Low Power BIST for Wallace Tree-based Fast Multipliers

D. Bakalis[1,2], E. Kalligeros[1], D. Nikolos[1,2], H. T. Vergos[1] and G. Alexiou[1,2]

[1]*Dept. of Computer Engineering and Informatics, University of Patras, 26 500, Rio, Greece*
[2]*Computer Technology Institute, 3, Kolokotroni Str., 262 61 Patras, Greece*

e-mail : bakalis@cti.gr, kalliger@ceid.upatras.gr, {nikolosd, vergos, alexiou}@cti.gr

Abstract

The low power as a feature of a BIST scheme is a significant target due to quality as well as cost related issues. In this paper we examine the testability of multipliers based on Booth encoding and Wallace tree summation of the partial products and we present a methodology for deriving a low power Built In Self Test (BIST) scheme for them. We propose several design rules for designing the Wallace tree in order to be fully testable under the cell fault model. The proposed low power BIST scheme for the derived multipliers is achieved by : (a) introducing suitable Test Pattern Generators (TPG), (b) properly assigning the TPG outputs to the multiplier inputs and (c) significantly reducing the test set length with respect to earlier schemes. Our results indicate that the total power dissipated during test can be reduced from 64.8% to 72.8%, while the average power per test vector can be reduced from 19.6% to 27.4% and the peak power dissipation can be reduced from 16.8% to 36.0%, depending on the implementation of the basic cells and the size of the multiplier. The test application time is also significantly reduced, while the introduced BIST scheme implementation area is small.

1. Introduction

In complex ICs, low controllability and observability of embedded blocks impose serious testability problems. In order for the whole chip to become a viable product such blocks must be well tested. Built In Self Test (BIST) structures are well suited for testing such embedded blocks, since they can cut down the cost of testing by eliminating the need of external testing as well as apply the test vectors at speed. Multipliers are commonly used as embedded blocks in both general purpose datapath structures and specialized digital signal processors.

Traditionally, high fault coverage, small area overhead and small application time have been the objectives of BIST designers. While these objectives still remain important, a new BIST design objective, namely low power dissipation during test application, is emerging [1 - 5], and is expected to become one of the major objectives in the near future [6].

There are quality as well as cost related issues that make the power dissipated during test application an important factor :

a) Reliability. Although there is a significant correlation between consecutive vectors applied to a circuit during its normal operation, the correlation between consecutive test vectors is significantly lower. Therefore the switching activity in the circuit can be significantly higher during testing than that during its normal operation [2, 9]. Since heat and power dissipation in CMOS circuits is proportional to switching activity, the latter may cause a circuit under test to be permanently damaged due to excessive heat dissipation or give rise to metal migration (electro-migration) that causes the erosion of conductors and leads to subsequent failure of circuits [7]. This is even more severe in circuits equipped with BIST since such circuits are tested frequently in the field.

b) Technology. The multi-chip module (MCM) technology that is becoming highly popular requires sophisticated probing to bare dies for fully testing them [8]. Absence of packaging of these bare dies precludes the traditional heat removal techniques. In such cases, power dissipated during testing can adversely affect the overall yield, increasing the production cost.

c) Cost. Consumer electronic products typically require a plastic package which imposes a strong limit on the energy dissipated. Excessive dissipation during testing may also prevent periodic testing of battery operated systems that use an on-line testing strategy.

Several research has been carried out in the past on the reduction of the power dissipated during test. In [9] a modified PODEM was presented which derives a test set with reduced switching activity between consecutive test vectors. In [2], a TPG for reducing switching activity has been presented based on the use of two LFSRs operating at different speeds. [3] describes a method for synthesizing a counter in order to reproduce on chip a set

of pre-computed test patterns, derived for hard to detect faults, so that the total heat dissipation is minimized. However, a test set targeting the hard to detect faults of a circuit C has some characteristics not available to a test set targeting all faults of C. In a BIST scheme some vectors generated by the TPG circuit are not useful for testing purposes. A technique that inhibits such consecutive test vectors, by the use of a three-state buffer and the associated control logic was proposed in [5]. The drawbacks of this method are that it fails to reduce test application time and suffers from high implementation cost. In [10] a programmable low power ATPG is proposed, implemented by linear cellular automata with external weighting logic, by determining the optimal signal probabilities and activities. The drawback of this method is also the high implementation cost.

