International Journal For Technological Research In Engineering
Volume 2, Issue 7, March-2015 *ISSN (Online): 2347 - 4718*

DYNAMIC BUILT-IN SELF-REPAIR FOR EMBEDDED SRAM USING REDUNDANCY ARCHITECTURE

Md Baba Fareeduddin[1], Dr. Ameet Chavan[2]
[1]Department of Digital System and Computer Electronics
[2]((B.E. (Pune Univ.), M.S., Ph.D. (Univ. of Texas ELP)), Professor, School of electronics (ECE)
Sreenidhi Institute of Science and Technology (SNIST)
Hyderabad, India.

Abstract: ***In this paper, design of dynamic Built-in Self-Repair for Embedded SRAM is proposed. Built-in Self-repair is used to enhance the yield for embedded memories for effective memory diagnosis and fault analysis. BISR mainly consists of Built-in Self-test , Built-in fault-analysis and Multiplexer(MUX). In the proposed BISR, each fault can be saved only once . The main aim of the proposed BISR to repair a fault using redundancy by forming one – to-one mapping of a faulty location to redundancy location. By dynamic redundancy architecture we can repair more number of faults by replacing even single bit fault with single bit redundancy bit. There are few rows, columns, blocks and bits used as redundancy from normal rows, columns, blocks and bits of the memory***
Keywords: SRAM; Built-in Self-Test (BIST); Built-in Repair-Analysis(BIRA), Redundancy

I. INTRODUCTION

Due to rapid growth in VLSI technology, it is possible to integrate several billions of transistors on a single chip. Process variation causes failures like read failure, access failure, hold stability failures and write failure. As part of the development of VLSI technology, System-on chip (SoC) is developed which is capable to integrate a entire system on a single chip. It may possible to integrate many processor cores with embedded memory, interconnect infrastructure and application specific circuits embedded on a single chip which reduces the size of the system from a board to chip. SoCs gives high performance and higher reliability at low cost with low power consumption. According to international technology roadmap for semiconductor (ITRS) over 90% of the chip is occupied by embedded memories in System-on chip (SOC). In figure.1 architrcture of an Embedded SoC is shown, it is observed in the figure that large part of the architecture consists of embedded memories. So it is important that embedded memories should have high fault tolerance and high reliability which improves the yield of the memory. It is compulsory to test the correctness of the memory and diagnosis the faults. Testing is a measurement of defects and quality level.

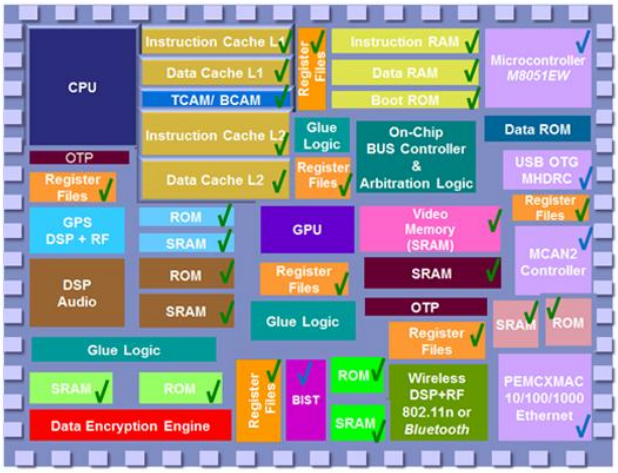


Figure1: Architecture of an Embedded SoC

There are two levels of testing. They are Production testing and Field testing. Production Testing is testing of newly manufactured component, this type of testing is carried out using automatic test equipment (ATE), which is very expensive and bulky. It is external hardware testing which is much expensive and increases test application time.

Fabrication industry → ATE → manufacturing defects

Another level of testing is Field Testing, testing is done on field of chip in a system. It is difficult to test each embedded memory on a board separately using ATE , as ATE test is costly and each chip has to be tested one after another which take much time. So in field testing we use extra built-in hardware and software circuitry to test the chip by it-self which is known as Built-in Self-test (BIST) which is used to check a memory by itself. BIST reduces the time to test a chip.

