An Efficient Fault Detection and Diagnosis Methodology for Volatile and Non-Volatile Memories

Suren Martirosyan
Yerevan State University
Yerevan, Armenia
e-mail: suren.martirosyan.92@gmail.com

Gurgen Harutyunyan
Synopsys
Yerevan, Armenia
e-mail: gurgen.harutyunyan@synopsys.com

ABSTRACT
Memory reliability and testability are considered as primary requirements for achieving high production yield in nowadays system on chips (SoCs). For that purpose, different testing methods and diagnosis flows were proposed in the past. The fault models and test mechanisms can be different when dealing with volatile and non-volatile memories. This paper describes an efficient test methodology for detection and diagnosis of faults in both volatile and non-volatile types of memories.

Keywords
Static random access memory, flash memory, memory faults, test algorithm, March test, fault detection and diagnosis.

1. INTRODUCTION
Memory reliability is an important and critical requirement for SoCs. Embedded memories are growing rapidly to a large amount in terms of size and density. As they use more and more complex design structures, the occurrence probability of manufacturing defects becomes increasingly high. Therefore, testing of embedded memories using advanced nodes is a real challenge. Memories are divided into two categories: volatile memories, which require connection to power source to maintain the stored data and non-volatile memories, which can store the data regardless of power supply connection). The testing methodologies of volatile and non-volatile memories are different since these memories have different structures and functions. Non-volatile memories are popular for portable IoT devices due to their low power consumption and flexibility. On the other hand, volatile memories (such as static random access memories – SRAMs) are much faster and can be used in applications where maintaining the memory data is not needed after the system power off. Usually SoCs are using both volatile and non-volatile types of memories in the same design and there is actually a need to have a unified architecture that allows testing both volatile and non-volatile memories re-using the same test infrastructure.
The main problems of testing memory devices are fault detection and fault diagnosis. The aim of fault detection is to check whether the given memory contains a fault or not. Faults can be divided into simple faults (such as single-cell faults) or complex faults (such as coupling or linked faults). The role of fault diagnosis is to provide complete information about the memory faults, which includes: fault type, physical location of the fault, aggressor cell information in case of coupling faults, etc. There are numerous articles proposing different kind of test methods for solving the problem of fault diagnosis [1]-[5]. However, very little research was done so far on fault diagnosis for non-volatile memories especially Flash.
This paper presents a unified solution of fault detection and diagnosis for designs, which contain both volatile and non-volatile types of memories. In addition, it provides comprehensive multi-level diagnosis information using advanced test algorithms and memory scrambling information.

2. MEMORY FAULTS AND TEST APPROACHES
For volatile memory testing March test algorithms [6] are most popular since March-based testing algorithms have linear complexity w.r.t. memory cells meanwhile providing high fault coverage. Therefore, those kind of test algorithms are widely used in majority of modern memory built-in self-test (BIST) schemes [7]. For non-volatile memories, conventional March test algorithms are not sufficient because the operations and testing requirements for non-volatile memories are different. In order to test non-volatile memories extended March-like test algorithms [8] are used.

2.1. Testing Volatile Memories
The popular type of volatile memory is the random access memory (RAM). There are two types of RAMs: static (SRAM) and dynamic (DRAM). In [9] the architecture and difference between these memories is described.
Figure 1 shows the bit-cell of SRAM. It consists of 6 transistors. Access transistors (A1 and A2) are located at the edges. Each inverter from a pair of cross-coupled inverters contains a driver transistor (D1 and D2) and a pull-up transistor (P1 and P2). When the word line is activated (during write and read operations), the access transistors open access from the bit lines to cell internal nodes (Q and ~Q).

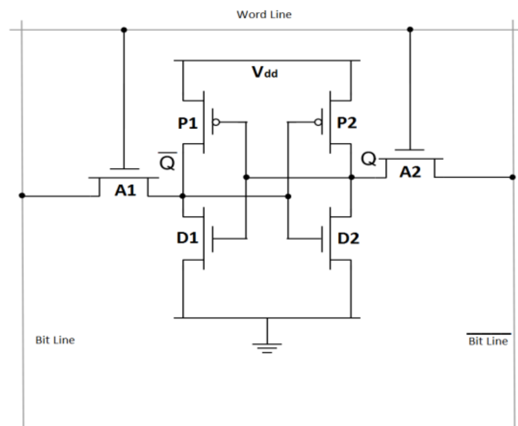

Figure 1. SRAM bit-cell

In standby state the access transistors are turned off since word line is not asserted, so the cell cannot be accessed, and the data is being hold. When reading from the cell, the word line is asserted and turns on the access transistors. The stored data is driven onto bit lines. A voltage difference occurs between bit lines and the sense amplifier detects the cell value. When writing to the cell, the word line is asserted, the value is applied to bit lines and the access transistor discharges one of the internal nodes.

