

Design of BIST using Self-Checking Circuits for Multipliers

Nishant Govindrao Pandharpurkar* and V. Ravi

School of Electronics Engineering, VIT University, Chennai - 600127, Tamil Nadu, India;
nishant.govindrao2013@vit.ac.in, ravi.v@vit.ac.in

Abstract

Background: Current technologies results in gradual increase in sensitiveness towards faults causing malfunctioning of the circuit. This paper presents the novel design of Built-In-Self-Test (BIST) using self-checking circuits for bit array multipliers. **Methods:** The design of BIST comprises of self-checking full adder which ensures fault detection on the same chip area. Each regular full adders and half adders in bit array multipliers are replaced by self-checking full adder so that any transient or permanent faults can be detected and recovered. The proposed BIST design also allows power saving procedures in Power Efficient-Test Pattern Generator (PE-TPG). **Findings:** Simulation results shows that implementation of this self-checking full adder into standard bit array multiplier minimizes the area overhead and power consumption by 25%-30% as compared to previous self-checking designs. The proposed BIST can handle up to ten faults with 70% probability of error detection, which is higher than earlier Double Modular Redundancy (DMR) as well as Triple Modular Redundancy (TMR) technique with handling of six faults with 60% error detection probability. **Conclusion:** The proposed BIST design forms the base of area and power efficient testing methodologies for digital circuits. The architecture of BIST can be modified according to the data path of multiplier under test.

Keywords: Built-In-Self-Test (BIST), Fault Detection, Self Checking, Stuck-at Fault, Test Pattern Generation (TPG)

1. Introduction

As microelectronic technologies advance and approach towards below nano-metric technology nodes, digital System-on-Chip (SoC) has become very sensitive to multiple faults. These faults may be introduced during production that can affect the obvious functioning of system and thereby result in producing the undesired outputs¹. In recent years, several fault tolerance approaches are emphasized to cope with increasing number and levels of defects that are affecting functionality with current and future nanometer CMOS technologies². With recent advancements of testing strategies, fault detection and recovery techniques are one of major research areas dealing with DFT procedure for digital systems. Most of the DSP architectures are composed of wide range of multipliers. The timing constraints of the multiplier decide the maximum operational frequency of a circuit data path for

optimum efficiency of circuit. The commonly employed multiplier designs used in high-speed data path structure refer to the category of array multipliers³.

Nowadays, multiplier serves an integral functional block in complex VLSI DSP processors, which constitutes major part of System-on-Chip (SoC) architectures. These ultra-complex structures generate testability issues for the multiplier blocks as well as to other functional modules within the processor. This problem is mainly produced due to the reduction in accessibility of the entire system. Such types testing difficulties also prevents integrated circuit for digital system from being launched in market for application. Built-In Self-Test (BIST) architectures for functional array multipliers^{3,4} and for other functional modules have emerged as an effective testing alternative in terms of economy, efficiency and fault detection probability. Several failure mechanisms in ultra-complex CMOS circuits are insufficiently being exhaustively tested

*Author for correspondence

by conventional fault models such as generic stuck-at fault model. The most common errors imposed during the fabrication flow of a CMOS circuit design^{5,6} can alter behavioural and functional properties and allows to convert it from a combinational circuit faults to a sequential circuit faults⁴. The major fault models are based on sequential fault models, such as the stuck-at fault model, the gate delay, the path delay, and the segment delay fault models can explain sequential behaviour of faults in CMOS circuit design^{5,6}.

In past few years, a prominent model has been emerged as a fault model for testing sequential faults. It is referred as Realistic Sequential Cell Fault Model (RS-CFM) and it is proved to be reliable for a wide range of arithmetic circuits such as multipliers and dividers⁷. It essentially works on the generation of Single Input Change (SIC) doublets of test vectors which are fed to each of cells of digital circuit, SIC refers to doublets of test vectors differing only in single bit. RS-CFM allows test strategies which are independent of gate level testing and assures maximised sequential fault coverage with minimum invalidation. This paper also concerned with circuit under test, i.e., multipliers to be totally self-checking which is based on RS-CFM cell fault model. For this, self-checking full adder is employed in design of multipliers. A single full adder performs addition in such a way that it performs addition as well as equivalence functionality computation by following distinct logical paths for both procedures, and finally the equivalence comparison of the final testing bits will indicate the faulty data path within the circuit design^{8,9}. The key characteristics of the design are reliability and a decrement in area overhead using distinct paths for a single addition operation at each time interval. In addition to this, DFT modifications are done in a way that operational speed of CUT remains same. It has been presumed that the result will be forwarded to next stage after the first computation of output bits is done. Therefore, suppose if there is a fault in first computed result, that result will be used for other computations, before the detection of faults. This can be avoided by the fault tolerant self-checking adder design.