The above mentioned techniques try to solve the general problem. However there are cases where exploiting the inherent properties of a class of circuits leads to a more efficient low power BIST scheme. Effective low power BIST schemes for both Carry Save Array Multipliers and Modified Booth Multipliers have recently been proposed in [4, 11].

Wallace tree summation along with Booth encoding are the most common techniques for designing fast multiplier blocks. Booth encoding aims to reduce the number of partial products whereas Wallace tree summation and carry look-ahead (CLA) addition in the final stage of the multiplier aims at the fastest addition of the partial products. A BIST scheme for such multipliers has recently been proposed in [12]. This BIST scheme does not take the low power dissipation objective into account. We will use this BIST scheme as the basis for our comparisons.

In this paper we will first introduce several rules for designing a Wallace tree which is fully testable under the Cell Fault Model [13] when it receives the test vectors produced by an 8-bit binary counter. The cell fault model is also used for all other modules except the CLA where single stuck at faults are considered. Next, starting from the basic BIST [12] we will describe a methodology that leads to a new BIST targetting low power dissipation during test. Our methodology is based on (a) suitably modifying the original TPG, (b) properly assigning the TPG outputs to the multiplier inputs and (c) significantly reducing the test set length.

2. Easily Testable Fast Multipliers

We consider nxn multipliers with inputs A ($A_{n-1}...A_0$) and B ($B_{n-1}...B_0$). They consist of three units :

1. The Booth Encoding Unit for the multiplier encoding and the partial products formation (we assume that input B is used for the Booth encoding)

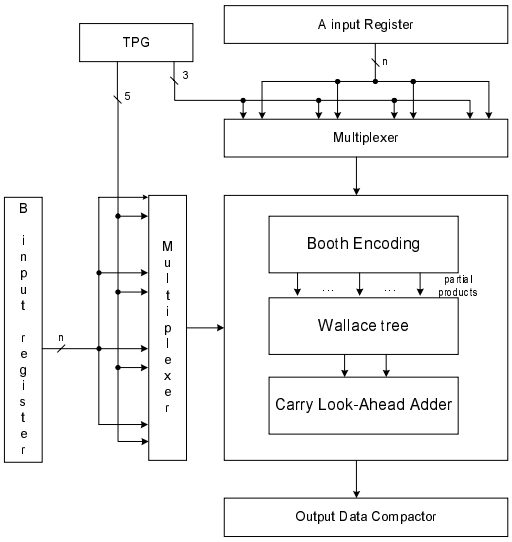

Figure 1: The BIST scheme

2. The Wallace Tree Unit which sums the partial products and produces the sum and carry vector and
3. The Carry Look-ahead Adder (CLA) Unit that produces the final result.

The TPG proposed in [12] for such multipliers (see Figure 1) consists of an 8-bit counter whose 3 outputs are repeatedly used to feed the A input of the multiplier while the remaining 5 bits are repeatedly used to form the input B of the multiplier. Multiplexers are used to select between normal inputs and BIST inputs.