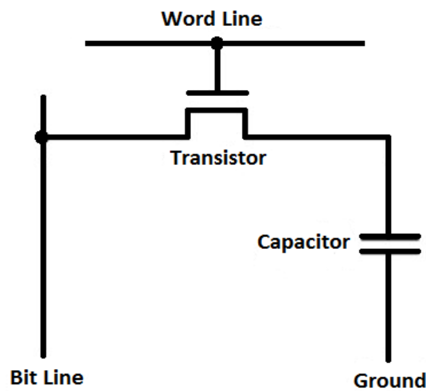

Figure 2. DRAM bit-cell

Figure 2 shows the bit-cell of DRAM. It consists of 1 transistor and 1 capacitor. The information is stored as a charge in capacitor. The transistor gives read and write access. When performing write operation, the state that the capacitor should take on is in word lines. The word line is opened, and the sense amplifier is forced to corresponding voltage state. When reading, if the charge in capacitor is more than 50%, the value is considered as 1, otherwise the value is 0. The charge is determined by the sense amplifier. Depending on SoC manufacturer, the set of faults in memory may vary. Fault is the logical representation of a physical defect (line is broken, short between lines, etc.) in the memory.

The sequence of write and read operations resulting in memory incorrect behavior is called a sensitizing operation sequence (S) and the behavior is called faulty behavior (F). The fault is described with combination of S, F and R (S/F/R), where R is the logical output of the read operation. This combination is called Fault Primitive (FP) [10]. The most common faults are classified in two groups:

1. Static faults: The faults that are being activated by performing a single operation. The common static fault is Stuck-at fault, when the cell is stuck at concrete value and write operation does not make any change on it. Transition faults, Read destructive faults, and coupling faults [10] also belong to the group of static faults.
2. Dynamic faults: The faults that require more than one consecutive operation to be activated. For example, two consecutive read operations applied to the same cell can flip the cell value (single-cell dynamic fault), or two consecutive write operations applied to a cell (aggressor cell) can change the value of another cell (victim cell) [10].

A single March test algorithm, which covers all static and dynamic faults may be unacceptable for some manufacturers because of its complexity. Therefore, there are a lot of different March test algorithms proposed in the literature with different fault coverage and complexities. The test algorithms for RAM fault testing are described in [6] and [10].

2.2. Testing Non-Volatile Memories

Flash memory is a common type of non-volatile memory. Figure 3 shows the transistor architecture of the Flash memory cell. It is based on floating gate (FG) concept. The cell value is determined by the charge in the FG. There are 2 ways of charging and discharging the FG: Fowler-Nordheim tunneling and channel hot electron injection (CHE) [11]. Unlike RAMs, Flash memories support three operations: read, program and erase. The read operation reads the stored data from the memory word. Program operation charges the cells of the word. Erase operation can discharge the cells of a

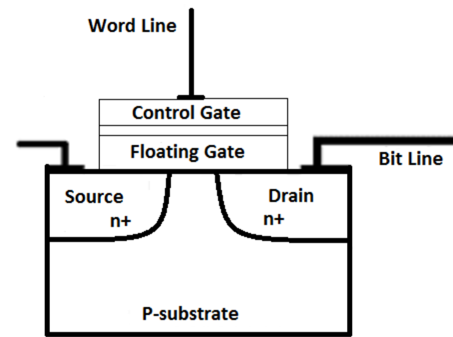

Figure 3. Flash bit-cell

sector or the cells of whole memory depending on memory architecture. The programming is performed by CHE, with drain and control gate (CG). The drain is connected to high voltage. The erase is performed by Fowler-Nordheim tunneling. A high voltage is applied to source diffusion and the electrons are being tunneled from FG to source diffusion by grounding CG. There are two types of Flash memories: NAND Flash and NOR Flash. They differ from each other with their architecture. NAND Flash provides faster program and erase compared to NOR Flash. On the contrary, NOR Flash provides random read access to memory word instead.

There are some common types of traditional RAM faults that can occur also in Flash memories such as stuck-open fault (SOF), address-decoder fault (AF), state-coupling fault (CFst), etc. However, since Flash memories have different cell architecture from RAMs, there are Flash specific faults that do not occur in RAMs. The set of these types of faults is called disturbance faults, which have the following subtypes:

- Word-line erase disturbance (WED);
- Bit-line erase disturbance (BED);
- Word-line program disturbance (WPD);
- Bit-line program disturbance (BPD);
- Source-line program disturbance (SPD);
- Gate read erase disturbance (GRE);
- Channel read-program disturbance (CRP);
- Read program disturbance (RPD);
- Read erase disturbance (RED);
- Over erase disturbance (OED);
- Over program disturbance (OPD);
- Read disturb (RD).

