Design and Implementation of BISR for 3d Multiple SRAMS with Redundancies in a SOC

K.Rukiya Begum[1], S.Lakshmi Kanth Reddy[2]

[1]*M.Tech(VLSI), Dept of ECE(VLSI), Global College of Engineering and Technology, Kadapa, Andhra Pradesh,*

[2]*Assistant Professor, Dept of ECE, Global College of Engineering and Technology, Kadapa, Andhra Pradesh*

Abstract- **Error correction code (ECC), built-in-self-repair (BISR) techniques by using redundancies has been widely used for improving the yield and reliability of embedded memories. The target faults of these two schemes are soft errors and permanent (hard) faults, respectively. In recent works, there are also some techniques integrating ECC and BISR to deal with soft errors and hard defects simultaneously. However, this will compromise reliability, since some of the ECC protection capability is used for repairing single hard faults. To cure this dilemma, I propose an ECC-enhanced BISR (EBISR) technique, which uses ECC to repair single permanent faults first and spares for the remaining faults in the production/power-ON test and repair stage (PTR). However, techniques are proposed to maintain the original reliability during the online test and repair stage. I also propose the corresponding hardware architecture for the EBISR scheme. A simulator is implemented to evaluate the hardware overhead (HO), repair rate, reliability, and performance penalty. Experimental results show that the proposed EBISR scheme can improve yield and reliability significantly with negligible HO and performance penalty.**

I.INTRODUCTION

Constructing an integrated circuit, or any semiconductor device, requires a series of operations—photolithography, etching, metal deposition, and so on. As the industry evolved, each of these operations were typically performed by specialized machines built by a variety of commercial companies. This specialization may potentially make it difficult for the industry to advance, since in many cases it does no good for one company to introduce a new product if the other needed steps are not available around the same time. A technology roadmap can help this by giving an idea when a certain capability will be needed. Then each supplier can target this date for their piece of the puzzle. With the progressive externalization of production tools to the suppliers of specialized equipment, the need arose for a clear roadmap to anticipate the evolution of the market and to plan and control the technological needs of IC production. For several years, the Semiconductor Industry Association (SIA) gave this responsibility of coordination to the United States, which led to the creation of an American style roadmap, the National Technology Roadmap for Semiconductors (NTRS).

In 1998, the SIA became closer to its European, Japanese, Korean, and Taiwanese counterparts by creating the first global roadmap: The International Technology Roadmap for Semiconductors (ITRS). This international group has 936 companies which were affiliated with working groups within the ITRS. The organization was divided into Technical Working Groups (TWGs) which eventually grew in number to 17,each focusing on a key element of the technology and associated supply chain. Traditionally, the ITRS roadmap was updated in even years, and completely revised in odd years.

The last revision of the ITRS Roadmap was published in 2013. The methodology and the physics behind the scaling results for 2013 tables is described in transistor roadmap projection using predictive full-band atomistic modeling which covers double gate MOSFETs over the 15 years to 2028.

With the generally acknowledged sunsetting of Moore's law and, ITRS issuing in 2016 its final roadmap, a new initiative for a more generalized roadmapping was started through IEEE's initiative Rebooting Computing, named the International Roadmap for Devices and Systems (IRDS).

LOGIC BUILT-IN SELF-TEST (OR) LBIST:LBIST is a form of built-in self-test (BIST) in which hardware and/or software is built into integrated circuits allowing them to test their own operation, as opposed to reliance on external automated test equipment.

Memory Bist(Built-In-Self-Test):MBIST, as its name implies, is used specifically for testing memories. It typically consists of test M circuits that apply, read, and compare test patterns capabilities .BIST can be used perform these special tests with additional on-chip test circuits, eliminating the need to acquire such high-end testers.

Memory testing is a more and more important issue because RAMs are key components for electronic systems and Memories represent about 30% of the semiconductor market, embedded memories are dominating the chip yield. Memory testing is more and more difficult due to Growing density, capacity, and speed, Emerging new architectures and technologies, embedded memories: access, diagnostics & repair, heterogeneity, custom design, power & noise, scheduling, compression, etc. Cost drives the need for more efficient test methodologies for fault modeling and simulation, test algorithm development and evaluation, diagnostics, DFT, BIST, BIRA, BISR, etc.

