

# Built-In Self Test and Self-Repairing Circuit for Array Multipliers

Jins Alex\* and S. Umadevi

Department of SENSE, VIT University, Chennai Campus, Chennai - 600 048, Tamil Nadu, India;  
jinsalex3221@gmail.com, umadevi.s@vit.ac.in

## Abstract

**Background/Objectives:** As the size of the chip reduces, nanoscale devices have become more susceptible to manufacturing faults, interference from radiations and transient faults. Many of these errors are not permanent but it causes malfunctioning of circuit either due to the complexity of the circuit or due to the interaction with the software. In this paper, an area and power efficient BIST with self repairing technique has been proposed, which detect and repair the faults in the circuit. **Methods/Statistical Analysis:** In this research work, a novel Built-In Self-Test (BIST) architecture with self repairing circuit is proposed. The novelty of this architecture is that testing is done along with the self repairing. The Self repairing circuit repairs the fault in the circuit during testing phase itself which increases the reliability of the circuit. Since the insertion of test pattern externally, BIST architecture does not alter the basic multiplier structure. **Findings:** Average power dissipation of the proposed Built-in self test and self repairing of array multiplier architecture is reduced by 36% since the use of a power efficient test pattern generator. Self repairing has been accomplished by the use of hardware redundancy technique. Also a TMR based self repairing architecture for real time self repairing has been proposed and its area and power dissipation is compared with the other self repairing architecture. Result shows that the BIST with repairing technique is good for low power applications while the TMR based self repairing method is good for real time self repairing applications. The proposed technique can be extended to self repairing processor.

**Keywords:** Array Multipliers, BIST, DMR, Low Power, TMR

## 1. Introduction

Advanced microelectronic technologies have allowed present digital systems to become more susceptible to faults. It is being observed that the issue of single event change in digital systems has become more influential with the increasing complexity of chip and decreasing number of clock cycle in order to obtain high speed<sup>1,2</sup>. The design of complex system on a chip is advantageous in terms of noise but it creates more problems in terms of reliability<sup>3,4</sup>. Researches show that the reduction in hardware size may increase the failure rate of hardware in future process<sup>5</sup>.

In order to deal with the problems that has mentioned above, a concept called fault tolerance and self checking has been introduced. A system is said to be fault secure if its output is unaffected by the fault or if it detects the

fault when it occurs<sup>6</sup>. A system will be self checking if it produces an error signal for every generated faults. A system is said to be self checking and self repairing if it detects the faults as well as correct it before it makes changes in the output of the system.

Multiplication operation is one of the most important operations among the frequently used functional operations in terms of circuit area, speed and power consumption. Basically, frequency of the system is determined by the propagation delay of multiplier. The most commonly used multipliers in terms of speed are array multipliers. The regular, cell-based design of array multipliers makes possible not only their efficient Very Large Scale Integration (VLSI) implementation, but also their design verification and testing phases. Today, multipliers are embedded in processor cores, which in turn are embedded in system-on-chip. This structure produces

\*Author for correspondence

serious testability problems not only in the multiplier module but also in the other functional units. Built in Self Test for array multipliers and for other multipliers is an effective testability solution<sup>7</sup>. It allows testing at actual frequency and provides very high fault coverage. Since there is no external expensive hardware is used, cost of the Built in Self Test becomes low. Defects during the fabrication of CMOS circuits<sup>8</sup> leads to stuck at faults and it converts a combinational circuit into sequential circuit. Only sequential fault models like the segment delay, stuck open, path delay and gate delay fault models can describe such a sequential behaviour.

A fault can be defined as an incorrect state of hardware or software which is part of a system. Fault can leads to physical defects, design flaws and single event upset due to radiations. According to the duration or source of faults, it can be divided as permanent, intermittent and transient faults.

