Middle-East Journal of Scientific Research
24 (Special Issue on Innovations in Information, Embedded and Communication Systems): 68-74, 2016
ISSN 1990-9233; © IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.IIECS.23142

A Novel Vectorized Built-In Self-Repair and Diagnosis for Bist (Built in Self Test)

[1]N. Ashok Kumar and [2]A. Kavitha

[1]Faculty of Electronics and Communication Engineering,
RVS College of Engineering, Dindigul-624005. TamilNadu, India
[2]Faculty of Electronics and Communication Engineering,
Chettinad College of Engineering, Karur-639114. TamilNadu, India

Abstract: Vectorized BIST schemes are emerging trends for generation of test sequence under normal and test mode operation of test circuit. Hardware Over Head(HOH) and Concurrent Test Latency(CTL) evolution of existing method having major drawback because of HOH and CTL are high. To overcome this and improving the performance and reducing the power we introduce new novel approach which is based on specific considered level of input vector sequences are applied to test circuit or monitoring window for Built-In self Repair under the level of normal operation. In this proposed novelty provides an efficient for fail Pattern identification and BISD(Built-In Self Diagnosis)to incorporate fault syndrome compression scheme.

Key words: Built In Self Test(BIST) · Circuit Under Test(CUT) · Response Verifier(RV) · Built-in Self Diagnosis(BISD)

INTRODUCTION

Built –in self test (BIST) is an emerging trends of design for testing technique [1]. The test parts of a circuit are used under the selftest. Built in self test (BIST) is technology to test circuit in which capable to diagnosis themselves. Built In self test (BIST) compose a class of schemes [2], on condition that potential for performing at speed testing with high fault coverage and concurrently relax the reliance on pricey external testing equipment [1]. Hence, they compose an gorgeous solution to the problem of testing VLSI devices. The basic BIST architecture [3] requires the addition of three hardware blocks to a digital circuit: a test pattern generator (tpe), a response Verifier or Response Analyser and a test controller. The test pattern generator generates the test patterns for the Circuit Under Test (CUT) [4]. In such that the pattern generators are the memory devices that are stored in patterns memory, a counter and a linear feedback shift register (LFSR) [5]. A typical response verifier is a comparator with stored responses or an LFSR [5, 6] used as a signature analyser. It compacts and analyses the test responses to determine correctness of the CUT. A control block(TCB) of test control is necessary for activate the test and examine the responses; several test-related functions should be executed during a test controller circuit.

BIST techniques are typically off the record into line of on and off. Basically Offline technique to operate in which ever normal mode or test mode. For the duration of test mode [7], test generator module outputs are applied as inputs of the circuit under test (CUT) and a response verifier (RV) capture corresponding output response [1]. Whenever under normal mode BIST is idle, therefore to perform the test, the normal operation of the CUT is stalled and, accordingly, the performance of the system in which the circuit is included, is degraded. Considered level of vectorized monitoring concomitant BIST techniques having to avoid this performance degradation. Circuit Under Test (CUT) in concert with its normal operation by exploiting vectoring input appearing to the CUT; if the arriving vector belongs to a set called lively test set, the RV is enabled to capture the CUT response [3].

Existing Bist Architecture: The existing BIST architecture is shown in Fig. 1. It consists of a multiplexer, CUT (Circuit Under Test), Concurrent Bist Unit(CBU) and a RV or RA(Response Verifier or Response Analyser) [8].

Corresponding Author: N. Ashok Kumar, Faculty of Electronics and Communication Engineering, RVS College of Engineering, Dindigul-624005. TamilNadu, India.