This paper emphasizes on the BIST design for standard array multiplier using self-checking technique. The basic principle of fault detection in BIST is to introduce self-checking adder circuit within the multiplier design. This means that the total architecture of multiplier has been redesigned in order to obtain high fault coverage capabilities. Detection of fault using self-checking adder includes

the monitoring of error function (E_i) signal within the each adder module. The major advantage of this method is that any transient or permanent fault can be detected online at any stage of multiplier^{10,11}. To implement low power methodologies, a power efficient Automatic Test Pattern Generator (ATPG) has been incorporated in the BIST design to check the authenticity and reliability of the total BIST structure¹². The ATPG is called as power efficient because a low power methodology is used such that generated test vectors will be having least hamming distance. This helps in incorporating low power techniques to design for testability procedures for wide range of testing applications¹³⁻¹⁵. In this paper, Section 2 illustrates earlier BIST design approaches that are previously designed in accordance with standard array multipliers. Section III illustrates the self-checking circuits employed in the design of proposed BIST structure. Section V presents the methodology and work flow of BIST architecture redesigned techniques. To elaborate the power efficiency of proposed BIST, BIST architecture with self-checking methodology has been implemented using 180nm technology using Industry Standard Cadence Virtuoso. Further, the proposed design has been compared with popular techniques like DMR and TMR^{14,15}.

2. Previous Design Techniques

2.1 Testable Carry save Adder Multipliers

In most DSP applications, the fastest multiplication possible can be performed using the method of carry save addition in which the partial products are being added using carry save adders. Hence, the multiplier is called a CSAM^{5,6}. The building elements consists of 2-input AND gate, the Full Adder (FA) and the Half Adder (HA). The input line ppi, j represents the output line of the 2-input AND gate receiving input bits as x_i and y_j of the multiplier and multiplicand bits respectively, shown in Figure 1. In this design, the output of and gate ppi, j are fed to successive full adders in order to save the carry bit during propagation. The carry saved are again fed in a diagonal fashion to generate the final partial products. These partial products are then added to obtain the final product bits.

Conventional BIST design for the CSAM was designed on the basis of cell fault model as shown in Figure 2. This design highlights on the consideration that a single cell or element can be faulty and that only combinational faults are possible at simulation run time. In this design, the

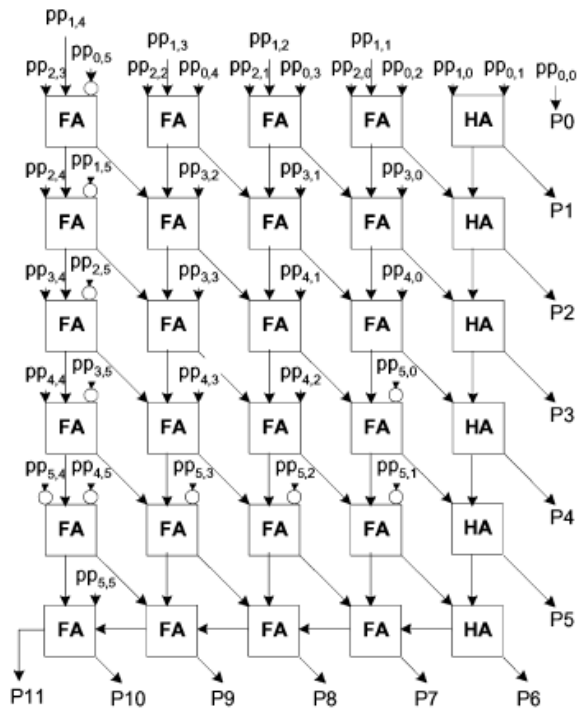


Figure 1. Carry Save Multiplier.

