An Advanced and more Efficient Built-in Self-Repair Strategy for Embedded SRAM with Selectable Redundancy

A. Sharone Michael.1[#1], K.Sivanna.2[#2]

#1. M.tech student Dept of Electronics and Communication, Stanley Stephen College of Engineering and Technology, Kurnool, AP, India.

#2. Associate professor, Dept of Electronics and Communication, Stanley Stephen College of Engineering and Technology, Kurnool, AP, India.

Abstract: **Built-in self-test (BIST) refers to those testing techniques where additional hardware is added to a design so that testing is accomplished without the aid of external hardware. Usually, a pseudo-random generator is used to apply test vectors to the circuit under test and a data compactor is used to produce a signature. To increase the reliability and yield of embedded memories, many redundancy mechanisms have been proposed. All the redundancy mechanisms bring penalty of area and complexity to embedded memories design. To solve the problem, a new redundancy scheme is proposed in this paper. Some normal words in embedded memories can be selected as redundancy instead of adding spare words, spare rows, spare columns or spare blocks.**

Built-In Self-Repair (BISR) with Redundancy is an effective yield-enhancement strategy for embedded memories. This paper proposes an efficient BISR strategy which consists of a Built-In Self-Test (BIST) module, a Built-In Address-Analysis (BIAA) module and a Multiplexer (MUX) module. The BISR is designed flexible that it can provide four operation modes to SRAM users. Each fault address can be saved only once is the feature of the proposed BISR strategy. In BIAA module, fault addresses and redundant ones form a one-to-one mapping to achieve a high repair speed. Besides, instead of adding spare words, rows, columns or blocks in the SRAMs, users can select normal words as redundancy.

Keywords: **SRAM; Built-In Self-Repair (BISR); Built-In Self Test (BIST); Built-In Address-Analysis (BIAA); Compiler.**

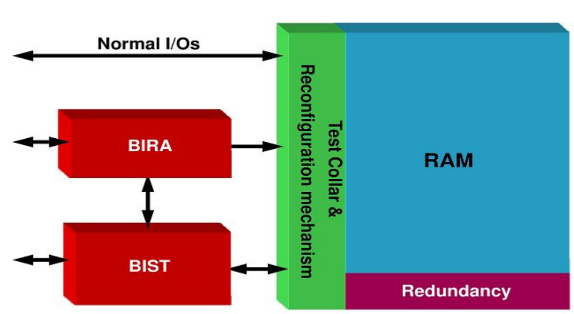
I. INTRODUCTION

In Present era, the area occupied by embedded memories in System-on-Chip (SoC) is over 90%, and expected to rise up to 94% by 2014. Thus, the performance and yield of embedded memories will dominate that of SoCs. However, memory fabrication yield is limited largely by random defects, random leakage defects, random oxide pinholes, gross processing and assembly faults, specific processing faults, gross photo defects, misalignments and other faults and defects.

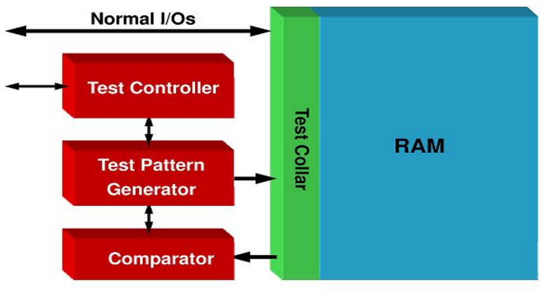
To increase the reliability and yield of embedded memories, many redundancy mechanisms have been proposed. Both redundant rows and columns are incorporated into the memory array. The spare words, rows, and columns are added into the word-oriented memory cores as redundancy. All these redundancy mechanisms bring penalty of area and complexity to embedded memories design. Considered that compiler is used to configure SRAM for different needs, the BISR had better bring no change to other modules in SRAM. To solve the problem, a new redundancy scheme is proposed in this paper. Some normal words in embedded memories can be selected as redundancy instead of adding spare words, spare rows, spare columns or spare blocks.