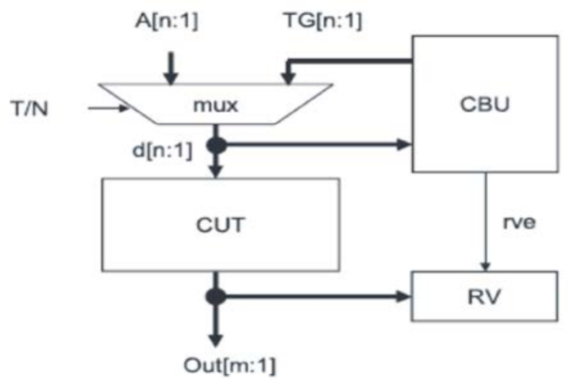


Fig. 1: Existing BIST architecture

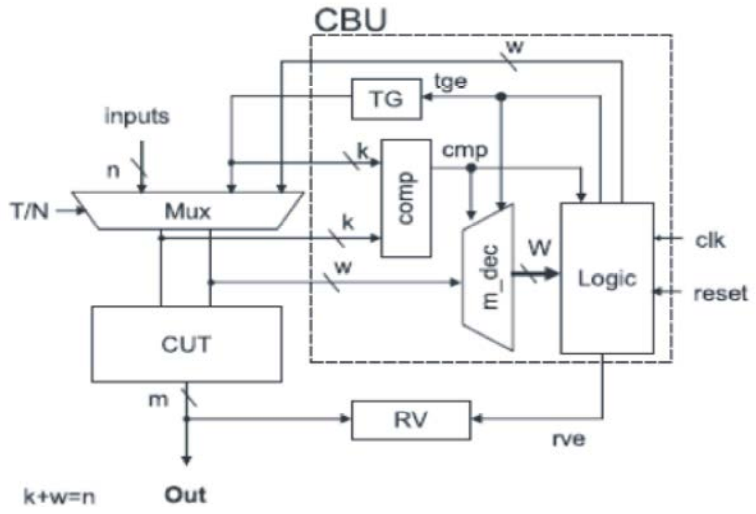


Fig. 2: Proposed BIST architecture

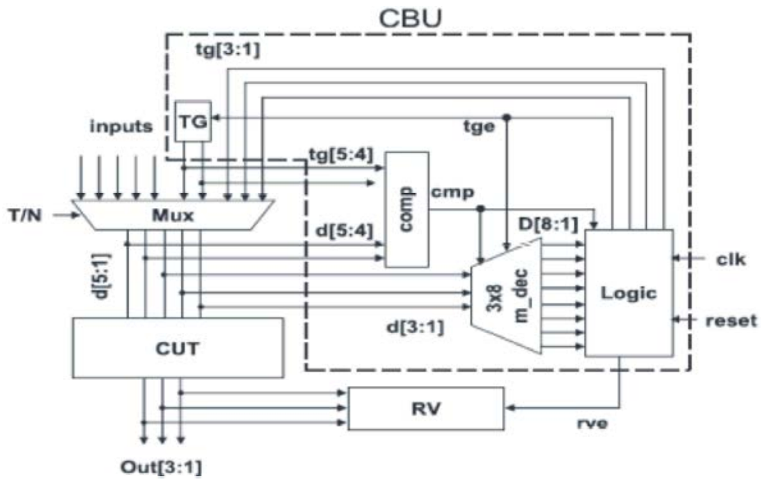


Fig. 3: Overall BIST architecture

The circuit can operate in both test mode and normal mode. Input vector is given to both the CUT and CUB. Then the output of the CUT and CUB is stored in the response verifier [5]. The response verifier compares both the outputs and returns the value whether the CUT is working correctly or not.

Proposed Bist Architecture: This scheme is based on window monitoring vectors. The circuit have n number of inputs. Then the number of inputs is splitted into *w* and *k* number of bits where $w + k = n$. Here *w* represents the lower order bits and *k* represents the higher order bits. The window size is *W*, with W, where w is an integer number $w < n$. The comparator compares the k bits form the output of the TG and the output of the MUX and returns the value 1 if both are equal otherwise 0.

The decoder decodes the w bits and sends to the logic module. The output of the MUX is also sent to the CUT. The CUT perform its operation and the result is stored in the response verifier (RV) and the value of rv from the logic module is also stored in the RV. The response verifier compares both values and return the output. If both the values are same the circuit is working properly otherwise not.

