Design Architect[®] tool by Mentor Graphics[®] in 0.5 μm CMOS technology, the RAS flip-flop design had an addition of 16 transistors compared to serial scan. Hence we can formulate the transistor overhead similar to the gate overhead calculation as follows:

$$\text{Transistor overhead of serial scan} = \frac{10 \times n_{ff}}{n_t + 28 \times n_{ff}} \times 100\% \quad (4.3)$$

Here ' n_t ' is the number of transistors in the circuit without the flip-flops and each flip-flop is made up of 28 transistors. There are 16 extra transistors in RAS compared to serial scan, hence the equation becomes:

$$\text{Transistor overhead of RAS} = \frac{26 \times n_{ff} + 4 \times \sqrt{n_{ff}}}{n_t + 28 \times n_{ff}} \times 100\% \quad (4.4)$$

4.4 Testing

The tests target all the stuck-faults in the CUT. Consistently dominant faults are modeled on the tri-state buffers in the circuit [46, 11, 35, 55, 60, 31]. The decoder is first tested using the MATS++[59] test. The flip-flops are cleared initially since it is assumed that a clear operation is possible on all the flip-flops to initialize them and then the test is performed.

$$\{ \Downarrow(w0); \Uparrow(r0,w1); \Downarrow(r1,w0,r0) \}$$

where:

\Downarrow - Addressing order can be either increasing or decreasing

\Uparrow - Increasing memory addressing order

\Downarrow -Decreasing memory addressing order

This test adequately tests for address decoder faults (AF) unlinked with transition faults (TF) and all AFs linked with TFs. All the stuck at faults (SAF) are detected because, from each cell a '0' and a '1' are read uniquely.

After the test-circuitry is tested for fault free operation, the flip-flops are set up to perform the routine tests. The initial states are loaded into the flip-flops and the combinational inputs are applied at the primary inputs. The vector sequences required to test the decoder and flip-flops, are linearly proportional to the number of flip-flops in the circuit.

4.5 Algorithm to compact test vectors

The 'toggle' RAS is a new method to implement DFT and the maximum compaction of test vectors may only be possible by using an algorithm specifically suited to it. A greedy algorithm has been developed to compact the test vectors. Here the vectors for the

combinational circuit are obtained using an ATPG¹. The vectors are sequenced based on the response captured by the flip-flops for an input vector along with the change in state of those flip-flops that are read where the faults have propagated during the application of the previous vector. The algorithm is as follows:

1. *Obtain the combinational vectors along with good circuit responses and store the results in a stack*
2. *Find the flip-flops where faults are propagated at each vector*
3. *While number of vectors > 0*
 - (a) *Read all the flip-flops where the faults are detected*
 - (b) *Choose the next vector from the stack that has the least Hamming distance from current flip-flop states*
4. *End While*

The algorithm can be explained with an example as follows: First the compacted test set is obtained using a combinational ATPG. A list of all the flip-flops where the faults are propagated is stored for every vector in the test set. Now an initial vector is selected which has the states of the flip-flops close to the circuit start-up state or clear state. The vector is applied and the response is captured in the flip-flops. Those flip-flops are read where the faults are propagated. Now the present states of the flip-flops in the circuit are those of the response captured from the previous vector except those, whose values are toggled due to a read operation performed on them. Then a search is performed on the remaining vectors to determine the vector which has the least Hamming distance from the present state of the

¹Vectors were obtained from HITEC/PROOFS [44, 43] and circuit responses and outputs where faults were detected on each vector were obtained using AUSIM [57]

Table 4.3: Results of Vector Compaction for various Benchmark Circuits.

Circuit	No. of FFs	No. of Combi. vectors	No. of SS vectors	No. of RAS vectors	Test time red. (%)
s208	8	64	584	301	48.46
s349	11	42	687	366	46.72
s386	6	138	972	450	53.70
s420	16	128	2192	1056	51.82
s510	6	110	776	344	55.67
s641	19	142	2859	1148	59.85
s838	32	240	7952	3595	54.79
s1196	18	344	6554	2447	62.66
s1269	37	118	4521	1981	56.18
s3271	116	264	31004	12540	59.55
s3384	183	260	48759	21119	56.69
s5378	179	618	111419	48677	56.31
s13207	638	1138	727820	309132	57.53

circuit. This will need minimum clock operations to set up the next test vector. The same procedure is followed until all the vectors have been applied.

4.6 Results

The proposed architecture was modeled and tested on ISCAS '89 [8] benchmark circuits. The algorithm was implemented and the fault coverage was observed to be the same as serial scan. A reduction in test vectors up to 60% can be observed (Table 4.3) in most of the circuits. Maximum reduction is achieved when the average number of faults per combinational vector is small and the number of flip-flops is proportionally higher, since in these cases the setup time of scan flip-flops would increase compared to RAS. The reduction in test time is slightly lower than that described in [7]. This is because of the improvement that we made in the design, by minimizing the number of signals that needs to be routed to every flip-flop.

During scan-in, the CUT is subject to unnecessary activity and all the flip-flops are subject to change state. Various methods are presented in the literature to mask the flip-flop transitions during test mode [24, 67]. Let us assume that the power dissipation in the CUT is directly proportional to the number of transitions in the primary inputs and the transitions in the states of flip-flops. The power dissipation in RAS is reduced drastically, since, the only activity during scan mode is a transition in the state of a single flip-flop under consideration and transitions at the primary input pins that control the decoder.

Relative reduction of power dissipation in the circuit is calculated assuming that, the power dissipated is directly proportional to the number of transitions in the primary inputs and states of flip-flops. The results were obtained for both serial scan and RAS (Table 4.4). It can be observed that, as the size of the circuits increases, reduction in power dissipation up to 99% is achieved using RAS.