Testing Memory is compulsory before using redundancy to repair. Design for test (DFT) techniques proposed in 1970 improves the testability by including additional circuitry. The DFT circuitry controlled through a BIST circuitry is more time-saving and efficient compared to that controlled by the external tester (ATE). However, memory BIST does not address the loss of parts due to manufacturing defects but only the screening aspects of the manufactured parts.

Typical Memory BISR Architecture



Typical Memory BIST Architecture



BISR techniques aim at testing embedded memories, saving the fault addresses and replacing them with redundancy. All the previous BISR techniques can repair memories, but they didn't tell us how to avoid storing fault address more than once. This paper proposes an efficient BISR strategy which can store each fault address only once.

The rest of this paper is organized as follows. Section II outlines SRAM fault models, test algorithms and BIST design. Section III introduces the proposed BISR strategy. We present the details of the proposed BISR strategy including the architectures, procedures and the features. In section IV, the experimental results are reported. Finally, Section V concludes this paper.

II. FAULT MODELS , TEST ALGORITHMS AND BIST

A fault model is a systematic and precise representation of physical faults in a form suitable for simulation and test generation [11]. Applying the reduced functional model, SRAM faults can be classified as follows in [12]:

AF ---- Address Fault
ADOF ---- Address Decoder Open Faults
CF ---- Coupling Faults

- o CFin ---- Inversion Coupling Faults
- o CFid ---- Idempotent Coupling Faults
- o BF ---- Bridge Coupling Faults
- o CFst ---- State Coupling Faults

DRF ---- Data Retention Faults
SAF ---- Stuck-at Faults
SOF ---- Stuck Open Faults
TF ---- Transition Faults

The details of the fault models can be referred in. They are the foundations of the memory test. An Efficient and economical memory test should provide the best fault coverage in the shortest test time [13]. BIST is used to test memories in the paper and its precision is guaranteed by test algorithms. The algorithms in most common use are the March tests. March tests have the advantage of short test time but good fault coverage. There are many March tests such as March C, March C-, March C+, March 3 and so on. TABLE I compares the test length, complexity and fault coverage of them. 'n' stands for the capacity of SRAM.

TABLE 1. COMPARISION OF DIFFERENT MARCH TESTS

Algorithms	Test length	Complexity	Fault coverage
March C	11n	O(n)	AF,SAF,TF,CFin,CFid and CFst
March C-	10n	O(n)	AF,SAF,TF,CFin,CFid and CFst
March C+	14n	O(n)	AF,SAF,TF,SOF,CFin and CFid
March 3	10n	O(n)	AF,SAF,SOF, and TF

As shown in TABLE I, March C- has better fault coverage than March 3 and shorter test time than March C and March C+. So March C- has been chosen as BIST algorithm in this paper. Its algorithm steps are as follows:

1 up - write 0
2 up - read 0, write 1
3 up - read 1, write 0
4 down - read 0, write 1
5 down - read 1, write 0
6 down - read 0

In above steps, "up" represents executing SRAM addresses in ascending order while "down" in descending order.

The BIST module in the paper refers to the MBISR design of Mentor Graphics. It mainly consists of a BIST controller, a test vector generator, an address generator and a comparator. It can indicate when memory test is done and weather there is fault in memory.

A. *Redundancy Architecture*

The proposed SRAM BISR strategy is flexible. The SRAM users can decide whether to use it by setting a signal. So the redundancy of the SRAM is designed to be selectable.In another word, some normal words in SRAM can be selected as redundancy if the SRAM needs to repair itself. We call these words Normal-Redundant words to distinguish them from the real normal ones. We take a 64 × 4 SRAM for example, as shown in Figure 1. There are 60 normal words and 4 Normal-Redundant words. When the BISR is used, the Normal-Redundant words are accessed as normal ones. Otherwise, the Normal-Redundant words can only be accessed when there are faults in normal words. In this case, the SRAM can only offer capacity of 60 words to users. This should be referred in SRAM manual in details.

