Low Power Built-In Self-Test Schemes for Array and Booth Multipliers*

D. BAKALIS[†], X. KAVOUSIANOS[‡], H. T. VERGOS[¶], D. NIKOLOS[§]
and G. Ph. ALEXIOU[#]

Department of Computer Engineering and Informatics, University of Patras, 26 500, Patras, Greece

Recent trends in IC technology have given rise to a new requirement, that of low power dissipation during testing, that Built-In Self-Test (BIST) structures must target along with the traditional requirements. To this end, by exploiting the inherent properties of Carry Save, Carry Propagate and modified Booth multipliers, in this paper we propose new power-efficient BIST structures for them. The proposed BIST schemes are derived by: (a) properly assigning the Test Pattern Generator (TPG) outputs to the multiplier inputs, (b) modifying the TPG circuits and (c) reducing the test set length. Our results indicate that the total power dissipated during testing can be reduced from 29.3% to 54.9%, while the average power per test vector applied can be reduced from 5.8% to 36.5% and the peak power dissipation can be reduced from 15.5% to 50.2% 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 schemes implementation area is small.

I. INTRODUCTION

The ever-increasing trend towards denser and faster ICs has resulted in embedded logic blocks with low controllability and observability that need to be tested at speed in order for the whole chip to become a viable product. Built-In Self-Test (BIST) structures are well suited for testing such

---

blocks, since they can cut down the cost of testing by eliminating the need for external testing equipment as well as by applying the test vectors at the desired operating speed.

The main objectives of BIST designers have traditionally been high fault coverage, small area overhead and small application time. While these objectives still remain important, a new objective, namely low power dissipation during test application, has recently emerged [1 – 6], and is expected to become one of the major objectives in the near future [7].

There are cost, reliability as well as technology related issues motivating for power and heat dissipation minimization during test application:

Cost Issues: Consumer electronic ICs typically require a plastic package. This imposes a strong limit on the energy that can be dissipated. Excessive dissipation during testing may also prevent periodic testing of battery operated systems that use an on-line testing strategy.

Reliability Issues: 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 [3 – 4]. 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 [8].

Technology Related Issues: The multi-chip module (MCM) technology which is becoming highly popular requires sophisticated probing to bare dies for fully testing them [9]. Absence of packaging of these bare dies precludes the traditional heat removal techniques. In such cases, the power dissipated during testing can adversely affect the overall yield, increasing the production cost.

A more detailed presentation of the motivations for low power dissipation during test application can be found in [3, 10].

The authors of [10] present a modification of the well-known PODEM algorithm for deriving a test set with reduced switching activity between consecutive test vectors. A BIST technique for reducing the switching activity has been presented in [4]. The technique is based on the use of two LFSR TPGs operating at different speeds. In [5] a counter synthesis method is presented that reproduces on-chip a set of pre-computed test patterns, derived for hard to detect faults, so as the total heat dissipation to be minimized. However, a test set targeting the hard to detect faults of a circuit has some characteristics not common to the test set targeting all faults of the circuit. Therefore, this method can not be used for the whole test set of the circuit. In a BIST scheme some vectors generated by the TPG circuit are not useful fo testing purposes. A technique that inhibits such consecutive vectors, generated by an LFSR, from being applied to the circuit under test was proposed in [6]. The technique uses a three state buffer and the associated control logic. Its drawbacks are that it fails to reduce the application time and suffers from high implementation cost.

The above-mentioned solutions target the general problem. However, there are cases where exploiting the inherent properties of a class of circuits may lead to the design of very efficient low power BIST schemes. Multiplier circuits belong to that circuits. Moreover, multipliers are commonly used as embedded blocks in both general purpose data-path structures and specialized digital signal processors.

In this work we address the problem of designing low power BIST schemes for Carry Propagate Array Multipliers (CPAMs), Carry Save Array Multipliers (CSAMs) and sign generate Modified Booth Multipliers (MBMs), that use carry save arrays of adders to add the partial products. MBMs with the final result forming adder implemented as: (a) ripple carry or (b) a group carry look ahead with ripple carry between groups, are considered and the notation RC-MBMs and CL-MBMs for the (a) and (b) case will be used respectively.

For the above circuits we adopt the cell fault model [11] which is considered to be more general than the single stuck-at fault model. Based on the cell fault model, only one cell of the circuit can be faulty at a time and only faults that do not change the combinational behavior of the circuit can occur. In the case of CL-MBMs we use the cell fault model for all cells of the multiplier except the final result forming carry look-ahead adder where single stuck-at faults are considered.

The paper is organized as follows: Preliminaries with respect to low power dissipation, array multipliers and modified-Booth multipliers are given respectively in Sections II.1, II.2 and II.3. The assignment of the Test Pattern Generator (TPG) outputs to the multiplier inputs in order to achieve power dissipation reduction during testing is addressed in Section III. In Section IV we introduce a new Test Pattern Generator for each multiplier structure that reduces the test application time by not applying test vectors that are not useful for testing purposes and therefore reduces the total power dissipation during testing. In Section V we discuss the power dissipation and testing characteristics of the each proposed BIST scheme as well as its area overhead.

II. PRELIMINARIES

II.1. Low Power Dissipation

Charging and discharging of capacitance is the dominant factor of power dissipation in full static CMOS circuits [12], the dominant today's technology. It has been reported (p. 8 of [12]) that in high frequency CMOS circuits this accounts for at least 90% of the total power dissipation.

