ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of Optimized Reconfigurable Built-in Self-Repair Scheme for RAMs in SOCs

Manchikanti Divyasree
M.Tech (VLSI Design),
Dept ECE,
TKREC.

Nalika Aravind
Assistant Professor,
Dept ECE,
TKREC.

Dr.P.Ram Mohan Rao
FIE,CE(I),MISTE,MISH,MISCEE,MASCE(I),MISN,
Principal,
TKREC.

Abstract:

Input vector monitoring concurrent built-in self test (BIST) schemes perform testing during the normal operation of the circuit without imposing a need to set the circuit offline to perform the test. These schemes are evaluated based on the hardware overhead and the concurrent test latency (CTL), i.e., the time required for the test to complete, whereas the circuit operates normally. In this brief, we present a novel input vector monitoring concurrent BIST scheme, which is based on the idea of monitoring a set (called window) of vectors reaching the circuit inputs during normal operation, and the use of a static-

RAM like structure to store the relative locations of the vectors that reach the circuit inputs in the examined window; the proposed scheme is shown to perform significantly better than previously proposed schemes with respect to the hardware overhead and CTL tradeoff. As an extention,in the CUT we are using dadda multiplier(8*8).we are using Xilinx version to verify the output.the performance analysis of the dada multiplier is verified using BIST.

Keywords:

Built-in self-test (BIST), Circuit Under test (CUT), Response verifier (RV), testing.

I. INTRODUCTION:

With the advance of VLSI technology, the capacity and density of memories is rapidly growing. The yield improvement and testing issues have become the most critical challenges for memory manufacturing.

The BIST is used to detect and locate faulty cells of circuit under test. Input vector monitoring concurrent BIST techniques have been proposed to avoid this performance degradation.

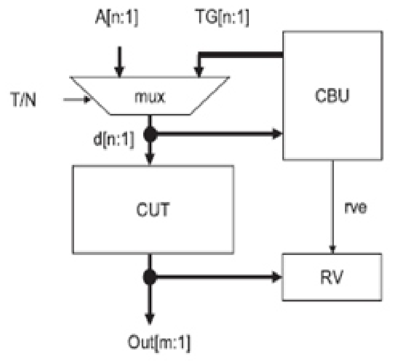
These architectures test the CUT concurrently with its normal operation by exploiting input vectors appearing to the inputs of the CUT; if the incoming vector belongs to a set called active test set, the

RV is enabled to capture the CUT response.The block diagram of an input vector monitoring concurrent BIST architecture. BIST utilizes a Test Pattern Generator (TPG) to generate the test patterns that are applied to the inputs of the Circuit Under Test (CUT). In off-line BIST, the normal operation of the CUT is stalled in order to perform the test.

Thus, if the CUT is of critical importance for the function of the circuit, the total circuit performance is degraded. To avoid such performance degradation, input vector monitoring concurrent BIST

techniques have been proposed, that exploit input vectors arriving at the inputs of the CUT during normal operation. Linear Feedback Shift Registers (LFSRs) have been by far the most popular devices for pseudorandom test pattern generation in BIST schemes .

They have the advantage of very low hardware overhead. However, for circuits with random pattern resistant faults, high fault coverage cannot be achieved within an acceptable test length.



Input vector monitoring concurrent BIST.

Fig 1: CBIST

A.CUT (Circuit Under Test):

The CUT has n inputs and m outputs and is tested exhaustively; hence, the test set size is N = 2n. Let us consider a combinational CUT with n input lines, as shown in Fig. 2; hence the possible input vectors for this CUT are 2n. The proposed scheme is based on the idea of monitoring a window of vectors, whose size is W, with W = 2w, where w is an integer number w < n. Every moment, the test vectors belonging to the window are monitored, and if a vector performs a hit, the RV is enabled.