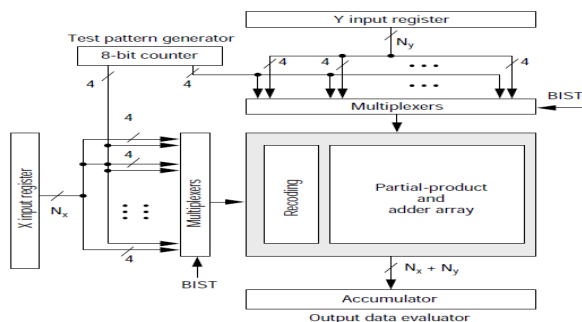


Figure 2. BIST structure for CSAM.

Test Pattern Generator (TPG) comprises of a counter of 8-bits that access through all of the possible 256 cycles. During BIST operation, the four MSB of the TPG outputs are allowed to generate the multiplier input X and the remaining four bits are allowed to generate the multiplier input Y in repetitive fashion. The TPG test vectors are applied to CSAM multipliers for exhaustive testing. The speed of exhaustive testing is totally dependent on the optimum speed of generation of counter outputs and their respective propagation delay at the input node of multipliers. Figure 2. shows basic architecture of BIST structure used for CSAM multiplier or booth multiplier⁵.

In case of booth multiplication, recording of bits is essential to generate partial products, which on successive addition gives final product bits.

2.2 Fault Tolerant Triple Modular Redundancy (TMR)

A sufficient number of transient faults produced within the design are temporary in nature, and their presence in run time, can damage the functioning of total system. To prevent the ultimate functional authenticity of Circuit Under Test (CUT), i.e. the digital circuit being exhaustively tested for proper functionality and optimum efficiency. A fault tolerant scheme such as Triple-Modular Redundancy (TMR) has been designed with the combined aim of designing a voter circuit which ensures fault tolerance along with generation of error free results and also maximising the system reliability⁷. The voter circuit is capable of determination of transients or permanent fault by simple comparing within its input stream. The major drawback of this system is that reliability of voter circuit is presumed to be very high, which may not be possible in each case. Also, TMR can tolerate only a single element within the total design to be subjected with error, which may or may not be practically possible. Considering a situation in voter circuit itself is sensitive to errors, the total redesigned scheme of TMR is assured to be failed⁷. In addition to this TMR scheme is not flexible with the technological advancements. As technology approaches toward complex design, the voter circuit needs to be redesigned according to the desired application.

3. Self-Checking Circuits

This section briefly summarizes the features of self-checking circuits to provide the essential concepts for the design of BIST using self-checking circuits. This self-checking full adder and other self-checking circuitry are purposely introduced in BIST and CUT to attain high fault coverage and tolerant system without affecting the operational speed of system. In recent years, many cell based testing strategies have been developed either by using hardware or by using time-based redundancy. The major setback of these strategies is that determination of fault location is not possible during fault detection. This is due to fault propagation due to carry. Self-checking circuits are one the alternative ways. Following subsections explains about the methodologies and cell structures employed to attain the final BIST design.

3.1 Self Checking Full Adder Circuit

In digital architecture design, the adder provides a module that can be used for wide range of digital system architecture. In each of every architecture design, adder can be considered as a fundamental block of data transmission mainly in multipliers. Recent advancements in testing techniques include efficient testing strategies that have been aimed to implement self-checking in adder circuits. This self-checking strategy can be introduced in adders using redundancy techniques based on either hardware or time constraints.