Let us consider a combinational CUT with n=5 input lines, hence the possible input vectors are 2n. The proposed scheme is under the level of monitoring vectors, whose size is *W*, with $W = 2w$, where *w* represents an integer number $w < n$. The test vectors belonging to the window are monitored at regular intervals and if the vector performs a hit [9], the response verifier is enabled. The vectorized input bits are wrecked into two distinct sets comprising w and k bits, correspondingly, such that w + k = n. The k bits belongs to the window under consideration of input vector and the remaining w bits confirm the relative location of the arriving input vector in the existing window. During the examination of existing window. The incoming vector belongs to the existing window and has not been received for assessment then the vector has performed a hit and the RV is clocked to capture the CUT's response [8]. All the input vectors are reached the existing window under the test the next window will be examine.

Test Pattern Generator: The purpose of using test pattern generator is to generate test pattern for the BIST circuit [10]. The TPG is designed by using flip flops and mod-2 adder. For m number of inputs the TPG generates 2 □ -1number of bit sequence for a test pattern. It requires m number of flip flops to generate the test pattern. Consider that the number of input bits m=3. Three flip flops are used to generate the test pattern of 7 bit sequence. The output of FF-A and FF-B are added using a mod-2 adder and given as input to the FF-A [9].

Condition for Test Pattern: Two conditions should be satisfied to check whether the bit sequence generated by the TPG is a test pattern or not? i) the number of 1's should be greater than the number of 0's in the test pattern sequence. ii) the difference between the number of 1's and number of 0's should be 1. For example consider the bit sequence 1011010. Here the number of 1's is 4 and 0's is 3. So, the number of 1's is greater than the number of 0's and also the difference between the number of 1's and number of 0's can be calculated as 4-3=1. 4.3.1 Therefore the two conditions can be satisfied by the above bit sequence. So, 1011010 can be considered as a test pattern.

Decoder Design: The design of the m dec module for $w = 3$ is shown in Fig. For $w = 3$, 3 to 8 decoder is used to generate 8 bit output. The operation of the decoder is explained as follows. When test generator enable (tge) is enabled, all outputs of the decoder are equal to one. When comparator (cmp) is disabled (and tge is not enabled) all outputs are disabled. When tge is disabled and comparator is enabled, the module operates as a normal decoding structure [9].

Logic Module Design: The logic module comprises of w cells, sense amplifier (SA), two D flip flops and a w-stage counters [9]. The output of the decoder is given as input to the logic module. Initially the reset and clock values are set as 1. It receives the data from 3:8 decoder and delivers the enable signal for TG (Test Pattern Generator).

Test Vector Compression

Binary Matrix Lossless Compression Scheme: Mass level of test data is required for circuit under test(CUT) and it is more level of complication in integration technology of VLSI chip and functionality and compatibility [6]. Testing process has the challenging process for manufacturers. Data volume and Test power are the main parameters for analysis of the BIST complaints. In our proposed level , a binary lossless data compression scheme is proposed. This technique has a significant reduction effect on the test storage needs and the overall test time. Binary matrix lossless compression scheme used for lossless compression . In our proposed level of the test vectors are compressed as binary elements matrix. Before that we can calculate the amount of memory space required for the uncompressed test vectors generated by the test pattern generator. The pattern generator generates the test vector having the length of 7 bits. So each test vector can occupy 7 bits of memory space in the memory. The total number of test vectors generated is calculated as $2^7 = 128$ test vectors.