The principle of fault-tolerance is to hide the effect of faults in the system by the use of redundant components. As a result, a fault tolerant system is one which is capable of maintaining the fault free operation even if a fault occurs due to either internal or external causes. Although, not all the faults cause error and not all the errors leads to malfunctioning of system. There are two ways to increase the reliability of digital systems. One method is known as fault prevention or fault intolerance and second method is fault tolerance. In the traditional fault prevention technique, the objective is to increase the reliability of the system by elimination of all the faults. But it is practically impossible. Impracticability of the first method leads to the second method, fault tolerance. It is a technique where faults are expected to occur in the system during operation, but its effects can be countered by the use of extra hardware- that is, additional resources – so that valid operation has been done even in the presence of faults. These resources can be extra hardware (Hardware redundancy), additional software (software redundancy) additional information (information redundancy) and more time (time redundancy) or the combination of all. They are redundant means it can be omitted from the system without affecting the normal operation of the system. In this paper, stuck-at fault has been discussed mainly. Two important techniques for fault masking are Double Modular Redundancy (DMR) and Triple Modular Redundancy (TMR). Advantages of both TMR and DMR technique has been used in this project.

## 2. Review of Existing Array Multipliers

Consider the multiplication of X and Y with size M and N respectively. The operation of array multiplier can be divided into two parts. The first part produces the partial products and the second part adds the partial product in order to get the final multiplied output.

### 2.1 Carry Propagate Array Multiplier

Consider a 6\*6 Carry Propagate Array Multiplier<sup>9</sup> shown in Figure 1 is having two inputs X and Y of 6-bit each. The first part is the production of all the 36 partial products using AND gates where the inputs of AND gates are X and Y operands. These partial inputs can be applied as the input of full adders and half adders. The 6\*6 carry save array multiplier consists of 24 full adder and 6 half adder. In this architecture, carry of each adder is propagated to the next adder which is situated in the same stage.

The Standard non-recoded array multipliers calculate partial products and it consists of n bits. The value of the j<sup>th</sup> bit of the i<sup>th</sup> partial product is the logic AND operation of

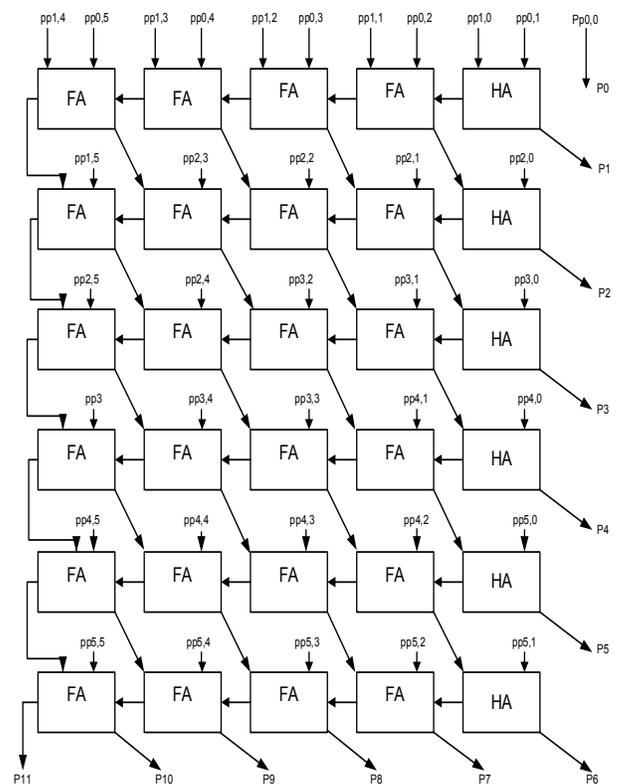


Figure 1. The 6\*6 carry propagate array multiplier.

bits  $X_i$  and  $Y_j$  of the multiplier and multiplicand operand, respectively. Partial products generated is named as  $PP_{ij}$ . The addition of  $M$  partial products realised using an array of Full Adders (FA) and Half Adders (HA).

### 2.2 Carry Save Array Multiplier

In carry save array multiplier<sup>9</sup> shown in Figure 2, carry from each adder transmit to the adder which is situated in the next row. Thus it can reduce the overall delay of the multiplier. In the carry-propagate configuration, carries ripple between the adders at the same row while in the carry-save method addition of carries realized in the next row.

