ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

Methodology to Detect and Diagnose Faults in Memories using BIST

S. Gayathri[1], V. Senthil Kumaran[2]

PG Scholar, ECE, Mahendra Engineering College, Namakkal, India [1]

Asst. Professor, ECE, Mahendra Engineering College, Namakkal, India [2]

Abstract: This project presents a scheme for detecting and diagnosing faults that are commonly occurred random access memory and Read only memory. The build in self test technique is used to identify permanent failures in embedded memories. The target of the project is fault detection and diagnosis in ROM and RAM such as single cell faults, row and column wise faults using Build in self test. In all the proposed test methods, Design for Test and Built-In Self-Test techniques have been proven to be very effective by the meaning of increasing the observability and controllability of the Circuit under Test thus fault is detected so as to increase the reliability of the memories. The proposed approach offers a simple test flow and does not require intensive interactions between a Build in Self Test controller and a tester. The scheme rests on partitioning of rows and columns of the memory array by employing low cost test logic. The signature is created for comparing the faulty cell and it is compared with the array of rows and columns. If there present the mismatch of signature already available with the testing cell then the fault is detected. Thus this project is designed to meet requirements of high-speed automatic test thus enabling detection of timing defects. It produces efficient way of identifying faults in memories.

Keywords: SRAM Read Only Memory (ROM), Random Access Memory (RAM).

I. INTRODUCTION

General

According to Moore's law, the number of transistors integrated per square inch one die has doubled every year and half since the integrated circuit was invented. Also, every few years the size of the transistors employed is shrunken and the frequency of circuits increases. As these trends continue, several new challenges become relevant in the testing of very-large-scale-integrated (VLSI) circuits. With the advance of semiconductor manufacturing technology, the requirements of digital VLSI circuits have led to many challenges during manufacturing test. This is because of the large and complex chips which require a huge amount of test data and dissipate a substantial amount of power during test, resulting in considerable increases in the test cost. This study addresses the problems of test and the problem of keeping the test data volume and test application time moderate. The main objective of this project is to introduce novel techniques that detect faults in Read only memory and Random access memory. This chapter introduces some important concepts in testing of digital VLSI memory circuits.

DFT Methods

Deploying reliable integrated circuits depends strongly on testing to eliminate defective circuits caused by the manufacturing process. Manufacturing test is performed after a circuit comes out of the manufacturing line to screen defective parts. The basic principle of manufacturing testing with its three basic components: Circuit under test (CUT), Automatic test equipment (ATE), and ATE memory to store test patterns or test vectors and expected responses obtained by automatic test pattern generation (ATPG) tools. To test a digital circuit several test vectors are applied to its inputs. Then, CUT response is analysed. If the CUT responses match the fault-free responses, then the circuit is considered to be functioning properly. The input test vectors and their responses are stored in an Automatic Test Equipment (ATE) which applies the tests to the CUT and analyze its responses.

The manufacturing test of a circuit composed of only combinational logic is a relatively easy task. The primary inputs can be set to the desired values and the primary outputs can be observed. However, the test of circuits containing sequential elements such as flip-flops or latches is a more complicated task. Sequential elements in the circuit need to be set to the desired values. In this case, the ATPG needs to create test sequences over many clockcycles to justify desired assignments to circuit inputs thatincreases run times and complexity of the test generation.Design for testability (DFT) refers to design techniques that make products easier to test. DFT techniques improve testability, in increasing controllability and observability of sequential elements by adding test hardware to the CUT. The most popular DFT techniques for testing VLSI circuits include scan design, Built-In Self-Test (BIST) and test data compression. In this sub-section, we briefly describe each of these techniques. Fig 1.2 Scan Based Circuit Design. The most common DFT methodology is scan design where sequential elements are modified to scan cells to obtain controllability and observability for flip-flops. This is performed by adding a test mode to the circuit such that when the circuit is in this mode, all flip-flops functionally form one or more shift

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

registers which is called scan chains. The inputs and outputs of these shift registers are made into primary inputs and primary outputs respectively. Thus using the test mode, all flip-flops can be set to any desired states by shifting appropriate logic values into the shift register. Similarly, the states of flip-flops are observed by shifting out the contents of the scan chains. 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 chain.

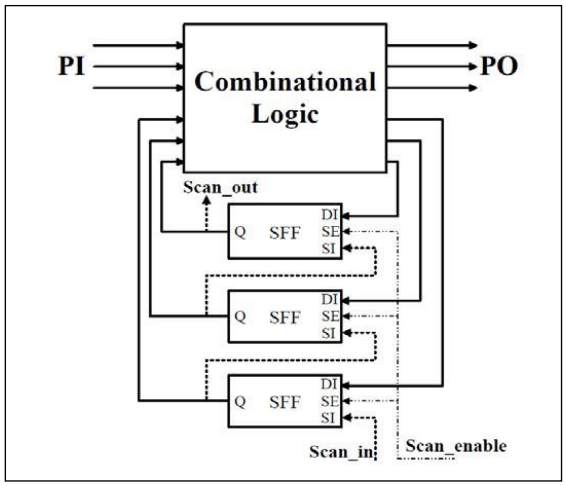

Fig. 1: Scan based circuit

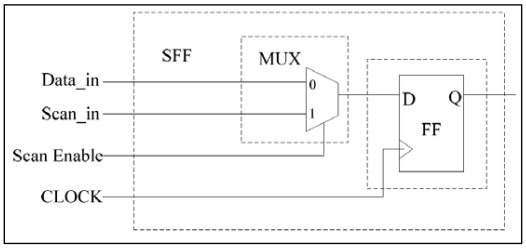

Fig. 2: Multiplexer based scan cell

Scan design can be further divided into full scan and partial scan designs. The main advantage of full scan (Fig. 1) is that by modifying all the sequential elements to scan cells it reduces the sequential TPG to combinational TPG. On the other hand, partial scan modifies only a small subset of sequential elements leading to lower test area overhead at the expense of more complex TPG.There are more than one possible implementations of a scan cell; the most common is shown in Fig. 2. This scan cell is composed of a D flip-flop and a multiplexer. Also the state of the circuit sequential elements can be observed by shifting out the values stored in scan cells through Scan_out. This enables the test of previously un-testable faults but it may set the circuit into non-functional states.