In¹, a new fault tolerant self-checking full adder has been proposed as shown in Figure 3., in which the faults on full adder can be detected online using conventional cell fault model. Self-checking and repair can be achieved considering the corresponding Boolean relationship between adder output bits. It has been observed that Sum and Carry bits always complement each other except when all adder input bits are equal in logic. Using this functionally equivalency, a full adder can be redesigned as self-checking at the expense of an extra area overhead of an Equivalence Tester (Eq_t). The equivalence testing circuit is aimed for the equivalence checking at input node of a full adder.

$$\text{Sum} = A \oplus B \oplus C \tag{1}$$

$$\text{Cout} = A.B + \text{Cin}(A + B) \tag{2}$$

$$\text{Eq}_t = \text{Not}((A'B'Cin') + (ABCin)) \tag{3}$$

The first XNOR gate (G1) is used for bit to bit comparison between two bits namely Sum or Count. This means that it is used to determine logic level comparison of Sum and Carry-out bits, being equal or complemented. In addition to this, the goal of a second XNOR gate (G2)

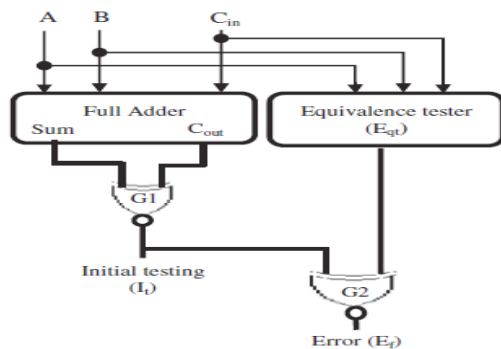


Figure 3. Self Checking Adder Design.

is used to check the output bit of G1 by comparing it with an equivalence tester, thereby generation of error function (E_f) signal, which on active low level signifies fault-free circuit. Active high signal on E_f signal indicates that presence of transient or permanent fault within the circuit design. Thus fault detection capability of full adder has been introduced with the expense of extra area overhead caused due to equivalent tester.

3.2 Self Checking Half Adder Circuit

On the same ground as of full adders, the half adder can also be employed with self-checking design techniques. The internal design of half adder is changed according to cell fault model. In same way as of self-checking full adder, this modification includes introduction functional equivalence tester. The functions of XNOR gates G1 and G2 remains the same as of self-checking full adder. Boolean expressions that can be used for this purpose are started below.

$$\text{Sum} = A \oplus B \tag{4}$$

$$\text{Cout} = AB \tag{5}$$

$$\text{Eq}_t = \text{NOT}(A'B') \tag{6}$$

$$\text{Error Function } E_f = S \odot \text{Cout} \odot \text{Eq}_t \tag{7}$$

3.3 Self Checking Multiplexer

In order to produce optimized level of fault coverage system, each and every of the module and element of system are needed to be redesigned on self-checking technique. Hence, the multiplexer are essentially are needed to be redesigned internally in accordance with self-checking methodology. It consists of four transmission gates along with an inverter as shown in Figure 4.

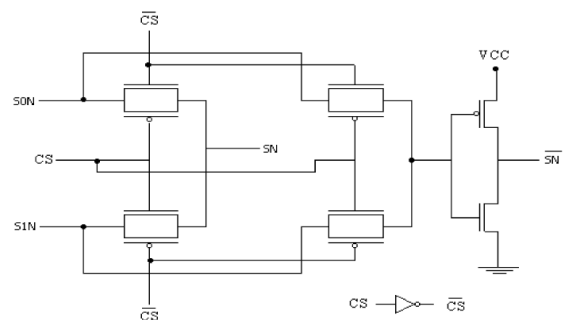


Figure 4. Self Checking Multiplexer.

A self-checking multiplexer is made of four transmission gates and an inverter. In these self-checking multiplexers, the basic idea of separable valid code word is used. This means that, the presence of fault within the design causes the appearance of valid code words at output nodes. The valid code words refer to set of complemented output bits SN and SN'. Similar logic level at SN and SN' indicates the presence of transient faults.

4. BIST using Self-Checking Circuit

The BIST structure is designed in such a way that it causes power requirement to a possible minimised level when exhaustive testing of multipliers being taken into consideration. This is allowed due to conventional automatic test pattern generator has been replaced by Power Efficient-Automatic Test Pattern Generator (PE-ATPG). Also modified registers X and Y have been introduced and implemented in place of simple registers to perform as a shift registers in BIST design described in this section.