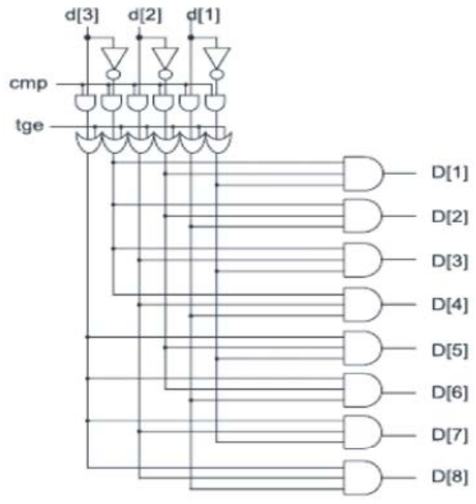


Fig. 4: Decoder unit

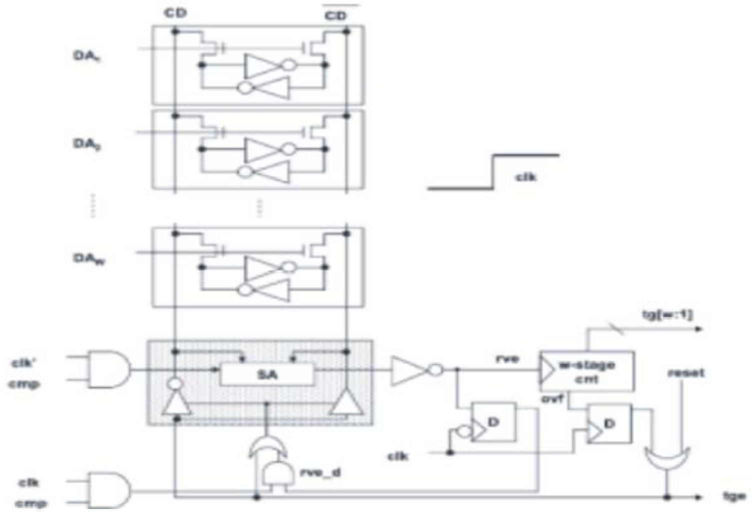


Fig. 5: Structure of logic module

$$BM(L) = \begin{pmatrix} 0\,0\,0\,0 \\ 0\,0\,0\,1 \\ 0\,0\,1\,0 \\ 0\,0\,1\,1 \\ 0\,1\,0\,0 \\ 0\,1\,0\,1 \\ 0\,1\,1\,0 \\ 0\,1\,1\,1 \\ 1\,0\,0\,0 \\ 1\,0\,0\,1 \\ 1\,0\,1\,0 \\ 1\,0\,1\,1 \\ 1\,1\,0\,0 \\ 1\,1\,0\,1 \\ 1\,1\,1\,0 \\ 1\,1\,1\,1 \end{pmatrix}$$

Fig. 6: *Binary matrix*

The total number of bits required for storing all the test vectors is equal to 128 * 7 = 896 bits. If the length of the test vector is high the amount of memory required is also very high. This degrades the performance of the circuit. To avoid this problem the binary matrix lossless compression scheme is used. The test vector generated by test pattern generator is separated into two parts as higher order bits and lower order bits. For the vector having 7 bit length, it can be separated into higher order bits (MSB=3 bits) and lower order bits (LSB= 4 bits). Then the binary matrices for MSB and LSB were generated [3].

The initial step of data compression is splitting of lower and higher order bits of the test vector. Then they are stored separately in their respective matrices. The next vector is scanned and separated as LSB and MSB bits and stored space can be reduced by this method. It will be repeated until all the test vectors have been scanned and compressed. in their respective matrices. If both the vectors having same MSB or LSB the same value is not stored again. It takes the previous stored value as their reference in the matrix. Repeated this above until the input vector is scanned and compressed finally the space of memory reduced. For example the test vector 0001101 is scanned and separated into MSB having the value as 000 and LSB having the value as 1101 were stored in the MSB and LSB binary matrices.

Then the next having the value as 1011. The MSB of the two vectors (000 and 001) were compared. Here they are not same. So the MSB of the second so the MSB of the second vector is also stored in another memory location of the MSB binary matrix [3]. Likewise the LSB of the two vectors were compared and returns the value in the LSB binary matrix. If the MSB of the two vectors were same the value was stored for only one time. The binary matrix for MSB = 3 bits is shown in Fig. 4.6. for MSB = 3 bits $(2)^3$=8 combination of vectors are generated.