DESIGN FOR TESTING:DFT consists of IC design techniques that features to a hardware product design. The added features make it easier to develop and apply manufacturing tests to the designed hardware. The purpose of manufacturing tests is to validate that the product hardware contains no manufacturing defects that could adversely affect the product's correct functioning.

Tests are applied at several steps in the hardware manufacturing flow and, for certain products, may also be used for hardware maintenance in the customer's environment. The tests are generally driven by test programs that execute using automatic test equipment (ATE) or, in the case of system maintenance, inside the assembled system itself. In addition to finding and indicating the presence of defects (i.e., the test fails), tests may be able to log diagnostic information about the nature of the encountered test fails. The diagnostic information can be used to locate the source of the failure.

ERROR CORRECTION CODE (ECC):In information theory and coding theory with applications in computer science and telecommunication, error detection and correction or error control are techniques that enable reliable delivery of digital data over unreliable communication channels. Many communication channels are subject to channel noise, and thus errors may be introduced during transmission from the source to a receiver. Error detection techniques allow detecting such errors, while error correction enables reconstruction of the original data in many cases. error correction code, sometimes error correcting code, (ECC) is used for controlling errors in data over unreliable or noisy communication channels.

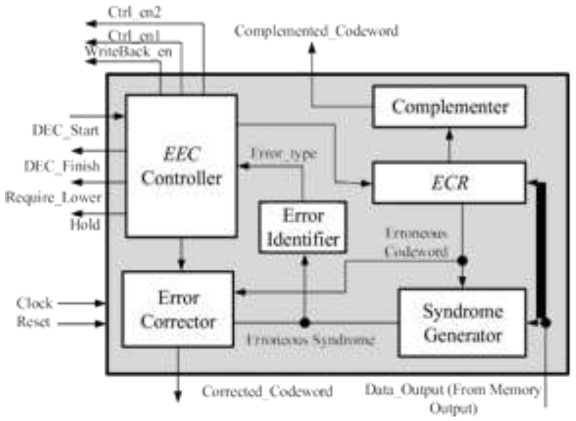


Fig. 4. Block diagram of the EEC.

II.PROPOSED SYSTEM

In computing, data recovery is a process of salvaging (retrieving) inaccessible, lost, corrupted, damaged or formatted data from secondary storage, removable media or files, when the data stored in them cannot be accessed in a normal way. The data is most often salvaged from storage media such as internal or external hard disk drives (HDDs), solid-state drives (SSDs), USB flash drives, magnetic tapes, CDs, DVDs, RAID subsystems, and other electronic devices. Recovery may be required due to physical damage to the storage devices or logical damage to the file system that prevents it from being mounted by the host operating system (OS).

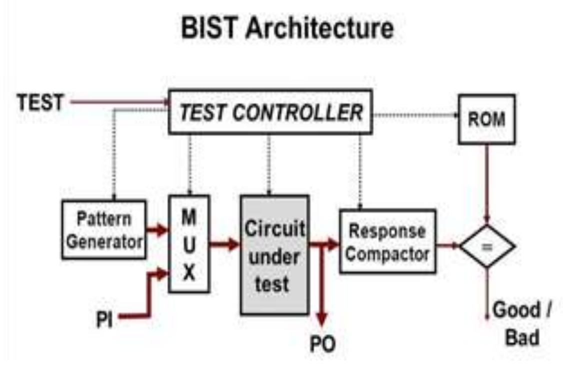
Recovery Techniques

Recovering data from physically damaged hardware can involve multiple techniques. Some damage can be repaired by replacing parts in the hard disk. This alone may make the disk usable, but there may still be logical damage. A specialized disk-imaging procedure is used to recover every readable bit from the surface. Once this image is acquired and saved on a reliable medium, the image can be safely analyzed for logical damage and will possibly allow much of the original file system to be reconstructed.