4.1 Test Pattern Generation (TPG)

Figure 5 shows the TPG design dependent on operation of an 8-bit counter. The TPG allows the counter output bits applied with specific patterned test vectors to the multiplier inputs during testing mode by BIST signal. This can be done in two ways: in first method, the four LSBs of output bits of the 8bit-counter are fed for the creation of X operand test pattern generation and the four MSBs outputs are fed for the creation of Y operand test pattern generation. The performance and reliability of counter based TPGs for array multipliers had already stated in previous research papers for a genuine one-pattern testing algorithms³.

In test pattern generation for modelling BIST, a power efficient scheme of two pattern testing has been implemented. In this scheme, as name suggests, a sequence of two-pattern test vectors are generated. The input register

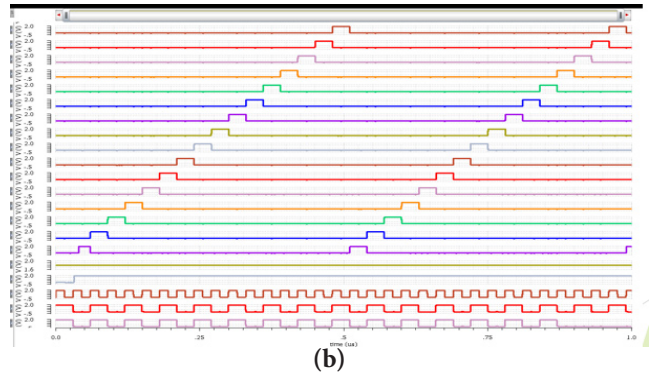


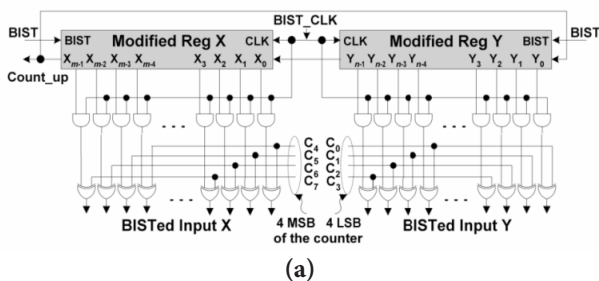
Figure 5. (a) PE-TPG in BIST. (b) Test Vectors in PE-TPG.

organization of the multiplier inputs in combination with controller circuit block is creditable for the pattern initialisation of the two-pattern testing sequence. In addition to this, the input modifications are done in input registers, so that it functions as a shift registers whenever testing under BIST signal is needed. The basic idea of implementing a robust design for BIST is that two vector patterns in succession are allowed to differ at very few bits. This can be inferred that hamming distance between successive patterns of test vectors are minimum. Considering the multiplication of m×n bits, the control logic of robust design consists of (m+n) AND gate and (m+n) XOR gates. The four LSBs, viz., C3, C2, C1 and C0 of the counter outputs are applied with data Y through array of XOR in a repetition. In same way, the four MSBs viz., C7, C6, C5 and C4, of counter outputs are applied with data X operands through array of XOR gates. The functions at the input side during testing under BIST signal is given as

$$BIST_{Yi} = (Yi.Clk) \oplus Ci(mod4) \tag{8}$$

$$BIST_{Xi} = (Xi.Clk) \oplus Ci(mod4 + 4) \tag{9}$$

This can be elaborated with following example. When the BIST clock is at active low state, the BISTed inputs viz., Y0, Y4, Y8 etc. gets the value of C0, while the BISTed inputs X0, X4, X8, etc., gets the value C4. On other hand, when BIST clock is at active high level, the BISTed inputs Y0, Y4, Y8 etc. gets the value C0 or C0' depending on respective outputs of the DFT redesigned or modified register Y gains the value of 0 or1 respectively. On the similar grounds, the BISTed inputs X0, X4, X8 etc., gains the value C4 or C4' depending on outputs of the DFT modified register X gains the value 0 or 1. It must be noticed that the BIST clock is taken at the half range to that of frequency range of the multiplier clock.