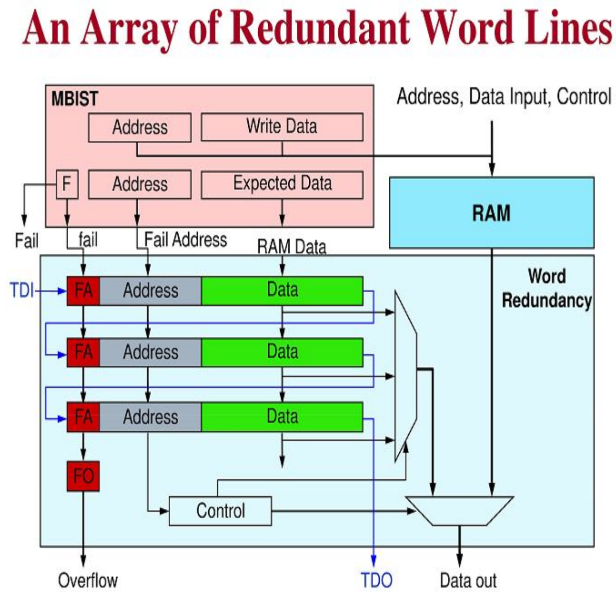


Figure 1. Architecture of redundancy in SRAM

This kind of selectable redundancy architecture can save area and increase efficiency. After BISR is applied, other modules in SRAM can remain unchanged. Thus the selectable redundancy won't bring any problem to SRAM compiler.

B. Proposed BISR Architecture

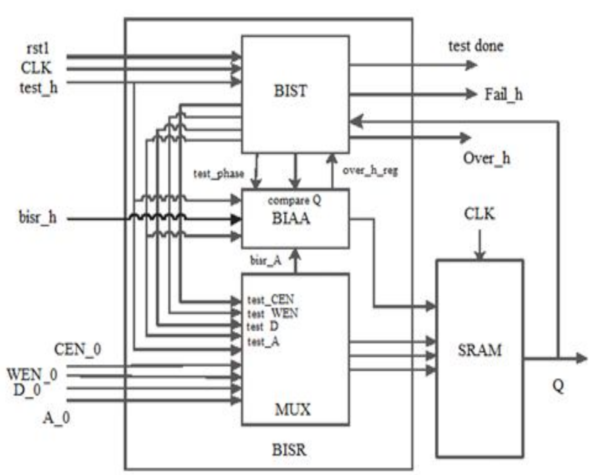


Figure2. Proposed BISR architecture

The architecture of the proposed BISR strategy is shown in Figure 2. It consists of three parts: BIST module, BIAA module and MUX module. We call the SRAM with BISR a system. The BIST module uses March C- to test the addresses of the normal words in SRAM. It detects SRAM failures with a comparator that compares actual memory data with expected data. If there is a failure (compare = 1), the current address is considered as a faulty address. The BIAA module can store faulty addresses in a memory named Fault_A_Mem. There is a counter in BIAA that counts the number of faulty addresses. When BISR is used (bisr_h = 1), the faulty addresses can be replaced with redundant addresses to repair the SRAM. The inputs of SRAM in different operation modes are controlled by the MUX module. In test mode (bist_h = 1), the inputs of SRAM are generated in BISR while they are equal to system inputs in access mode (bist_h = 0).

C. BISR procedure

Figure 3 shows the proposed BISR block diagram. The BISR starts by resetting the system (rst_l = 0). After that if the system work in test mode, it goes into TEST phase. During this phase, the BIST module and BIAA module work in parallel. The BIST use March C- to test the normal addresses of SRAM. As long as any fault is detected by the BIST module, the faulty address will be sent to the BIAA module. Then the BIAA module checks whether the faulty address has been already stored in Fault-A-Mem. If the faulty address has not been stored, the BIAA stores it and the faulty address counter adds 1. Otherwise, the faulty address can be ignored. When the test is completed, there will be two conditions. If there is no fault or there are too many faults that overflow the redundancy capacity, BISR goes into COMPLETE phase. If there are faults in SRAM but without overflows, the system goes into REPAIR&TEST phase. The same as during TEST phase, the BIST module and BIAA module work at the same time in REPAIR&TEST phase.