BUILT- IN SELF –TEST:

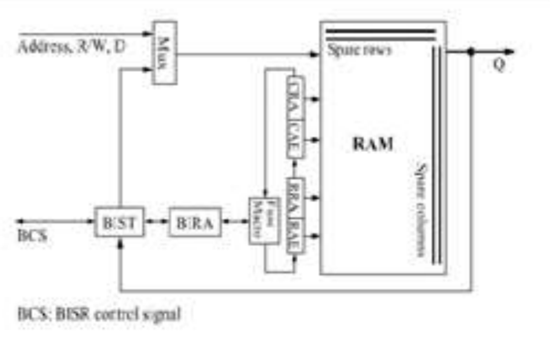
Logic built-in self-test (BIST) is a design for testability (DFT) technique in which a portion of a circuit on a chip, board, or system is used to test the digital logic circuit itself. Logic BIST is crucial for many applications, in particular for life critical and mission-critical applications. These applications commonly found in the aerospace/defense, automotive, banking, computer, healthcare, networking, and telecommunications industries require on-chip, on-board, or in-system self-test to improve the reliability of the entire system, as well as the ability to perform remote diagnosis.

With recent advances in semiconductor manufacturing technology, the production and usage of very-large-scale integration (VLSI) circuits has run into a variety of testing challenges during wafer probe, wafer sort, pre-ship screening, incoming test of chips and boards, test of assembled boards, system test, periodic maintenance, repair test, etc. Traditional test techniques that use automatic test pattern generation (ATPG) software to target single faults for digital circuit testing have become quite expensive and can no longer provide sufficiently high fault coverage for deep submicron or nanometer designs from the chip level to the board and system levels.



BUILT-IN SELF-REPAIR (BISR)

With the trend of SOC technology, high density and high capacity embedded memories are required for successful implementation of the system. In modern SOCs, embedded memories occupy the largest part of the chip area and include an even larger amount of active devices. As memories are designed very tightly to the limits of the technology, they are more prone to failures than logic. Thus, memories concentrate the large majority of defects. That is, RAMs have more serious problems of yield and reliability. Keeping the memory cores at a reasonable yield level is thus vital for SOC products. As a matter, Built-In Self-Repair is gaining importance. Built-in self-repair (BISR) technique has been widely used to repair embedded random access memories (RAMs). If each repairable RAM uses one self contained BISR circuit (Dedicated BISR scheme), then the area cost of BISR circuits in an SOC becomes high. This, results in converse effect in the yield of RAMs. This paper presents a reconfigurable BISR (ReBISR) scheme for repairing RAMs with different sizes and redundancy organizations. An efficient redundancy analysis algorithm is proposed to allocate redundancies of defective RAMs. In the ReBISR, a reconfigurable built-in redundancy analysis (ReBIRA) circuit is designed to perform the redundancy algorithm for various RAMs. The ReBISR structure has been synthesized and found that the area cost when compared with the Dedicated BISR structure is very small. This paper is implemented using Verilog HDL. Simulation and Synthesis is done using Model Sim and Xilinx ISE 12.4 Tools.



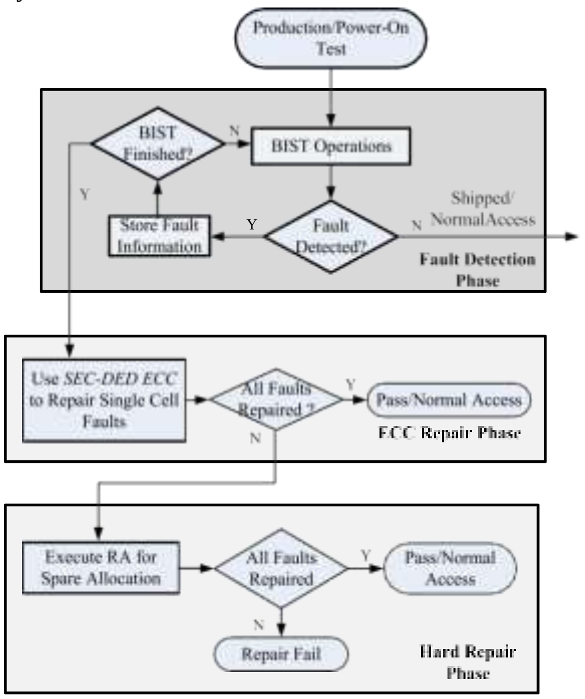
POWER ON SELF TEST

POST is a process performed by firmware or software routines immediately after a computer or other digital electronic device is powered on.