II. LITERATURE SURVEY

A. Fault Diagnosis for Embedded Read-Only Memories

N. Mukherjee, A. Pogiel, J. Rajski, J. Tyszer in their paper they presented a BIST-based scheme for fault diagnosis that can be used to identify permanent and address independent failures in embedded read-only memories. The proposed approach offers simple test flow and low power consumption. The scheme rests on partitioning of rows and columns of the memory array by employing low cost test logic. It is designed to meet requirements of at speed test thus enabling detection of time-related faults.

B. Diagnostic Testing of Embedded Memories Using BIST

Timothy J. Bergfeld, Dirk Niggemeyer, Elizabeth M. Rudnick in their research paper effective diagnostic memory tests of linear order O(N) are proposed that enable memory reconfiguration, and their diagnostic capabilities are analysed. In particular, these tests allow single-cell faults to be distinguished from multiple cell faults, such as coupling faults. In contrast to conventional O(N) tests, all cells involved in a fault are detected and localized, which allows complete reconfiguration using minimal-area BIST hardware that compares favourably with other BIST designs. The increasing use of large embedded memories in Systems-on-Chips requires automatic memory reconfiguration to avoid the need for external accessibility

C.Enabling Embedded Memory Diagnosis Via Test Response Compression

John T Chen, JanuszRajski, JitendraKhare, Omar Kebichi, WojciechMaly in their paper they introduced a method that enables failure diagnosis of BIST ed memories by compression of test responses. This method has been tested by simulation of memories with various specifications, fail patterns and test algorithms. The proposed method has been implemented in 0.18µ CMOS IC.Built-In Self-Test (BIST) is less often applied to random logic than to embedded memories due to the following reasons: a.)For satisfiable fault coverage it may be necessary to apply additional deterministic patterns, which cause additional hardware costs. b.) The BIST-signature reveals only poor diagnostic information.Recently; the first issue has been addressed successfully. The paper at hand proposes a viable, effective and cost efficient solution for the second problem. The paper presents a new method for Built-In Self-Diagnosis (BISD). The key advantage of this architecture is that all data, which is relevant for a subsequent diagnosis, is gathered during just one test session. The BISD method comprises a hardware scheme, a test pattern generation approach and a diagnosis algorithm. Experiments conducted with industrial designs substantiate that the additional hardware overhead introduced by the BISD method is on average about 15% of the BIST area, and the same diagnostic resolution can be obtained as for external testing.

D. A Microcode-based Memory BIST - Implementing Modified March Algorithm

DongkyuYoun_, Taehyung Kim and Sungju Park in their paper a new microcode-based BIST (Built-In Self Test) circuitry for embedded memory components were proposed. The memory BIST implements march algorithms which are slightly modified by adopting DOF (Degree of Freedom) concept to detect ADOFs (Address

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

Decoder Open Faults) on top of conventional stuck faults. Furthermore it is shown that the march BIST modified can capture a few NPSFs (Neighbourhood Pattern Sensitive Faults) coupled with the Cellular Automata address generator and patterns. The microcode-based memory BIST proposed lends itself to performing different combinations of March and retention tests with less microcode storage than the other approaches.

E. Built-in Self-test Technique for Selective Detection of Neighbourhood Pattern Sensitive Faults in Memories

Rajeshwar S. Sable, Ravindra P. Saraf, Rubin A. Parekhji and Arun N. Chandorkar in their research traditional tests for memories are based on conventional fault models, involving the address decoder, individual memory cells and a limited coupling between them is proposed. The algorithms used in these tests have been successively augmented to consider stronger coupling conditions. Built-in self-test (BIST) solutions for testing memories today incorporate hardware for test pattern generation and application for a variety of these algorithms. In this paper presents a BIST implementation for detection of neighbourhood pattern sensitive faults (NPSFs) in random access memories (RAMs). These faults are of different classes and types. More specifically, active, passive and static faults for distance 1 and 2 neighbourhoods, of types 1 and 2, are considered. It is shown how the proposed address generation and test pattern generation schemes can be made scalable for the given fault type under consideration.

III. BIST

A. Built In Self Test (BIST)

Built-In Self-Test (BIST) is a technique of designing additional hardware and software features into integrated circuits to allow performing self-testing. BIST is a DFT technique which employs on chip test pattern generator (TPG) and signature analyzer (SA). Fig. 3 shows a CUT with BIST. When the circuit is in test mode, a test pattern generator (TPG) generates patterns that are loaded into the CUT and a 5 signature analyzer (SA) examines the CUT response to the test patterns. The signature analyzer has an output to indicate if the circuit has passed or failed the test. In most BIST architectures, linear feedback shift registers (LFSRs) is usually used as a TPG because LFSR can generate sequence of good random property with little area overhead. The typical components of an LFSR are memory elements (latches or flip flops) and exclusive OR (XOR) gates. The signature analyzers (SAs) are commonly constructed from multiple-input signature registers (MISRs).

BIST is a good solution for testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In BIST, typically up to 95% coverage of stuck-at faults can be achieved provided that test points are employed to address random pattern resistance. Other types of fault models, such as transition or path- delay faults,

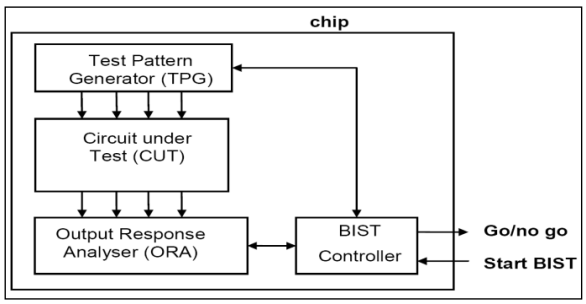

Fig. 3: High level view of the BIST scheme

are not handled efficiently by pseudorandom patterns. In BIST, all test responses have to be known. Unknown values corrupt the signature and, therefore, have to be bounded by additional test logic. Also, deterministic tests are almost always needed to target the remaining random pattern resistance faults.