B.CBU (Concurrent BIST Unit):

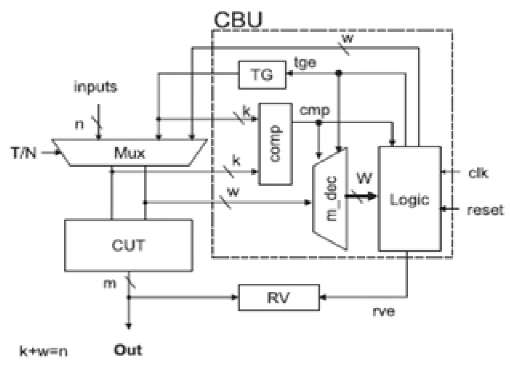
A typical BIST architecture that consists of Test Pattern Generator (TPG) usually implemented as a LFSR, Test Response Analyzer (TRA), Multiple Input Signature Register (MISR), CUT and BIST control unit. The approach uses the concept of reducing the transitions in the test pattern generated by conventional LFSR. The transition is reduced by increasing the correlation between the successive bits. The technique can operate in either normal or test mode, depending on the value of the signal labeled T/N.

During normal mode, the vector that drives the inputs of the CUT (denoted by d[n:1] in Fig. 1) is driven from the normal input vector (A[n:1]). A is also driven to a concurrent BIST unit (CBU), where it is compared with the active test set. If it is found that A matches one of the vectors in the active test set, we say that a hit has occurred.

In this case, A is removed from the active test set and the signal response verifier enable (rve) is issued, to enable the m-stage RV to capture the CUT response to the input vector. C-BIST has low hardware overhead but very high concurrent test latency, since in every clock cycle the input vector is compared against only one active test vector. To drive down the concurrent test latency, four techniques have been proposed so far, namely Multiple Hardware Signature Analysis Technique (MHSAT, [3]), Order Independent Signature Analysis Technique (OISAT, [4]), windowed-Comparative Concurrent BIST (w-CBIST, [5]) and RAM-based concurrent BIST (RCBIST [6]). These techniques accomplish to decrease the Concurrent Test Latency by increasing the number of active test vectors.

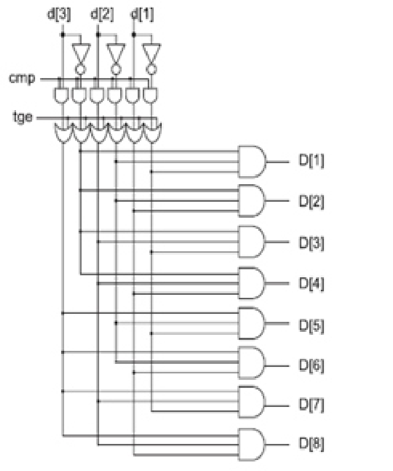
C. RV (Response Verifier):

When all input vectors have performed hit, the contents of RV are examined. During test mode, the inputs to the CUT are driven from the CBU outputs denoted TG[n:1]. The concurrent test latency (CTL) of an input vector monitoring scheme is the mean time (counted either in number of clock cycles or time units) required to complete the test while the CUT operates in normal mode. The Active test set Generator and Comparator of CBIST is a single Linear Feedback Shift Register (LFSR) and a comparator and thus the active test set consists of only one active test vector, which is the current value of the LFSR. During normal mode, the input vector is compared with the unique active test vector. If the two vectors match, the LFSR proceeds to the next state changing the active test vector and the Response Verifier is enabled.



Proposed architecture.

Fig 2: Proposed Architecture



Modified decoder design used in the proposed architecture.

Fig 3: Decoder Architecture

A.Architecture of Proposed Scheme:

The bits of the input vector are separated into two distinctsets comprising w and k bits, respectively, such that w + k = n. The k (high order) bits of the input vector show whether the input vector belongs to the window under consideration. The w remaining bits show the relative location of the incoming vector in the current window. If the incoming vector belongs to the current window and has not been received during the examination of the current window, we say that the vector has performed a hit and the RV is clocked to capture the CUT's response to the vector. When all vectors that belong to the current window have reached the CUT inputs, we proceed to examine the next window. The module implementing the idea is shown in Fig. 2. It operates in one out of two modes, normal, and test, depending on the value of the signal T/N. When T/N = 0 (normal mode) the inputs to the CUT are driven by the normal input vector. The inputs of the CUT are also driven to the CBU as follows: the k (high order) bits are driven to the inputs of a k-stage comparator; the other inputs of the comparator are driven by the outputs of a k-stage test generator TG. The proposed scheme uses a modified decoder (denoted as m_dec in Fig. 2) and a logic module based on a static-RAM (SRAM)-like cell, as will be explained shortly. The design of the m_dec module for w = 3 is shown in Fig. 3 and operates as follows. When test generator enable (tge) is enabled, all outputs of the decoder are equal to one. When comparatot (cmp) is disabled (and tge is not enabled) all outputs are disabled. When tge is disabled and cmp is enabled, the module operates as a normal decoding structure.

II OVERVIEW OF A BISR SCHEME WITH BIRA:

The BISR circuit consists of a BIST circuit, a BIRA, and reconfiguration mechanism. The BIST can generate testpatterns to test the RAM and locate the faulty cells. The reconfiguration mechanism can swap the defective elements with the spare elements, which is usually realized by multiplexers and storage elements. The control signals of the multiplexers are stored in registers. The registers include row repair address (RRA), column repair address (CRA), row address enable (RAE), and column address enable (CAE) registers. The fuse macro typically consists of nonvolatile storage cells that are used to store the repaired addresses when the power is turned OFF.

The BIRA can optimize the allocation of 2-D redundancy for replacing the faulty cells in a defective RAM. Typically, a BIRA uses a local bitmap to collect fault information and then allocates redundancy using the implemented RA rules according to the corrected faults. Fig. 2(a) shows an example of a 4 × 4-bit local bitmap for an 8 × 4 × 4-bit RAM (i.e., the RAM has 3-bit row address, 2-bit column address, and 4-bit data width) with five faulty cells shown in Fig. 2(b), where RAR, CAR, and BAR denote the row address register, column address register, and bit address register, respectively. The bit address is used to identify the location of faulty bit in a faulty word. A local bitmap can be considered as a compressed device image memory. Hereafter, the local bitmap is called bitmap for short. During the testing of the RAM, the fault information of the detected faulty cells are stored in the bitmapon the fly. Once the fourth faulty cell is detected and stored, the bitmap is full, i.e., either the CAR or RAR has no space. If the bitmap is full, the BIST is halted and the BIRA performs redundancy allocation. This process is iterated until the test and repair process is completed.

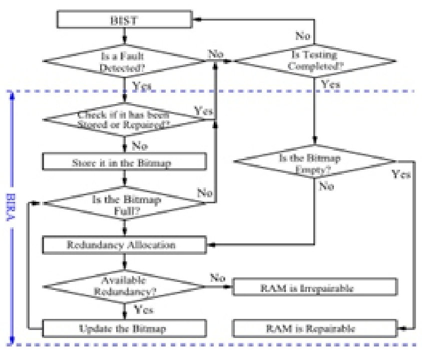


Fig. 3. Operations between the BIST and BIRA.

Fig. 3 shows a operation flow between the BIST and BIRA. The BIST generates test patterns for the RAM. When a fault in the RAM is detected, the information of the fault is sent the BIRA. The BIRA first checks if the fault has been stored in the bitmap or repaired. If not, the fault is stored in the bitmap. Then, the BIRA checks if the bitmap is full or not. A bitmap is called full if either its CAR or RAR has no space for storing a fault. If yes, the BIRA allocates the redundancies according to the stored fault information and the implemented RA algorithm. If the test process is completed, the BIRA checks if the bitmap is empty or not. A bitmap is called empty if no fault information of faulty cells is stored in the bitmap. If not, the BIRA performs the redundancy allocation process to repair the remained faults. Once the BIRA process is completed and the RAM is repairable, the repaired addresses are stored in the fuse macro. Fig. 4 shows a simplified block diagram of a BIRA with k × l-bit bitmap, where some functional blocks are not shown. Some of interaction signals between the BIST and BIRA are shown in the figure, which are faulty address (FA), syndrome (SYN), and repair (REP). Once the BIST detects a fault, the corresponding FA and Hamming syndrome are sent to the BIRA through the FA and SYN, where Hamming syndrome is defined as the modulo-2 sum of the fault-free data output vector and the output vector from the RAM under test [32]. The signal REP is used to indicate the RA result of the BIRA is repairable or unrepairable.