## 3. Review of Existing BIST Architectures

Key elements of the BIST design are Test pattern generator, Circuit under test and output response analyzer. BIST places the job of device testing inside the device itself. It generates its own stimulus and analyzes its own responses. Main advantages of the BIST technique are low test cost, better fault coverage, short test time and test can be performed throughout the operational life of the chip. Disadvantages of this technique are the use of extra pins and area overhead. This section provides the analysis of previously implemented Built-in Self-test architectures.

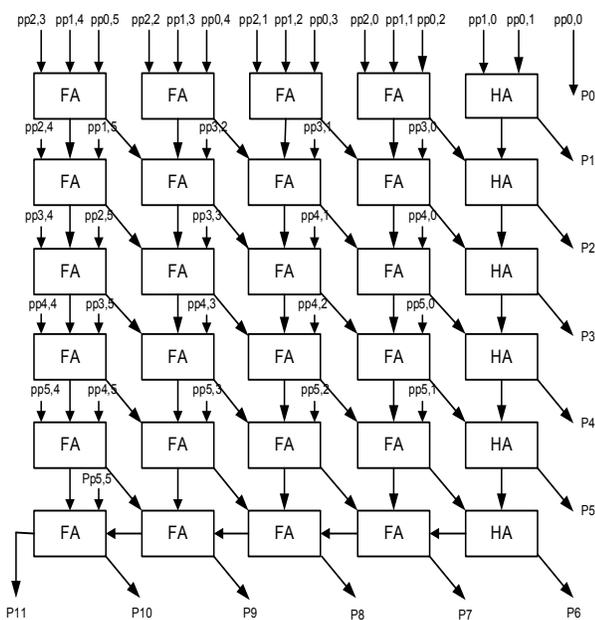


Figure 2. 6\*6 Carry save array multiplier.

## 3.1 Built-in Sequential Fault Self-testing of Array Multipliers

### 3.1.1 Test Pattern Generator

There are many circuits available for test pattern generator such as linear feedback shift register, counter and low speed test pattern generators. The Test Pattern Generator (TPG) which is shown in Figure 3 is a power efficient TPG. The modified registers of this TPG produces a sequence of  $(M+N)$  states where  $M$  and  $N$  are the size of  $X$  and  $Y$  modified register respectively. For every 64 states of the counter, TPG produces  $2(M+N)$  test vectors. That means it generates 1536 test patterns with minimum hamming distance. Power consumption of this test pattern generator is less since the reduction in hamming distance. The basic concept of this robust TPG implementation is that each two successive vectors must differ in very few bit position, in other words hamming distance of the applied two pattern test must be minimum. This TPG design consists of a 6 bit counter, modified  $X$  and  $Y$  register and control unit. Initial values of the modified  $X$  and  $Y$  register determines the test vector. Here registers are initialised to 00...1. Six bit counter generates 64 output patterns and each one of the 64 states the robust TPG generates a sequence of two-pattern test vector that contain one, two or three bit transitions. The Boolean equations for the BIST ed output are shown below.

$$BIST\ ed\ X_i = (X_i * CLK) \oplus C_j \tag{1}$$

$$BIST\ ed\ Y_i = (Y_i * CLK) \oplus C_j \tag{2}$$

### 3.1.2 Output Response Analyzer

The block diagram shown in Figure 4 is the accumulator based output response analyzer. The main function of the

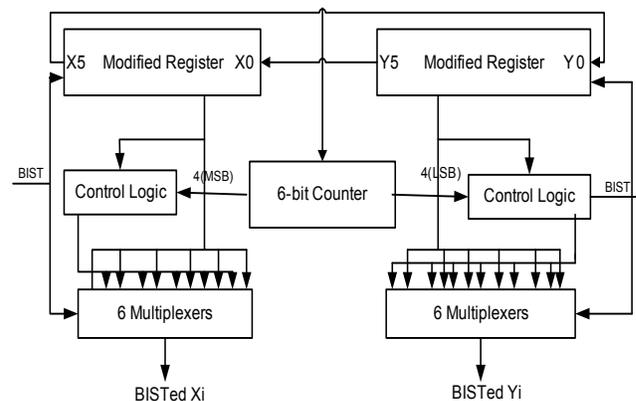


Figure 3. Test pattern generator architecture.