B. Test Compression

As devices grew in gate count, scan test data volume and application time grew as well. Test compression techniques have been developed to reduce test data volume and test application time. Test compression techniques are easy to adopt in industry because they are based on scan. Test compression is achieved by adding some additional on-chip hardware before the scan chains to decompress the test stimuli coming from the tester and after the scan chins to compact the response going to the tester (Fig. 4).

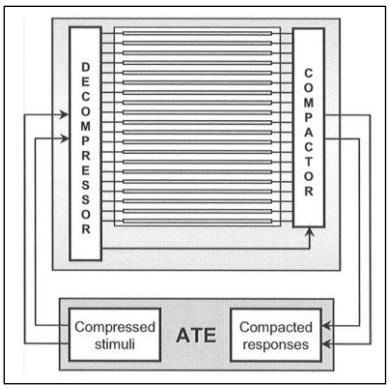

Fig. 4: Architecture for test compression

Here a technique called Embedded Deterministic Test (EDT) is described: EDT is based on adding a data decompressor at the inputs and response compactor at the outputs of the circuit. The data decompressor is implemented by a ring generator (optimized LFSR) and a phase shifter (Fig. 4). The phase shifter is necessary to drive a large number of scan chains and to reduce linear dependencies between sequences 7 entering the scan chains. The circuits scan chains are divided evenly, if possible, into several shorter chains. For the purpose of producing the desired output, the ring generator is continuously seeded with data. The ratio between the inputs of the ring generator and the outputs of the phase shifter determines the maximal compression possible. Fig. 5 shows the internal schematic of an on-chip

DOI 10.17148/IJARCCE.2015.4806

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

decompressor and Fig. 6 shows an implementation of a four-output 8-bit decompressor.

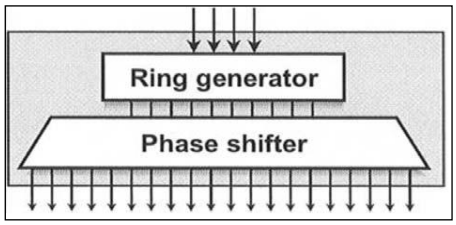

Fig. 5: On-chip decompressor

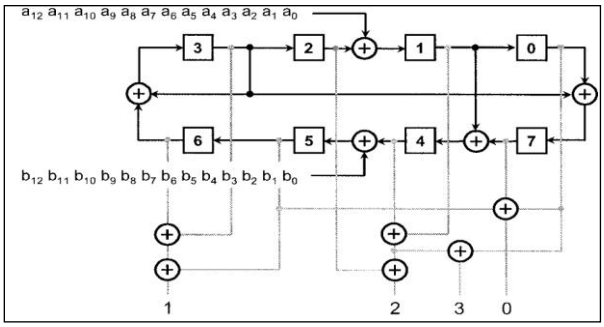

Fig. 6: Example of four-output 8-bit decompressor

In EDT the compactor consists of an XOR tree and masking logic. Since XORs always propagate fault effects (when no unknown values exist), every scan chain can be observed at the same time using a reduced numberof outputs which effectively reduce the response data. Since unknown values can be present in the CUT response to a test, AND-gates are placed at the outputs of every scan chain to selectively block these unknown values.

IV. FAULT MODELS

In this sub-section some most popular fault models, the stuck-at fault model, the transition fault model and the path delay fault model, will be reviewed.

A. Stuck at Fault Model

The stuck-at fault model is the earliest fault model, and still the most common. A stuck-at fault happens when a line in the circuit is stuck at a fixed logic value. To test for stuck-at faults, two steps are involved: one to generate a test vector that excites the fault and the other to propagate the faulty effect to a primary output or a scan flip-flop. Research has shown that stuck-at fault model covers a large percentage of physical defects. However, with the continuously shrinking sizes of the transistors employed in modern designs, increasing clock speed and decreasing power supply voltage, other types of defects not covered by the tests for stuck-at faults are beginning to appear in the CUTs. For this reason, tests for other types of faults are being applied, such as the transition fault model, which is presented next.

B. Transition Fault model

Certain types of defects in the manufacturing of the transistors that comprise the circuit gates may cause the gate to have a higher than normal delay. This abnormal delay causes the gate to switch at a lower than normal speed when its inputs change. When this delay is large enough the defect is modeled as a transition delay fault. The transitional delay fault model is the first delay fault model to be developed and is also the simplest. A transition delay fault occurs when the time required for switching outputs from 0 (1) to 1 (0) in the gate, due to a change in the gate's inputs, takes longer than its normal time. Depending on how the transition is launched and captured, there are three transition fault pattern generation methods: launch-off-shift or skewed load test method, launch-off-capture or broadside test method and enhanced scan which are briefly explained below.

C. Launch off Shift Method (LOS)

In launch-off-shift (LOS) approach, the transition at the gate output is launched in the last shift cycle during the shift operation. Fig. 7 shows the launch off- shift method waveform. The launch clock is a part of the shift operation and is immediately followed by a fast capture pulse. The scan enable (SEN) is high during the last shift and must go low to enable response capture at the capture clock edge. Since the capture clock is applied at the full system clock speed after the launch clock, the scan enable signal, which typically drives all scan flip-flops in the CUT, should also switch in the full system clock cycle. This requires the scan enable signal to be driven by a 10 sophisticated buffer tree or strong clock buffer. Such a design requirement is often too costly to meet.