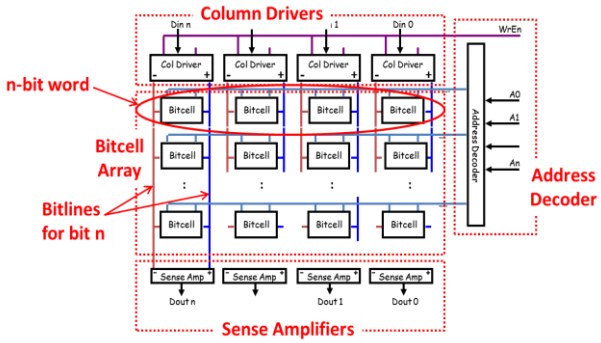


Figure2: Memory Architecture of SRAM cell

Typical SRAM memory architecture consists of the following sub-blocks (Figure 2).

- Column Driver—contains circuitry required for reading from and writing to the Bitcell Array.
- Bitcell Array—consists of a two-dimensional arrangement of bitcells, each of which stores one bit worth of data. A bitcell row comprises a word.
- Sense Amplifier—commonly called a sense amp, it contains analog circuitry for reading the contents of a selected bitcell row.
- Address Decoder—decodes words (bitcell rows) from a specified memory address.

II. FAULT MODELS AND TEST ALGORITHM

BIST is a design for testability (DFT) which has ability to test the faults by itself. BIST consists of test pattern generation, test controller and output response analysis. Test patterns are used to test the chip. To test Memory BIST, March algorithms are used to generate test patterns. As shown in the figure.3 Test patterns are applied from BIST to the circuit under test (CUT) using test controller. Output response analysis (ORA) is used to analyze the result with actual input patterns. A typical response analyzer is a comparator with stored responses and gives the result as the memory faulty or fault free. BIST adds complexity to the design but reduces time and cost of test.

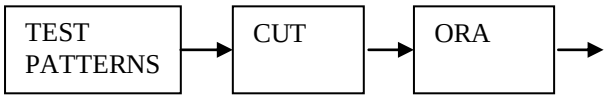


Figure.3: Basic architecture of BIST

A failure in memory cell can be occurs due to an increase in inability to hold the cell content, unstable read/write operation and access time. The mismatch in device parameters will increase the probability of their failures.

The various faults exist in memory are

- SAF -- Struck-at Faults
- SOF -- Stuck Open Faults
- DRF – Data Retention Faults
- AF -- Address Fault
- ADOF -- Address Decoder Open Faults
- TF -- Transition Faults
- CF -- Coupling Faults
    - Cfin –Inversion Coupling Faults
    - Cfid -- Idempotent Coupling Faults
    - CFst -- State Coupling Faults
    - BF -- Bridge Coupling Faults

There are March algorithms used to generate test patterns to test a memory. There are different march algorithms, few march algorithms have less test length but can detect only few faults and few other march algorithms have more test length but can detect more number of faults. So it is necessary to choose a correct algorithm for our design.

TABLE.I: VARIOUS MARCH ALGORITHMS AND ITS FAULT MODELS

Algorithm	**Test lenght**	**Fault coverage**
March 3	10n	AF,SAF,SOF and TF
March C	11n	SAF,TF,CFin,CFid,CFst and AF
March C-	10n	AF,SAF,TF,CFin,CFid and CFst
March C+	14n	AF,SAF,TF,SOF,CFin and CFid
March SS	22n	SAF,TF,WDF,RDF,DRDF,IRF,CFst, CFds and CFtr,CFwd,CFrd,CFdrd,CFir

Here “n” stands for the capacity of SRAM.

In this paper March SS algorithm is used to generate test patterns and test the faults.

March SS algorithm:

f m (w0) ;
M0
* (r0; r0; w0; r0; w1) ; * (r1; r1; w1; r1; w0) ;
M1 M2
+ (r0; r0;w0; r0; w1) ; + (r1; r1; w1; r1; w0) ;
M3 M4
m (r0) g
M5

From figure.3 comparing to other algorithms SS-algorithm have greater test length but it can detect more number of faults, so SS-algorithm is used to test the faults and detect more number of faults.

III. BUILT-IN SELF TEST(BIST)

Built-in Self Test, or BIST, 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. 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.

A. Advantages of Implementing BIST:

1) Lower cost of test, since the need for external electrical testing using an ATE will be reduced, if not eliminated;
2) Better fault coverage, since special test structures can be incorporated onto the chips;
3) Shorter test times if the BIST can be designed to test more structures in parallel;
4) Easier customer support;
5) Capability to perform tests outside the production electrical testing environment.

Built-in self-test (BIST) has been proven to be one of the most cost-effective and

The last advantage mentioned can actually allow the consumers themselves to test the chips prior to mounting or even after these are in the application boards.