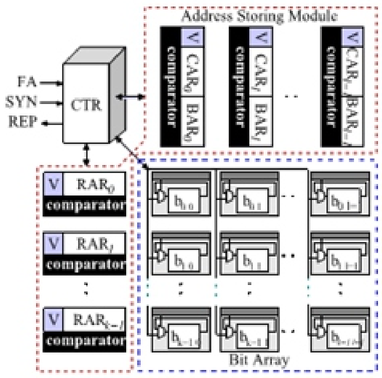


Fig. 4. Simplified block diagram of a BIRA with 3 × 3 bitmap.

III. PROPOSED HRE-BISR SCHEME:

A.Architecture of HRE-BISR Scheme:

As aforementioned, the BIRA of a BISR circuit for a RAM with 2-D redundancy typically has a bitmap for storing fault information during the process of redundancy allocation.

The bitmap is a cache-like element which has the parallel comparison and storing functions. Since the bitmap is only used in test mode for RA, we can modify it into a spare memory in normal mode. Fig. 5 shows the block diagram of the proposed HRE-BISR scheme for RAMs by reusing the bitmap as a spare memory. Here, we assume that the RAM has 2-D redundancy, e.g., spare rows and columns, and the reconfiguration of redundancies is done by shift redundancy technique [9]. The BIRA consists of a multiple-fault-bit detector (MFBD), a finite state machine (FSM), a repaired fault checker (RFC), a repair signature register (RSR), and a modified bitmap. The FSM realizes the RA algorithm. The MFBD is used to check if the number of faulty bits of a detected faulty word are larger than 2. The RSR is used to store the repaired addresses during the process of RA. Once the BIST detects a fault, the RFC checks if the detected fault has been stored in the RSR. If yes, the detected fault has been repaired. The modified bitmap is used to collect fault information in test mode and serves as spare bits in normal mode.

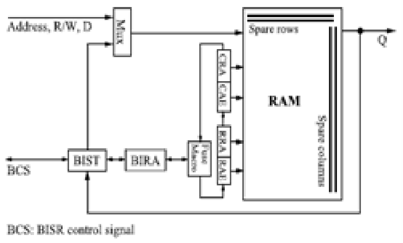


Fig. 5 Typical BISR scheme for a repairable RAM.

Since the size of bitmap is typically much smaller than the size of a word of RAMs, the bitmap is used as a spare-bit memory. To reuse a bitmap as a spare memory, the bitmap should be modified such that it can support the read/write operation in normal mode. Fig. 6(a) shows the simplified block diagram of a modified BIRA with a k × l-bit modified bitmap. A mode setting signal MODE is used to set the BIRA to be operated in normal or test mode. Normal IOs of the RAM are connected to the BIRA as well. In test mode, the operation of the modified bitmap is the same as that of the original bitmap. In normal mode, the operation of the modified bitmap is described as follows. If the RAM is accessed, the applied address is compared with the stored addresses including the row and column addresses. If one of stored row addresses is the same as the row address of the input address, then the corresponding row address hit (RAH) signal is set to 1, i.e., RAHi = 1.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Similarly, if one of stored column addresses is the same as the column address of the input address, then the corresponding column address hit (CAH) signal is set to 1, i.e., CAH j = 1. Therefore, when the accessed address is matched with one of the stored addresses, i.e., RAHi = 1 and CAH j = 1, one bit of the addressed word is repaired by the spare bit.To support the read/write operation, each storage element of the bit array is modified, as shown in Fig. 6(b). If a write operation is executed, i.e., R/W = 1, the repair data input din is written into the cell b ij Otherwise, the data stored in the cell bij is read out while R/W = 0. Furthermore, a repair bit selection (RBS) module is added to select the repair bit from the data input D [m – 1 : 0] for the data input din of bit array and generate the control signal C [m – 1 : 0] determining which bit of data output is replaced by the data output of the bit array qr. Fig. 6(c) shows the simplified circuit diagram of the RBS module. In test mode (i.e., Test = 1), the din is set to logic 1 such that the hit bit can be set to logic 1. In addition, the C [m – 1 : 0] is set to all-0 state, which forces Q [m – 1 : 0] = q [m – 1 : 0], as shown in Fig. 5. In normal mode (i.e., Test = 0), once the input address is matched with one of the stored addresses, RAHi and CAH j , the hit signal is asserted and the decoder is enabled. Simultaneously, the value of the corresponding BAR BAR j is exported to the decoder. Then, the decoder decodes the BAR j into m-bit control signal C [m – 1 : 0] for controlling the data input and data output multiplexers.