Denoting the power supply voltage by V_{dd}, the load capacitance at line l by C_l, and the total number of transitions at line l by $T(l)$, the above mentioned power dissipation component (denoted by PD) is given by the relation:

$$PD = \frac{1}{2} V_{dd}^2 \sum_l C_l T(l) \qquad (1)$$

It is evident that the power dissipation can be reduced by reducing $T(l)$. By reducing the number of transitions at the primary inputs of the circuit, it is expected that the total number of transitions at the lines of the circuit will also be reduced leading to lower power dissipation. However, depending on the circuit structure, the transitions at some primary inputs cause more transitions at internal lines than those caused by transitions at other primary inputs. A procedure has been presented in [4–5] to identify among the primary inputs of the circuit those that cause more transitions at internal lines. Let $f(l)$ denote the function of line l, and $(\partial f(l)/\partial in_i)$ denote the Boolean difference of $f(l)$ with respect to input in_i. This Boolean function indicates whether $f(l)$ is sensitive to changes of input in_i. Let $f(l)_{in_i}$ (respectively $f(l)_{in_i'}$) denote the cofactor of $f(l)$ with respect to input variable in_i (respectively in_i') and $\oplus$ be the XOR operator. The Boolean difference is precisely:

$$\frac{\partial f(l)}{\partial in_i} = f(l)_{in_i} \oplus f(l)_{in_i'} \qquad (2)$$

Let $P(\partial f(l)/\partial in_i)$ denote the probability that function $(\partial f(l)/\partial in_i)$ evaluates to 1. The power dissipation is then estimated as:

$$PD = \frac{1}{2} V_{dd}^2 \sum_l C_l \sum_i P\left(\frac{\partial f(l)}{\partial in_i}\right) T(in_i) \qquad (3)$$

Equation (3) shows that the total power dissipation of a circuit can be reduced by reducing the total number of transitions on inputs. Once the probability $P(\partial f(l)/\partial in_i)$ is computed, each input in_i is assigned a weight:

$$w(in_i) = \sum_l C_l P\left(\frac{\partial f(l)}{\partial in_i}\right) \qquad (4)$$

Weights $w(in_i)$ are a good metric of how many lines of the circuit, weighted by the associated capacitance, are affected by input in_i.

Relation (3) implies that 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 very significant. To this end, we address the subject of correct assignment in Section II. Also the reduction of the cardinality of the test set will reduce the number of transitions and thus the power dissipation. We investigate the reduction of the cardinality of test sets, without affecting their fault coverage, in Section IV.

II.2. Array Multipliers and Built-In Self-Test

Parallel multiplication implies the simultaneous generation of all partial products which have to be added together for getting the final result. When carry-propagate adders are used to perform the required additions then the multiplier is called a CPAM. One way of speeding up the multiplication is based on the addition of the partial products by using carry save-adders. When addition is performed using an array of carry-save adders, the multiplier is called a CSAM. The building cells of a CPAM or a CSAM are the 2-input AND gate, the Full Adder (FA) and the Half Adder (HA). The AND gates are used to derive the partial products bits whereas the FAs and HAs are used for the addition of the partial products.

The designs of 8×8 CPAM and CSAM circuits are respectively presented in Figures 1 and 2. The input line $pp_{i,j}$ represents the output of an AND gate that receives as inputs the bits x_i and y_j of the multiplier and multiplicand inputs respectively.

The testability of the CPAMs and CSAMs, under the cell fault model [11], was investigated in [13–14]. Recently, a BIST scheme for them was proposed [15] taking into account the cell fault model. It was considered that only one cell can be faulty at a time and that only combinational faults can happen. In [15], the Test Pattern Generator (TPG) circuit consists of an 8-bit counter that goes through all of its 256 states (see Fig. 3). During testing, the 4 most significant bits of the TPG outputs are used repeatedly to form the multiplier input X while the remaining 4 bits are used

repeatedly to form the multiplier input Y. During application of the 256 vectors, all cells of the CSAM and the CPAM are exhaustively tested with all their input combinations, except for a few that do not receive all possible input combinations. Multiplexers are used to select between normal inputs and BIST inputs. An accumulator-based response compaction scheme is used for Output Data Compaction (ODC).

In this work we consider $n \times n$ CPAMs and CSAMs. We denote the outputs of the counter TPG as $c_7 c_6 c_5 c_4 c_3 c_2 c_1 c_0$, with c_7 representing the most significant output of the counter. We will also denote as $x_{n-1} \ldots x_1 x_0$ and $y_{n-1} \ldots y_1 y_0$ the bits of X and Y inputs of the multiplier respectively. x_{n-1} and y_{n-1} correspond to the most significant bits of X and Y respectively.

II.3. Modified-Booth Multipliers and Built-In Self-Test

Another way of speeding up the multiplication process is the reduction of the partial products. Multipliers implementing the modified Booth algorithm with 2-bit recoding feature regularity, short execution time and small area compared to other implementations of multipliers for signed multiplication [16]. In this work we consider $n \times n$ MBMs ($n = 2^k$), with sign generate that use carry save arrays of adders to add the partial products. A $n \times n$ MBM is a combinational circuit with inputs $x_{n-1} \ldots x_1 x_0$, $y_{n-1} \ldots y_1 y_0$ and outputs $p_{2n-1} \ldots p_1 p_0$. Figure 4 presents an 8-bit MBM. A $n \times n$ MBM is composed by: (i) recoding cells, denoted as r-cells in Figure 4, (ii) product formation cells, denoted as pp-cells, l_pp-cells (the leftmost cell in a pp-cell row) and r_pp-cells (the rightmost cell in a pp-cell row) (iii) full adders, (iv) half adders, (v) 2-input OR gates and (vi) the 2n-bit final result forming adder.