4.2 DFT Modifications to Registers X and Y

The design of the DFT modified input registers at the periphery, in case of testing circuitry, depends on the type of the multiplier and its length. The robust design of BIST needs to cover all the testing requirements and characteristics of a wide range of bit array multipliers, including the standard array multiplier and the array multiplier based on modified booth algorithm. This can be accomplished by reconfiguring internal design of the registers such that a different sequencing of states for input pattern is generated for particular architecture of array multiplication design. The DFT modified input registers for array multipliers are shown in Figure 6. For every specific design of the array multipliers, input registers are “modified” in accordance with Design for Testability (DFT) to act as a shift register (Hence referred as DFT modified registers). The register is initialized to the bit values “000...0001”. When standard array multipliers is being exhaustively tested, the modified register acts as a single bit shift register. The serial output is applied to the serial input bit position of the register.

The two registers are connected in such a way that it constitutes an overall ring counter. The DFT modifications in input registers include the introduction of multiplexers at the periphery, which selects the input bits to CUT depending on the BIST signal of the control logic. This means that when BIST signal is at high level, BISTed input are fed to CUT and when it is at low level, normal user defined inputs can be fed to the CUT. Another modification includes applying a distinct sequence of patterns for every design of the array multipliers. All possible testing parameters of the array multipliers have been monitored and these DFT modifications are suggested. The DFT modified registers for the array multipliers are shown in Figure 6.

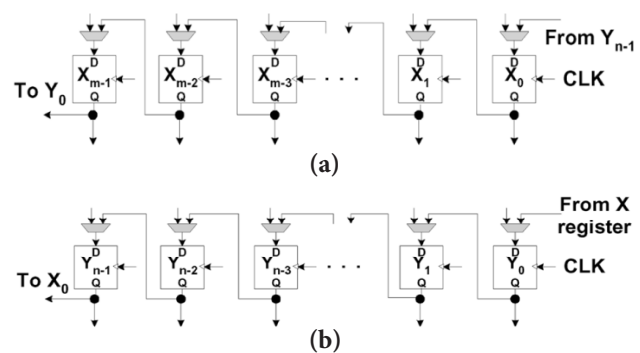


Figure 6. (a) DFT Modifications to X register. (b) DFT Modifications to Y register.

5. Total BIST Design

The generalised BIST structure is shown in the Figure 7. As depicted in the figure, total structure of BIST includes PE-TPG followed by CUT i.e. multipliers. These multipliers which comprises of self-checking full adders and other elements, will generate series of Error Flag E_f signal at each stage. This means that each row of standard array 8×8 multiplications consists of about eight E_f signals, totalling 64 E_f signals for total multiplier design. In other words, for an array multiplication of $n \times n$ data words, a total of n^2 error flag signals are generated which when fed to n bit encoding stage, which on encoding E_f signals, can determine the location of faults. The location of fault indicates the particular stage of array multiplication. This means the fault detection can be achieved using BIST with self-checking circuits at the expense of extra area overhead of equivalence testing circuits. This area overhead may range from 20%-35%. In addition to this, the requirement of data compaction circuit (ODC) or accumulator circuit has been replaced by encoding circuit. It must be noted that the area overhead due to separate ODC or accumulator will be higher than encoding circuit employed in BIST design enabling high fault coverage

6. Simulation Results

Table 1 shows the power requirements for ATPG design methodologies, which proves that power savings will be more in case of Power Efficient-Automatic Test Pattern Generator (PE-ATPG) as compare to earlier design approaches like counter based ATPG and LFSR based ATPG. On other hand, due to employing the PE-ATPG in BIST design with self-checking technique, overall power requirement get reduced to 25%-30% as shown in Table 2.

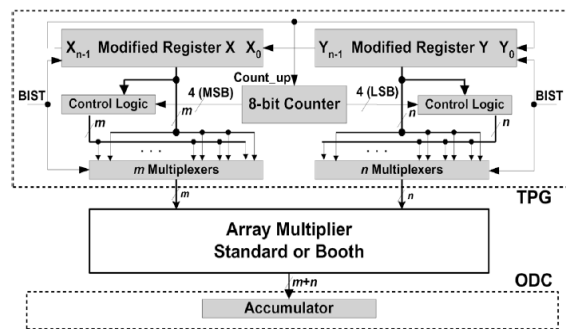


Figure 7. BIST Design.