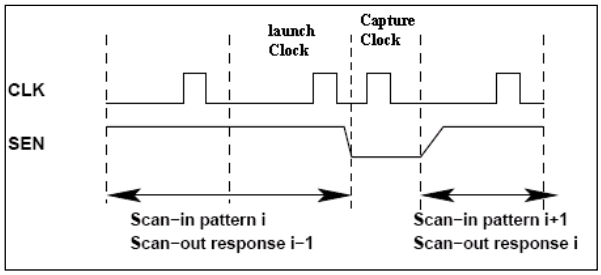

Fig. 7 Waveform for Launch-off-Shift delay test

D. Launch off Capture Method (LOC)

In the launch-off-capture approach, the launch cycle is separated from the shift operation. Fig. 4 shows the waveforms of the launch-off-capture (LOC) method. At the end of scan-in (shift mode), pattern V1 is applied and CUT is set to an initialized state.. The launch-off-shift method is more preferable based on the ATPG complexity and pattern count compared to LOC method. The LOC technique is based on a sequential ATPG algorithm, while the LOS method uses a combinational ATPG algorithm. This will increase the test pattern generation time in case of LOC, and also, a high fault coverage cannot be guaranteed due to the correlation between the two patterns, V1 and V2; note that V2 is the functional response of pattern V1. The main concern about the LOS is its requirement to at-speed SEN signal.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

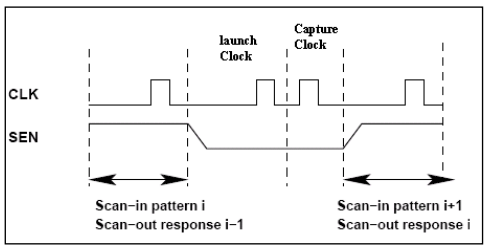

Fig. 8: Waveform for Launch-off-Capture delay test

E. Enhanced Scan Method

In the enhanced scan approach, two vectors V1 and V2 are shifted into the scan flip–flops simultaneously in order to initialize and propagate the fault. The drawback on enhanced scan is that it needs hold-scan flip–flops which make it unattractive for application specific integrated circuit (ASIC) designs.

F. Path Delay Fault Method

The path delay fault model focuses on the testing of a set of predefined structural paths in order to detect the accumulated delays along these paths. A path is defined as an ordered set of gates {go, g1, gn}, where go and gn are primary input and primary output, respectively and gate gi is an input to gate gi+1 ($0 < i < n-1$)). A delay defect on a path can be observed by propagating a transition through the path. The path delay fault model takes the sum of all delays along a path into accounts, while the transition fault model accounts for localized faults (delays) at the inputs and outputs of each gate. Test for path delay fault model can detect small distributed delay defects caused by statistical process variations. A major limitation of this fault model is that the number of paths in the circuit can be very large. Therefore testing all path delay fault in the circuit is not practical.

V. TESTING

A. Memory Testing

The memory built-in self-test (MBIST) has established itself as one of the mainstream design for test (DFT) methodologies as it allows one to generate, compress, and store on chip very regular test patterns and expected responses by using a relatively simple test logic. The available input/output channels, moreover, suffice to control built-in self-test (BIST) operations, including at-speed testing and detection of timing defects.Non-volatile memories are among the oldest programmable devices, but continue to have many critical uses. ROM, PROM, EPROM, EEPROM, and flash memories have proved to be very useful in a variety of applications. New non-volatile memories such as ferroelectric, magneto resistive, and phase changed RAMs retain data when powered off but are not restricted in the number of operation cycles. They may soon replace other forms of non-volatile memory as their advantages, e.g., reduced standby power and improved density, are tremendous.

No longer, however, is it sufficient to determine whether a memory failed or not. In ROM and RAM defect analysis and fine-tuning of a fabrication process, the ability to diagnose the cause of failure is of paramount importance. In particular, new defect types need to be accurately identified and well understood. It is also a common desire to verify if the programming device that is writing the Memory is working correctly. The method and accuracy of the diagnostic technique, therefore, is a critical factor in identifying failingsites of a memory array. It can be performed either on chip or off-line after downloading compressed test results.

Until recently, the main strategy for memory diagnosis was to have users provide an initialization file that describes the content of the memory. The initialization sequence can be random as far as the test is concerned. During the MBIST session, the content of the ROM or RAM is read multiple times using different addressing schemes and compressed into a signature. Current techniques either rely on downloading the signature value at certain intervals (based on binary search techniques) such that one can corner the test step when the MISR gets corrupted. Some other techniques suggest downloading the content of the entire memory when a failure occurs. Such techniques can get to the failing address and data, but they are complex, time consuming, and often prohibitive in practice.

Therefore, additional hardware is added to allow downloading the content of the entire memory. As the memory needs to be stopped after every read operation, the time needed to diagnose memory failures increases significantly.In this project, a low-cost test and diagnostic scheme that allows uninterrupted test response collection to perform accurate identification of failing rows, columns, and cells in read-only memories. The method utilizes a concept of partitioning, originally introduced for scan- based fault diagnosis in BIST environment. The proposed scheme partitions rows and columns of a memory array deterministically and records signatures corresponding to array segments being currently read (observed), every time narrowing down possible error locations until the failing rows and columns are determined.

Such approach neither requires interactions between BIST and automatic test equipment (ATE) nor interrupts a test flow.