C-Testable MBM designs have been proposed in the past for the cell fault model [17] as well as for stuck-at, transistor stuck-open and stuck-closed fault models [18]. A BIST scheme, under the cell fault model, for RC-MBMs was proposed in [19].
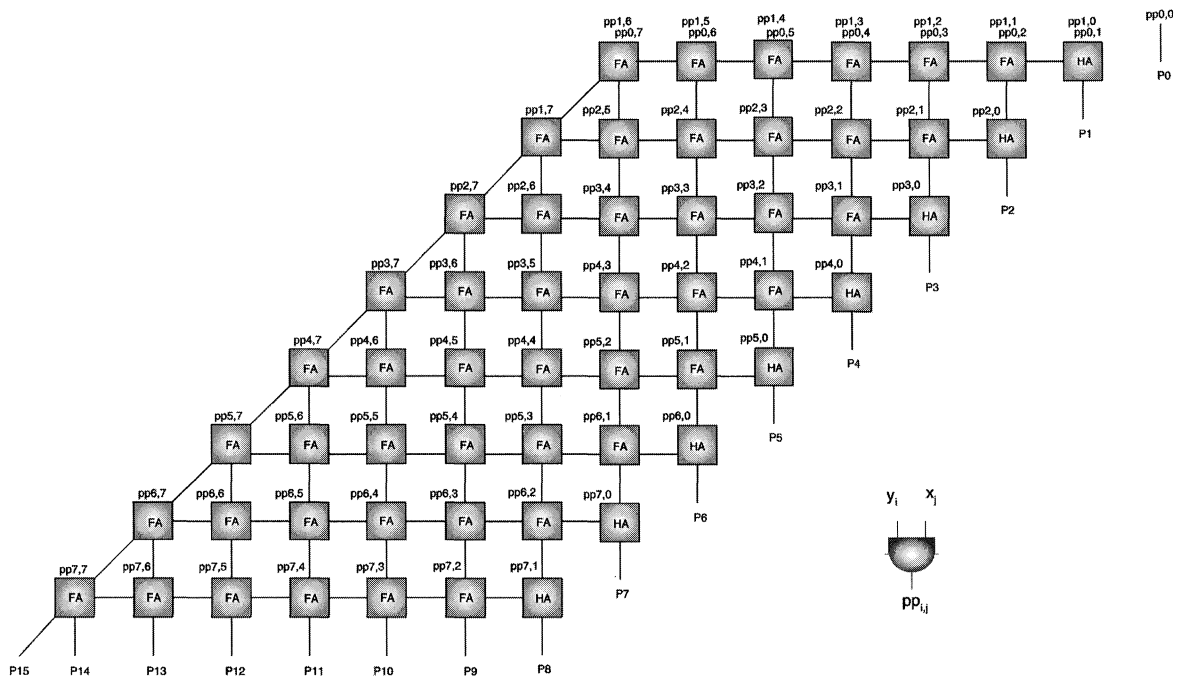
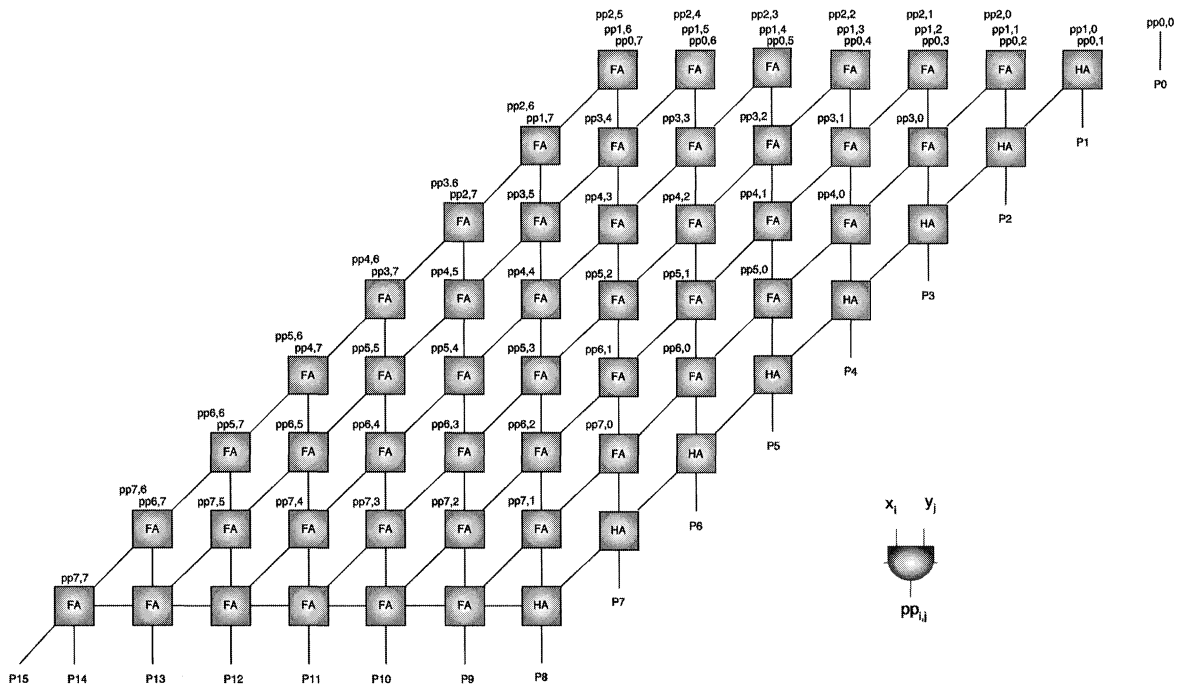FIGURE 1 An 8-bit carry-propagate array multiplier (CPAM).