B. BIST techniques:

BIST techniques are classified in a number of ways, but two common classification of BIST are the Logic BIST (LBIST) and the Memory BIST (MBIST).

LBIST, 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.

MBIST, 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 as shown in figure-4. There now exists a variety of industry-standard MBIST algorithms, such as the "March" algorithm, the checkerboard algorithm, and the varied pattern background algorithm.

Memory testing is more and more difficult

−Growing density, capacity, and speed

−Emerging new architectures and technologies

−Embedded memories: access, diagnostics & repair, heterogeneity, custom design, power & noise, scheduling, compression, etc

Widely used solutions for memory testing for the following reasons:

(1) No external test equipment
(2) Reduced development efforts;
(3) Tests can run at circuit speed to yield a more realistic test time;
(4) On-chip test pattern generation to provide higher controllability and observability.
(5) On-chip response analysis;
(6) Test can be on-line or off-line;
(7) Adaptability to engineering changes;
(8) Easier burn-in support.

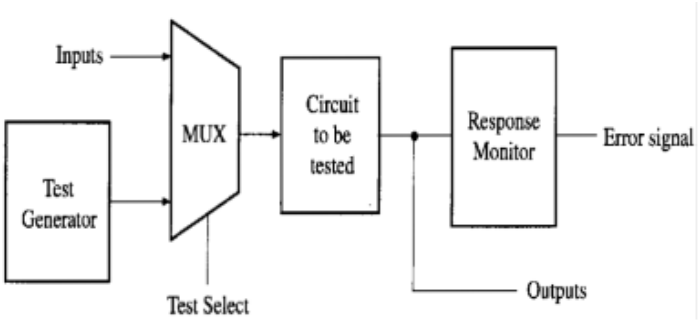


Figure.4: BIST architecture

In the above figure test patterns are generated using Test Generator and applied to a memory which is to be tested then the using comparator the inputs and test patterns are compared in response monitor

C. BIST Controller:

BIST controllers are mainly used to control pattern generators and comparators, and to strictly allow the controller to perform write and read operations in compliance with the Memory BIST Protocol.

D. Pattern Generator:

Pattern generators are mainly used to generate different patterns required for a Memory BIST operation. These Patterns can be Marching ones and zeros, Alternations ones and zeros, random numbers and fixed patterns.

E. Comparator:

Comparators are mainly used to compare the Memory Read data and compare against the expected values, the values generated by the pattern generator. Any failing comparisons will be flagged as BIST fail by the BIST Controller.

IV. BUILT-IN SELF REPAIR(BISR)

Firstly, the BISR strategy is flexible. TABLE II lists the operation modes of SRAM. In access mode, SRAM users can decide whether the BISR is used base on their needs. If the BISR is needed, the Normal-Redundant words will be taken as redundancy to repair fault. If not, they can be accessed as normal words.

TABLE II. SRAM OPERATION MODES

Modes	Repair Selection	Operation
Test mode (test_h=1)	Default: repair (bisr_h=1)	Access normal words. Repair faults and test.
	Don't repair (bisr_h=0)	Access normal words. Test only.
Access mode (test_h=0)	Repair (bisr_h=1)	Access normal words. Repair faults and write/read SRAM.
	Don't repair (bisr_h=0)	Access Normal-Redundant and

		normal Words Write/read SRAM only.

Secondly, the BISR strategy is efficient. On one hand, the efficiency reflects on the selectable redundancy which is described as flexible above. No matter the BISR is applied or not, the Normal-Redundant words are used in the SRAM. It saves area and has high utilization. On the other hand, each fault address can be stored only once into Fault-A-Mem. As said before, March C- has 6 steps. In another word the addresses will be read 5 times in one test. Some faulty addresses can be detected in more than one step. Take Struck- at-0 fault for example, it can be detected in both 3rd and 5th steps. But the fault address shouldn't be stored twice. So we propose an efficient method to solve the problem in BIAA module. Figure 4 shows the flows of storing fault addresses. BIST detects whether the current address is faulty. If it is, BIAA checks whether the Fault-A-Mem overflows. If not, the current fault address should be compared with those already stored in Fault-A-Mem. Only if the faulty address isn't equal to any address in Fault-A-Mem, it can be stored. To simplify the comparison, write a redundant address into Fault-A-Mem as background. In this case, the fault address can be compared with all the data stored in Fault-A-Mem no matter how many fault addresses have been stored. At last, the BISR strategy is high-speed. As shown in Figure 5, once a fault address is stored in Fault-A-Mem, it points to a certain redundant address. The fault addresses and redundant ones form a one-to-one mapping. Using this method, the BISR can quickly get the corresponding redundant address to replace the faulty one.