Likewise if the value of LSB of the two vectors were same the value was stored only one time. All the test vectors generated by the test pattern generator were scanned and compressed by this method. After compressing all the incoming test vectors the binary matrix for MSB and LSB can be generated as shown in Fig. The MSB of the test vector having 3 bits can generate binary matrix for $(2)^3$=8 combination of the MSB bits. So the total number of bits required to store MSB binary matrix are 8*3=24 bits.

The LSB of the test vector having 4 bits can generate binary matrix for $(2)^4$=16 combination of the MSB bits. So the total number of bits required to store MSB binary matrix are 16*4=64 bits. So after compressing the test set the total number of bits required for storing the compressed test set is the sum of number of bits in MSB binary matrix and number of bits in LSB binary matrix. This can be equal to 24 + 64 = 88 bits. Here we need only 88 bits instead of 896 bits after compression.

The compression ratio between uncompressed and compressed test set can be calculated as follows.

$$\text{CR} = \frac{\text{uncompressed bit size}}{\text{compressed bit size}} = \frac{896}{88}$$

Memory space after compression.

$$\text{memory space reduced}(\%) = \frac{\text{UC bit size} - \text{C bit size}}{\text{UC bit size}} * 100$$

From the above it is cleared that about 90.178% of memory space is reduced by this compression method.

Decompression of Test Vector: The compressed data can be decompressed to get the original data. The data can be retrieved from the compressed data without any loss of the original data. The original test vector can be retrieved by combining the MSB and LSB binary matrices. The steps for the decompression of the test vector are shown as follows. The first row of the MSB binary matrix (000) is combined with the first row of the LSB binary matrix (0000). The first row of the MSB binary matrix occupies the higher order bit positions of the test vector having 7 bits length and the first row of the LSB binary matrix occupies the lower order bit position of the test vector having 7 bits length (0000000). Now the first test vector is decompressed from the compressed test vector. The second test vector can be retrieved by combining first row of the MSB binary matrix (000) and second row of the LSB binary matrix (0001). The remaining vectors can be retrieved by combining the first row of the MSB binary matrix and the remaining rows of the LSB binary matrix.After combining MSB's first row elements and all the LSB's row elements, the second row of the MSB binary matrix (001) is combined with the LSB's first row elements (0000) to create a test vector (0010000) and with the LSB's second row elements to create the next test vector (0010001). This can be continued until the second row element of the MSB binary matrix is combined with all rows of the LSB binary matrix. Likewise all the test vectors can be retrieved by combining all the elements in the MSB and LSB binary matrices. By this all combinations of test vectors (from 0000000 to 1111111) could be retrieved without any loss of data by this method. The original data can be retrieved from the compressed data in a very efficient manner by using this method.