This article mainly deals with personal computers, but many other embedded systems such as those in major appliances, avionics, communications, or medical equipment also have self-test routines which are automatically invoked at power-on.

The results of the POST may be displayed on a panel that is part of the device, output to an external device, or stored for future retrieval by a diagnostic tool. Since a self-test might detect that the system's usual human-readable display is non-functional, an indicator lamp or a speaker may be provided to show error codes as a sequence of flashes or beeps. In addition to running tests, the POST process may also set the initial state of the device from firmware.

As part of the starting sequence the POST routines may display a prompt to the user for a key press to access built-in setup functions of the BIOS. This allows the user to set various options particular to the mother board before the operating system is loaded. If no key is pressed, the POST will proceed on to the boot sequence required to load the installed operating system.

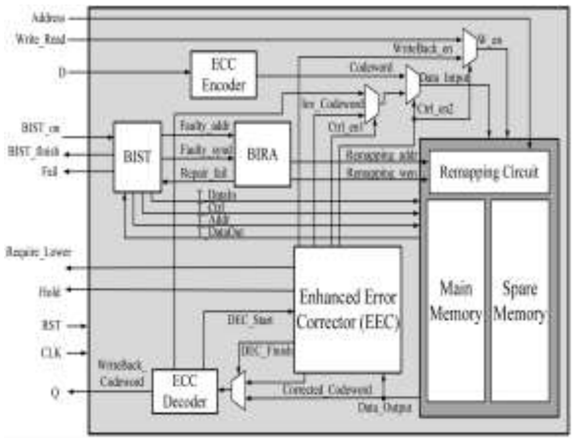


Test and repair flow of the PTR stage.

ENHANCED BUILT-IN SELF –REPAIR

It mainly consists of two stages—the roduction/power-ON test and repair (PTR) stage and the online test and repair (OLTR) stage. The PTR stage uses the SEC-DED code to repair single permanent faults first and spares for the remaining faults during production test and the power-ON test. The main target is the permanent faults. During the OLTR stage, the main targets are soft errors and unintentional permanent faults due to aging and process variations. The OLTR stage can detect the second fault or error by the incorporated SEC-DEDcode. The second fault/error can be corrected by the fault discrimination and correction (FDC) phase in this stage. Unlike the conventional techniques that integrate both ECC and hard repair, the protection capability of the SEC-DED code will not be compromised even it has been used to correct a permanent fault in a codeword. Therefore, the original reliability can be maintained when a second fault occurs.

The EBISR is based on the time redundancy concept. When a second faulty bit is detected , extra read and write operations are required. According to the first and the second readout codewords, our EBISR can discriminate the fault categories (permanent faults or soft errors) and, then, correct the second faults. Although the SEC-DED code can only correct an SCF in a codeword, however, EBISR can correct up to two faulty bits with almost negligible performance degradation and hardware overhead (HO). Therefore, the reliability will not be reduced even when the ECC is used to protect permanent SCFs.



Block diagram of the EBISR architecture

It mainly consists of the BIST module, the BIRA module, the ECC encoder/decoder, the enhanced error corrector (EEC), the memory array, the spare memory, the remapping circuit, and spare memory during the PTR stage when the BIST_ON signal is asserted. This signal is activated by either the automatic test equipment (ATE) in the production test and repair phase or the power-ON signal during the power-ON test and repair phase. When the BIST module detects any faults in the memory, the BIST operations will be suspended. The fault information are sent to the BIRA module through the Faulty_addr and the Faulty_synd signals. According to the received syndromes, SCFs are not corrected by spares and are subjected to the correction capability of the ECC during online operations.

III.SIMULATION RESULTS

TEST AND REPAIR FLOW OF EBISR

It mainly consists of the fault detection phase, the ECC repair phase, and the hard repair phase. In the fault detection phase, the BIST module executes the adopted March test algorithms to test the memory array. If no faults are detected, the memories can be shipped or used for normal access. If there are any faults detected, the BIST operations will be suspended, and the fault information including the fault addresses and the fault syndromes, should be stored.