The design of the Wallace Tree Unit can be done in several ways (using 3-2 compressors, 4-2 compressors, etc). The authors of [12] claim that the 256 vectors of their proposed TPG are capable of providing all possible input combinations to the inputs of every full or half adder cell. Since they do not propose a specific method for designing the Wallace Tree Unit we could conclude that the above is valid no matter which structure is used for the Wallace tree. However this conclusion is not correct. In [14] we have indicated several Wallace tree structures for which this does not hold. In other words, the Wallace tree structure must follow certain design rules in order for its cells to receive all possible input combinations. These design rules are :

i) The partial product bits (PP bits) are grouped in triplets and summed at the first level of each Wallace tree. If the number of PP bits modulo 3 is non zero then the remaining PP bits are summed at next levels of the Wallace tree along with carry bits.

ii) If a carry occurs at the i-1 level of a certain Wallace tree, then this carry should be inserted at a level k adder of the succeeding most significant tree, such that $k \geq i$.

iii) Every Wallace tree has at most one half adder which either resides at the last or the previous to the last level of the tree.

iv) The sign extension bits are summed either at the last or the previous to the last level of each tree. In the latter case a half adder should not be used.

v) Carry bits that are the outcome of trees which sum a lot of carry bits of less significance should be propagated at the highest possible level of the succeeding tree and if possible added with the outcome of subtrees that receive only a very small number of carry bits.

We have verified the validity of these design rules by constructing various multipliers (with operand sizes of n = 8, 12, 16, 24, 32). Their Verilog descriptions can be found in [14].

After the description of the way fast multipliers can be designed to be easily testable under the 256 vectors produced by the basic BIST [12], we will focus in the next section on the production of a new BIST scheme taking also the low power objective into account.

3. Low Power Dissipation during Testing

3.1. Preliminaries

The primary source of power dissipation in CMOS circuits is due to charging and discharging the capacitances. It is expected that by reducing the number of transitions at the primary inputs of a circuit, the total number of transitions at internal lines will also be reduced, leading to lower power dissipation. However, depending on the circuit structure, the transitions at some primary inputs may cause more transitions at internal lines than those at other primary inputs.

In [2, 3] a procedure has been presented for identifying those primary inputs that cause more transitions at internal lines. Let $f(l)$ denote the function of line l, and $\frac{\partial f(l)}{\partial in_i}$ the Boolean difference of $f(l)$ with respect to primary input in_i. The latter function indicates whether $f(l)$ is sensitive to changes of input in_i. Let $P(\frac{\partial f(l)}{\partial in_i})$ denote the probability that function $\frac{\partial f(l)}{\partial in_i}$ evaluates to 1. The power dissipation is then estimated as:

$$P = (1/2)V_{dd}^2 \sum_l C_l \sum_i P(\frac{\partial f(l)}{\partial in_i}) T(in_i) \quad (1),$$

with C_l denoting the capacitance of line l, V_{dd} the power supply voltage and $T(in_i)$ the number of transitions of the primary input in_i. Therefore, if we assign a weight w to every primary input in_i such that $w(in_i) = \sum_l C_l P(\frac{\partial f(l)}{\partial in_i})$, then weights are a good metric of how many lines of the circuit, weighted by the associated capacitance, are affected by primary input in_i.

Relation (1) implies that the power dissipation can be reduced by cutting down the number of transitions at the inputs of the circuit. The reduction is larger when the number of transitions at the inputs with greater weights is reduced. Therefore, the assignment of the TPG outputs to the circuit inputs is significant. The reduction of the cardinality of the test set will also reduce the total number of transitions and thus the power dissipation.

3.2. Assignment of the TPG outputs to the multiplier inputs

In this subsection, we address the problem of properly assigning the TPG outputs to the multiplier inputs for achieving low power dissipation. The error aliasing calculation of the Output Data Compactor (ODC) circuit and the estimation of the power dissipation during testing enforce us to take into account specific cell implementations. Since we consider the Cell Fault Model, more than one cell implementation must be taken into account. Specifically three distinct implementations of the half and full adder cells, presented respectively in [15, 7, 16] are considered. We will refer to these implementations as Cell 1, Cell 2 and Cell 3 respectively. The implementations considered for the Booth encoding logic were those presented in [17].