Disturbance fault types and their most common causes are discussed in [12].

In addition to the above discussed faults, Flash memories have two characteristics that also need to be tested properly.

1. Endurance: the parameter that measures the number of sequential program-erase cycles the memory can handle without any failure. The cycle may generate a defect, which can prevent from successful operation on memory cells.
2. Retention: the ability of memory cells to retain charged (programmed) state for a concrete period of time.

Endurance and retention are reliability issues and stress tests [13] are required for their testing. The March like test algorithms for testing disturbance faults and common faults with RAMs are discussed in [12].

3. PROPOSED TEST METHODOLOGY

In this paper, a unified BIST architecture is proposed, which allows to perform fault detection and diagnosis for volatile and non-volatile memories (see Figure 4). “RAM BIST” blocks contain Test Algorithm Register (TAR) for storing RAM fault detection and diagnosis algorithms and FSM (Finite State Machine), which performs the execution of the

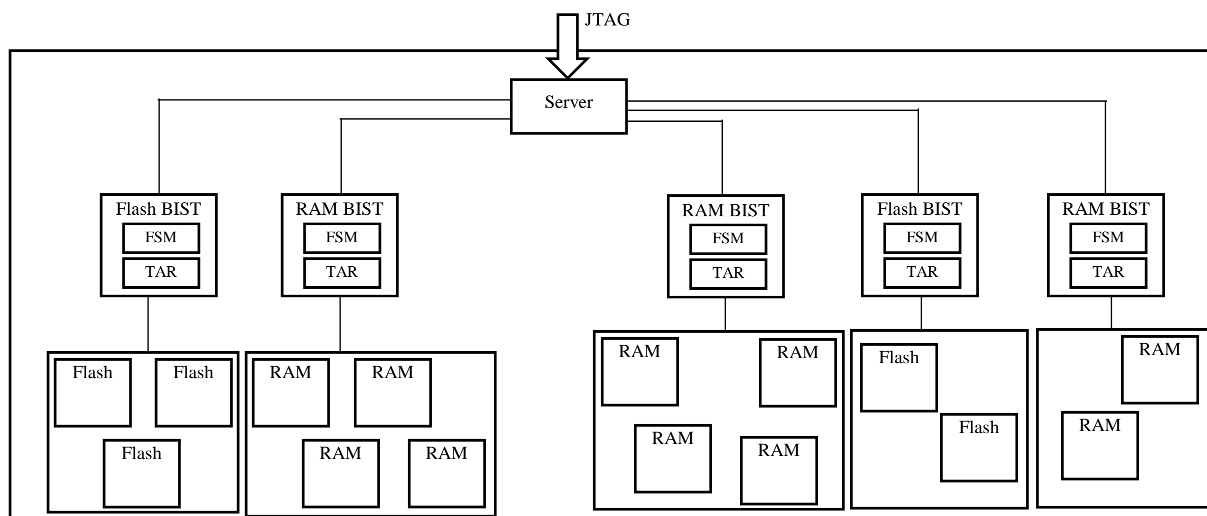

Figure 4. Proposed architecture

test algorithms on RAMs in two modes: normal mode (fault detection) and diagnosis mode. Similarly, "Flash BIST" blocks contain the corresponding TAR for storing Flash fault detection and diagnosis algorithms and the FSM for test algorithm execution in normal and diagnosis modes. Due to large number of memories in nowadays SoCs, multiple "RAM BIST" and "Flash BIST" blocks are used, which are shared across the memories. The grouping of memories can be done based on different criteria, such as memory frequency, proximity of memories in the chip, limit on power consumption or number of memories per BIST block, etc. On the top, all the BIST blocks are connected to the Server, which is in charge of test scheduling, i.e., it allows to run the BIST blocks in parallel or in serial based on resource and test limitations available for a given design. Server is connected to JTAG standard interface [14], which allows to apply test patterns from ATE (automatic test equipment). After running the test on ATE if a fault is detected then it will generate a datalog, which contains information on the following:

- at which cycle of test the fault was detected;
- which March element and which operation in March element were running at that time;
- in which memory address the fault was observed;
- in which data bit the fault was observed;
- etc.