The fault syndromes can be used to identify if the faulty words contain SCFs or multiple cell faults. After storing the fault information, the BIST operations are resumed to perform the we then enter the ECC repair phase. The SEC-DEC ECC code is used to repair all permanent SCFs. Since the SCFs occupy the largest proportion of all the possible fault types, correcting them by ECC can increase the efficiency of spare usage, and therefore, the fabrication yield and reliability can be raised significantly. We assume that during the very short BIST operation time, the probability of SEU attacks is very low.

Therefore, during the PTR stage, we do not have to consider the effects of soft errors, and the inherently incorporated ECC can be merely used for correcting SCFs.

We assume that the 8-bit codeword b7b6b5b4b3b2b1b0) is affected by two stuck-at-1 faults at bit positions b2 and b3. The correct codeword is “0000 0000,” as shown in the second row. Due to these two stuck-at faults, the readout erroneous codeword becomes “0000 1100,” as shown in the third row and can be detected by the SEC-DED ECC. However, these two faults are uncorrectable by merely using SEC-DED ECC.

Therefore, we complement the erroneous codeword and get the complemented codeword, as shown in the fourth row.We write back it into the memory and read the codeword again. Due to the effects of these two faults, the second readout codeword becomes “1111 1111.” We conduct an XOR (compare) operation for the first and the second readout Code words and get the syndrome, as shown in the seventh row. The second and third syndrome bits are both zero. We can conclude that b2 and b3contains stuck-at faults. Therefore, we can make a simple complement operation for b2 and b3 of the first readout codeword to get the correct codeword “0000 0000,” as shown in the eighth row. Therefore, these two stuck-at faults can be corrected.

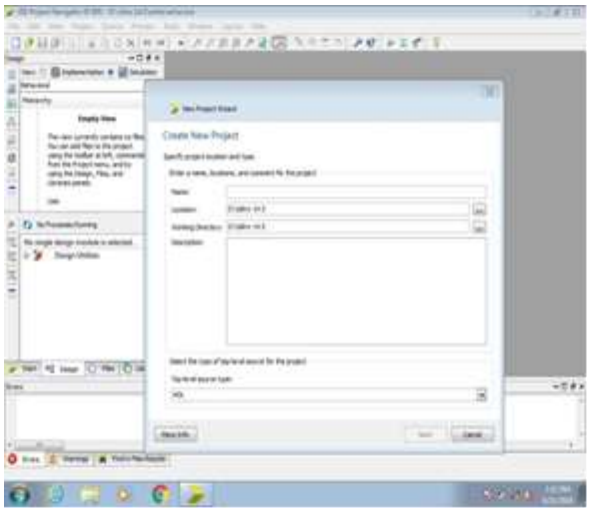
Evolution and Operation	Codeword
Correct Codeword	0000 0000
1st Read Out Erroneous Codeword	0000 1100
Complemented Erroneous Codeword	1111 0011
Write Back	1111 0011
2nd Read Out Codeword	1111 1111
Syndrome	1111 0011
Complement b_2 and b_3 of the Erroneous Codeword	0000 0000

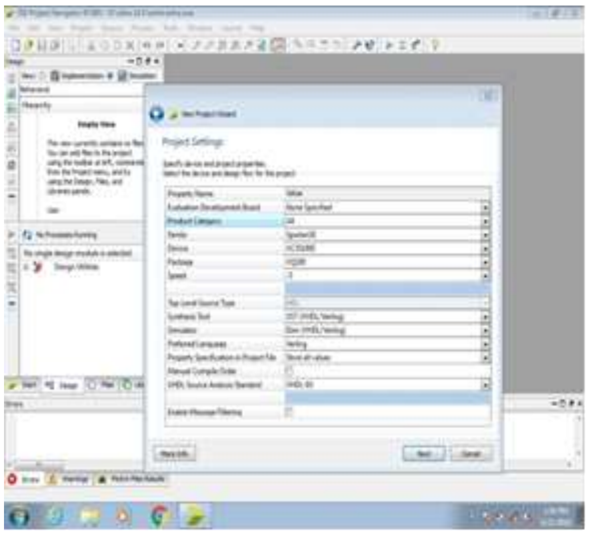
EXAMPLE CODEWORD CONTAINING TWO SOFT ERRORS