4.7 Modifying ATPG to Further Decrease the Number of Vectors

The results presented in this paper are based on the vectors obtained using existing ATPG algorithms. A slight modification in the form of an added constraint in the ATPG algorithms can further decrease the number of test vectors needed using RAS. The following algorithm can be employed to obtain this further compaction of test vectors;

1. *Set the cost function of modifying the value of a flip-flop to be the highest*
2. *Generate a vector to target a fault*
3. *Perform Fault simulation*
4. *While the number of faults > 0*
 - (a) *Read all the flip-flop where the faults are detected*

(b) *Target a fault and Generate the next vector with minimum changes to be made in the flip-flops from the current states considering the change of state due to a read operation.*

(c) *Perform Fault simulation*

5. *End While*

Consider modifying PODEM's [26] back-trace algorithm, such that the pseudo-controllability of the flip-flop (pseudo primary inputs) is set very high. Thus during back-trace, a minimal set of flip-flops is assigned for each targeted fault. This will require the least number of flip-flops to be set at test. Furthermore, the test for the next fault is generated with minimum changes to the test response captured in the scan chain from the current test, again to minimize test application time. Early experimentation on the smaller benchmark circuits indicates that such a strategy can show a 30-40% improvement in test time. It's worthwhile noting that better vector compaction can be achieved for larger circuits using this algorithm.

4.8 Summary

In this chapter we introduced the concept of 'toggle' RAS. The design and working was explained in detail. The routing and area overhead of the proposed architecture was derived analytically. The decoder design is described in detail along with the method to test the circuitry. We have presented the results of the experiments we performed on benchmark circuits implementing this architecture.

Table 4.4: Power estimation based on number of transitions at the inputs for various Benchmark Circuits.

Circuit	No. of Tansitions in SS tests	No. of Transitions in RAS tests	Test power saving (%)
s208	1866	1209	35.21
s349	4755	1233	74.07
s386	2495	1515	39.28
s420	11587	4708	59.37
s510	3141	2382	24.16
s641	27715	7924	71.41
s838	72914	17782	75.61
s1196	57409	10601	81.53
s1269	77755	7880	89.87
s3271	1744149	45971	97.36
s3384	4299362	77665	98.19
s5378	8947677	175710	98.04
s13207	230176409	211048	99.91

CHAPTER 5

SCAN-OUT DESIGN

We have designed a novel mechanism for the scan-out of the flip-flops. This is a hierarchical structure that ensures there is no loading on the flip-flops while driving the output bus. The idea is illustrated in Figure 5.1. A cluster of flip-flops in close proximity feed a common bus. The bus control signals from the flip-flops are ORed together to produce a signal to control the next stage of the bus. This function is performed by the scan-out macro-cell.

5.1 Macro-cell design

The design of the scan-out macro cell is given in Figure 5.2. The tri-stated signal from each flip-flop feeds a bus. The maximum number of signals that can be placed on a single bus depends on the specific technology that is used to implement the design. The bus control signals from the flip-flops are ORed together and used to control the next level tri-state buffer. These scan-out macro-cells can be replicated at several stages before the bus signal reaches the primary output. This is illustrated in Figure 5.3, where a single 4×4 block is a structure similar to the one shown in Figure 5.1. 4×4 is just shown as an illustration. The number could be as large as the maximum number of tri-state buffers that can be placed on a bus in that particular technology.

To avoid a slow read during test, normal D-flip-flops can be inserted after a given number of stages of scan-out macro-cells so that the values are preserved for a multi-cycle