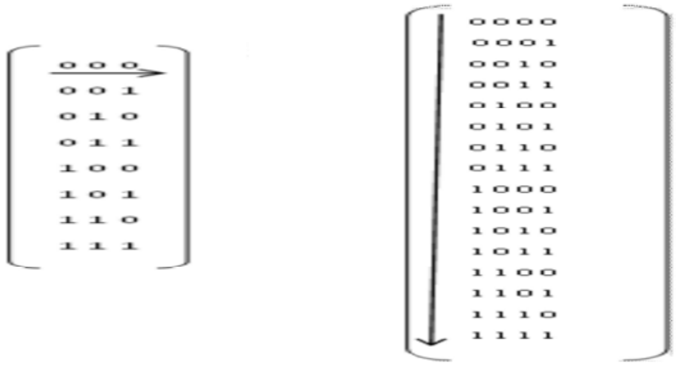


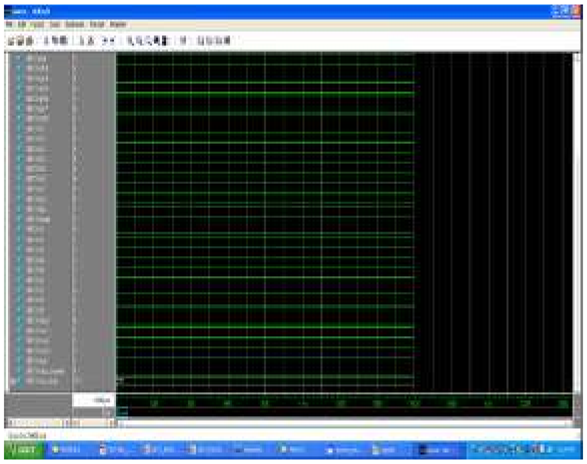
Fig. 6: Decompression of test vector

Built In Self Diagnosis And Repair Scheme: Built-in self test detects the presence of error. But, it did not diagnose or repair the error present in the test set. In this paper we propose the Novel Built-in Self Diagnose (BISD) and Built-in Self Repair (BISR) based on the fail pattern identification. The test pattern is decompressed and the syndrome pattern is calculated for each test pattern in the test set.

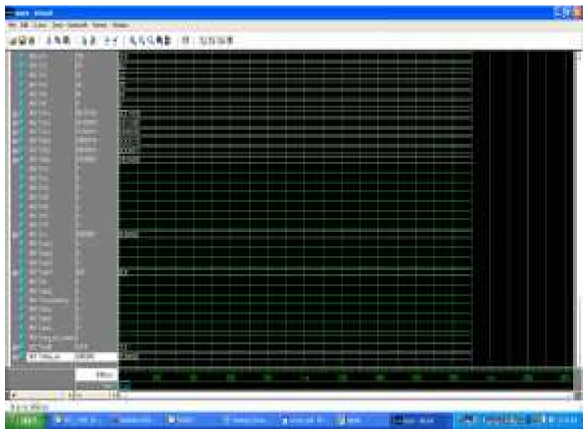
Syndrome1=rx[2]^rx[3]^rx[4]^rx[7];
Syndrome2=rx[1]^rx[2]^rx[3]^rx[6];
Syndrome3=rx[3]^rx[4]^rx[1]^rx[5]; Depending on the value obtained after the syndrome calculation the error bit is located and then the repairing of test vector is done. The error bit is calculated as follows.
rx=rx[1]^rx[2]….^rx[7]. The faulty test pattern can be corrected and the new test pattern can be calculated as follows.
New pattern = old pattern + rx

Simulation Result:

Logic Module

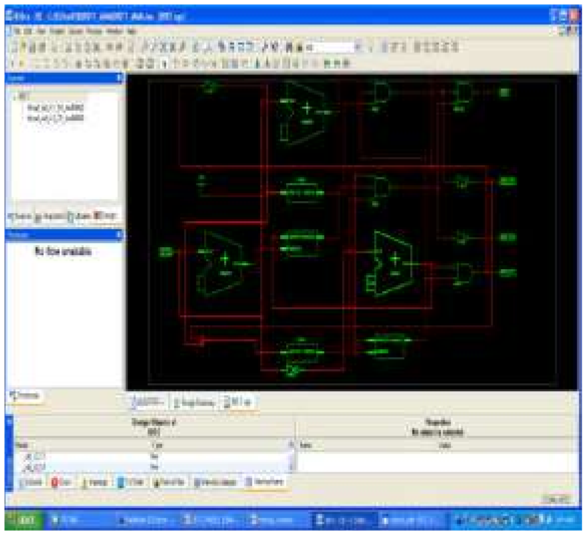


BIST Circuit



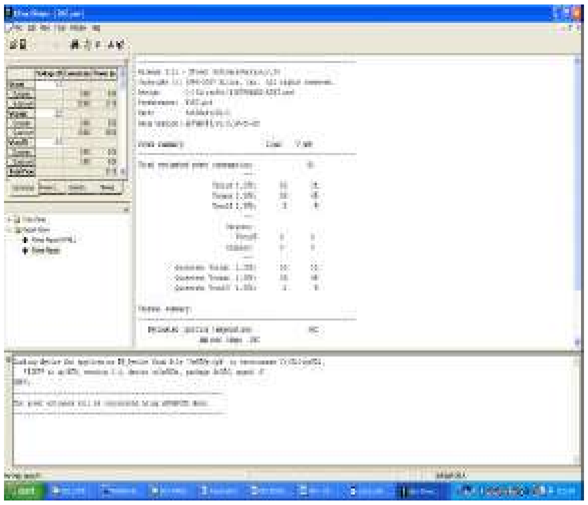
Synthesis Result:

RTL Schematic



The above figure shows the RTL schematic diagram of the BIST circuit. The word RTL stands for Register Transfer Logic. The RTL schematic displays the top view of the IC.