TABLE 1 Basic Parameters of a MEMORY Array

R-The number of rows
B- The word size (the number of bits)
M-The number of words in a row (mux factor)
C-The number of columns ($C = B \times M$)
Memory Array Organization

Fig. 9 shows the salient architectural features of a memory array. Every row consists of M words, each B-bit long. Bits be-longing to one word can be either placed one after another or interleaved forming segments, as illustrated in the Fig.9.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

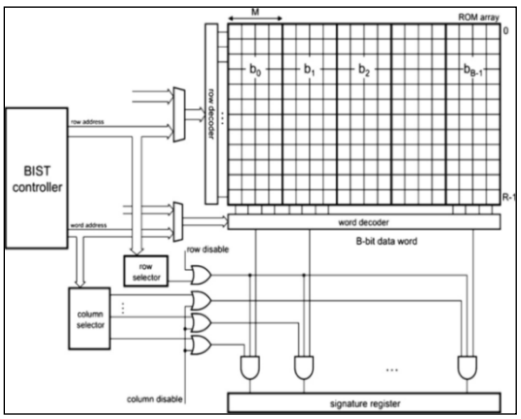

Fig. 9 Test Logic Architecture of Memory Array Organization

B. Collection of Diagnostic Data

The same Fig. 9 summarizes the architecture of a test environment used to collect diagnostic data from the memory arrays. Assuming permanent failures, the BIST controller sweeps through all memory addresses repeatedly while the row and column selectors decide which data arriving from the memory rows and/or columns is actually observed by the signature register. Depending on a test scenario, test responses are collected in one of the following test modes.

1) Row disable = 0 and column disable = 1; the row selector may enable all bits of the currently received word, thereby selecting a given row; this mode is used to diagnose row failures and, in some cases, single cell faults.

2) Row disable = 1 and column disable = 0; assertion of the row disable signal effectively gates the row selector off; the column selector takes over as it picks a subset of bit lines to be observed (this corresponds to selecting desired columns and is recommended to diagnose column and single cell failures).

Filter out failing sites accurately depends on how selection of observable rows and columns is carried out. Our scheme employs an enhanced version of deterministic partitioning originally proposed for scan-based diagnosis. It assures the fastest possible identification of fault sources down to the array nodes that cannot be recognized as fault-free ones. Details of the partitioning procedure will be presented in next chapters.

C. Signature Register

A signature register is used to collect all test responses arriving from selected memory cells. The register is reset at the beginning of every run (test step) over the address space. Similarly, the content of the register is downloaded once per run. A multiple input ring generator (MIRG) driven by the outputs of gating logic is used to implement the signature register. The design of Fig. 8 features the injector network handling the increasing number of input channels. It is worth noting that connecting each input to uniquely selected stages of the compactor makes it possible to recognize errors arriving from different input channels. In principle, selection of rows and columns that should be observed during a single diagnostic test run

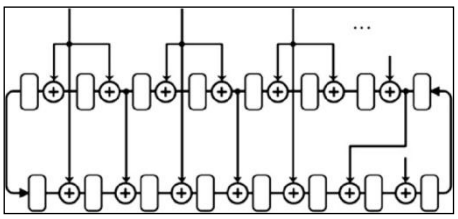

Fig. 10 MIRG-based signature register

proceeds in accordance with deterministic schemes and, for the sake of completeness, briefly summarized is as follows. The set of memory rows or columns is decomposed several times into groups of 2n disjoint partitions of approximately same size. In order to reduce test time, the number of partitions within each group should be small. Consequently, the same applies to the value of n. Hence, if the total number of memory rows or columns v is an even power of 2, then the value of n can be computed as 0.5 log2 v. Otherwise, $n = \lceil 0.5 \log_2 v \rceil$. As a result, the size of partitions may vary from 2n−1 to 2n. This rule guarantees the most time-efficient tracking down of faulty rows or columns. Indeed, if the array has x failing elements, then it suffices to run a test as indicated by x + 1 groups to determine the faulty items.

D. Row and Column selection

In this section, we introduce several hardware solutions for row and column selection. In particular, after presenting separate row and column selectors that implement a deterministic partitioning of a memory array, we introduce a scheme that allows one to partition rows and columns simultaneously.

E. Row Selection

We start by introducing the general structure of the row selector shown in Fig.5.4. Essentially, it is comprised of four registers. The up counters partition and group, each of sizen = $\lceil 0.5 \log_2 R \rceil$, keep indexes of the current partition and the current group, respectively. They act as an extension of the row address register that belongs to the BIST controller (the leftmost part of the counter in Fig.5.4). In principle, the circuit shown in Fig. 12 implements the following formula used to determine members r of partition p within group g:

$$r = S \cdot k + (p \oplus (g \otimes k)), k = 0,1,\ldots,P-1 \qquad (5.1)$$

where S is the size of partition, P is the number of partitions, $\oplus$ is a bit-wise addition modulo 2, and $g \otimes k$ is a state that the diffract or reaches after k − 1 steps assuming that its initial state was g. If k = 0, then $g \otimes k = 0$. For example, (1) yields successive partitions of Fig. 4.4 for S = 4 and k =0,1,2,3, assuming that the diffract or cycles through the following states: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$. Let g = 3 and p = 2. Then we have

k = 0: r = 4·0 + (2⊕(3⊗0)) = 0 + (2⊕0) = 2
k = 1: r = 4·1 + (2⊕(3⊗1)) = 4 + (2⊕3) = 5
k = 2: r = 4·2 + (2⊕(3⊗2)) = 8 + (2⊕1) = 11
k = 3: r = 4·3 + (2⊕(3⊗3)) = 12 + (2⊕2) = 12. (5.2)

With the ascending row address order, selection of rows within a partition, a group, and finally the whole test is

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

done as follows. The offset counter is reloaded periodically every time the n least significant bits of the row address register become zero (this is detected by the NOR gate N1). Once loaded, the counter is decremented to reach the all-0 state after p⊕ (g ⊗k) cycles. Hence, its asserted output enables observation of a single row within every S successive cycle.