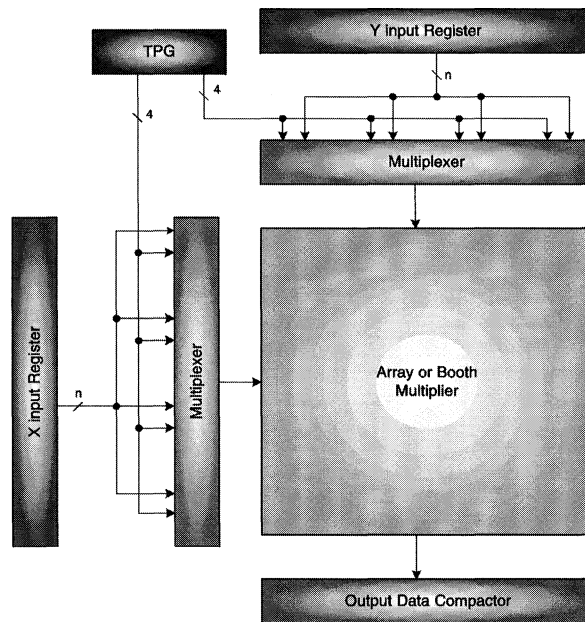FIGURE 2 An 8-bit carry-save array multiplier (CSAM).
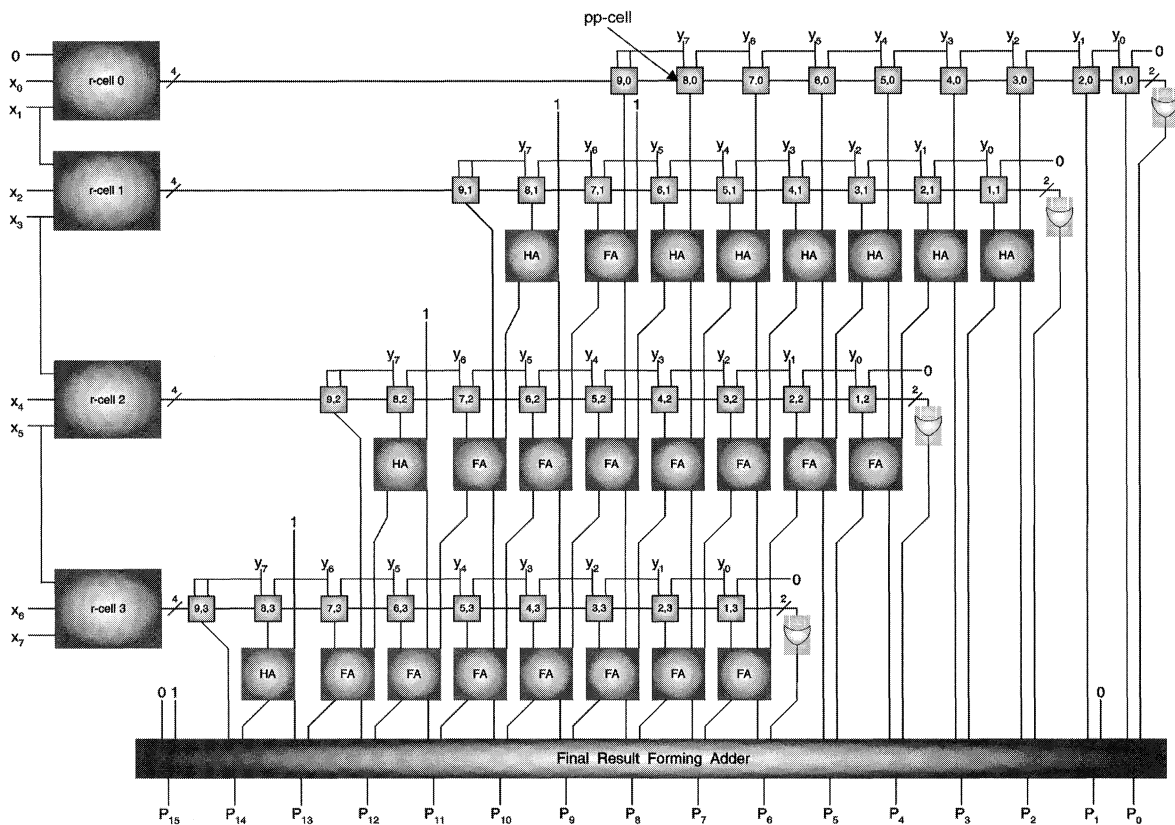
FIGURE 3   The BIST scheme for multipliers.

FIGURE 4   An 8-bit modified-Booth multiplier with Sign Generate (MBM).

Unfortunately, the authors of [19] use carry propagate arrays of adders to add the partial products and furthermore they do not consider the low power dissipation objective. The TPG circuit of [19] is an 8-bit counter that goes through all of its 256 states as in the case of CPAM and CSAM (see Fig. 3). During testing, the low nibble (that is, the 4 least significant bits of the TPG outputs) is used repeatedly to form the multiplier input Y while the high nibble (that is, the 4 most significant bits of the TPG outputs) is used repeatedly to form the multiplier input X. During application of the 256 vectors, all cells of the MBM, except few, are exhaustively tested with all their input combinations. Multiplexers are used to select between normal inputs and BIST inputs. An accumulator with rotate carry [20] or multiple rotate carry adders [19] is used for Output Data Compaction. The length of the test set was reduced to 225 vectors by avoiding the all 0's patterns in any nibble of the counter TPG in [21].

III. ASSIGNMENT OF THE TPG OUTPUTS TO THE MULTIPLIER INPUTS

In this section, we address the problem of properly assigning the TPG outputs to the multiplier inputs for achieving low power dissipation. Although we consider the cell fault model, two reasons enforce us to take into account specific implementations of the cells: (a) the error aliasing calculation of the ODC circuit and (b) the estimation of the power dissipation during testing.

However, in order to derive results not depending on a specific implementation, the analysis of the multiplier structures, that will lead us to the new BIST schemes, as well as the evaluation of them, must be based on more than one implementations of the adder cells. Hence, we consider three distinct gate implementations of the half and full adder cells, presented respectively in [22], [8] and [23]. We will refer to these implementations as Cell 1, Cell 2 and Cell 3 respectively. The same implementations were used for the adders of the ripple carry adder at the last stage of the MBM.

The implementations considered for the r-cells, the pp-cells, the r_pp-cell, the l_pp-cell of the MBMs were taken from [24]. The carry look-ahead adder circuit considered in the case of CL-MBMs was taken from [25].

The primary inputs weights for multipliers of various sizes for each of the possible cells were computed using relation (4). Table I lists the weights for the 8×8 multiplier inputs for all the cells considered and indicates that the distribution of weights is independent of the specific full and half adder cell implementation. Comparing any possible pair of inputs, the one with the larger weight contributes more than the others to the power dissipation. Similar distribution of weights has also been observed in the larger multipliers. Hence, the same conclusions are also valid for the larger multipliers.

From Table I we can easily see that the sum of weights of X inputs is greater than the sum of weights of Y inputs. Therefore, the 4 most significant outputs of the TPG, that is, the outputs with the less number of transitions, should drive the X inputs while its 4 least significant outputs should drive the Y inputs.

The next step is to assign the low nibble of the TPG ($c_3c_2c_1c_0$) to specific inputs y_i, with $i = 0$, $1, \ldots, n-1$. Since this nibble is repeatedly assigned to the Y multiplier inputs, we sum the weights of the inputs that receive the same TPG output bit. The results for the 8×8 multiplier are listed in Table II. Larger multipliers also present similar behavior. For maximum reduction of the number of transitions, the signals with the least number of transitions should be assigned to the inputs with the largest sum of weights. We assign the TPG output bit having the most transitions (that is c_0) to the inputs with the smallest sum of weight (that is, y_{4i}, in the case of CSAM and MBM and y_{4i+3} in the case of CPAM, with $i = 0, 1, 2, \ldots$). The assignment of the remaining bits of the low nibble of the counter is made in the same way. We will use the notation "Best Assignment" for referring to this assignment.