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

The BIAA module replaces the faulty addresses stored in Fault-A-Mem with redundant ones and the BIST module tests the SRAM again. There will be two results: repair fail or repair pass. By using the BISR, the users can pick out the SRAMs that can be repaired with redundancy or the ones with no fault.

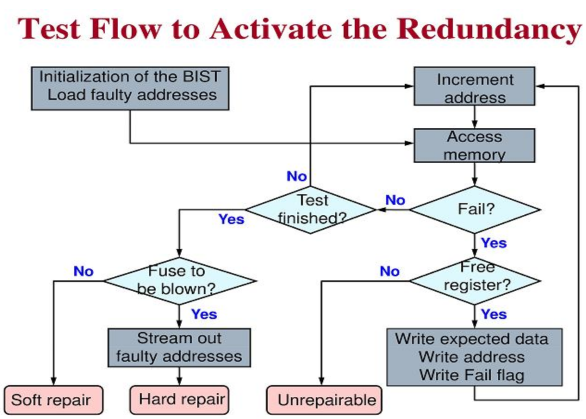


Figure3. Block Diagram of BISR

D. Features of the 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

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 add resses can be detected in more than one step. Take Stuck- 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.

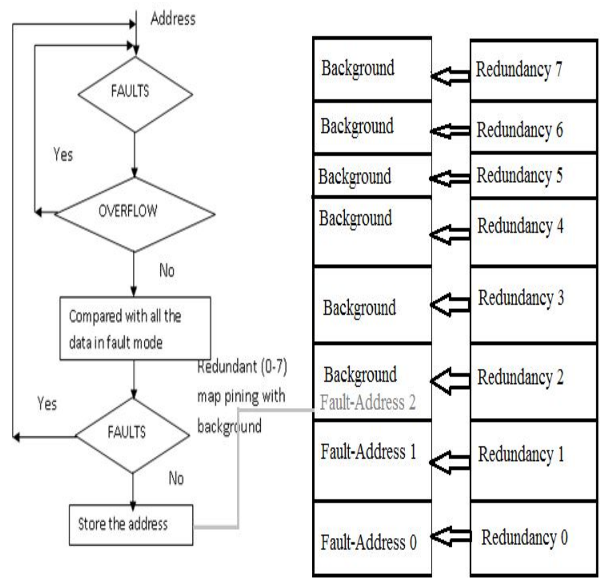


Figure 4. Flows of Storing Fault Addresses

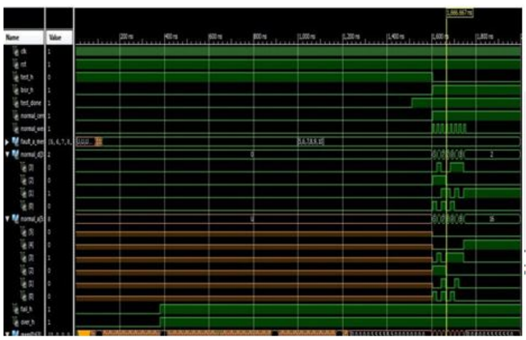
At last, the BISR strategy is high-speed. As shown in Figure 4, once a fault address is stored in Fault-A-Mem, it points to a certain redundant address. The fault addresses and redundant ones form a one-to-one mapping. Using this method, the BISR can quickly get the corresponding redundant address to replace the faulty one.