This architecture has the following advantages:

- Unified interface for RAM and Flash BIST blocks;
- BIST block logic sharing across different memories;
- Possibility to test multiple memories in parallel;
- Possibility to run multiple BIST blocks in parallel.

For a given March test algorithm and a given fault, the corresponding March syndrome [1] is a vector of 0s and 1s, the size of which matches the number of read operations in the March test algorithm. The i-th bit of the March syndrome is 1 if i-th read operation of the March test algorithm detects the given fault, otherwise it is 0.

Figure 5 shows the proposed fault detection and diagnosis flow for embedded SRAM and Flash memories. The proposed flow is implemented in Synopsys DesignWare STAR Memory System product [15] and has already been used in many production chips using leading edge technology nodes including 16nm, 7nm and 5nm. It has the following functionalities:

1. Read a given design and create a database to include:
    a. the distribution of BIST blocks and their types (RAM or Flash BIST);
    b. the distribution of memories along with the information on how the memories are grouped under each BIST block;
    c. the test algorithms for Flash and RAM testing available in TARs and their purpose (for fault detection or fault diagnosis).
2. Create a dictionary of March syndromes for the fault diagnosis algorithms;
3. Based on the provided datalog (generated by ATE), report fault information (fault type, fault location, etc.).

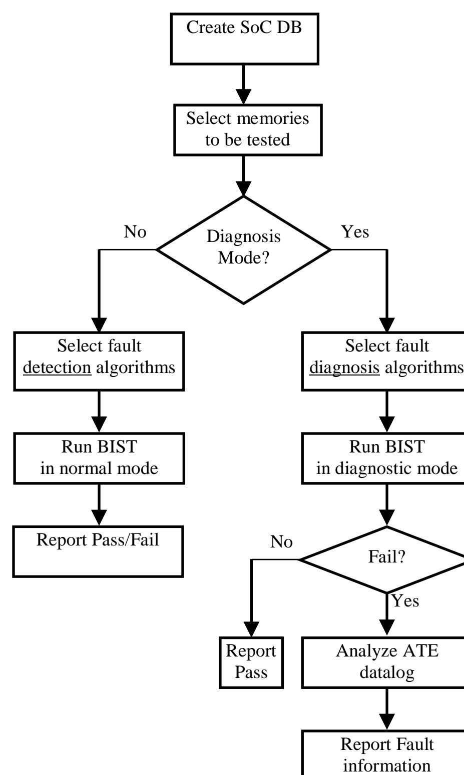

Figure 5. Fault detection and diagnosis flow

Table 1 lists a set of test algorithms that can be used by default (maybe length can also be mentioned). In addition, DesignWare STAR Memory System has a library of test algorithms, which can be used in case more complex faults need to be covered. While March MSL, March-FTE, March FD are well known test algorithms published in our previous works, March Flash-FD test algorithm has been developed within the scope of this work.

Table 1. Used Test Algorithms

March test	Memory	Purpose
March MSL [17]	SRAM	Detection
March-FTE [18]	Flash	Detection
March FD [16]	SRAM	Diagnosis
March Flash-FD	Flash	Diagnosis

Figure 6 shows multiple levels of the proposed fault diagnosis flow [16] supported by the tool. It starts from identifying whether a given memory instance has a fault or not. Then logical address and physical location of the fault are identified including the row/column and X, Y coordinates of the faulty cell. The next step is to identify the defect classification (i.e., whether it is single bit, pair bit, column/row failure, etc.). Finally, the flow allows to do fault classification and localization, i.e., to identify the fault type and location of aggressor cell in case of coupling faults.
Information on the memory scrambling (mapping between memory logical address and physical location of the bit-cell in the memory array) is necessary for creating optimal fault detection and diagnosis solutions. e.g., for generating accurate physical background patterns, reporting the exact physical coordinates of failed cells in the memory, etc.
In [19], it is stated that the lack of memory scrambling information can lead up to 35% test escapes.

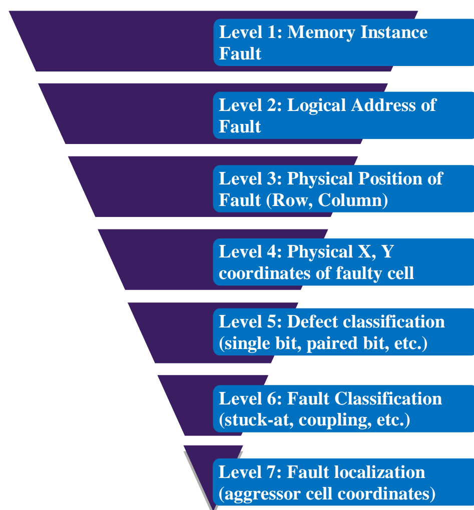

Figure 6. Multi-level fault diagnosis results

4. CONCLUSION

This paper presents an efficient test methodology for detection and diagnosis of faults in volatile and non-volatile memories. It starts with the discussion on fault types and testing approaches for both types of memories. Then the actual proposed flow is presented demonstrating how fault detection and diagnosis is done. The solution is implemented in Synopsys DesignWare STAR Memory System and has been used in many production chips.

