International Journal of Electronics Communication and Computer Engineering
Volume 3, Issue 3, ISSN 2249 –071X

A Study of Built- in Self-test of RAMs

Hari Prasad Anasuri
Nova College of Engg. & Technology,
A.P., India
hariprasad.2591@gmail.com

G. Sravya Kukanakuntla
Assoc. Professor
Nova CET, JRG, A.P., India
sravya.ece.jrg@gmail.com

***Abstract* — The task of testing a RAM chip to guarantee its functionality is extremely complex and often very time consuming. In addition to the problem of testing the RAM chips themselves, the incorporation of the chips into systems has caused test generation's cost to grow exponentially. A widely accepted approach to deal with the testing problem at the chip level is to incorporate built- in self- test(BIST) capability inside a RAM chip. This increases the controllability and observability of the chip, thereby making the test generation and fault detection easier. In conventional testing, test patterns are generated externally by using computer- aided design (CAD) tools. The test patterns and the expected responses of the circuit under test to these test patterns are used by an automatic test equipment(ATE) to determine if the actual responses match the expected ones. On the other hand, in BIST, the test pattern generation and the output response evaluation are done on chip. Thus the use of expensive ATE machines to test chips can be avoided. In this paper, we present an overview of the types of BIST, and RAM functional fault models and various march tests for RAM as well as the advantages of BIST scheme.**

***Keywords* — Built- in self- test, March test, RAM, Test pattern .**

I. INTRODUCTION

Random access memory or RAM most commonly refers to computer chips that temporarily store dynamic data to enhance computer performance. By storing frequently used or active files in random access memory, the computer can access the data faster than if it to retrieve it from the far-larger hard drive. Random access memory is also used in printers and other devices. Random access memory is volatile memory, meaning it loses its contents once power is off.

Built-in Self Test, or BIST [1][2], is the technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing, i.e., testing of their own operation (functionally, parametrically, or both) using their own circuits, thereby reducing dependence on an external automated test equipment (ATE). BIST is a Design-for-Testability (DFT) technique, because it makes the electrical testing of a chip easier, faster, more efficient, and less costly. The concept of BIST is applicable to just about any kind of circuit, so its implementation can vary as widely as the product diversity that it caters to. As an example, a common BIST approach for DRAM's includes the incorporation onto the chip of additional circuits for pattern generation, timing, mode selection, and go-/no-go diagnostic tests. The main drivers for the widespread development of BIST techniques are the fast-rising costs of ATE testing and the growing complexity of integrated circuits. It is now common to see complex devices that have functionally diverse blocks built on different technologies inside them. Such complex devices require high-end mixed-signal testers that possess special digital and analog testing capabilities. BIST can be used to perform these special tests with additional on-chip test circuits, eliminating the need to acquire such high-end testers. BIST is also the solution to the testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In the near future, even the most advanced tester may no longer be adequate for the fastest chip, a situation wherein self-testing may be the best solution for.

In this paper we present an overview of the Built- in self- test scheme of RAMs. This paper is organized as follows: Section 2 presents the BIST techniques. Section 3 presents the RAM functional fault models. Section 4 presents the various March tests. Section 5 presents the advantages of BIST. And finally section 6 presents the conclusion.

II. BIST TECHNIQUES

BIST techniques are classified in a number of ways, but two common classification of BIST are

- Logic BIST(LBIST)
- Memory BIST(MBIST)

A. Logic BIST

LBIST [3], which is designed for testing random logic, typically employs a pseudo-random pattern generator(PRPG) to generate input patterns that are applied to the device's internal scan chain, and a multiple input signature register (MISR) for obtaining the response of the device to these test input patterns. An incorrect MISR output indicates a defect in the device.