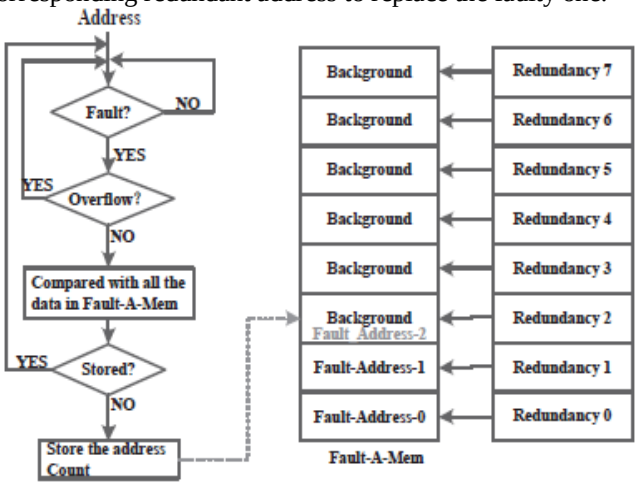


Figure5: Flows of Storing Fault Addresses

V. PROPOSED DYNAMIC BISR

Built-in Self-repair (BISR) is an extra circuitry used to repair the faults by itself using built-in circuitry. BISR mainly consists of BIST, Built-in Fault analysis (BIFA) and Multiplexer. When the faults are tested and detected using BIST, they are repaired by BISR using redundancy architecture. Same faults cannot be stored more than once in the proposed BISR. To mask the faults with redundant location redundancy is used. Faults masking of small size memories, triple modular redundancy (TMR) can be used. It is not preferable for large size memories because of additional large hardware overhead. The faulty location and redundant location forms one-to-one mapping to achieve high repair rates. In our proposed BISR we are using March SS-Algorithm to detect the faults which can detect more number of faults compare to other algorithms shown in table-1. The detected faults are analyzed by BIRA and categorized into different categories as per the number of faults. These faults are repaired by dynamic Built-In Self repair using redundant bits and address locations. If there are more number of faults in a address location then the faulty address will be mapped with redundant address location and if there is few bits faults in a address then that particular faulty bit will be mapped with redundant bits.

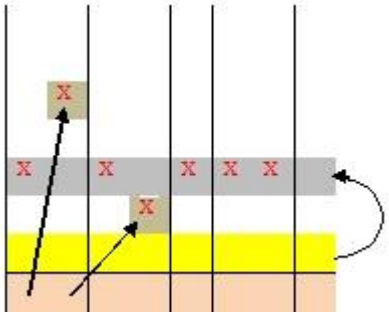
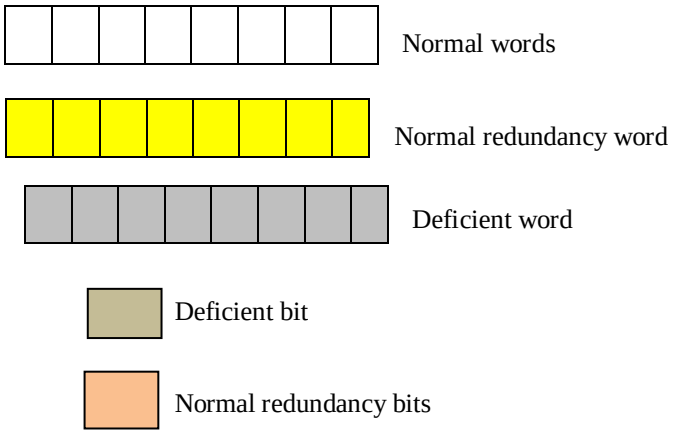


Figure.6: Redundancy architecture

In the above figure 8x8 memory which has normal words, normal-redundant bits or words and deficient bits or words is shown as



In this type of redundancy architecture instead of using spare words/bits, normal words/bits are used as normal redundancy words or normal redundancy bits, so spare memory is not required. As normal words/bits are used as normal redundant words or normal redundant bits instead of spare words/bits, it does not make any change in compiler design. The main advantage of using this built-in repair analysis is we can save the faults only once and increment the counter value by one. The faulty word/bit form one-to one mapping which increases repair rate. This type of test and repair of a chip is known as internal testing, which mean the test and repair is done by the circuit it-self.