Furthermore, the number of transitions at the primary inputs of the multipliers can be reduced

TABLE I Weights of the 8 × 8 multiplier inputs

	X input weights								Y input weights							
	$w(x_7)$	$w(x_6)$	$w(x_5)$	$w(x_4)$	$w(x_3)$	$w(x_2)$	$w(x_1)$	$w(x_0)$	$w(y_7)$	$w(y_6)$	$w(y_5)$	$w(y_4)$	$w(y_3)$	$w(y_2)$	$w(y_1)$	$w(y_0)$
Carry Propagate Array Multiplier																
Cell 1	89	102	112	120	125	127	127	127	31	48	65	82	99	116	130	130
Cell 2	175	202	224	240	251	258	260	261	65	98	131	165	198	231	262	262
Cell 3	158	182	201	215	224	229	229	228	52	83	115	146	178	209	237	237
Carry Save Array Multiplier																
Cell 1	39	57	75	92	108	122	122	117	86	98	104	96	85	71	56	39
Cell 2	75	113	150	187	224	255	256	251	172	199	217	201	179	152	120	86
Cell 3	68	102	134	165	197	226	226	215	154	179	189	175	155	130	103	72
Modified Booth Multiplier – Ripple Carry Adder as the Final Result Forming Adder																
Cell 1	57	56	94	78	105	86	102	81	73	74	78	76	74	69	63	57
Cell 2	97	96	156	140	182	160	183	154	128	130	140	140	134	128	117	106
Cell 3	89	88	144	127	167	144	163	136	117	119	127	126	120	113	102	91
Modified Booth Multiplier – Group Carry Look Ahead as the Final Result Forming Adder																
Cell 1	68	68	107	90	119	99	115	92	84	86	90	88	84	80	73	66
Cell 2	86	85	142	125	163	140	160	129	113	119	127	124	115	106	93	79
Cell 3	83	83	135	118	156	133	149	121	109	113	120	118	110	102	90	77

TABLE II Sum of weights of the 8 × 8 multiplier inputs

	Sum of weights for input X				Sum of weights for input Y			
	$\sum_{i=0}^{1} w(x_{(4i+3)})$	$\sum_{i=0}^{1} w(x_{(4i+2)})$	$\sum_{i=0}^{1} w(x_{(4i+1)})$	$\sum_{i=0}^{1} w(x_{(4i+0)})$	$\sum_{i=0}^{1} w(y_{(4i+3)})$	$\sum_{i=0}^{1} w(y_{(4i+2)})$	$\sum_{i=0}^{1} w(y_{(4i+1)})$	$\sum_{i=0}^{1} w(y_{(4i+0)})$
Carry Propagate Array Multiplier								
Cell 1	214	230	240	246	130	164	195	212
Cell 2	426	460	483	501	263	329	393	427
Cell 3	382	412	430	442	230	292	352	383
Carry Save Array Multiplier								
Cell 1	147	179	196	209	170	169	160	135
Cell 2	299	367	406	438	351	351	337	287
Cell 3	265	328	380	380	309	309	291	247
Modified Booth Multiplier – Ripple Carry Adder as the Final Result Forming Adder								
Cell 1	162	142	196	158	147	143	141	133
Cell 2	278	256	339	294	262	257	256	246
Cell 3	256	232	307	262	237	232	229	217
Modified Booth Multiplier – Group Carry Look Ahead as the Final Result Forming Adder								
Cell 1	187	166	222	182	168	166	163	153
Cell 2	249	225	301	254	228	225	219	202
Cell 3	238	214	284	239	219	215	210	194

TABLE III Total power dissipation reduction percentages over the reference architectures

	8 × 8			16 × 16			32 × 32			64 × 64		
	Cell1	Cell2	Cell3	Cell1	Cell2	Cell3	Cell1	Cell2	Cell3	Cell1	Cell2	Cell3
Carry Propagate Array Multiplier												
Best Assignment	8.7	8.0	9.4	5.0	4.4	5.2	3.9	3.6	4.0	3.5	3.4	3.6
Best Assignment + Gray	26.1	25.1	24.2	20.5	19.8	18.4	18.4	18.3	16.3	17.5	17.8	15.4
Carry Save Array Multiplier												
Best Assignment	6.8	6.6	7.3	3.6	3.1	3.8	2.0	1.6	2.1	1.3	1.1	1.3
Best Assignment + Gray	27.1	27.2	25.5	23.0	23.2	20.9	20.0	20.5	17.9	18.1	18.7	15.9
Modified Booth Multiplier – Ripple Carry Adder as the Final Result Forming Adder												
Best Assignment	11.5	11.1	11.1	4.3	2.6	4.0	1.5	0.2	1.4	1.3	0.7	1.4
Best Assignment + Gray	35.3	34.3	33.2	30.2	28.0	27.5	26.5	24.3	23.4	25.0	23.1	21.9
Modified Booth Multiplier – Group Carry Look Ahead as the Final Result Forming Adder												
Best Assignment	10.6	10.4	10.8	3.6	2.2	3.6	1.3	0.1	1.3	1.2	0.6	1.4
Best Assignment + Gray	33.9	33.6	33.4	29.1	27.8	27.5	25.9	24.2	23.5	24.7	23.0	21.9

using as TPG a Gray instead of a binary counter. To this end, we decided to use a Gray counter.

For verifying the above analysis, we have developed a gate-level power simulator. This simulator estimates the power dissipation of the whole circuit consisting of the multiplier and the BIST circuitry. Therefore, from this point forward when we refer to power reduction results, we will imply power dissipation results that also take into account the BIST circuitry as well. Table III presents the simulation results. We present comparisons against a reference architecture for CPAMs and CSAMs where the test set consists of all 256 vectors and against a reference architecture for MBMs where the test set consists of the 225 vectors presented in [21]. In the two reference architectures, the bits $c_3c_2c_1c_0$ are generated by a binary counter and the assignment of its outputs to the multiplier inputs is made in such a way that the counter bit with the least number of transitions is assigned to the multiplier inputs with the smallest sum of weights. The analysis presented in Subsection II.1 indicates that this assignment leads to the worst power dissipation results.

The X inputs of the CSAM and MBM are driven by the binary counter output bits c_7, c_6, c_5 and c_4 according to the assignment: $c_7 = x_{4i+3}$, $c_6 = x_{4i+2}$, $c_5 = x_{4i+1}$, $c_4 = x_{4i}$, with $i = 0, 1, 2, \ldots$ whereas the X inputs of the CPAM are driven by a binary counter with output bits c_7, c_6, c_5 and c_4 according to the assignment: $c_7 = x_{4i}$, $c_6 = x_{4i+1}$, $c_5 = x_{4i+2}$, $c_4 = x_{4i+3}$, with $i = 0, 1, 2, \ldots$

Each column of Table III presents the total power reduction percentage achieved over the reference architecture, when "Best Assignment" or "Best Assignment" along with Gray counting is used for the bits c_3, c_2, c_1 and c_0.