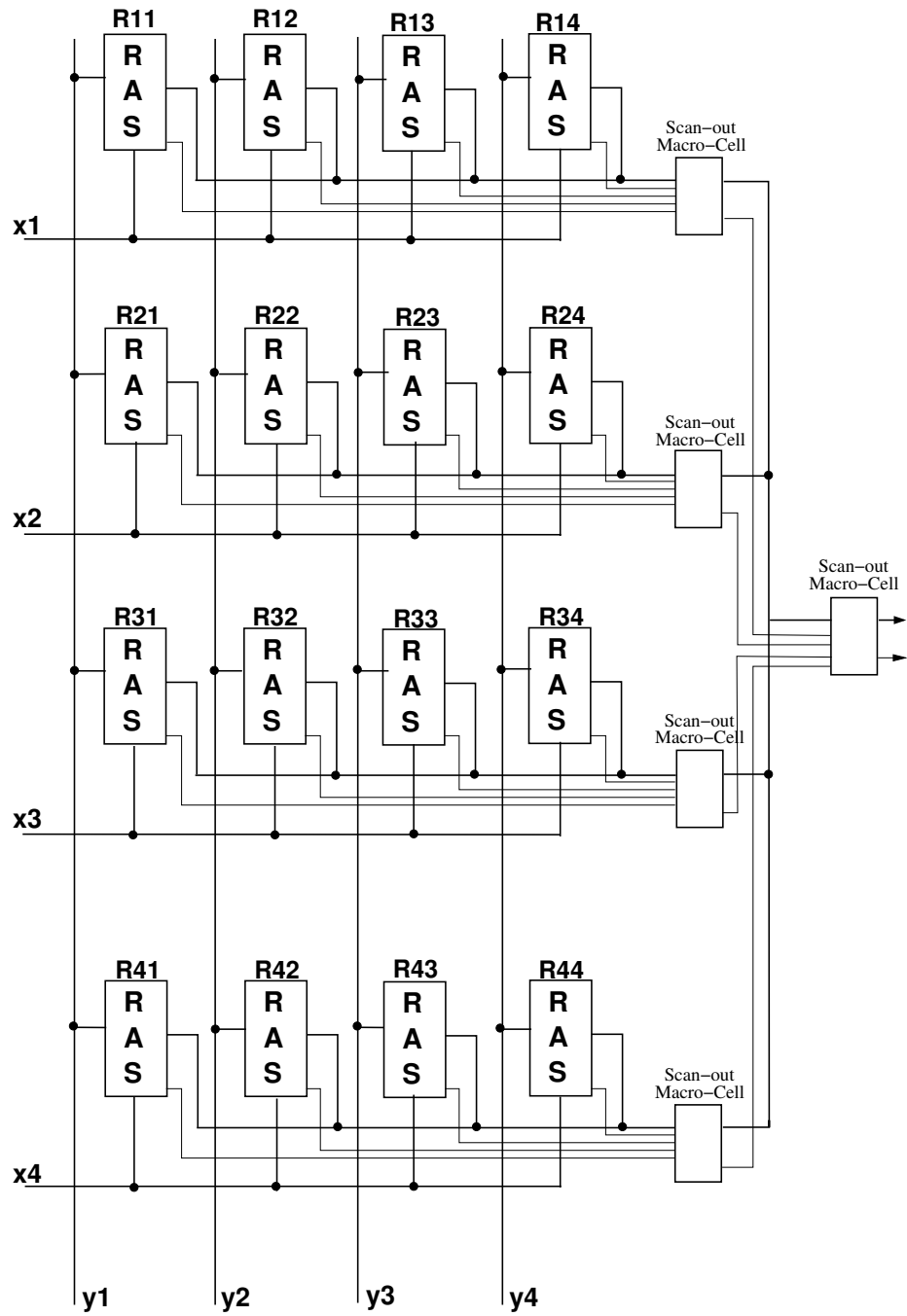


Figure 5.1: Macro level description of scan-out structure.

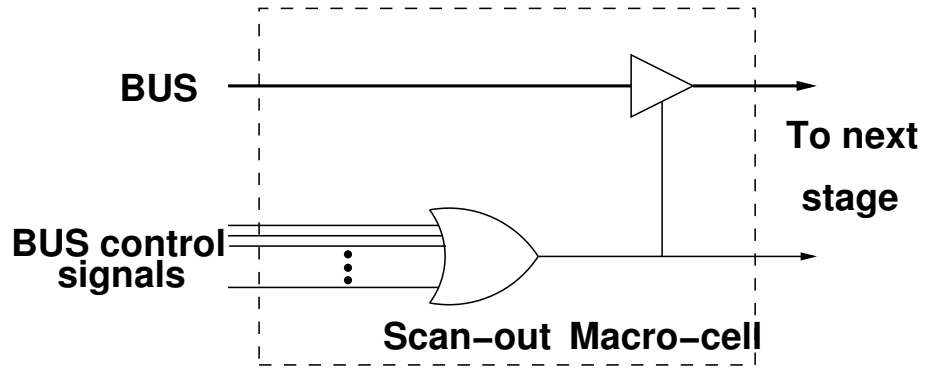


Figure 5.2: Scan-out Macro-cell.

read out. This is a novel scan-out design architecture that we have developed that is not limited to this design but may be extended to SOC cores as well.

As an advancement, another way to implement the scan-out structure is to implement a scheme with sense amplifiers and pre-charged lines, like in conventional memory, to read the contents of the flip-flops.

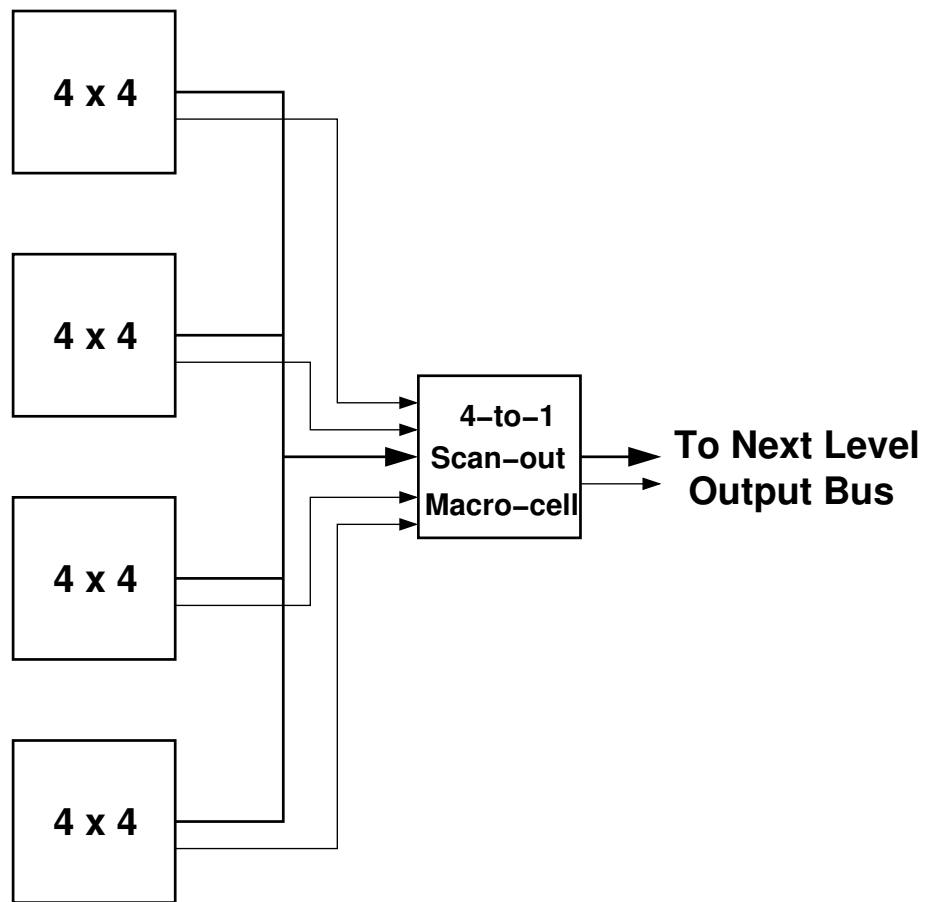


Figure 5.3: Hierarchical scan-out illustration.

CHAPTER 6

AN EXPERIMENTAL STUDY

An experimental study was performed at Texas Instrument India Pvt. Ltd. to implement this work in an industrial circuit. The duration of work was 3 months and the outcome was very promising. The most significant results of the work is described in this chapter.