As indicated by (1), the initial values of the offset counter are obtained by adding the actual partition number to the current state of the diffract or. Subsequently, the diffract or changes its state every time the offset register is reloaded. As the period of the LFSR-based diffract or is 2n − 1, and the offset counter is reloaded 2n times, the missing all-0 state is always generated at the beginning of a test run by means of the AND gates placed at the outputs of the diffract or.

a)

group	partition 0	partition 1	partition 2	partition 3	row address 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0, 4, 8, 12	1, 5, 9, 13	2, 6, 10, 14	**3, 7, 11, 15**				x				x				x				x
1	0, 5, 10, 15	1, 4, 11, 14	**2, 7, 8, 13**	3, 6, 9, 12								x								

b)

group	partition 0	partition 1	partition 2	partition 3	row address 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0, 4, 8, 12	**1, 5, 9, 13**	**2, 6, 10, 14**	**3, 7, 11, 15**		x	x	x		x	x	x		x	x	x		x	x	x
1	**0, 5, 10, 15**	**1, 4, 11, 14**	2, 7, 8, 13	3, 6, 9, 12		x				x					x	x			x	x
2	**0, 6, 11, 13**	**1, 7, 10, 12**	2, 4, 9, 15	**3, 5, 8, 14**		x				x					x	x			x	
3	0, 7, 9, 14	1, 6, 8, 15	**2, 5, 11, 12**	**3, 4, 10, 13**						x					x	x				

Fig. 11 Partition groups for 16-word memory. (a) Single faulty row. (b) Three faulty rows.

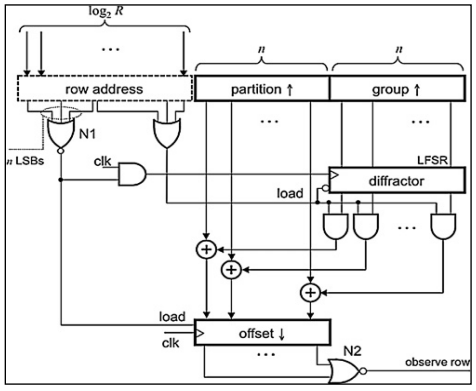

Fig. 12 Row selector.

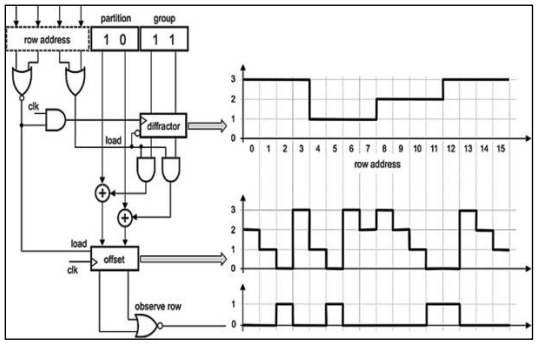

Fig. 13 Row selector operation.

F. Column Selector

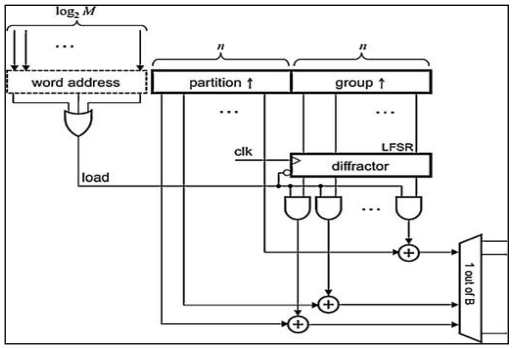

Fig. 14 Column selector.

Fig. 14 shows the column selector used to decide, in a deterministic fashion, which columns should be observed. Its architecture resembles the structure of the row selector as both circuits adopt the same selection principles. The main differences include the use of a BIST column address register and a diffract or clocking scheme. Moreover, the offset counter is now replaced with a combinational column decoder, which allows selection of one out of B outputs of the word decoder (see Fig.14). It is worth noting that the diffract or advances every time the column address increments. Its content added to the partition number yields a required column address in a manner similar to that of the row selection.If the size B of the memory word is equal to M (the number of words per row), it suffices to select one out of B columns at a time to cover all columns of the memory array for one partition group. Typically, however, we observe that B > M. This requires more than one column of each word to be selected at a time, as far as the single test run is concerned for every partition. The number t of columns observed simultaneously can be determined by dividing the maximal number of columns in a partition, which is 2n, by the number M of memory words per row

T =2n/M. (5.3)

It is important to note that columns observed in parallel cannot be handled by a single "t out of B" selector, as in such a case certain columns would always be observed together, thereby precluding an effective partitioning. Consequently, the output column decoder is divided into t smaller "1 out of B" decoders fed by phase shifters (PS), and then the diffractor, as shown in Fig.5.7. The phase shifters transform a given input combination in such a way that the resultant output values are spread in regular intervals over the diffractor state trajectory. Fig. 15 demonstrates this scenario for a 3-bit diffractor driving three phase shifters and using primitive polynomial x3 + x + 1. Let the diffractor be initialized to the value of 1. The phase shifters PS1, PS2, and PS3 are then to output states of the original trajectory, but starting with the values of 4, 6, and 5, respectively. When various partitiongroups are examined, the diffractor traverses the corresponding parts of its state space while the phase shifters produce appropriate values that ensure generation of all 2n − 1 combinations. The missing all-0 state is again obtained by means of AND gates.

DOI 10.17148/IJARCCE.2015.4806

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

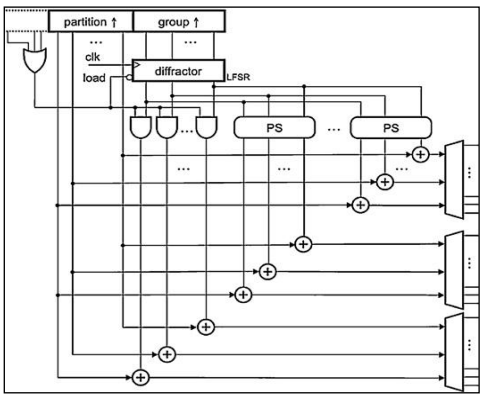

Fig. 15 Enhanced column selector.