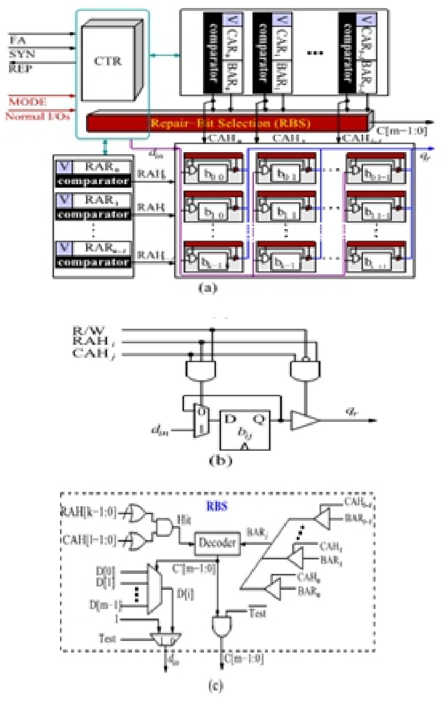


Fig. 6. (a) Simplified block diagram of the modified BIRA with k × l-bit modified bitmap. (b) Modified cell structure. (c) Simplified circuit diagram of RBS.

To reduce the hardware complexity, only one bit addressstored in the BARs is selected, as shown in Fig. 6(c). Therefore, the bitmap cannot repair multiple-faulty-bit words. If the bitmap stores a multiple-faulty-bit word, for example, then there are two CARs that store the same column address and the corresponding BARs that store different bit addresses. Thus, if the column address stored in the CARs is accessed, the two different bit addresses are exported to the bus, which results in bus congestion. One straightforward approach to support the repair of multiple-faulty-bit words is to design an individual decoder for each pair of CAHi and BARi . Then, the result of bitwise OR operation of the outputs of all decoders is the C [m – 1 : 0]. However, the area cost is drastically increased.Consequently, we modify the bitmap to support the repair of single-faulty-bit word only here.

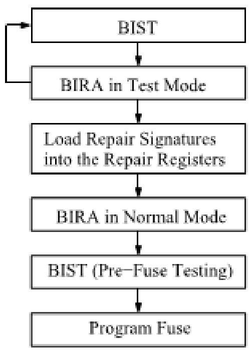


Fig. 7. Test and redundancy allocation flow

B.Test and Repair Flow:

In test mode, the test and redundancy allocation flow is as shown in Fig. 7. First, the BIST tests the RAM under repair and the BIRA is in test mode. If a fault is detected, the fault information is sent to the BIRA for analysis. This process is iterated until the testing of the RAM is completed. If the RAM is repairable, then the repair signatures are loaded into the repair registers and the BIRA are set to normal mode. Subsequently, the BIST is used to test the repaired RAM again for the prefuse testing. If the prefuse testing is completed and the repaired RAM is fault-free, then the fuses in the fuse macro can be programmed and the test and repair process is completed.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Here, we assume that the electrically programmed fuse (e-fuse) is used to realize the fuses in the fuse macro [34]. In addition, the fuse macro has a fuse CTR to handle the programming of e-fuse. The e-fuse enables the function of on-chip self-repair, whereby repair data is programmed into e-fuse when the test and repair process is completed. Therefore, if the BIST and the fuse CTR are asserted by the power on reset signal, on-chip self-repair can be achieved [10].