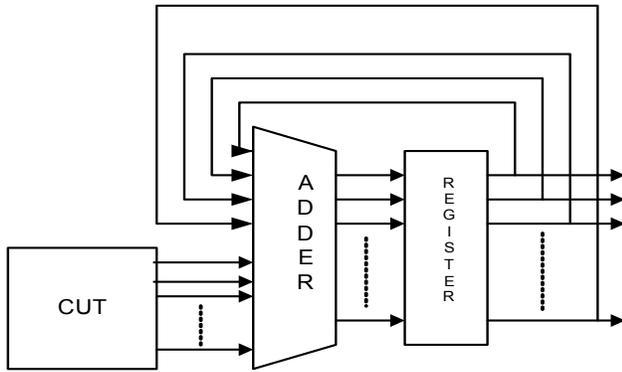


Figure 4. Output response analyzer architecture.

output response analyzer is to differentiate faulty and fault free output response of the multiplier. Accumulator based compactor consist of adder, accumulator and comparator. The binary adder connected with accumulator in such a way that each binary number transferred to the adder is added to the previous content of the register. The circuit, called binary accumulator, performs the function that can be considered as a compaction of responses generated by an external circuit under test. A fault is said to be injected to the accumulator, if a respective input vector, i.e., a test response, is different than that obtained from the fault free circuit.

The working of BIST can be described as follows. The test pattern generator which is explained in Figure 4 generates test vectors and is applied to circuit under test that is array multiplier. According to the test patterns which applied to the circuit under test, multiplier will produce responses. Accumulator gives a value, after applying all test vectors will give as one of the input to comparator. This value is considered as the signature of the particular fault free multiplier. Faults can be applied after the analysis of fault free multiplier. The fault which is applied to the fault free multiplier is stuck-at fault. In order to apply a stuck at fault either connect a wire whose value is one to ground or connect a zero valued wire to power supply. After applying all the test vectors it can be confirmed that the value which is present at the accumulator is different from fault free accumulator value. This is called the propagation of fault.

### 3.2 Effective Built-in Self-test for Booth Multipliers

The BIST architecture shown in Figure 5 can be divided into three parts. They are 8-bit counter, booth multiplier

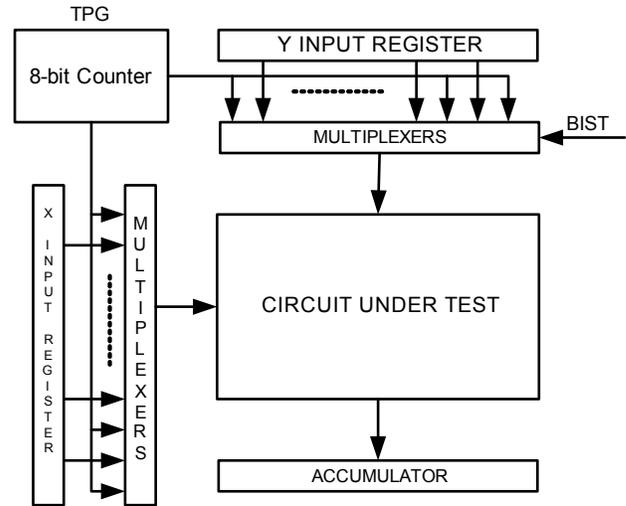


Figure 5. BIST architecture.

and accumulator. Eight bit counter works as a test pattern generator, accumulator function as output response analyser and circuit under test is booth multiplier.

8-bit counter produces 256 random test patterns and will apply to booth multiplier. Output coming from booth multiplier act as input to the accumulator. Accumulator which is present at the output of the architecture will indicate whether booth multiplier is faulty or not. Disadvantages of this circuit are higher power consumption and lack of self repairing circuit. Multiplexer present in the architecture will switch in between BIST mode and normal mode according to the select line.