REFERENCES

[1] J.-F. Li, K.-L. Cheng, C.-T. Huang, and C.-W. Wu, "March based RAM diagnostic algorithms for stuck-at and coupling faults", *IEEE ITC*, 2001, pp. 758-767.

[2] Z. Al-Ars, S. Hamdioui, "Fault Diagnosis Using Test Primitives in Random Access Memories", *IEEE Asian Test Symposium*, 2009, pp. 403-408.

[3] M. de Carvalho, P. Bernardi, M. Sonza Reorda, N. Campanelli, et al, "Optimized Embedded Memory Diagnosis", *IEEE International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, 2011, pp. 347–352.

[4] N. Campanelli, T. Kerekes, P. Bernardi, M. de Carvalho, et al, "Cumulative embedded memory failure bitmap display & analysis", *IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, 2010, pp. 255-260.

[5] T.J. Bergfeld, D. Niggemeyer, E.M. Rudnick, "Diagnostic Testing of Embedded Memories Using BIST", *Design, Automation and Test in Europe*, 2000, pp. 305-309.

[6] A.J. van de Goor, "Testing semiconductor memories: Theory and Practice", *John Wiley & Sons*, Chichester, England, 1991.

[7] Y. Zorian, S. Shoukourian, "Embedded-Memory Test and Repair: Infrastructure IP for SoC Yield", *IEEE Design and Test of Computers*, pp. 58-66, 2003.

[8] J.-Ch. Yeh, K.-L. Cheng, Y.-F. Chou, Ch.-W. Wu, "Flash Memory Testing and Built-In Self-Diagnosis with March-Like Test Algorithms", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 6, Jun. 2007, pp. 1101-1113.

[9] https://www.microcontrollertips.com/dram-vs-sram/

[10] S. Hamdioui, Z. Al-Ars, A.J. van de Goor, "Testing Static and Dynamic Faults in Random Access Memories", *IEEE VLSI Test Symposium*, 2002, pp. 395-400.

[11] F.-Ch. Hsu, K.-Y. Chiu, "A comparative study of tunneling, substrate hot-electron and channel hot-electron injection induced degradation in thin-gate MOSFET's", *International Electron Devices Meeting*, 1984.

[12] K.-L. Cheng, J.-Ch. Yeh, Ch.-W. Wang, Ch.-T. Huang, Ch.-W. Wu, "RAMSES-FT: A Fault Simulator for Flash Memory Testing and Diagnostics", *IEEE VLSI Test Symposium*, 2002, pp. 281-286.

[13] H. Aziza, J-M. Portal, J. Plantier, "Non volatile memory reliability prediction based on oxide defect generation rate during stress and retention tests", *International Semiconductor Device Research Symposium*, 2011, pp. 1-2.

[14] IEEE Std. 1149.1, IEEE Standard for Test Access Port and Boundary-Scan Architecture, 2001.

[15] K. Darbinyan, G. Harutyunyan, S. Shoukourian, V. Vardanian, Y. Zorian, "A Robust Solution for Embedded Memory Test and Repair", *IEEE Asian Test Symposium*, 2011, pp. 461-462.

[16] G. Harutyunyan, S. Martirosyan, S. Shoukourian, Y. Zorian, "Memory Physical Aware Multi-Level Fault Diagnosis Flow", *IEEE Transactions on Emerging Topics in Computing*, 2018.

[17] G. Harutunyan, V. A. Vardanian, Y. Zorian, "Minimal March Test Algorithm for Detection of Linked Static Faults in Random Access Memories", *IEEE VLSI Test Symposium (VTS)*, 2006, pp. 120-125.

[18] S. Martirosyan, G. Harutyunyan, S. Shoukourian, Y. Zorian, "An Efficient Testing Methodology for Embedded Flash Memories", *IEEE East-West Design and Test Symposium (EWDTS)*, 2017, pp. 422-425.

[19] A.J. van de Goor, I. Schanstra, "Address and Data Scrambling: Causes and Impact on Memory Tests", *IEEE International Workshop on Electronic Design, Test and Applications*, 2002, pp. 128-137.