We have computed the primary inputs weights for multipliers of various sizes for each of the possible cells and we have verified that their distribution is independent of the specific full and half adder cells. Comparing any possible pair of inputs, the one with the larger weight contributes more than the other to the power dissipation. Since the sum of weights of B inputs is greater than the sum of weights of A inputs, the 5 most significant outputs of the TPG should drive the B inputs while its 3 least significant outputs should drive the A inputs.

Since the outputs of the TPG are repeatedly assigned to both A and B multiplier inputs, in order to assign them to specific inputs, we sum the weights of the inputs that receive the same TPG output bit. The results for the sum of weights for nxn multipliers with n = 8, 16 or 32 are listed in Table 1. For maximum reduction of the power dissipation, the signals with the least number of transitions should be assigned to the inputs with the largest sum of weights. This assignment is presented as "Best Assignment" in Table 2, along with the savings in power dissipation over a "Random Assignment".

Table 1. Sum of weights of the multiplier inputs

	Sum of weights for input B					Sum of weights for input A		
	$\sum_{i=0}^{n/5} w(b_{(5i+4)})$	$\sum_{i=0}^{n/5} w(b_{(5i+3)})$	$\sum_{i=0}^{n/5} w(b_{(5i+2)})$	$\sum_{i=0}^{n/5} w(b_{(5i+1)})$	$\sum_{i=0}^{n/5} w(a_{(5i+0)})$	$\sum_{i=0}^{n/3} w(a_{(3i+2)})$	$\sum_{i=0}^{n/3} w(a_{(3i+1)})$	$\sum_{i=0}^{n/3} w(a_{3i+0)})$
	8x8 multiplier							
Cell 1	88	115	167	183	192	172	244	241
Cell 2	129	159	234	251	268	243	337	332
Cell 3	116	151	219	235	247	227	315	310
	16x16 multiplier							
Cell 1	757	821	781	844	1002	1211	1196	1395
Cell 2	1222	1307	1253	1354	1621	1947	1910	2210
Cell 3	1096	1190	1137	1226	1455	1755	1731	2002
	32x32 multiplier							
Cell 1	4372	4463	4499	5011	5019	6851	7375	7389
Cell 2	7614	7772	7857	8722	8755	11901	12752	12822
Cell 3	6800	6960	7019	7785	7791	10629	11400	11415

Depending on the specific cell implementation and the size of the multiplier "Best Assignment" can lead to power dissipation savings from 2.0 up to 13.1%. For obtaining the results of Table 2, we used the zero-delay gate level power simulator developed in [4]. The power simulator estimates the power dissipation of the whole circuit consisting of both the multiplier and the BIST circuitry.

Using as TPG a Gray instead of a binary counter we can further reduce the number of transitions at the primary inputs of the multiplier. Therefore we decided to encode in Gray both the 3 and 5 output bits of the 8-bit binary counter that drive A and B multiplier inputs respectively. Table 2 lists the savings in power dissipation that can be accomplished by using both "Best Assignment" as well as a Gray counter TPG. As can be observed by Table 2 the use of Gray code can increase the power dissipation savings by approximately 20% in all cases.

Table 2. Total Power Dissipation reduction percentages using best assignment and Gray encoding

	8x8			16x16			32x32		
	Cell1	**Cell2**	**Cell3**	**Cell1**	**Cell2**	**Cell3**	**Cell1**	**Cell2**	**Cell3**
Best Assignment	13.1	12.7	12.3	8.1	7.8	7.5	2.4	2.0	2.1
Best Assignment + Gray Encoding	30.5	29.7	29.0	28.7	27.0	26.5	24.7	22.0	22.0

3.3. Test length reduction

It is well known that in BIST schemes some vectors generated by the TPG circuits are not useful for testing purposes. Therefore another way for reducing the power dissipated during the test application is to reduce the number of vectors applied to the circuit under test.