The circuit used for the experiment was a module belonging to a Texas Instruments (TI) System On Chip (SOC). The application of the SOC was targeted for high performance video applications such as videophones, image processing, video CODECs and streaming media. It is one of the fastest DSP chips made at TI.

A synthesized Verilog netlist was used, which was originally DFT-ready with scan flip-flops inserted. The synthesis was performed using TI's internal 90nm technology component library. Synopsys'® Design Compiler (DC) was used for synthesis.

Scan stitching was performed in DC on the original netlist to have one scan chain. This netlist had 5,321 scan flip-flops contributing to 31,321 NAND gate equivalents, while the total NAND gate equivalent count for the entire circuit was 107,315. This Verilog netlist with a single scan chain will be referred to as the Serial Scan netlist.

Initially the RTL models for the RAS-cell, scan-out structure and row and column decoders were synthesized using the same library. Tcl scripts were written and run in DC to replace every scan flop by the RAS-cell from the original netlist. Tcl scripts were also written to insert the scan-out structure and the row and column decoders. The final netlist

was saved in Verilog format. This Verilog netlist will be referred to as the RAS netlist from here on.

The row and column decoders in the RAS netlist contributed to 507 NAND equivalent gates. The 5,321 RAS-cells contributed to 83,805 NAND gates, while the scan-out structure contributed to 20,919 NAND gates. The total NAND gate equivalent count for the entire circuit was 181,224.

The scan-out structure was designed assuming only 4 drivers could drive a single bus. But, for the module under consideration, which runs at 333Mhz, we could have 25 drivers driving a single bus in the scan-out structure. So, instead of contributing to 20,919 NAND gate equivalent counts, now it would contribute to just 2,636 gate counts. This would decrease the area overhead.

A RAS standard cell could not be built but would have reduced the area overhead significantly, the reason for which is justified below:

The scan flip-flop, if implemented using standard gates, resulted in 9.75 NAND equivalent gates where 2 AND gates equals 2.5 NAND gates, 1 OR gate equals 1.25 NAND gates, 2 NOT gates equals 1.5 NAND gates and finally 2 LATCHES equals 4.5 NAND gates. While the scan flip-flop standard cell used only 5.75 NAND equivalent gates, which is a 41% reduction.

In the case of RAS, a standard gate implementation used 15.75 NAND equivalent gates. The split is 3 AND gates which equals 3.75 NAND gates, 2 OR gates which equals 2.5 NAND gates, 4 NOT gates which equals 3 NAND gates, 2 LATCHES which equals 4.5 NAND gates and 1 BUFFER which equals 2 NAND gates. An extra inverter was used in synthesis since the library did not contain an active high enable BUFFER. Assuming that extra inverter is removed, the gate count would be 15.

Also, let us assume that a 41% reduction is possible for a RAS standard cell. The gate count would be down to a mere 8.85. But we chose 9 to be a little pessimistic in our estimation.

Using these values, we could calculate the gate area overhead of the circuit. The 5321 flip-flops were replaced by RAS cells which would contribute to 47,889 NAND equivalent gates. The scan-out structure had 25 drivers on a single bus. The total increase in the gate count would be 127,026. Hence the gate overhead of RAS over serial scan could be summarized as:

$$Gate\ overhead = \frac{127,026 - 107,315}{107,315} \times 100\% = 18.4\% \quad (6.1)$$

Synopsys[®] Tetramax[®] was used for generating the test patterns. DC was used to generate the STIL files required by Tetramax[®] for test generation.

For the serial scan netlist, Tetramax[®] generated 612 patterns. The X-fill option was used for test pattern generation, which filled the unspecified bits in the test pattern with Xs and simulated the patterns using these X-filled vectors. Now to convert these serial scan vectors to RAS vectors, we required to know which flip-flops captured a useful value for a given pattern, i.e. the flip-flops that assisted in fault detection. But, no commercial tool was able to give us this information. Hence, we assumed that all flip-flops capturing a value other than an X were useful (please note that this is a pessimistic estimate). A vector is applied and its response is captured. The useful flip-flops are then toggled (scanned out). Now the new vector to be applied was chosen to be at a minimum *Hamming* distance from the new state of the flip-flops to reduce the test time. A script was written to order the vectors using this strategy.

The total test time for serial scan and RAS is calculated as follows:

$$\begin{aligned}
\textit{Total Test Time for Serial Scan} &= \textit{No. of vectors} \\
& \quad (\textit{No. of clock cycles}) \quad \times (\textit{No. of scan flip-flops} + 1) \\
& \quad \quad \quad \quad \quad \quad \quad \quad + \textit{No. of scan flip-flops} \\
&= 612 \times (5,321 + 1) + 5,321 \\
&= 3,262,385 \tag{6.2}
\end{aligned}$$

$$\begin{aligned}
\textit{Total Test Time for RAS} &= \textit{No. of scanout operations} \\
& \quad (\textit{no. of clock cycles}) \quad (\textit{i.e. Cumulative No. of useful flip-flops}) \\
& \quad \quad \quad \quad \quad \quad \quad \quad + \textit{Cumulative No. of toggles performed} \\
& \quad \quad \quad \quad \quad \quad \quad \quad \textit{to apply new vectors after vector reordering} \\
& \quad \quad \quad \quad \quad \quad \quad \quad + 612 (\textit{test mode clock cycles}) \\
&= 363,584 + 456,989 + 612 \\
&= 821,185 \tag{6.3}
\end{aligned}$$

$$\begin{aligned}
\textit{Test Time Reduction} &= \frac{3,262,385 - 821,185}{3,262,385} \times 100\% \\
&= 74.82\%
\end{aligned}$$

This amounts to a speedup of 4X. This number can be increased with careful analysis. We found that after collapsing, the faults are reduced to 166,261. So a maximum of 166,261

scan-out operations will be required instead of 363,584. This can only be achieved if the ATPG tool gives us the useful flip-flop information. Now with this new number, the speedup would be 5.3X.