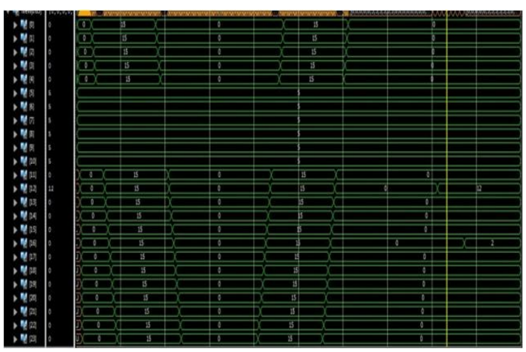
III. EXPERIMENTAL RESULTS

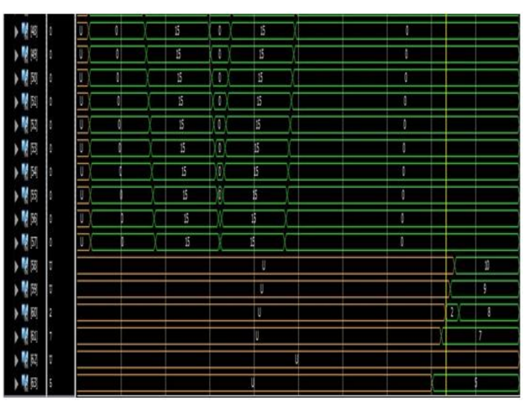
The proposed BISR was designed at RT level and it was synthesized to gate-level using Synopsys DC compiler. We use Cadence SOC Encounter to complete physical design of a 4Kx32 SRAM with BISR. The post simulation results show that the frequency of SRAM with BISR is at least 150MHz. The SRAM was implemented based on a 55nm CMOS process. The 32 addresses form H'FE0 to H'FFF were selected as Normal-Redundant addresses. To verify the function of BISR, a Stuck-at-0 fault was set in the SRAM. Figure 5 shows the layout view of the SRAM with BISR circuitry. BISR brings about 20% area penalties.

SIMULATION RESULTS:

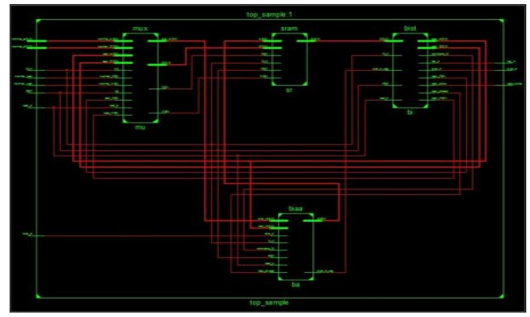
TOP MODULE







TOP MODULE WITH INTERNAL MODULES:



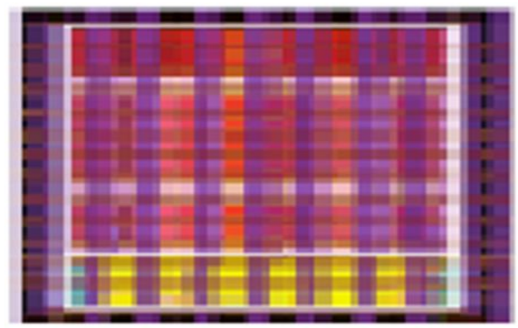


Fig: Layout view of SRAM with BISR

IV. CONCLUSIONS

An efficient BISR strategy for SRAM IP with selectable redundancy has been presented in this paper. It is designed flexible that users can select operation modes of SRAM. The BIAA module can avoid storing fault addresses more than once and can repair fault address quickly. The function of BISR has been verified by the post simulation. The BISR can work at up to 150MHz at the expense of 20% greater area.

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]. H. C. Kim, D. S. Yi, J. Y. Park, and C. H. Cho, "A BISR (built-in self- repair) circuit for embedded memory with multiple redundancies," VLSI and CAD 6th International Conference, pp. 602-605, Oct. 1999.

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

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

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

[9]. 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.

[10]. 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.

[11]. 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.

[12]. Mentor Graohics, MBISTArchitect Process Guide, Software Version 8.2009_3, Aug 2009, pp. 113-116.

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

ABOUT AUTHOR:

A.SHARONE MICHAEL, Graduated from G.Pulla Reddy Engineering College, Kurnool, in ECE Dept. Now pursuing Masters in VLSI System Design in Stanley Stephen College of Engineering & Technology, Kurnool,AP,INDIA.

Sri.K.SIVANNA, Associate Professor (ECE Dept) and for his constant co-operation, support and for providing necessary facilities throughout the M. Tech program. He has more than 10 Years of Experience, at B.Tech and M. Tech Level and Now he is working as a Associate Professor in Stanley Stephen College of Engineering & Technology, Kurnool,AP,INDIA.