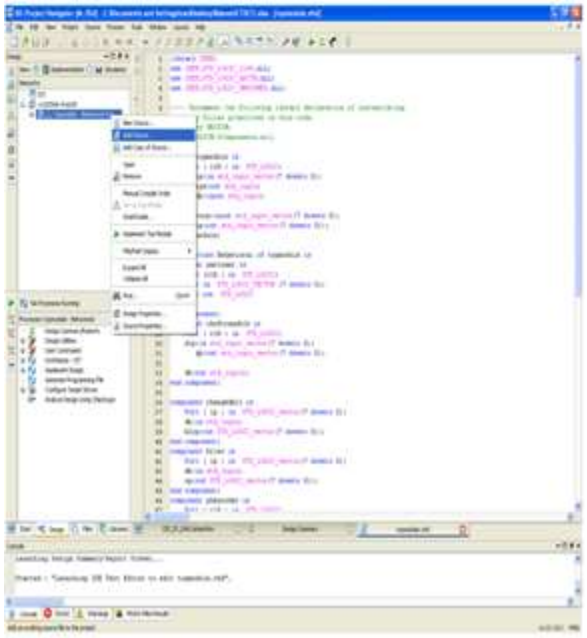
CREATING A NEW PROJECT

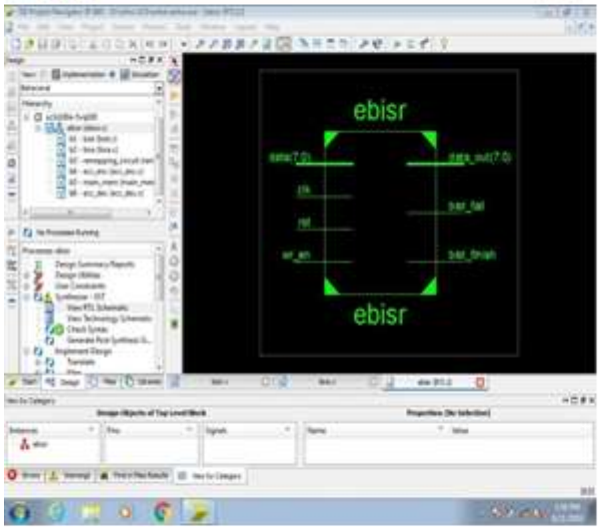
Xilinx Tools can be started by clicking on the Project Navigator Icon on the Windows desktop. This should open up the Project Navigator window on your screen. This window shows the last accessed project.

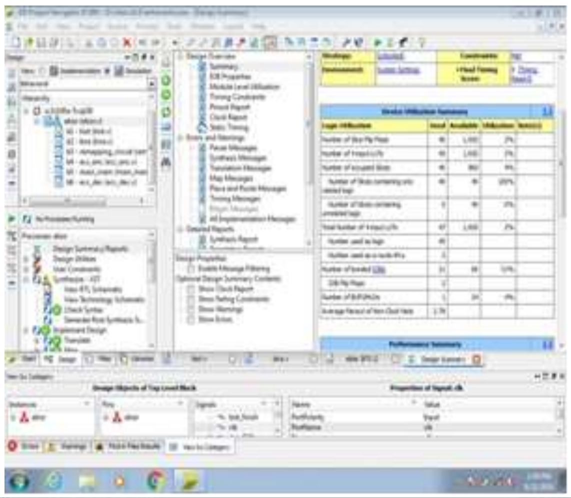
Opening a project Select File->New Project to create a new project. This will bring up a new project window on the desktop. Fill up the necessary entries as follows:

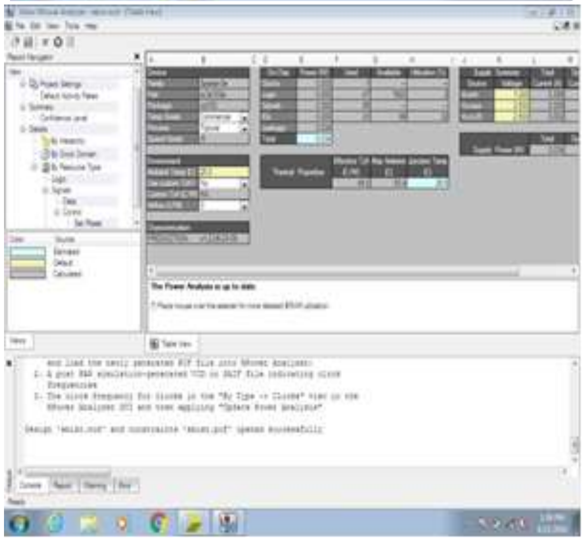












REFERENCES

[1] International Technology Roadmap for Semiconductors (ITRS),Semicond. Ind. Assoc., Washington, DC, USA, 2009.

[2] L.-T. Wang, C.-W. Wu, and X. Wen, VLSI Test Principles and Architectures: Design for Testability. San Francisco, CA, USA: MorganKaufmann, 2006.

[3] S. Hamdioui, G. Gaydadjiev, and A. J. van de Goor, “The state-ofartand future trends in testing embedded memories,” in Proc. IEEEInt. Workshop Memory Technol., Design Test. (MTDT), Aug. 2004, pp. 54–59.

[4] N. Derhacobian, V. A. Vardanian, and Y. Zorian, “Embedded memory reliability: The SER challenge,” in Proc. IEEE Int. Workshop

MemoryTechnol., Design Test. (MTDT), Aug. 2004, pp. 104–110.
[5] X. Du, S. M. Reddy, W.-T. Cheng, J. Rayhawk, and N. Mukherjee, “At-speed built-in self-repair analyzer for embedded word-oriented memories,” in Proc. IEEE Int. Conf. VLSI Design, Jan. 2004, pp. 895–900.
[6] H.-N. Liu, Y.-J. Huang, and J.-F. Li, “Memory built-in self test in multicore chips with mesh-based networks,” IEEE Micro, vol. 29, no. 5, pp. 46–55, Sep./Oct. 2009.
[7] L.-M. Denq, Y.-T. Hsing, and C.-W. Wu, “Hybrid BIST scheme for multiple heterogeneous embedded memories,” IEEE Des. Test Comput.,vol. 26, no. 2, pp. 64–73, Mar./Apr. 2009.
[8] C.-T. Huang, C.-F. Wu, J.-F. Li, and C.-W. Wu, “Built-in redundancy analysis for memory yield improvement,” IEEE Trans. Rel., vol. 52, no. 4, pp. 386–399, Dec. 2003.
[9] T. Kawagoe, J. Ohtani, M. Niiro, T. Ooishi, M. Hamada, and H. Hidaka, “A built-in self-repair analyzer (CRESTA) for embedded DRAMs,” in Proc. IEEE Int. Test Conf., Oct. 2000, pp. 567–574.
[10] V. Schober, S. Paul, and O. Picot, “Memory built-in self-repair using redundant words,” in Proc. IEEE Int. Test Conf. (ITC), Oct. 2001, pp. 995–1001.
[11] J. F. Li, J.-C. Yeh, R.-F. Huang, and C.-W. Wu, “A built-in self-repair scheme for semiconductor memories with 2-D redundancy,” in Proc. Int.Test Conf. (ITC), vol. 1. Oct. 2003, pp. 393–402.
[12] S.-K. Lu, C.-L. Yang, Y.-C. Hsiao, and C.-W. Wu, “Efficient BISR techniques for embedded memories considering cluster faults,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 184–193, Feb. 2010.
[13] S.-K. Lu, H.-H. Huang, J.-L. Huang, and P. Ning, “Synergistic reliability and yield enhancement techniques for embedded SRAMs,” IEEE Trans.Comput.-Aided Design Integr. Circuits Syst., vol. 32, no. 1, pp. 165–169,Jan. 2013.
[14] “Enhanced Built-In-Self-Repair Techniques for Improving Fabrication Yield and Reliability of Embedded Memories”, Shyue-Kung Lu,Member,IEEE,Cheng-Ju Tsai, and Masaki Hashizume,Member,IEEE Aug.2016.