### 3.3 An Efficient Test Pattern Generator in Built-in-Self-test Application

The BIST architecture shown in Figure 6 is having look up table as an extra module compared to the above discussed architecture. The look up table is used to store the true

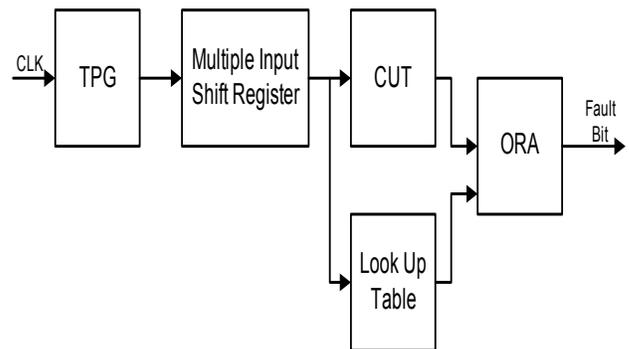


Figure 6. BIST architecture.

responses of multiplier. The output response analyser compares the real time response of the multiplier with true response and will produce corresponding true or false signal. Main disadvantage of this circuit is higher area due to the use of look up table.

## 4. Review of Existing Self Repairing Techniques

### 4.1 Triple Modular Redundancy

Triple-Modular Redundancy (TMR) shown in Figure 7 is used as a most common hardware masking technique in practical systems<sup>10</sup>. This is done by making three copies of the same circuit and connecting its outputs to a voter circuit. The specially designed voter circuit will transmit the majority output to the output. For example, if two or more inputs of the voter circuit is zero, then the output of the voter circuit will be zero. If the input of the voter circuit is defined as X, Y and Z and output of the circuit is defined as V, then the Boolean output function for voter circuit is:

$$V = XY + XZ + YZ \tag{3}$$

Voter circuit can be realised using three AND gates and two OR gates. One of the disadvantages of this circuit is that voter circuit can be faulty. In order to get rid of this problem fault tolerant voter circuit can be used.

### 4.2 Self Repairing Adder using Fault Localisation

Self repairing adder is an adder where real time detection of fault is possible<sup>10</sup>. The basic principles behind this self repairing adder are:

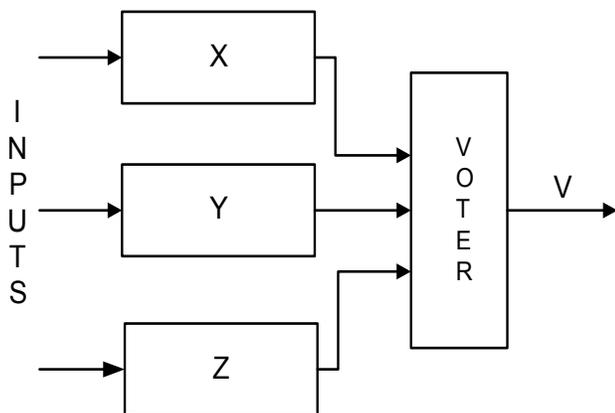


Figure 7. Triple modular redundancy with voter.

- Sum and carry bits of full adder are same when all the inputs are same
- Sum and carry bit is complimented when any of the input is different

By using the above property, a full adder can be self checked with the expense of an equivalence circuit as shown in the Figure 8.

According to the property which explained above boolean equation of equivalence tester is:

$$Eq_t = \overline{(\bar{A} * \bar{B} * \bar{Cin})} + (A * B * Cin) \tag{4}$$

Whenever all the inputs of full adder is same, G1 and G2 becomes logic 1 and whenever different input comes to the adder input G1 becomes logic 0 and G2 becomes logic 1. And all the other situations value of G2 becomes logic 0. For example, consider a stuck at 1 fault comes in sum bit of full adder. Then the output of G2 will become logic 0. That is, fault detected. Advantage of this circuit is the ability to find the location of the fault. Main disadvantage is that internal circuitry of adder is getting changed.

## 5. Proposed Self Repairing Techniques

### 5.1 Proposed BIST with Self Repairing Technique

In this research work, a new architecture is been proposed for self repairing of multipliers as shown in Figure 9