A straightforward approach to this problem is to use a fault simulator along with a test set compression program. The result that would be obtained by this method would be an optimal test set in terms of its cardinality but totally inappropriate for implementation as a TPG circuit. This is because the vectors that would be selected would have no straight correlation between them. So for their generation a specially designed circuit would be required (in most cases this would be a Finite State Machine). Such circuits apart from requiring large implementation area may also destroy any possible power dissipation gains attained by the elimination of the original redundant test vectors. Another approach, that is not so straightforward, is to modify the original TPG so as to only go through some of its states. This solution does not guarantee to generate an optimal number of test vectors but as long as the original state sequence is not disrupted a lot it can be implemented in approximately the same area as the original BIST TPG.

We will follow the second approach. First we will try to reduce the number of test vectors required for exhaustively testing the basic cells of the multiplier (Booth encoding cells and Wallace tree full and half adders) except for the CLA at the last stage. By using a simple simulator, we constructed a table with 256 rows and X columns, with X indicating the number of cells used in the multiplier. The content of each cell of the table indicates the specific input combination that the specific test vector applies to the corresponding cell of our design. The "Best Assignment" and the Gray encoding of the TPG define the order of the test vectors (rows in the table). Another ordering of the test vectors would lead to a different result.

The next goal is to find a subset of rows in this table, able to apply all the input combinations at every cell. Since in our case the original TPG is a counter, this means that we must select large subsets of consecutive rows. We start by dividing the 256 rows into 8 groups of 32 consecutive vectors each. The idea behind this selection is that if a whole group of 32 vectors can be omitted, then this corresponds to the skipping of 4 consecutive states of

the 5-bit counter driving input B of the multiplier. Therefore it can be easily implemented without increasing substantially the area of the TPG circuit. Then, the same procedure is executed for groups of 16 or 8 vectors each. We stop at groups of 8 vectors since smaller groups would require a significant increase of the implementation area of the TPG because this will destroy the Gray encoding on the 3 least significant bits of the TPG. The above analysis leads to an area of the TPG vs power dissipation reduction tradeoff. For example for the 16x16 multiplier one can make a selection between the removal of 1 group of 32 vectors or the removal of 5 groups of 16 vectors (see Table 3). The first solution will result in a smaller area overhead but will also lead to a smaller reduction in the power dissipation.

Table 3. Number of redundant groups vs group size for 8x8, 16x16 and 32x32 multipliers

Group Size	**Number of groups**	**Number of redundant groups**		
		8x8	**16x16**	**32x32**
32	8	0	1	0
16	16	6	5	6
8	32	20	20	18

Among the various possibilities, we found that the removal of groups of 8 consecutive vectors lead to good results within a reasonable area increase. For the 16x16 and 32x32 multipliers there are respectively 20 and 18 redundant groups of 8 vectors [14] while the additional area overhead of the TPG circuit is negligible (1.0% in 16x16 and 0.3% in 32x32 multiplier). Therefore the test set cardinality can be reduced from the original 256 to respectively 96 and 112.

Although the reduced test sets guarantee exhaustive testing of the basic cells, the fault coverage may drop due to undetected faults of the CLA unit for which we consider the single stuck-at model and the increased error aliasing. Therefore, we need to verify that the fault coverage attained by the reduced test sets, with respect to single stuck-at faults, remains at high levels. Table 4 lists the fault coverage achieved, assuming either Rotate Carry Adder or Cascaded Compaction [18] as the ODC, for 8x8 or 16x16 multipliers and the three cell implementations.

From Table 4 we can see that there is a slight increase in the number of undetectable faults with the proposed test set compared to the application of the 256 initial vectors. These are located at the CLA unit. Moreover, we can see that due to increased error aliasing the fault coverage may drop below the acceptable level of 99% when a Rotate Carry Adder is used as ODC. The use of Cascaded Compaction solves the aliasing problem in all cases.