The results of Table III indicate that the maximum power reduction is achieved by using a Gray counter and by assigning its outputs with the most transitions to the multiplier inputs that have the less sum of weights. When both techniques are used, the power savings vary from 15.4% to 35.3% depending on the specific multiplier structure, size

and specific cell implementation. From Table III we can easily see that the influence of a specific cell implementation on power dissipation reduction is very small.

Let us now consider the high nibble of the counter ($c_7c_6c_5c_4$) and the inputs X of the multiplier. The weights for the x_i 8×8 multiplier inputs are also listed in Table I. Following an analysis similar to the one that was presented earlier, we can encode these counter outputs in Gray and assign them to the X multiplier inputs. In this case though, the power reduction is expected to be very small since the bits of the high nibble have much less transitions than those of the low nibble. Using our power simulator, we confirmed the above. The differences in power dissipation when using different assignments and/or Gray counting are negligible. Hence, taking into account that the hardware overhead for the implementation of a binary counter is slightly smaller than that for the implementation of a Gray counter, we decided to use a binary counter for producing c_7, c_6, c_5 and c_4. The assignment chosen was $c_7 = x_{4i+3}$, $c_6 = x_{4i+2}$, $c_5 = x_{4i+1}$, $c_4 = x_{4i}$, for $i = 0,1,2,\ldots$ for the case of CSAMs and MBMs and $c_7 = x_{4i}$, $c_6 = x_{4i+1}$, $c_5 = x_{4i+2}$, $c_4 = x_{4i+3}$, with $i = 0, 1, 2, \ldots$ for the case of CPAMs.

IV. TEST LENGTH REDUCTION

It is well known that among the vectors that a Test Pattern Generator circuit produces, some are not useful for testing purposes and are therefore redundant. Another way for reducing the power dissipated during the test application is to reduce the number of vectors applied to the circuit under test by removing these redundant states from the TPG circuit.

However, in order to remove redundant vectors without increasing the area of the TPG a lot, we have to find long sequences of consecutive redundant vectors. Two alternatives may be followed at this point. The first one is to consider each multiplier structure and size separately and

find a different set of redundant sequences for each one. In this case the smaller multipliers will have more redundant sequences whereas the bigger multipliers will have less redundant sequences. The second alternative is to examine all multiplier sizes of each structure in parallel and find a unique set of redundant sequences, independent of the multiplier size. This alternative is more suitable for highly regular structures such as the array and modified-Booth multipliers. Therefore, in this work we adopt the second alternative, keeping in mind however that this will result in a smaller reduction of the number of test vectors.

IV.1. Carry-Save Array Multiplier

Based on the information given in [15], we have verified that among the 256 vectors that the counter TPG generates, counter outputs 0000 xxxx, 1000 xxxx and 1011 xxxx, where xxxx stands for all possible values, are redundant. Also in the range from 0100 0000 to 0100 1111 only the output 0100 1111 is required. This vector is used to apply the input 010 to some of the full adders of the right half part of the multiplier. We verified that this is achieved also by the counter output vector 1100 1111, hence the output 0100 1111 of the counter can also be removed. Therefore, the counter outputs 0100 xxxx can also be considered redundant.

From the above, we conclude that from the 256 vectors that the 8-bit counter applies to a CSAM, 64 vectors can be removed. If we omit these vectors, an improved on-chip TPG circuit can be designed based on a counter. When the signal RESET is applied, the counter initializes to state 0001 0000. Then, at every CLK cycle, the low nibble counts in Gray code whereas the high nibble counts in straight binary omitting the unnecessary values.

IV.2. Carry-Propagate Array Multiplier

For determining if all 256 vectors, produced by the TPG circuit, are necessary for providing all the possible input combinations to the inputs of the CPAM cells we have developed a cell fault simulator. This simulator accepts the circuit's cell level description, a test set and each distinct cell's simulation model. Given a test vector, it identifies the input combination that gets applied in each cell of the circuit. After the application of all test vectors, the cell fault simulator is capable of producing results about the input combinations that were applied to each cell.

Using the developed simulator we verified that the values $c_7c_6c_5c_4 = 0000$, 0010, 0100 and 1000, are redundant, since the remaining values of $c_7c_6c_5c_4$ are capable of applying to every cell of the CPAM the same input combinations with those applied when $c_7c_6c_5c_4$ get all their possible values. The above was verified for all realistic CPAM sizes (with operands length of 8, 16, 32, 64 and 128 bits).

So, in this case also, 64 out of the original 256 vectors can be removed. A circuit that only produces the 192 necessary counter states can be easily synthesized. When the signal RESET is applied, the counter initializes to state 0001 0000. Then, at every CLK cycle, the low nibble counts in Gray code whereas the high nibble counts in straight binary omitting unnecessary values.

IV.3. Modified-Booth Multiplier

Using the cell-fault simulator, and starting off by the 256 vector TPG, we verified that the values $c_7c_6c_5c_4 = 0000$, 0010, 0101, 0111, 1001 and 1111, are redundant. The remaining values of $c_7c_6c_5c_4$ are capable of applying to every cell of the MBM the same input combinations with those applied when $c_7c_6c_5c_4$ get all their possible values. The above was verified for all realistic MBM sizes (with operands length of 8, 12, 16, 20, 24, 32, 40, 64 and 128 bits).

Therefore, 96 out of the 256 vectors that the TPG of [19] applies to the MBM, can be removed. A circuit that only produces the 160 necessary counter states can be easily synthesized. The circuit is initialized to state 0001 0000 and at every cycle,

its low nibble counts in Gray code whereas its high nibble in straight binary omitting unnecessary values.

V. EVALUATION OF THE PROPOSED SCHEMES

The above analysis has shown that for each multiplier structure we can produce a new TPG circuit with reduced number of vectors. A Verilog description of the proposed TPG circuits for the three multiplier structures (CSAM, CPAM and MBM) is given in Figure 5. The proposed TPG circuits were synthesized and their correct operation was verified by simulation.

In Tables IV–VI we give the power dissipation reduction of the BIST schemes, that use the proposed TPG circuits against the reference BIST schemes defined in Section II. We present results for total power dissipation, average power dissipation per test vector applied and peak power dissipation.