B. Memory BIST

MBIST [4], as its name implies, is used specifically for testing memories. It typically consists of test circuits that apply, read, and compare test patterns designed to expose defects in the memory device. There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

One may also encounter the acronym "ABIST", which stands for two totally different BIST techniques: the Array BIST, which is a form of MBIST used for embedded memories, and the Analog BIST, which is a BIST approach for analog circuits. BIST is fast becoming an alternative solution to the rising costs of external electrical testing and increasing complexity of devices. This approach will find greater deployment in a wider variety of circumstances as more and better BIST techniques are developed. This does not mean, however, that BIST will eventually replace external electrical testing altogether. Still, BIST proponents are optimistic that BIST will

Copyright © 2012 IJECCE, All right reserved

someday be the preferred mode of testing, instead of being merely an alternative to external ATE testing as it is today.

III. RAM Functional Fault Models

In memory functional testing, we normally need to characterize the device first in order to determine the most likely failure modes of the *circuit under test* (CUT). After the dominant failure modes are identified, we can select a set of tests to detect these failure modes.

A. RAM Functional Fault Models

Functional fault models[5][6] are commonly used for memories. They define the functional behavior of the faulty memory. More and more fault models are being proposed to cover defects and failures in modern memory circuit and deep-submicron process technologies. Test algorithms are also being developed to detect these faults. Some of the faults that may occur in address decoder, read/write circuitry, and memory cell array of the RAM.

Address decoder faults (AFs) can be categorized as follows according to their functional behavior:

1. no cell can be accessed by a certain address.
2. multiple cells are accessed simultaneously by a certain address.
3. a certain cell is not accessible by any address and
4. a certain cell is accessible by multiple addresses.

As to the *read/write circuitry* (including buses, sense amplifiers, and write buffers), the typical faults are equivalent to faults in the memory cell array. Typical faults in the *memory cell array* are as follows:

1. Stuck-at fault (SAF)—A cell is stuck-at-1 or 0.
2. Stuck-open fault (SOF)—A cell is not accessible due to, *e.g.*, a broken wordline or a permanent open switch.
3. Transition fault (TF)—A cell fails to transit.
4. Data retention fault (DRF)—A cell fails to retain its logic value after a prespecified period of time.
5. Read disturb fault (RDF)—The cell value will flip when being read (repeatedly).

B. RAM Dynamic Faults

A static fault is one that has a static behavior; that is, its behavior does not change over time. A dynamic fault, on the other hand, has a dynamic behavior that may change over time. We give a few examples here.

1. A recovery fault occurs when some part of the memory cannot recover fast enough from a previous state.
2. A retention (refresh) fault occurs when the memory loses its content spontaneously, not caused by the read or write operation.
3. Another dynamic fault is the imbalance fault, where the bit-line voltage imbalance causes read errors.

IV. March Tests

March tests[7] are used to detect various RAM faults. A March test consists of a finite sequence of March elements, while a March element is a finite sequence of operations applied to every cell in the memory array before proceeding to the next cell. An operation can consist of writing a 0 into a cell (w0), writing a 1 into a cell (w1), reading an expected 0 from a cell (r0), and reading an expected 1 from a cell (r1).

Note that any March element can be done in either one of two address orders: the ascending order (⇑) or descending order (⇓). When the address order is irrelevant (*i.e.*, it can be either ascending or descending) then the symbol ⇕ is used.

The following are the different march tests for esting RAM:

Modified algorithmic test sequence (MATS): {⇕ (w0); ⇕ (r0,w1); ⇕ (r1}.

MATS^{+} : { ⇕ (w0); ⇑(r0,w1); ⇓(r1,w0)}

Marching 1/0: {⇑(w0);⇑(r0,w1,r1);⇓(r1,w0, r0); ⇑ (w1);⇑(r1,w0,r0);⇓ (r0,w1, r1)}.