G. Combined Row and Column Selection

In order to reduce the area overhead, some components of the row selector and the column selector can be shared. Since the word address increments prior to the row address, the memory array is read in the fast column addressing mode. As a result, the proposed scheme allows reading memory at-speed, and thus detection of timing defects. Finally, as the combined selector makes it possible to collect the row and column signatures in parallel, such an approach allows one to reduce the diagnostic time by half. In this mode, however, two signature registers are required.

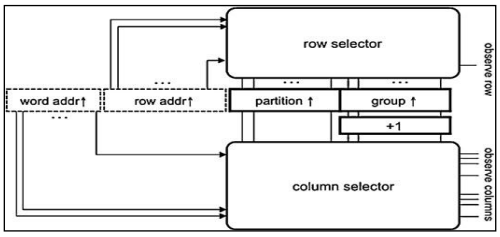

Fig. 16 Combined rows and column selector.

H. Single Cell Failure

Since the compactor (signature register) is a linear circuit, we work with so-called error signature E, which replaces the actual signature A, and can be obtained by adding modulo 2 a golden (fault-free) signature G to A, i.e., E = A⊕ G.

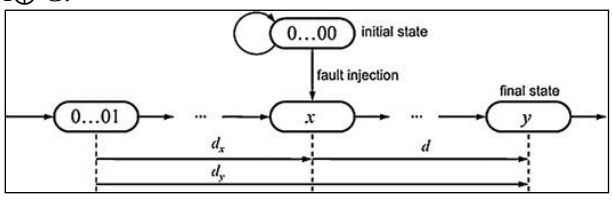

Fig. 17 Single cell failure diagnosis.

In terms of error signatures, the compactor remains in the all-0 state (Fig. 92) till a fault injection that moves the compactor to a certain state x determined by the compactor injector network. Subsequently, the compactor advances by additional d steps to reach state y. Typically, d is the number of steps required to complete a given memory run. The same value provides then the actual fault location which is the distance between states x and y, as recorded by the compactor.

The value of d, and hence a fault site, can be found by using a discrete logarithm-based counting. It solves the following problem: given an LFSR and a given state, determine the number of cycles needed to reach that state assuming that the compactor is initially set to 0. . . 001. When working with a failing row signature (most likely representing a single cell failure), a fault injection site (the compactor input) is unknown. Thus, d must be computed B times by applying repeatedly the following formula:

$$d = dy - dx \qquad (5.4)$$

wheredy and dx are distances between the state 0. . . 01 and states y and x, respectively. Recall that state x depends on where a fault is injected, so does dx. Finally, only $d < M \cdot R$ is considered an acceptable solution. It is worth noting that once accepted, the corresponding state x identifies uniquely the memory segment from which a fault arrives.

I. Execution of Testing

This section reports experiments are carried out to characterize performance of the proposed diagnostic scheme. In particular, a diagnostic coverage is used as a primary Fig. of merit. Assuming that we target up to x failing rows or columns, all numbers presented in this section have been obtained by adopting the following procedure.

1) Run tests for x + 1 column partition groups. Let xc be the resultant number of failing columns.
2) Repeat the same tests for x + 1 row partition groups. Let xr be the resultant number of failing rows.
3) If neither xc nor xr is less than or equal to x, then carry out the trellis selection and stop. Otherwise:
4) If xc ≤ x, then: Examine signatures (one per a failing column) collected in step (1) against single cell faults (by using the discrete logarithms-based counting).
5) If xr ≤ x, then: Examine signatures (one per a failing row) collected in step (2) against single cell faults (again by using the discrete logarithms-based counting).

On the other hand, in a rare case of multiple row and column failures, the trellis selection is the only feasible diagnostic approach, and thus the condition of step (3) must be checked before launching .The first group of experiments examines a relationship between the compactor size and the diagnostic coverage when attempting to identify single cell failures. Each entry to the table indicates a fraction of failures that were correctly diagnosed out of 100 K randomly generated single cell failures. In order to increase statistical significance of the experiments, the compactor injector network kept changing every 1000 failures.It is worth noting that only failing row signatures were considered as starting points to trace faulty cells. This experiment can be regarded therefore as the worst case analysis as far as the discrete logarithm-based counting is concerned. Typically, one may expect substantial improvement once failing column signatures are also available.

DOI 10.17148/IJARCCE.2015.4806

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

The size of the signature register can be crucial in achieving adequate diagnostic resolution and coverage. Interestingly, there is a coverage drop when comparing memories of the same capacity but having different number of segments. Apparently, the increasing number of segments adversely impacts the diagnostic coverage. Fortunately, this phenomenon is gradually diminishing with the increasing size of the compactor itself.It also appears that the discrete logarithms-based counting works fine even for memories greater than the compactor period. As an example, consider a 2 MB array comprising over16.8M memory cells. They may potentially produce 16.8M erroneous patterns. Nevertheless, a 32 input 20 bit compactor with the period of 1 048 575 guarantees almost 96% diagnostic coverage. This is because the diagnostic algorithm targets only memory cells belonging to the indicated failing rows.

The schemes proposed here were further tested on 128 kB and 2 MB memories working with 16 bit and 32 bit compactors, respectively. This group of experiments was aimed at determining the overall diagnostic coverage for faults commonly exhibited by memories. In principle, each entry to the table consists of two numbers. The first one is the percentage of faults of a given type that were correctly identified. The second number provides the percentage of test cases in which at least rows and/or columns that host the actual failure were part of the solution. Clearly, if the first number is 100%, the second assumes the same value and is, therefore, omitted in the table if the complete coverage was reached for all cases in a row. Note that each data presented here was obtained by injecting 10 K and 5 K randomly generated failures to 128 kB and 2 MB devices, respectively. In each case, the number of failures was chosen arbitrarily.

There trade off between accuracy of diagnosis and test application time (measured in memory runs—see the header third row). In particular, the increasing number of memory runs increases the diagnostic coverage as well. The best results are achieved for the largest partition groups. Predominantly, the coverage is complete. The proposed scheme always yields a solution that includes all columns and rows that host failing cells. It was meticulously verified during the experiments and is confirmed by the overwhelming presence of 100% numbers. A detailed analysis of the remaining test cases reveals that some diagnostic misses can be attributed to one of the following reasons.