6.1 Physical Design

Magma[®] Design Automations Blast Fusion[®] was used for doing the Physical Design. The floor plan of serial scan occupied a total floor area of 1.125 sq. mm. The total area used was 0.970 sq. mm., and the utilization was 86.2%. In case of RAS the same floor plan was used with a larger floor area. The total floor area was equal to 1.563 sq. mm. and the area that was used was 1.382 sq. mm., which gives a utilization factor of 88.4%.

$$\begin{aligned} \text{Area and Routing Overhead} &= \frac{1.563 - 1.125}{1.125} \times 100\% \\ &= 39\% \end{aligned}$$

So a 68.9% area overhead after synthesis had translated to 39% area and routing overhead after physical design. Now if we consider just the 18.4% area overhead after synthesis after using the RAS standard cell and new scan-out structure, we can say that this could effectively translate to an area and routing overhead of just 10.4% after physical design. Sometimes, it may so happen that even if there is an increased area overhead after synthesis, it could result in a zero overhead after physical design as all this increased gate overhead could be accommodated in the same floor plan with an increased utilization.

6.2 Design and Implementation Issues

Due to the toggle mechanism used in the design, there is difficulty in loading the first pattern. Either there has to be some added circuitry to clear or set the flops initially or,

there needs to be some kind of feedback loop during testing which will allow the first pattern to be loaded after reading out the contents of the flip-flops.

In the test mode, the clock is suppressed to the scan flip-flops while it is still applied to the rest of the circuit (Scanout structure). This could result in some added circuitry for suppressing the clock or some routing overhead, as we may have to route two separate clock trees.

We were not able to get an exact estimate of the area overhead as we did not have a RAS standard cell in the library. Also none of the commercial ATPG tools allowed us to get the information about the useful flip-flops assisting in fault detection. Finally, we were expecting to have an extra metal layer to route the column decoder signals, but the metal layers for a certain library and certain technology were fixed and hence we did not have any control over it.

CHAPTER 7

CONCLUSIONS

In conclusion, we have designed a novel toggle RAS flip-flop which eliminates two broadcast signals, namely scan-in and test control (mode control) signal from earlier RAS designs. The main constraint today is routing within the circuit. Transistors can be added at a very low cost in today's technology. We have met our target in reducing the routing overhead by eliminating the two broadcast signals.

We have shown the advantages of RAS over single chain serial scan by reducing the test time by 60% and reducing the test power by three orders of magnitude. We have also derived an analytical expression for the increase in gate area overhead compared to serial scan. This expression agrees well with the experiment performed on an industrial circuit.

We have developed a unique scan-out structure through which the values can be scanned out dynamically through a primary output pin. This structure can also be used for a multi cycle readout to reduce the slow scan-out time.

7.1 Future Work

An ATPG needs to be built specific to toggle RAS. Experiments related to delay testing needs to be performed. The design needs to be implemented and studied in its completeness.

7.1.1 Delay Testing

Delay testing in serial scan circuits is very constrained. The scan-FFs are modified and HOLD latches [20, 21] are often inserted between the FFs and the combinational logic.

The latches insert excess delays in the path and increase area overhead due to routing of an additional control (HOLD) signal. A one bit change in the consecutive vectors can be obtained very easily using RAS, which is very vital in the case of delay testing. A vector V1 is set up and vector V2 with a one bit change is applied. It is known that any testable path can be tested by a single input change vector pair [25]. These tests are easy to apply in RAS but cannot be guaranteed in serial scan. A change in state of a flip-flop only needs one clock and the circuit response is captured in the next clock cycle, thereby testing a desired path for delay. Hence delay testing can be performed using RAS with no additional hardware, and any combinationally generated delay test vector will work for sequential circuits using RAS.

7.1.2 Random-Pattern BIST using RAS

With the ability to control any flip-flop in the circuit, random patterns can be applied by just addressing any flip-flop through the primary inputs. While testing the flip-flops and decoder for faults initially, using the march test, a fault simulation will result in random pattern testing of the circuit and the results may be interesting to observe. BIST circuit to implement the march tests are relatively easier to implement and are commonly used to test random access memory [30, 36, 62, 59].

Error diagnosis, which is a lengthy process for serial scan, can be very efficient with RAS.