Fig. 8 shows the repair flow of the proposed HRE-BISR scheme in normal mode. The repair flow consists of two phases: the repair setup and the normal access phases. In the repair setup phase, once the power is turned on, the repair signatures stored in fuse macro are loaded into the repair registers and the RAR, CAR, and BARs of the bitmap. In normal access phase, the applied address is imported to the repairable RAM and bitmap simultaneously. If the applied address is not matched with the addresses stored in the bitmap, then the repairable RAM is accessed only. Otherwise, both the repairable RAM and the bitmap are accessed. Then, the faulty bit of accessed word in the repairable RAM is replaced by the hit spare bit of bitmap.

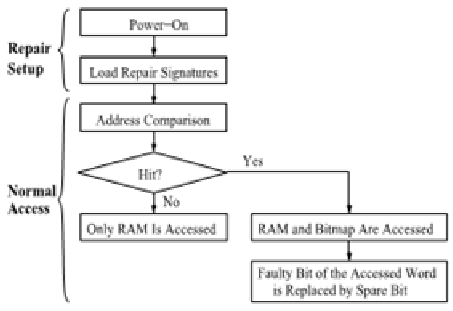


Fig. 8. Repair flow diagram.

SIMULATION RESULTS:

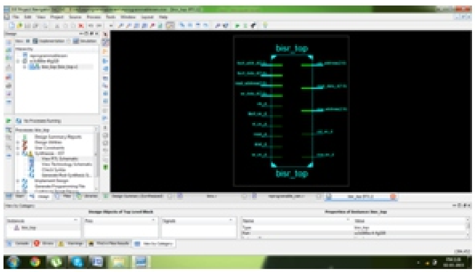


Fig: RTL SCHAMATIC:

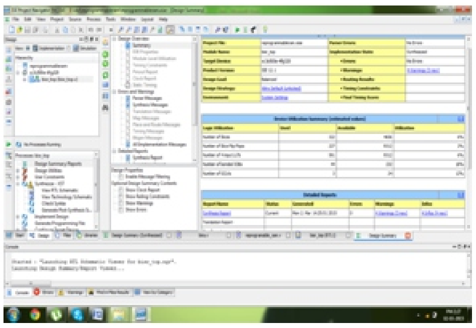


Fig: AREA CALCULATION

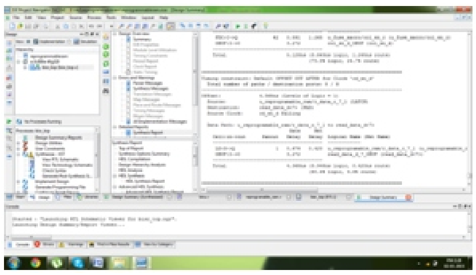


Fig: DELAY CALCULATION

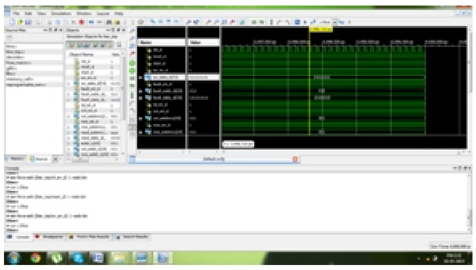


FIG: WAVEFORMS

CONCLUSION:

In this brief, a novel input vector monitoring concurrent BIST scheme is proposed, which compares favorably to previously proposed schemes with respect to the hardware overhead/CTL tradeoff. BIST schemes constitute an attractive solution to the problem of testing VLSI devices. Input vector monitoring concurrent BIST schemes perform testing during the circuit normal operation without imposing a need to set the circuit offline to perform the test, therefore they can circumvent problems appearing in offline BIST techniques. The evaluation criteria for this class of schemes are the hardware overhead and the CTL, i.e., the time required for the test to complete, while the circuit operates normally.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

In this brief, a novel input vector monitoring concurrent BIST architecture has been presented, based on the use of a SRAM-cell like structure for storing the information of whether an input vector has appeared or not during normal operation. The proposed scheme is shown to be more efficient than previously proposed input vector monitoring concurrent BIST techniques in terms of hardware overhead and CTL.