Table 4. Fault Coverage Percentage

	8x8			**16x16**		
	Cell1	***Cell2***	***Cell3***	***Cell1***	***Cell2***	***Cell3***
	Test set with 256 vectors					
UF	99.82	99.84	99.84	99.45	99.55	99.55
UF + ALS-RCA	98.72	98.89	98.89	98.99	99.18	99.18
UF + ALS-CC	99.36	99.44	99.44	99.26	99.40	99.40
	Test set with 96 vectors for 8x8 and 16x16					
UF	99.73	99.76	99.76	99.20	99.35	99.35
UF + ALS-RCA	98.81	98.97	98.97	98.71	98.97	98.95
UF + ALS-CC	99.31	99.40	99.40	99.16	99.30	99.33
	Test set with 168 vectors for 8x8, 16x16 and 32x32					
UF	99.54	99.60	99.60	99.30	99.43	99.43
UF + ALS-RCA	98.67	98.89	98.89	98.99	99.18	99.18
UF + ALS-CC	99.45	99.52	99.52	99.24	99.38	99.38

UF = Fault Coverage measuring only undetected faults
UF + ALS-RCA = Fault Coverage measuring undetected faults and aliasing with Rotate Carry Adder
UF + ALS-CC = Fault Coverage measuring undetected faults and aliasing with Cascaded Compaction

One may also want to have a uniform test set for all examined nxn multipliers (n = 8, 16 or 32) generated by a uniform TPG. We derived such a test set consisting of 168 test vectors [14]. The fault coverage attained by this test set is presented also in Table 4 and is in all cases above the acceptable level of 99% when Cascaded Compaction is used. In 32x32 multipliers, the Rotate Carry Adder provides always fault coverage greater than 99%.

Table 5 presents the savings in total, average per vector and peak power dissipation when the reduced test sets along with "Best Assignment" and Gray encoding are used. The total power reduction achieved varies from 64.8% to 72.8%. The average power dissipation reduction per vector varies from 19.6% to 27.4%, while the reduction of the peak power dissipation varies from 16.8% to 36.0%. The test application time is also reduced from 56.3% to 62.5%.

Table 5. Power Dissipation reduction percentages using best assignment, Gray encoding and test set reduction

Multiplier (Number of vectors)		**Total power reduction (%)**	**Average power reduction per vector (%)**	**Peak power reduction (%)**
8x8 (96)	Cell 1	72.8	27.4	17.5
	Cell 2	72.4	26.3	24.8
	Cell 3	72.1	25.5	16.8
16x16 (96)	Cell 1	72.6	26.8	24.1
	Cell 2	72.0	25.3	27.1
	Cell 3	71.7	24.4	22.4
32x32 (112)	Cell 1	66.1	22.5	31.0
	Cell 2	64.9	19.8	36.0
	Cell 3	64.8	19.6	30.2

To obtain the above comparison results, our gate – level simulator assumes a zero gate delay. The reductions in the total power dissipated are expected to be even greater if glitches were also taken into account, since the switching activity of the internal nodes of the multiplier is reduced during the application of the reduced test set by the proposed BIST.

Table 6 presents the savings in power dissipation that can be obtained by the uniform test set when combined with "Best Assignment" and Gray encoding. Comparing Tables 5 and 6 we can see that average power per vector and peak power savings are similar in both cases. We can also observe that the total power dissipation savings in this case are smaller but the uniform test set is still able to reduce the total power dissipation by half with respect to the original BIST scheme [12].