BIBLIOGRAPHY

- [1] V. D. Agrawal, K.-T. Cheng, D. D. Johnson, and T. Lin, "Designing Circuits with Partial Scan," *IEEE Design & Test of Computers*, vol. 5, pp. 8–15, Apr. 1988.
- [2] H. Ando, "Testing VLSI with Random Access Scan," in *Digest COMPCON*, Feb. 1980, pp. 50–52.
- [3] B. Arslan and A. Orailoglu, "Test Cost Reduction through a Reconfigurable Scan Architecture," in *Proc. International Test Conf.*, Oct. 2004, pp. 945–952.
- [4] D. H. Baik and K. K. Saluja, "Progressive Random Access Scan: A Simultaneous Solution to Test Power, Test Data Volume and Test Time," in *Proc. International Test Conf.*, Nov. 2005.
- [5] D. H. Baik and K. K. Saluja, "State-reuse Test Generation for Progressive Random Access Scan: Solution to Test Power, Application time and Data Size," in *Proc. 14th IEEE Asian Test Symp.*, Dec. 2005.
- [6] D. H. Baik and K. K. Saluja, "Test Cost Reduction Using Partitioned Grid Random Access Scan," in *Proc. 19th International Conf. VLSI Design*, Jan. 2006.
- [7] D. H. Baik, K. K. Saluja, and S. Kajihara, "Random Access Scan: A Solution to Test Power, Test Data Volume and Test Time," in *Proc. 17th International Conf. VLSI Design*, Jan. 2004, pp. 883–888.
- [8] F. Beglez, D. Bryan, and K. Komzminski, "Combinational Profiles of Sequential Benchmark Circuits," in *Proc. IEEE International Symp. on Circuits and Systems*, 1989, pp. 1929–1934.
- [9] B. Bhattacharya, S. Seth, and S. Zhang, "Double-Tree Scan: A Novel Low-Power Scan-Path Architecture," in *Proc. International Test Conf.*, 2003, pp. 470–479.
- [10] M. L. Bushnell and V. D. Agrawal, *Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits*. Boston, MA: Kluwer Academic Publishers, 2000.
- [11] S. T. Chakradhar, S. G. Rothweiler, and V. D. Agrawal, "Redundancy Removal and Test Generation for Circuits with Non-Boolean Primitives," *IEEE Trans. on CAD*, vol. 16, no. 11, pp. 1370–1377, Nov. 1997.
- [12] A. Chandra and K. Chakrabarty, "Combining Low-Power Scan Testing and Test Data Compression for System-on-a-chip," in *Proc. ACM/IEEE Design Automation Conf.*, 2001, pp. 166–169.
- [13] A. Chandra and K. Chakrabarty, "System-on-a-Chip test data compression and decompression architectures based on Golomb codes," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 3, pp. 355–368, Mar. 2001.
- [14] S. J. Chandra, T. Ferry, T. Gheewala, and K. Pierce, "ATPG based on a Novel Grid Addressable Latch Element," in *Proc. ACM/IEEE Design Automation Conf.*, 1991, pp. 282–286.
- [15] K.-T. Cheng, S. Devadas, and K. Keutzer, "Delay Fault Test Generation and Synthesis for Testability Under a Standard Scan Design Methodology," *IEEE Trans. on Computer-Aided Design*, vol. 12, pp. 1217–1231, Aug. 1993.

- [16] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Power Constraint Scheduling of Tests," in *Proc. 7th International Conf. VLSI Design*, Jan. 1994, pp. 271–274.
- [17] R. M. Chou, K. K. Saluja, and V. D. Agrawal, "Scheduling Tests for VLSI Systems Under Power Constraints," *IEEE Trans. VLSI Systems*, vol. 5, no. 2, pp. 175–185, June 1997.
- [18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*. New York: McGraw Hill, 2000.
- [19] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. M. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits During Test Application," in *IEEE Tran. on Computer-Aided Design of Integrated Circuits and Systems*, 1998, pp. 1325–1333.
- [20] S. DasGupta, P. Goel, R. G. Walther, and T. W. Williams, "A Variation of LSSD and its Implications on Design and Test Pattern Generation in VLSI," in *Proc. International Test Conf.*, 1982, pp. 216–219.
- [21] S. DasGupta, R. G. Walther, T. W. Williams, and E. B. Eichelberger, "An Enhancement to LSSD and Some Applications of LSSD in Reliability, Availability and Serviceability," in *Proc. International Fault-Tolerant Computing Symp*, 1981, pp. 32–34.
- [22] E. B. Eichelberger, E. Lindbloom, J. A. Waicukauski, and T. W. Williams, *Structured Logic Testing*. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1991.
- [23] H. Fujiwara, *Logic Testing and Design for Testability*. Cambridge, MA: The MIT Press, 1985.
- [24] S. Gerstendorfer and H. J. Wunderlich, "Minimized Power Consumption for Scan-based BIST," in *Proc. International Test Conf.*, 1999, pp. 77–84.
- [25] M. A. Gharaybeh, M. L. Bushnell, and V. D. Agrawal, "Classification and Test Generation for Path-Delay Faults Using Single Stuck-at Fault Tests," *J. Electronic Testing: Theory and Applications*, vol. 11, no. 1, pp. 55–67, Aug. 1997.
- [26] P. Goel, "An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits," *IEEE Trans. on Computers*, vol. C-30, no 3., pp. 215–222, Mar. 1981.
- [27] L. H. Goldstein, "Controllability/observability analysis of digital circuits," *IEEE Trans. Circuits and Systems*, vol. CAS-26, no 9., pp. 685–693, 1979.
- [28] I. Hamzaoglu and J. Patel, "Reducing Test Application Time for Full Scan Embedded Cores," in *FTCS*, 1999, pp. 260–267.
- [29] E. P. Hsieh, R. A. Rasmussen, L. J. Vidunas, and W. T. Davis, "Delay test generation," in *Proc. 14th ACM/IEEE Design Automation Conf.*, 1977, pp. 486–491.
- [30] C.-T. Huang, J.-R. Huang, C.-F. Wu, C.-W. Wu, and T.-Y. Chang, "A Programmable BIST Core for Embedded DRAM," *IEEE Design & Test of Computers*, vol. 16, no. 1, pp. 59–70, Jan. 1999.
- [31] N. Itazaki and K. Kinoshita, "Test Pattern Generation for Circuits with Tri-State Modules by Z-Algorithm," *IEEE Trans. Computer-Aided Design*, vol. 8, no. 12, pp. 1327–1334, Dec. 1989.
- [32] N. Ito, "Automatic incorporation of on-chip testability circuits," in *Proc. 27th ACM/IEEE Design Automation Conf.*, 1990, pp. 529–535.
- [33] S. Kajihara, K. Ishida, and K. Miyase, "Test Vector Modification for Power Reduction During Scan Testing," in *Proc. VLSI Test Symp.*, 2002, pp. 160–165.
- [34] K. Kishida, F. Shirotori, Y. Ikemoto, S. Isiyama, and Y. Hayashi, "A delay test system for high-speed logic lsi's," in *Proc. 23rd ACM/IEEE Design Automation Conf.*, 1986, pp. 786–790.
- [35] Y. Koseko, T. Ogihara, and S. Murai, "Ti-state bus conflict checking method for atpg using bdd," in *Proc. International Conf. Computer Aided Design*, 1993, pp. 512–515.