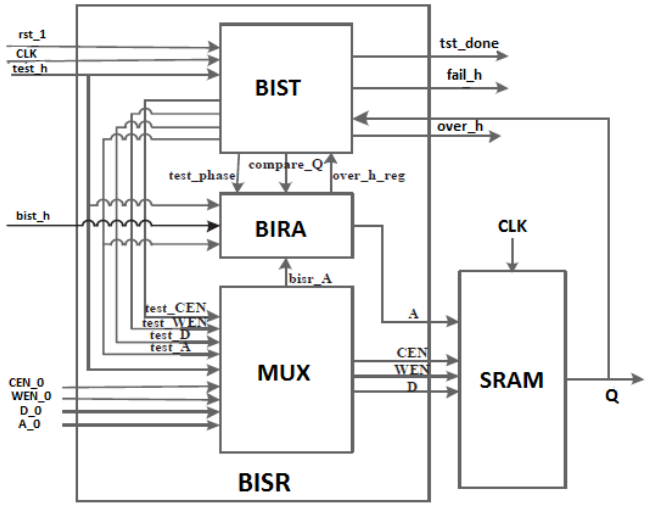


Figure.7: BISR Architecture

Built-in Repair Analysis (BIRA) is used to check the number of faults exits. If there are few faults then normal redundant bits are used to cope the fault and if many faults exits in a word then normal redundant word is used. The faulty locations and normal redundancy locations forms one-to-one mapping to achieve high repair rate. Fault can be occurred due to various reasons like if a bit is struck at zero or struck at one, the bit value will not change even if the input is changed. Many faults are detected during reading of the memory.

VI. CONCLUSIONS

Dynamic BISR strategy for SRAM with selectable redundancy has been presented in this paper. It is designed flexible that users can select operation modes of SRAM dynamically. More number of faults are detected using March –SS algorithm. The BIRA module can avoid storing fault addresses more than once and can repair fault address and bits quickly. Dynamic redundancy architecture repair more number of faults by replacing even single bit fault with single redundancy bit.

REFERENCES

[1] Semiconductor Industry Association, "International technology roadmap for semiconductors (ITRS), 2003 edition," Hsinchu, Taiwan, Dec.2003.

[2] C. Stapper, A. Mclaren, and M. Dreckman, "Yield model for Productivity Optimization of VLSI Memory Chips with redundancy and Partially good Product," IBM Journal of Research and Development, Vol. 24, No. 3, pp. 398-409, May 1980.

[3] W. K. Huang, Y. H. shen, and F. lombrardi, "New approaches for repairs of memories with redundancy by row/column deletion for yield enhancement," IEEE Transactions on Computer-Aided Design, vol. 9, No. 3, pp. 323-328, Mar. 1990.

[4] P. Mazumder and Y. S. Jih, "A new built-in self-repair approach to VLSI memory yield enhancement by using neuraltype circuits," IEEE transactions on Computer Aided Design, vol. 12, No. 1, Jan, 1993.

[5] Shyue-Kung Lu, Chun-Lin Yang, and Han-Wen Lin, "Efficient BISR Techniques for Word-Oriented Embedded Memories with Hierarchical Redundancy," IEEE ICIS-COMSAR, pp. 355-360, 2006.

[6] C. Stroud, A Designer's Guide to Built-In Self-Test, Kluwer Academic Publishers, 2002.

[7] Karunaratne. M and Oomann. B, "Yield gain with memory BISR-a case study," IEEE MWSCAS, pp. 699-702, 2009.

[8] I. Kang, W. Jeong, and S. Kang, " High-efficiency memory BISR with two serial RA stages using spare memories," IET Electron. Lett., vol. 44, no. 8, pp. 515-517, Apr. 2008.

[9] Heon-cheol Kim, Dong-soon Yi, Jin-young Park, and Chang-hyun Cho, "A BISR (Built-In Self-Repair) circuit for embedded memory with multiple redundancies," in Proc. Int. Conf. VLSI CAD, Oct. 1999, pp. 602-605.

[10] M. Sachdev, V. Zieren, and P. Janssen, " Defect detection with transient current testing and its potential for deep submicron CMOS ICs," IEEE International Test Conference, pp. 204-213, Oct. 1998

[11] Pavlov. Andrei and Sachdev. Manoj, CMOS SRAM Circuit Design and Parametric Test in Nano-Scaled Technologies, CA: Springer, 2008, pp. 85-86.

[12] Understanding SRAM is key to understanding the effects of various memory operations on a power grid., SEPTEMBER 12TH, 2013 - BY: MARKO CHEW