Future Scope:

In this paper we proposed the selfrepairing for RAM only In future we can implement the BISR for the whole circuit.

References:

[1] M. Lin, Ed., "1997 Semiconductor Industry Annual Report," (in Chinese), Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan, ITIS project report, 1997.

[2] I. Voyiatzis, A. Paschalis, D. Gizopoulos, N. Kranitis, and C. Halatsis, "A concurrent BIST architecture based on a selftesting RAM," IEEE Trans. Rel., vol. 54, no. 1, pp. 69–78, Mar. 2005.

[3] I. Voyiatzis and C. Halatsis, "A low-cost concurrent BIST scheme for increased dependability," IEEE Trans. Dependable Secure Comput., vol. 2, no. 2, pp. 150–156, Apr. 2005.

[4] K. K. Saluja, R. Sharma, and C. R. Kime, "A concurrent testing technique for digital circuits," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 7, no. 12, pp. 1250–1260, Dec. 1988.

[5] R. Sharma and K. K. Saluja, "Theory, analysis and implementation of an on-line BIST technique," VLSI Design, vol. 1, no. 1, pp. 9–22, 1993.

[6] K. K. Saluja, R. Sharma, and C. R. Kime, "Concurrent comparative built-in testing of digital circuits," Dept. Electr. Comput. Eng., Univ. Wisconsin, Madison, WI, USA, Tech. Rep. ECE-8711, 1986.

[7] I. Voyiatzis, T. Haniotakis, C. Efstathiou, and H. Antonopoulou, "A concurrent BIST architecture based on monitoring square windows," in Proc. 5th Int. Conf. DTIS, Mar. 2010, pp. 1–6.

[7] I. Voyiatzis, T. Haniotakis, C. Efstathiou, and H. Antonopoulou, "A concurrent BIST architecture based on monitoring square windows," in Proc. 5th Int. Conf. DTIS, Mar. 2010, pp. 1–6.

[8] M. A. Kochte, C. Zoellin, and H.-J. Wunderlich, "Concurrent self-test with partially specified patterns for low test latency and overhead," in Proc. 14th Eur. Test Symp., May 2009, pp. 53–58.

[9] S. Almukhaizim and Y. Makris, "Concurrent error detection methods for asynchronous burst mode machines," IEEE Trans. Comput., vol. 56, no. 6, pp. 785–798, Jun. 2007.

[10] S. Almukhaizim, P. Drineas, and Y. Makris, "Entropy-driven parity tree selection for low-cost concurrent error detection," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 25, no. 8, pp. 1547–1554, Aug. 2006

[11] 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.

[12] J.-F. Li, J.-C. Yeh, R.-F. Huang, and C.-W. Wu, "A built-in self-repair design for RAMs with 2-D redundancies," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 6, pp. 742–745, Jun. 2005.

[13] C.-D. Huang, J.-F. Li, and T.-W. Tseng, "ProTaR: An infrastructure IP for repairing RAMs in SOCs," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 15, no. 10, pp. 1135–1143, Oct. 2007.

[14] T.-W. Tseng and J.-F. Li, "A shared parallel built-in self-repair scheme for random access memories in SOCs," in Proc. Int. Test Conf. (ITC), Santa Clara, CA, USA, Oct. 2008, pp. 1–9.

[15] 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.

[16] L. R. Huang, J. Y. Jou, and S. Y. Kuo, "Gauss-elimination based generation of multiple seed-polynomial pairs for LFSR," IEEE Trans. Comput. Aided Design Integr. Circuits Syst., vol. 16, no. 9, pp. 1015–1024, Sep. 1997.