- [36] K.-J. Lin and C.-W. Wu, "Testing Content-Addressable Memories Using Functional Fault Models and March-like Algorithms," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 5, pp. 577–588, May 2000.
- [37] C. M. Maunder and R. E. Tulloss, *The Test Access Port and Boundary-Scan Architecture*. IEEE Computer Society Press, 1990.
- [38] M. R. Mercer and V. D. Agrawal, "A Novel Clocking Technique for VLSI Circuit Testability," *IEEE J. Sol. St. Circ.*, vol. SC-19, pp. 207–212, Apr. 1984.
- [39] K. Miyase, S. Kajihara, I. Pomeranz, and S. M. Reddy, "Don't-Care Identification on Specific Bits of Test Patterns," in *Proc. VLSI Test Symp.*, Sept. 2002, pp. 194–200.
- [40] A. S. Mudlapur, V. D. Agrawal, and A. D. Singh, "A novel random access scan flip-flop design," in *Proc. 9th VLSI Design & Test Symp. (VDAT'05)*, Aug. 2005, pp. 226–236.
- [41] A. S. Mudlapur, V. D. Agrawal, and A. D. Singh, "A random access scan architecture to reduce hardware overhead," in *Proc. International Test Conf.*, Nov. 2005.
- [42] S. Narayanan and M. A. Breuer, "Reconfigurable scan chains: A novel approach to reduce test application time," in *Proc. International Conf. Computer Aided Design*, 1994, pp. 271–274.
- [43] T. M. Niermann, W.-T. Cheng, and J. H. Patel, "PROOFS: A Fast, Memory-Efficient Sequential Circuit Fault Simulator," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 2, pp. 198–207, Feb. 1992.
- [44] T. M. Niermann and J. H. Patel, "HITEC: A Test Generation Package for Sequential Circuits," in *Proc. European Design Automation Conf.*, 1991, pp. 214–218.
- [45] Z. Plíva, O. Novák, and P. B. d'Aguerre, "Hardware Overhead of Boundry Scan and RAS Design Methodologies." <http://www.fm.vslib.cz/kes/pub/ecms03.pdf>.
- [46] T. J. Powell, "Consistently Dominant Fault Model for Tristate Buffer Nets," in *Proc. VLSI Test Symp.*, 1996, pp. 400–404.
- [47] J. Rajski and J. Tyszer, *Arithmetic Built-In Self-Test*. Upper Saddle River, NJ: Prentice-Hall, Inc., 1998.
- [48] J. Rajski, J. Tyszer, M. Kassab, and N. Mukherjee, "Embedded Deterministic Test," *IEEE Trans. on CAD*, vol. 23, pp. 776–792, May 2004.
- [49] S. Reda and A. Orailoglu, "CircularScan: A Scan Architecture for Test Cost Reduction," in *Proc. Design, Automation and Test in Europe (DATE'04)*.
- [50] K. Saluja, "An Enhancement of LSSD to reduce test pattern Generation Effort and Increase Fault Coverage," in *Proc. ACM/IEEE Design Automation Conf.*, 1982, pp. 489–494.
- [51] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," in *Proc. VLSI Test Symp.*, 2000, pp. 35–40.
- [52] J. Savir, "Skewed-Load Transition Test: Part I, Calculus," in *Proc. International Test Conf.*, 1992, pp. 705–713.
- [53] J. Savir, "Skewed-Load Transition Test: Part II, Coverage," in *Proc. International Test Conf.*, 1992, pp. 714–722.
- [54] J. Savir, "On Broad-Side Delay Testing," in *Proc. 12th VLSI Test Symp.*, 1994, pp. 284–290.
- [55] R. Schrift, "Digital Bus Faults Measuring Techniques," in *Proc. International Test Conf.*, 1998, pp. 382–387.
- [56] O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu, "Test power reduction through minimization of scan chain transitions," in *Proc. VLSI Test Symp.*, 2002, pp. 166–171.

- [57] C. E. Stroud, "AUSIM: Auburn University SIMulator - Version L2.2." Dept. of Electrical & Computer Engineering, Auburn University, Jan. 2004.
- [58] T. G. Susheel, J. Chandra, T. Ferry, and K. Pierce, "ATPG Based on A Novel Grid-Addressable Latch Element," in *Proc. ACM/IEEE Design Automation Conf.*, 2002, pp. 282–286.
- [59] A. J. van de Goor, *Testing Semiconductor Memories: Theory and Practice*. Chichester, UK: John Wiley & Sons, Inc., 1991.
- [60] J. T. van der Linden, M. H. Konijnenburg, and A. J. van de Goor, "Circuit Partitioned Automatic Test Pattern Generation Constrained by Three-State Buses and Restrictors," in *Proc. Asian Test Symp.*, 1996, pp. 29–33.
- [61] K. D. Wagner, "Design for testability in the amdahl 580," in *Digest COMPCON*, 1983, pp. 384–388.
- [62] C.-W. Wang, C.-F. Wu, J.-F. Li, C.-W. Wu, T. Teng, K. Chiu, and H.-P. Lin, "A Built-In Self-Test and Self-Diagnosis scheme for embedded SRAM," in *Proc. Asian Test Symp.*, 2000, pp. 45–50.
- [63] F. C. Wang, *Digital Circuit Testing*. San Diego, CA: Academic Press, Inc., 1991.
- [64] S. Wang and S. K. Gupta, "ATPG for Heat Dissipation Minimization During Scan Testing," in *Proc. ACM/IEEE Design Automation Conf.*, 1997, pp. 614–619.
- [65] N. Weste and K. Eshraghian, *Principles of CMOS VLSI Design*. Reading, PA: Addison-Wesley, 2nd edition, 1992.
- [66] M. J. Y. Williams and J. B. Angell, "Enhancing Testability of Large-Scale Integrated Circuits via Test Points and Additional Logic," *IEEE Trans. on Computers*, vol. C-22, no. 1, pp. 46–60, Jan. 1973.
- [67] X. Zhang and K. Roy, "Power Reduction in Test-Per-Scan BIST," in *International On Line Testing Workshop*, 2000, pp. 133–138.
- [68] Y. Zorian, E. J. Marinissen, and S. Dey, "Testing embedded-core-based system chips," *IEEE Trans. Computer*, vol. 32, no. 6, pp. 52–60, 1999.