Table 6. Power Dissipation reduction percentages using best assignment, Gray encoding and uniform test set of 168 vectors

Multiplier		**Total power reduction (%)**	**Average power reduction per vector (%)**	**Peak power reduction (%)**
8x8	Cell 1	52.5	27.6	24.0
	Cell 2	51.9	26.6	25.4
	Cell 3	51.2	25.7	21.1
16x16	Cell 1	51.7	26.4	24.1
	Cell 2	50.6	24.7	27.1
	Cell 3	50.1	23.9	22.4
32x32	Cell 1	48.9	22.2	28.4
	Cell 2	47.1	19.4	34.3
	Cell 3	47.0	19.2	27.5

4. Conclusions

Fast multipliers are used as embedded blocks in both general purpose datapath structures and specialized digital signal processors. In this paper we analyzed their testability characteristics and proposed several design rules in order to make their Wallace tree summation unit fully testable under the Cell Fault Model. As quality and cost related issues make the low power feature a necessity for BIST schemes, we presented a methodology for deriving a novel low power BIST scheme. We showed how the low power objective can be achieved by: a) proper assignment of the TPG outputs to the multiplier inputs, b) the use of Gray code and c) reducing significantly the test set. Our methodology is able to provide the BIST designer with alternate solutions, each having distinct power reduction, implementation area and fault coverage characteristics.

5. References

[1] Y. Zorian, "A Distributed BIST control scheme for Complex VLSI devices", VLSI Test Symp pp 4-9, 1993.

[2] S. Wang & S. K. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation", Proc. of Int. Test Conference, pp. 848-857, 1997.

[3] X. Kavousianos, D. Nikolos, S. Tragoudas, "On-Chip Deterministic Counter-Based TPG with Low Heat Dissipation", Proc. of Southwest Symposium on Mixed-Signal Design, pp. 87 - 92, 1999.

[4] D. Bakalis & D. Nikolos, "On Low Power BIST for Carry Save Array Multipliers", Proc. of 5th IEEE International On-Line Testing Workshop (IOLTW '99), pp. 86-90, 1999.

[5] P. Girard et. al., "A Test Vector Inhibiting Technique for Low Energy BIST Design", Proc. of VLSI Test Symposium, pp. 407-412, 1999.

[6] R. Roy, "Scaling Towards Nanometer Technologies: Design for Test Challenges", Panel, Design Automation and Test in Europe, 1999.

[7] N. H. E. Weste and K. Eshragian, *Principles of CMOS VLSI Design: A Systems Perspective*, Addison-Wesley, 2nd edition, 1992.

[8] R. Parkar, "Bare Die Test", Proc. of IEEE Multi-Chip Module Conference, pp. 24 – 27, 1992.

[9] S. Wang & S. K. Gupta, "ATPG for Heat Dissipation Minimization During Test Application", Proc. of International Test Conference (ITC '94), pp. 250-258, 1994.

[10] X. Zhang & K. Roy, "Design and Synthesis of Programmable Low Power Weighted Random Pattern Generator", Proc. of 5th IEEE International On-Line Testing Workshop (IOLTW '99), pp.82-85, 1999.

[11] D. Bakalis, et. al., "Low Power Dissipation in BIST Schemes for Modified Booth Multipliers", Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT '99), pg. 121-129, 1999.

[12] A. Paschalis et. al., "An Effective BIST Architecture for Fast Multiplier Cores", Proc. of Design, Automation and Test in Europe (DATE '99), pp. 117 – 121, 1999.

[13] W. Kautz, "Testing for faults in cellular logic arrays", Proc. of the 8th Annual Symposium on Switching and Automata Theory, pp. 161-174, 1967.

[14] D. Bakalis et. al., "Low Power BIST for Wallace Tree-based Fast Multipliers", Computer Technology Institute Technical Report (TR99/09/07), Patras, Greece, 1999.

[15] M. Morris Mano, *Digital Design*, Prentice – Hall 1991, 2nd edition.

[16] M. Abramovici, M. Breuer, A. Friedman, *Digital Systems Testing and Testable Design*, Computer Science Press, 1990.

[17] E. Kalligeros et. al., "Path Delay Fault Testable Modified Booth Multipliers", Proc. of XIV Design of Circuits and Integrated Systems Conference (DCIS '99), pp.301-306, 1999.

[18] J. Rajski, J. Tyszer, *Arithmetic Built-In Self-Test for Embedded Systems*, Prentice-Hall, 1998.