MATS^{++}: {⇕ (w0);⇑(r0,w1); ⇓(r1,w0,r0)}

March X: {⇕ (w0); ⇑(r0, w1);⇓(r1,w0);⇕ (r0)}

March C: {⇕ (w0); ⇑(r0,w1);⇑(r1,w0); ⇕ (r0); ⇓(r0,w1);⇓(r1,w0); ⇕ (r0)}

March C^{-} : { ⇕ (w0); ⇑(r0,w1);⇑(r1,w0); ⇓(r0,w1);⇓(r1,w0); ⇕ (r0)}

March A : {⇕ **(**w0); ⇑(r0,w1,w0,w1); ⇑(r1,w0,w1); ⇓ (r1,w0,w1,w0); ⇓ (r0,w1, w0)}

March Y :{ ⇕ (w0); ⇑(r0,w1, r1); ⇓(r1,w0, r0); ⇕ (r0)}

March B : {⇕ (w0); ⇑(r0,w1, r1,w0,r0,w1); ⇑ (r1,w0,w1);⇓(r1,w0,w1,w0); ⇓(r0,w1,w0)}

V. Advantages of BIST

The following are some of the advantages of BIST:

Lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated.

Better fault coverage, since special test structures can be incorporated onto the chips.

Shorter test times if the BIST can be designed to test more structures in parallel.

Easier customer support.

Capability to perform tests outside the production electrical testing environment. This advantage can actually allow the consumers themselves to test the chips prior to mounting or even after these are in the application boards.

VI. Conclusion

A built-in self-test (BIST) is a mechanism that permits a machine to test itself. BIST is of two types: Logical BIST(LBIST) and Memory BIST(MBIST). LBIST is

Copyright © 2012 IJECCE, All right reserved

mainly designed for testing random logic, whereas MBIST is specifically designed for testing memories (EX: RAM). In this paper we present overview of BIST for RAMs. Functional fault models are commonly used for memories like RAM. They are used to define the functional behavior of faulty RAMs. To detect these faults test algorithms like march tests are used.

There are several advantages of BIST scheme. Some of them are low cost and time for testing, better fault coverage, easier customer support and capability to perform tests outside the production testing environment.

REFERENCES

[1] P. Camurati, P. Prinetto, M. S. Reorda, S. Barbagallo, A. Burri, and D. Medina, Industrial BIST of embedded RAMs, *IEEE Des. Test Comput.*, 12(3), 86–95, 1995.

[2] C.T. Huang, J.-R. Huang, and C.-W. Wu, A programmable built-in self-test core for embedded memories, in *Proc. Asia and South Pacific Design Automation Conf.*, January 2000, pp. 11–12.

[3] B. Cheon, E. Lee, L.-T. Wang, X. Wen, P. Hsu, J. Cho, J. Park, H. Chao, and S. Wu, At-speed logic BIST for IP cores, in *Proc. Design, Automation and Test in Europe*, March 2005, pp. 860–861.

[4] K. Zarrineh and S. J. Upadhyaya, On programmable memory built-in self test architectures, in *Proc. IEEE Design, Automation, and Test in Europe Conf.*, March 1999, pp. 708–713.

[5] R. Dekker, F. Beenker, and L. Thijssen, Fault modeling and test algorithm development for static random access memories, in *Proc. IEEE Int. Test Conf.*, October 1988, pp. 343–352.

[6] A. J. van de Goor and C. A. Verruijt, "An overview of deterministic functional RAM chip testing," ACM Comput. Surveys, vol. 22, no. 1, pp. 5–33, Mar. 1990.

[7] K. Zarrineh, S. J. Upadhyaya, and S. Chakravarty, A new framework for generating optimal march tests for memory arrays, in *Proc. IEEE Int. Test Conf.*, October 1998, pp. 73–82.

AUTHOR'S PROFILE

Hari Prasad Anasuri
Nova College of Engg. & Technology,
A.P., India
hariprasad.2591@gmail.com

G. Sravya Kukanakuntla
Assoc. Professor
Nova CET, JRG, A.P., India
sravya.ece.jrg@gmail.com

Copyright © 2012 IJECCE, All right reserved