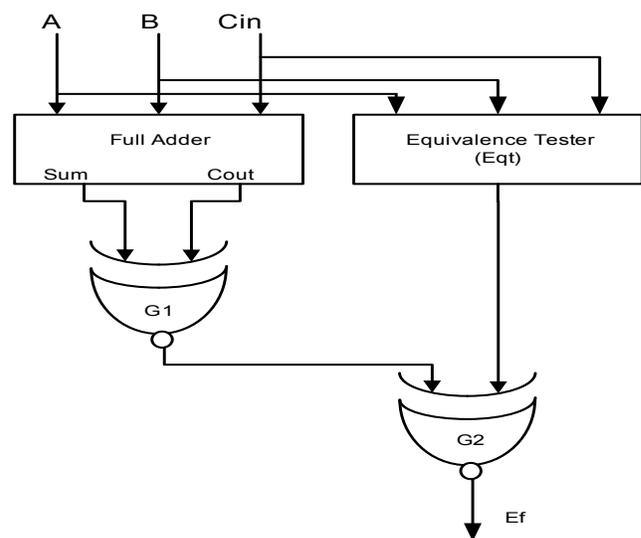
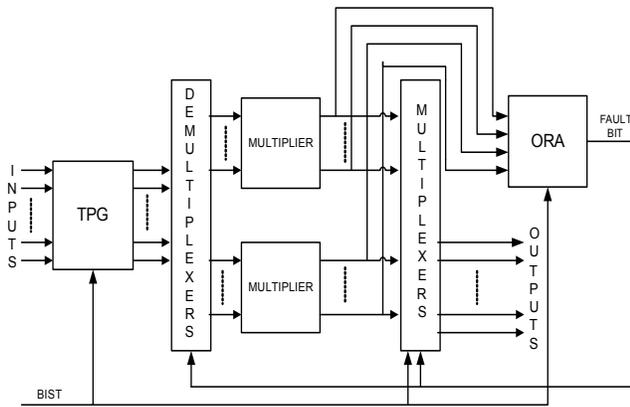


Figure 8. Self checking full adder.



**Figure 9.** Proposed BIST with self repairing architecture.

which is having less power and good fault coverage. The test pattern generator used here is shown in Figure 3 which has been designed<sup>9</sup>. Since hamming distance of the consecutive test vectors is low, power dissipation is also low. Random test patterns from TPG are connected to the input of multiplier. Output response of the multiplier is analysed by output response analyser and it will generate fault signal if error appears in the multiplier.

Repairing of digital circuit after finding fault is essential. For that, hardware redundancy technique is been used, where a copy of circuit under test is placed parallel to the normal multiplier. Whenever fault comes, circuit will change its path from faulty multiplier to fault free multiplier path by the use multiplexers. Compared to TMR based self repairing technique, this architecture consumes less power. It is needed to make BIST input as '1' for checking the circuit under test. Disadvantage of this technique is that real time detection of fault is not possible. Since the use of a power efficient TPG, power dissipation of the proposed architecture will also become low.

For example, consider the first multiplier is faulty. In order to check the faulty multiplier it is needed to make BIST input as logic '1'. So the comparator which is placed at the end of output response analyzer generates a fault signal. Repairing of the faulty multiplier is done by changing the faulty path to fault free path by the use of multiplexers. Demultiplexers are used to select outputs between two identical multipliers.

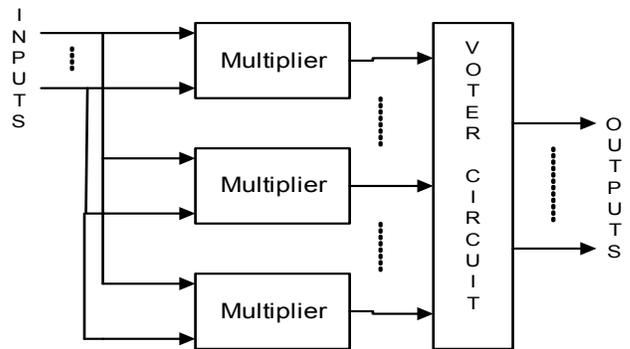
### 5.2 TMR based Self Repairing for Multiplier

In triple modular redundancy based self repairing, three identical copies of multiplier have to connect to voter

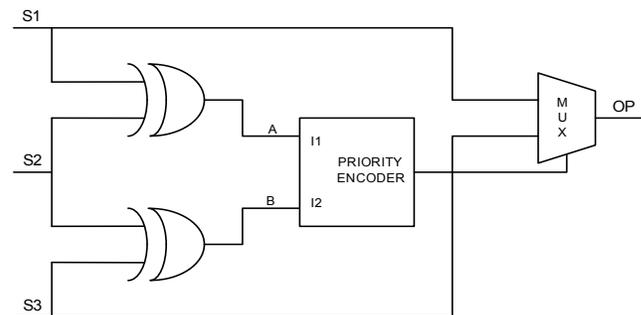
circuit. TMR based self repairing technique architecture is shown in Figure 10.