We can see that the total power dissipation reduction varies from 29.3% to 54.9% while the average reduction per test vector applied varies from 5.8% to 36.5%. Reductions depend on the multiplier size and specific cell implementation. The reduction of the peak power dissipation is presented in Table VI. The reduction varies from 15.5% to 50.2% depending on the specific multiplier structure, size and implementation of its basic cells.

For obtaining the above comparison results, our gate-level simulator assumes a zero delay model. The authors of [26] report that there is a correlation between the power dissipation of a circuit assuming a zero delay and the power dissipation assuming a general delay model. Hence, using a zero delay approximation is reasonable. The reductions in the total power dissipation during testing are expected to be even greater if glitches were also taken into account, since the switching activity in the nodes of the multiplier is reduced significantly during the

application of the test set by the proposed BIST schemes.

Besides the above, the test application time is also reduced by 25% in the case of CPAMs and CSAMs and by 28.9% in the case of MBMs with respect to the original BIST schemes proposed in [15, 19, 21].

Although the proposed BIST schemes can significantly cut down the power dissipated during test, the fault coverage may drop due to increased error aliasing, since every change of the test set implies new values for the error aliasing. Therefore, we need to verify that the fault coverage attained by the reduced test set, with respect to single stuck-at faults, remains at high levels.

To derive the fault coverage of the proposed schemes, with respect to single stuck-at faults, we applied fault simulation to 8×8, 16×16 and 32×32 multipliers. The results are presented in Table VII. Table VII lists the fault coverage achieved, assuming that the ODC is an accumulator implemented either as a Rotate Carry Adder (RCA) or as a Multiple-Input Non-Linear Feedback Shift Register (NLFSR) [27]. We consider 8×8, 16×16 and 32×32 multipliers and all three specific cell implementations. When we use RCA for ODC, the fault coverage is larger than 99% in most cases. However, we can observe that, due to increased error aliasing, it may drop below the acceptable level of 99% especially in the case of small multipliers. In order to reduce the error aliasing we have to use a more efficient response compaction algorithm. We can therefore use a NLFSR [27] as the ODC. In this case, the aliasing is either significantly reduced or totally eliminated, leading to a fault coverage always larger than 99%.

In the case of CL-MBMs, there is also a slight increase in the number of undetected faults (located at the group carry look-ahead adder) by the proposed test set compared to the application of the 225 vectors, but the fault coverage still remains at high levels.

The hardware overhead imposed by the proposed BIST schemes can be approximately esti-

```
module CSAM_high (clock, reset, out, enable);
input clock, reset, enable;
output [3:0] out; reg [3:0] out;
wire    T7,T6,T5,T4,C7,C6,C5,C4;
always @(posedge reset or posedge clock)
  if (reset) out <= 4'h1;
    else
      begin
        if (T7) out[3] <= !out[3];
        if (T6) out[2] <= !out[2];
        if (T5) out[1] <= !out[1];
        if (T4) out[0] <= !out[0];
      end
assign {C7,C6,C5,C4} = out;
assign T7 = (C6&C5&C4)&enable;
assign T6 = ((C7&C6&C5&C4)|(C7&!C6&C5&!C4)|
(!C7&C6&C5&C4)|(!C7&!C6&C5&C4))&enable;
assign T5 = ((C7&C6&C5&C4)|(C7&C6&!C5&C4)|
(C7&!C6&C5&!C4)|(C7&!C6&!C5&C4)|(!C7&C6&C5&C4)|
(!C7&C6&!C5&C4)|(!C7&!C6&C5&C4)|(!C7&!C6&!C5&C4)
)&enable;
assign T4 = ((C7&C6&C5&!C4)|(C7&C6&!C5&C4)|
(C7&C5&!C5&!C4)|(C7&!C6&!C5&C4)|(!C7&C6&C5&!C4)|
(!C7&C6&!C5&C4)|(!C7&!C6&C5&!C4)|(!C7&!C6&!C5&C4
))&enable;
endmodule


module CPAM_high (clock, reset, out, enable);
input clock, reset, enable;
output [3:0] out; reg [3:0] out;
wire    T7,T6,T5,T4,C7,C6,C5,C4;
always @(posedge reset or posedge clock)
  if (reset) out <= 4'h1;
    else
      begin
        if (T7) out[3] <= !out[3];
        if (T6) out[2] <= !out[2];
        if (T5) out[1] <= !out[1];
        if (T4) out[0] <= !out[0];
      end
assign {C7,C6,C5,C4} = out;
assign T7 = (C6&C5&C4)&enable;
assign T6 = ((C7&C6&C5&C4)|(C7&!C6&C5&C4)|
(!C7&C6&C5&C4)|(!C7&!C6&C5&C4))&enable;
assign T5 = ((C7&C6&C5&C4)|(C7&C6&!C5&C4)|
(C7&!C6&C5&C4)|(C7&!C6&!C5&C4)|(!C7&C6&C5&C4)|
(!C7&C6&!C5&C4)|(!C7&!C6&C5&C4)|(!C7&!C6&!C5&C4)
)&enable;
assign T4 = ((C7&C6&C5&!C4)|(C7&C6&!C5&C4)|
(C7&C6&!C5&!C4)|(C7&!C6&C5&C4)|(C7&!C6&!C5&!C4)|
(C7&!C6&!C5&C4)|(!C7&C6&C5&!C4)|(!C7&C6&C5&C4))
&enable;
endmodule


module MBM_high (clock, reset, out, enable);
input clock, reset, enable;
output [3:0] out; reg [3:0] out;
wire    T7,T6,T5,T4,C7,C6,C5,C4;
always @(posedge reset or posedge clock)
  if (reset) out <= 4'h1;
    else
      begin
        if (T7) out[3] <= !out[3];
```

```
        if (T6) out[2] <= !out[2];
        if (T5) out[1] <= !out[1];
        if (T4) out[0] <= !out[0];
      end
assign {C7,C6,C5,C4} = out;
assign T7 = (C6&C5&!C4)&enable;
assign T6 = ((!C6&C5&C4)|(C6&C5&!C4))&enable;
assign T5 = ((!C7&!C6&C4)|(!C7&C6&!C4)|
(C7&!C6&!C5&!C4)|(C7&!C6&C5&C4)|(C7&C6&!C5&C4)|
(C7&C6&C5&!C4))&enable;
assign T4 = ((!C6&C5&C4)|(C7&!C6&C5&!C4)|
(C7&C6&!C4)|(C7&C6&!C5&C4))&enable;
endmodule


module low (clock, reset, out, enable);
input clock, reset;
output [3:0] out; output enable;
reg    [3:0] out;
wire    T3,T2,T1,T0,C3,C2,C1,C0;
always @(posedge reset or posedge clock)
  if (reset) out <= 4'h0;
    else
      begin
        if (T3) out[3] <= !out[3];
        if (T2) out[2] <= !out[2];
        if (T1) out[1] <= !out[1];
        if (T0) out[0] <= !out[0];
      end
assign {C3,C2,C1,C0} = out;
assign T3 = (!C0&!C1&C2&!C3)|(!C0&!C1&!C2&C3);
assign T2 = (!C3&!C2&C1&!C0)|(C3&C2&C1&!C0);
assign T1 = (!C3&!C2&!C1&C0)|(!C3&C2&C1&C0)|
(C3&C2&!C1&C0)|(C3&!C2&C1&C0);
assign T0 = (!C3&!C2&!C1&!C0)|(!C3&!C2&C1&C0)|
(!C3&C2&C1&!C0)|(!C3&C2&!C1&C0)|(C3&C2&!C1&!C0)|
(C3&C2&C1&C0)|(C3&!C2&C1&!C0)|(C3&!C2&!C1&C0);
assign enable = (out[3:0]==4'h8) ? 1'b1 : 1'b0;
endmodule


module CSAM_TPG (clock, reset, out);
input clock, reset;
output [7:0] out;
wire enable;
low in0 (clock, reset, out[3:0], enable);
CSAM_high in1 (clock, reset, out[7:4], enable);
endmodule


module CPAM_TPG (clock, reset, out);
input clock, reset;
output [7:0] out;
wire enable;
low in0 (clock, reset, out[3:0], enable);
CPAM_high in1 (clock, reset, out[7:4], enable);
endmodule


module MBM_TPG (clock, reset, out);
input clock, reset;
output [7:0] out;
wire enable;
low in0 (clock, reset, out[3:0], enable);
MBM_high in1 (clock, reset, out[7:4], enable);
endmodule
```