APPENDICES

APPENDIX A

DESCRIPTION OF THE PROGRAMS USED TO IMPLEMENT THE VECTOR COMPACTING ALGORITHM

This appendix describes the various programs implemented to achieve the vector compaction. The programs were finally tied together using a script to automate the procedure. The flow of functions are described using Figure A.1.

The sequential circuit netlist is transformed to a combinational circuit by removing the flip-flops from the netlist and adding pseudo primary inputs and pseudo primary outputs. A compacted test set is obtained for the combinational circuit using an ATPG such as HITEC. Using the vectors obtained fault simulation is performed on a fault simulator that can provide detailed information about the detected faults such as the vector that detected the fault and the primary output (or pseudo primary output) at which it was detected. A vector re-ordering program is executed to obtain minimum scan operations. The vector re-ordering program can be explained using the following example:

Consider a circuit with three flip-flops. Assume the compacted vectors obtained from the ATPG as shown in Table A.1. The first column indicates the test vector number, the second column indicates the values of the pseudo primary input (PPI), the third column indicates the values of the pseudo primary outputs (PPO), the fourth column indicates the

Table A.1: Example vectors

Test set	PPI (i_i)	PPO (o_i)	Faults detected at	Modified PPO after read
t_1	000	111	FF1, FF3	010
t_2	011	101	FF2, FF3	110
t_3	110	110	FF1	010
t_4	010	001	-	001

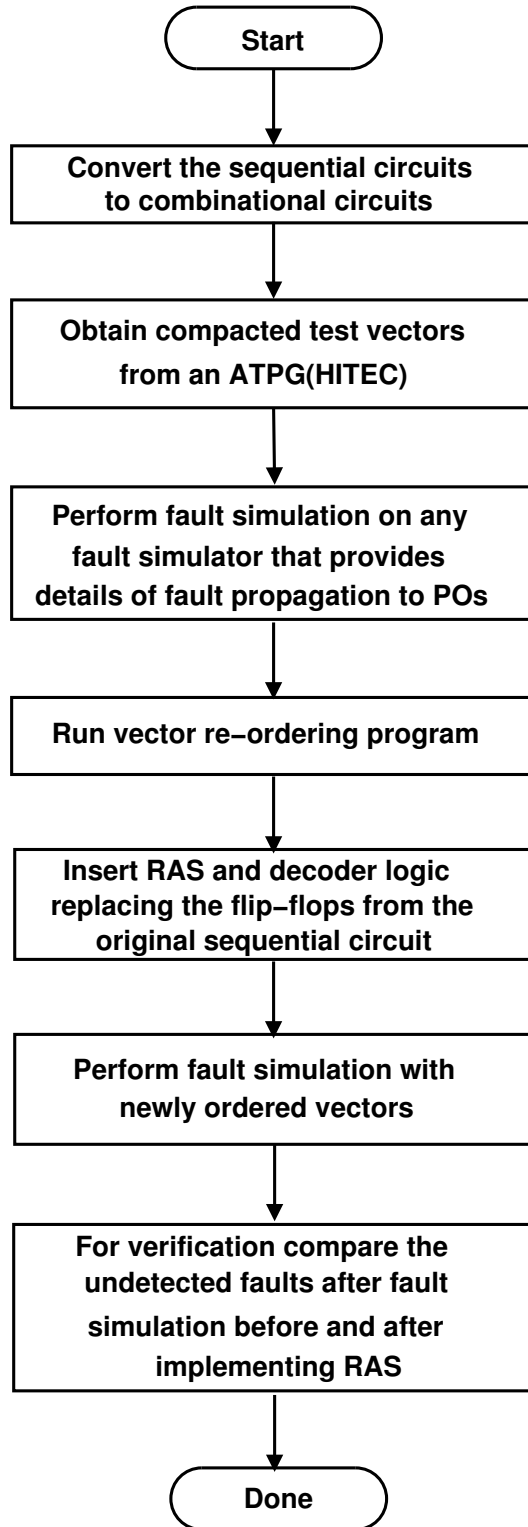


Figure A.1: Vector compaction program flow.

flip-flops or PPOs where the faults have been detected for which a read is to be performed and the fifth column indicates the value of PPOs after a read at the respective flip-flops.

The first vector is chosen to be the closest to the all zero state, which is t_1 . The program now searches the entire vector set for a closest match between the modified PPO value (which is “010” in this case) and a PPI state (which is t_4 in this case). The program stops searching when the first exact match is found. If an exact match is not found, the PPI with the least Hamming distance from the modified PPO state is chosen. It is now intuitive that the following vector after t_4 is t_2 and the last vector is t_3 . It is worth noting that we have reduced the number of scan operation from 2 to 1.

The next step in the flow is to remove the flip-flops from the netlist and add the RAS flip-flop and the decoders. A fault simulation is performed with the newly ordered vectors. To verify that all the combinational faults in the circuit are detected using the RAS architecture, the undetected faults from the two simulations (with and without RAS) are compared.

APPENDIX B

DESCRIPTION OF THE PROGRAMS USED TO CALCULATE THE POWER DISSIPATION DURING TEST

This appendix describes the program used to calculate the power dissipation during scan-in. We have assumed that the change in state at the input of any circuit is directly proportional to the activity in the circuit and hence proportional to the power dissipated by the circuit. A program to calculate the number of bit changes compared to the previous input during serial scan-in process was developed. This number for RAS is the change in the address bits during scan-in process. Essentially the program counts the number of transitions occurring at the primary and pseudo primary inputs of the circuit.