Table 1. ATPG power requirements

Sl. No.	ATPG Power Requirements	
	ATPG Design	Power (uW)
1	Normal ATPG(without Self Checking)	2x10 ⁵
2	Counter Based ATPG	723.2
3	LFSR Based ATPG	644.3
4	Power Efficient ATPG	613.2

Table 2. BIST power requirements.

Sl. No.	BIST Power Requirements	
	BIST Design	Power (mW)
1	Normal BIST(without Self Checking)	11.230
2	BIST with TMR	9.878
3	BIST with DMR	9.762
4	BIST with Self Checking	7.811

7. Conclusions

From Table 2, it can be inferred that with the same set of test vectors, BIST architecture with self-checking technique provides much better results than the pseudorandom BIST. The PE-ATPG contributes to sufficient amount of power saving due to generation of set of two pattern test vectors with minimum possible hamming distance. Self-checking technique within the array multiplier increases the probability of fault detection but it also causes an extra area overhead¹. The total increase in area overhead due to self-checking technique ranges from 30%-35%¹. For applied test vectors, the proposed BIST using self-checking technique achieves exhaustive testing of array multipliers with fault handling up to 10 faults with 70% probability of error detection. It can be concluded that the fault coverage has been maximized with considerable amount of power saving at the expense of 30%-35% area overhead.

8. References

1. Akbar MA, Lee J-A. Self-repairing adder using fault localization. *Microelectronics Reliability*. 2014 May; 54(6-7):1443-51.
2. Vasudevan DP, Lala PK, Parkerson JP. Self-checking carry-select adder design based on two-rail encoding. *IEEE Transactions on Circuits and Systems - I: Regular papers*. 2007 Dec; 54(12):2696-705.

3. Bakalis D, Kalligeros E, Nikolos D, Vergos HT, Alexiou G. On the design of low power BIST for multipliers with Booth encoding and Wallace tree summation. *Journal of Systems Architecture*. 2002 Dec; 48(4-5):125-35.
4. Bakalis D, Nikolos D. On low power BIST for carry saves array multipliers. *Proceedings of the 5th International On-Line Testing Workshop*; 1999. p. 86-90.
5. Bakalis D, Vergos HT, Nikolos D, Kavousianos X, Alexiou G. Low power dissipation in BIST schemes for modified Booth multipliers. *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*; 1999. p. 121-9.
6. Paschalis GA, Zorian Y. An effective BIST scheme for carry save and carry-propagate array multipliers. *Proceedings of 4th IEEE Asian Test Symposium*; 1995. p. 286-92.
7. Kshirsagar RV, Patrikar RM. Design of a novel fault-tolerant voter circuit for TMR implementation to improve reliability digital. *Microelectronics Reliability*. 2009; 49:1573-7.
8. Psarakis M, Gizopoulos D, Paschalis A. Built-in sequential fault self-testing of array multipliers. *IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems*. 2005 Mar; 24(3):449-60.
9. Wang L-T, Wu Paul Y-F. Patriot - A boundary-scan test and diagnosis system. *Proceedings of the 37th Annual IEEE International Computer Conference*; San Francisco, CA, USA. 1992. p. 436-9.
10. Abramovici M, Breuer MA, Friedman AD. *Digital System Testing and Testable Design*. IEEE Press; 1990.
11. Rajski J, Tyszer J. On the diagnostic properties of linear feedback shift registers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1991 Oct; 10(10):1316-22.
12. Aitken RC, Agarwal VK. A diagnosis method using pseudo-random vectors without intermediate signatures. *Proceedings of the IEEE International Conference on Computer-Aided Design*; 1989. p. 574-7.
13. Karpovsky MG, Chaudhry SM. Design of self-diagnostic boards by multiple signature analysis. *IEEE Transactions on Computers*. 1993 Sep; 42(9):1035-44.
14. Kanopoulos N, Vasanthavada N, Watterson J, Hallenbeck JJ. A new implementation of signature analysis for board fault isolation testing. *Proceedings of the IEEE International Test Conference*; 1987. p. 730-6.
15. Karpovsky MG, Roziner TD, Moraga C. Fault detection in multiprocessor systems and array processors. *IEEE Transactions on Computers*. 1995 Mar; 44(3):383-92.