1) Insufficient number of partition groups (mostly columns labelled 3 in the table). One may alleviate this drawback by simply collecting more signatures at the price of a longer test session.

2) Low diagnostic resolution due to a small compactor. It becomes apparent when looking for single cell failures. The diagnostic algorithm returns incorrect faulty sites that still belong to the same failing row/column as the actual faulty cell. As demonstrated earlier, a larger compactor can easily alleviate this problem.

3) Correlation between rows and columns (observed mainly when using the trellis selection). For larger memories this effect is negligible. For instance, there are 32 out of 1024 ≈3.1%) correlated rows and columns in 128 kB memory, whereas a 2 MB array lowers down this percentage to 1.6% (64 out of 4096).

It is also instructive to compare diagnostic times when employing the method proposed here and some conventional techniques.It is also instructive to compare diagnostic times when employing the method proposed here and some conventional techniques. In the rest of this section, we will delve into two of these techniques. According to the first one, each memory address location is read from and its content is dumped into an s-bit register, which is then immediately shifted out. This approach allows diagnosing any number of memory failures.

The second method follows a binary search scheme. The memory address space is divided in half, and the MBIST is run for both halves separately to collect two s bit signatures. Once the failing half is determined, one can continue running MBIST for the corresponding sub-halves to match signatures again. Clearly, this technique allows for correct identification of single memory failures only. Diagnostic time can be derived from the cycle time, memory size n (in terms of its words), and the number of cycles required to perform both read operations and serial download of the resultant signatures. The memory dump technique requires n read cycles and ns shift cycles to download a content of successive memory words. The binary search-based method proceeds as follows. First, it reads all n memory words and dumps two s-bit signatures. Next, it reads n/2 memory locations and again produces two signatures. This is roughly repeated log2 n times. Hence, it takes $n + n/2 + n/4 + \ldots \approx 2n$ cycles to carry out the read operations, and additional 2s log2 n cycles to download all relevant signatures.

The approach presented here reads all memory locations g times assuming that one targets at most g−1 faults. Since test time in this case is memory-architecture dependent, we will assume the worst-case scenario where the number of rows is equal to the number of memory words n. The three schemes discussed here would have then approximately the following diagnostic time:

1) the memory dump: $cn + rcns \approx rcns$;
2) the new method: $cg(n + rs\sqrt{n}).2n$; (5.5)

Clearly, the binary search offers the shortest test time. Unfortunately, it will only work for single failures. Let us now assume that n = 1024, s = 32 (the signature size), and r = 10 (the ratio of BIST and ATE clocks.) Then in order to locate faults of multiplicity 3 (which implies g = 4), it will take 327 680 cycles c to diagnose the memory by using the dump-based approach, whereas the method proposed here requires only 45 056 such cycles. For a4096-word here and otherwise the same conditions, the new method would be almost 23 times faster.

DOI 10.17148/IJARCCE.2015.4806

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 8, August 2015

VI. CONCLUSION

The BUILT IN SELF TEST is one of the most efficient ways of testing memories. Here the same technique is used for ROM and single cell failure and row cell failure and combined row and column failures is tested for given memory size. The similar technique is used for implementation of RAM also. Both memories are tested efficiently using the above technique and hence in future testing operation is carried out in low power and low cost by using this technique.

REFERENCES

[1] N. Mukherjee, A. Pogiel, J. Rajski, and J.Tyszer, "Fault diagnosis for embedded read-only memories," in IEEE transactionson computer-aided design of integrated circuits and systems, vol. 30, no. 7, July 2011 .
[2] R. D. Adams, "High Performance Memory Testing: Design Principles, Fault Modeling and Self-Test". New York: Kluwer, 2003.
[3] D. Appello, P. Bernardi, M. Grosso, M. Rebaudengo, and M. SonzaReorda, "Embedded memory diagnosis: An industrial workflow," in Proc. ITC, 2006, paper 26.2.
[4] S. Barbagallo, P. Camurati, A. Burri, D. Medina, P. Prinetto, and M.Sonza Reorda, "An experimental comparison of different approaches to ROM BIST," in Proc. Eur. Computer. Conf., 1991, pp. 567-571.
[5] I. Bayraktaroglu and A. Orailoglu, "The construction of optimal deterministic partitioning in scan-based BIST fault diagnosis: Mathematical foundations and cost-effective implementations," IEEE Trans. Computer., vol. 54, no. 1, pp. 61-75, Jan. 2005.
[6] T. J. Bergfeld, D. Niggemeyer, and E. M. Rudnick, "Diagnostic testing of embedded memories using BIST," in Proc. DATE, 2000, pp. 305-309.
[7] T. Boehler and G. Lehmann, "Using data compression for faster testing of embedded memory," U.S. Patent 6 950 971, Sep. 27, 2005.
[8] J. T. Chen and J. Rajski, "Method and apparatus for diagnosing memory using self-testing circuits," U.S. Patent 6 421 794, Jul. 16, 2002.[8]
[9] J. T. Chen, J. Rajski, J. Khare, O. Kebichi, and W. Maly, "Enabling embedded memory diagnosis via test response compression," in Proc. VTS, 2001, pp. 292-298.
[10] J. T. Chen, J. Khare, W. Maly J. Rajski, S. Shaikh and K. Walker,, "Test response compression and bitmap encoding for embedded memories in manufacturing process monitoring," in Proc. ITC, 2001, pp. 258-267.
[11] D. W. Clark and L.-J. Weng, "Maximal and near-maximal shift register sequences: Efficient event counters and easy discrete logarithms," IEEE Trans. Computer., vol. 43, no. 5, pp. 560-568, May 1994.

DOI 10.17148/IJARCCE.2015.4806