Power Report



The total power consumed by the BIST circuit is 81mW. The power consumption of existing BIST circuit is 159mW. Compared with the existing method the power consumption is low. The total power consumed by the BIST circuit is 81mW. The power consumption of existing BIST circuit is 159mW. Compared with the existing method the power consumption is low.Form the table it is given that the total power consumption of the BIST circuit is 159mW and extended work is 81mW. The power rate can be calculated by using the equation given below.

$$power\ rate = \frac{p2\ pwr}{p1\ pwr} * 100$$

Here p2 pwr refers to power consumption of the BIST unit in phase2 and p1 pwr refers to the power consumption of the BIST unit in previous work. This can be estimates by using the above equation.

$$power\ rate = \frac{81}{159} * 100 = 50.94$$

About 50.94% of power can be reduced in our proposed work.

CONCLUSION

In this paper, we provide a solution to the problem of vectorized testing of BIST circuits in VLSI devices. Vectorized BIST schemes performs the testing of circuit as normally and having without the need of circuit offline. The proposed scheme is shown to be more efficient performance improvement of existing work of input vector monitoring concurrent BIST techniques in terms of hardware overhead and CTL. The binary matrix lossless compression scheme and built in self repair and diagnosis scheme reduces the power is 50.94% and performances are improved.

REFERENCES

1. Woosik Jeong, 2009. “A Fast Built-in Redundancy Analysis for Memories With Optimal Repair Rate Using a Line-Based Search Tree”, IEEE Trans. On VLSI Systems, 17(12).
2. Chandra, A. and K. Chakrabarty, 2001. “System-on-a-chip test-data compression and decompression architectures based on Golomb codes, ” IEEE Trans. Computer Aided Design Integr. Circuits Syst., 20(3): 355-368.
3. Abramovici, M., C.E. Stroud and J.M. Emmert, 2004. “Online BIST and BIST-Based Diagnosis of FPGA Logic Blocks, ” IEEE Trans. Very Large Scale Integration (VLSI) Systems, 12(12): 1284-1294.
4. Voyiatzis, I. and C. Halatsis, 2005. “A Low Cost Concurrent BIST Scheme for Increased Dependability,” IEEE Trans. Dependable and Secure Computing, 2(2).
5. Lu, S.K., Y.C. Tsai, C.H. Hsu, K.H. Wang and C.W. Wu, 2006. “Efficient built-in redundancy analysis for embedded memories with 2-D redundancy, ” IEEE Trans. Very Large Scale Integr. Syst., 14(1): 34-42.
6. Chatterjee, M. and D.K. Pradhan, 2003. “A BIST Pattern Generator Design for Near-Perfect Fault Coverage, ” IEEE Trans. Computers, 52(12).
7. Paolo Bernardi, 2012. “An Adaptive Low-Cost Tester Architecture Supporting Embedded Memory Volume Diagnosis”, IEEE Trans. on Instrumentation and Measurement, 61(4).
8. Voyiatzis, I., A. Paschalis, D. Gizopoulos, N. Kranitis and C. Halatsis, 2005. “A Concurrent Built-In Self Test Architecture Based on a Self-Testing RAM, ” IEEE Trans. Reliability, 54(1): 69-78.
9. Chih-Tsun Huang, 2003. “Built-In Redundancy Analysis for Memory Yield Improvement”, IEEE Transaction on Reliability, 52(4).
10. Sobeeh Almukhaizim, 2006. “Entropy-Driven Parity-Tree Selection for Low Overhead Concurrent Error Detection in Finite State Machines”, IEEE Transaction on CAD Integrated Circuits and Systems, 25(6).