In this architecture, three copies of identical multiplier have been connected to voter circuit. Voter circuit will pass the majority output signal to the output. Real time checking of error is possible is the main advantage of this architecture. One disadvantage of this circuit is that error can be affected to voter circuit. In order to solve this problem a fault tolerant voter circuit has been designed and is shown in Figure 11.

The circuit shown in Figure 11 is the schematic diagram of fault tolerant voter circuit. It consists of a priority encoder multiplexer and a couple of XOR gates. Priority encoder is designed in such a way that the highest priority will be given to an input where logic '0' is present. If logic '0' is present in the wire A then zero will be the output of priority encoder and if logic '0' is present in the wire B then logic '1' will be the output of priority encoder. Consider a normal condition where S1, S2 and S3 are logic '1'. Then output of the voter circuit is expected to be logic '1'. And consider a stuck at 1 fault present in wire A of voter circuit. At this same time wire B will be zero since both S2 and S3 are zero. So output of the priority encoder will



**Figure 10.** TMR based self repairing multiplier architecture.



**Figure 11.** Fault tolerant voter circuit.

become logic '1'. That means output of the voter circuit is independent of the faulty path. A TMR based self repairing technique has been proposed here which uses the advantages of fault tolerant voter circuit. The architecture of the proposed design is shown in Figure 12.

There are three copies of multiplier and a fault tolerant voter circuit in the architecture shown in Figure 12. The proposed BIST with self repairing architecture requires more area compared to the self repairing multiplier with fault tolerant voter circuit represented in Figure 12. Real time repairing of faults is the main advantage of this method.

### 6. Results and Discussion

It can be inferred from Table 1 that the power consumption of PETPG is very less compared CTPG and it gives better fault coverage as well. The chart which is shown in Figure 13 is a pictorial representation of the results shown in Table 1. Since the use of power efficient ATPG in proposed BIST with self repairing model, power of the entire architecture is reduced

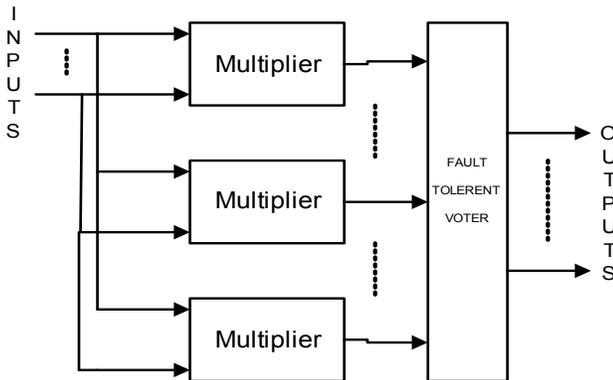


Figure 12. Self repairing multiplier with fault tolerant voter circuit.

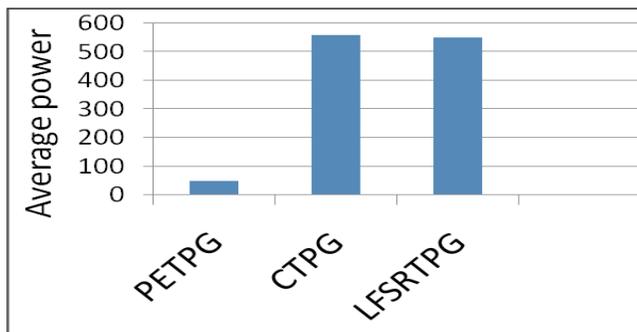


Figure 13. Average power dissipation of different test pattern generator.

Table 1. Average power dissipation of test pattern generators

ATPG	Average Power Dissipation	Total No. of faults assigned	Errors Detected	% of Fault Coverage
Power Efficient TPG (PETPG)	48.42 $\mu W$	24	23	96
Counter Based TPG (CTPG)	559.3 $\mu W$	24	22	92
LFSR Based TPG (LFSRTPG)	550 $\mu W$	24	22	92

From Table 2 it is clear that the power dissipation of the proposed BIST with self repairing technique is less than the other two models. So this architecture is suitable for low power applications. And if the application needs real time repairing, then TMR based self repairing technique is suitable. The internal structure of the CUT need not to be changed, low power dissipation and better fault coverage are the advantages of the proposed BIST with self repairing architecture. The chart which is shown in Figure 14 is a pictorial representation of the results shown in Table 2.

Table 2. Power and transistor count of different self repairing techniques

Self repairing techniques	Average power dissipation (mW)	Number of transistors
Built-in self test and self repairing of array multipliers	0.5393	6168
TMR based self repairing multiplier	0.8385	5172
Self repairing multiplier using fault tolerant voter circuit	3.540	5484

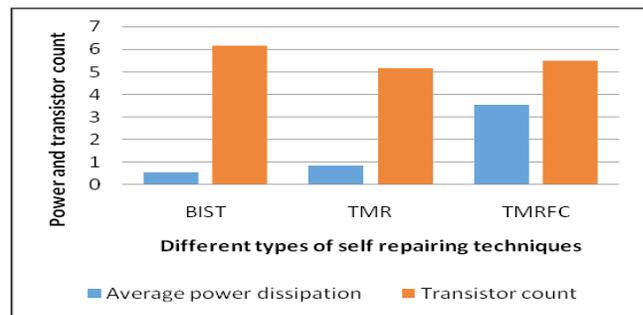


Figure 14. Power and transistor count of different self repairing techniques.

## 7. Conclusion and Future Work

In this paper, a power efficient BIST and self repairing circuit for array multipliers has been proposed. Power of the proposed architecture becomes low since the use of a power efficient test pattern generator. Self repairing has been accomplished by the use of hardware redundancy technique. Also a TMR based self repairing architecture for real time self repairing has been proposed and its area and power dissipation is compared with the other self repairing architecture. Result shows that the BIST with repairing technique is good for low power applications while the TMR based self repairing method is good for real time self repairing applications. The proposed technique can be extended to self repairing processor.

## 8. References

1. Mehdizadeh N, Shokrolah-Shirazi M, Miremadi SG. Analyzing fault effects in the 32-bit Open RISC 1200 microprocessor. Third international conference on availability, reliability and security; 2008 Mar 4-7; Barcelona. p. 648–52.
2. Meixner A, Bauer ME, Sorin DJ. Argus: low-cost, comprehensive error detection in simple cores. 40th Annual IEEE/ACM International Symposium on Microarchitecture; 2007 Dec 1–5; Chicago, IL. p. 210–22.
3. Nicolaidis M. On-line testing for VLSI: state of the art and trends. *Integrat VLSI J.* 1998 Dec; 26(1):197–209.
4. Psaiar H, Dustdar S. A survey on self-healing systems approaches and systems. *Computing.* 2011; 91(1):43–73.
5. Pellegrini A, Smolinski R, Chen L, Fu X, Hari SKS, Jiang J, et al. CrashTest'ing SWAT: accurate, gate-level evaluation of symptom-based resiliency solutions. *Proceedings of the Conference on Design, Automation and Test in Europe;* 2012. p. 1106–9.
6. Smith JE, Lam P. A theory of totally self-checking system design. *IEEE Trans Comput.* 1983 Sep; C-32(9):831–44.
7. K. Heragu, J. H. Patel, and V. D. Agrawal. Segment delay faults: a new fault model. *Proceedings 14th IEEE VLSI Test Symposium;* 1996 28 Apr–1 May; Princeton, NJ. p. 32–9.
8. Gizopoulos D, Paschalis A. Effective built-in self-test for Booth multipliers. *IEEE Design Test Comput.* 1999 Jul-Sep; 15(3):105–11.
9. 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.
10. Kshirsagar RV, Patrikar RM. Design of a novel fault tolerant voter circuit for TMR implementation to improve reliability in digital circuits. *Microelectronics Reliability.* 2009 Dec; 49(12):1573–77.