FIGURE 5 A Verilog description of the proposed TPG circuits.

TABLE IV Total power dissipation reduction percentages of the proposed TPGs

	CPAM				CSAM			
	8×8	16×16	32×32	64×64	8×8	16×16	32×32	64×64
Cell 1	34.1	31.4	31.5	31.7	38.9	35.6	34.2	33.3
Cell 2	33.2	31.1	31.8	32.2	38.7	35.7	34.7	34.0
Cell 3	32.0	29.3	29.6	29.8	37.2	33.6	32.3	31.4
	RC-MBM				CL-MBM			
Cell 1	54.9	50.4	47.6	46.4	53.6	49.5	47.2	46.2
Cell 2	54.0	48.4	45.5	44.5	53.4	48.2	45.5	44.5
Cell 3	53.3	48.3	45.3	44.1	53.3	48.3	45.4	44.1

TABLE V Average power dissipation reduction percentages per vector of the proposed TPGs

	CPAM				CSAM			
	8×8	16×16	32×32	64×64	8×8	16×16	32×32	64×64
Cell 1	12.1	8.5	8.7	8.9	18.5	14.1	12.3	11.1
Cell 2	11.0	8.1	9.0	9.6	18.3	14.3	12.9	12.0
Cell 3	9.3	5.8	6.1	6.3	16.2	11.5	9.7	8.6
	RC-MBM				CL-MBM			
Cell 1	36.5	30.3	26.4	24.7	34.8	28.9	25.7	24.4
Cell 2	35.2	27.4	23.4	21.9	34.4	27.2	23.3	21.9
Cell 3	34.3	27.3	23.1	21.4	34.4	27.3	23.2	21.4

TABLE VI Peak power dissipation reduction percentages of the proposed TPGs

	CPAM				CSAM			
	8×8	16×16	32×32	64×64	8×8	16×16	32×32	64×64
Cell 1	40.6	39.2	41.2	41.7	32.5	27.2	24.5	31.5
Cell 2	47.5	47.0	47.6	48.4	48.2	50.2	46.5	44.8
Cell 3	37.3	35.1	37.6	38.6	21.4	18.2	16.5	15.5
	RC-MBM				CL-MBM			
Cell 1	19.3	19.1	24.0	26.6	19.2	20.2	24.9	26.9
Cell 2	22.4	27.4	31.8	34.3	18.3	26.0	31.4	34.1
Cell 3	20.0	15.8	21.5	23.9	19.6	17.6	22.6	24.5

mated in gate equivalents, based on the information given in [28], as follows (we assume that 1 gate equivalent is equal to one 2-input NAND gate): a full and a half adder equal to 10 and 5 gate equivalents respectively, each r-cell, pp_cell, l_pp-cell and r_pp-cell respectively requires 15, 13, 13 and 8 gate equivalents for its implementation, one 4-bit CLA cell requires 80 gate equivalents, one flip-flop equals 10 gate equivalents, one multiplexer equals 3 gate equivalents, one 2 or 3-input AND or OR gate equals 2 gate equivalents and one 2-input XOR or XNOR gate equals 4 gate equivalents.

The design of an n-bit CPAM or CSAM consists of n^2 AND gates, n^2-2n full adders and n half adders. We assume that an accumulator is already part of the circuit so we add 2n full adders and an equal number of flip-flops for the Rotate Carry Adder. Thus the total number of gates is: $25n + 12n^2$.

TABLE VII   Fault Coverage Percentages of the proposed TPGs

		CPAM		CSAM	
		RCA	NLFSR	RCA	NLFSR
	Cell 1	99.32	99.94	98.30	99.77
8 × 8	Cell 2	99.36	99.36	99.11	99.11
	Cell 3	99.72	100.0	99.22	99.43
	Cell 1	99.61	99.99	98.79	99.84
16 × 16	Cell 2	99.72	99.72	99.47	99.47
	Cell 3	99.89	100.0	99.40	99.61
	Cell 1	99.67	100.0	99.09	99.90
32 × 32	Cell 2	99.87	99.87	99.71	99.71
	Cell 3	99.88	100.0	99.51	99.78
		RC-MBM		CL-MBM	
	Cell 1	98.66	100.0	97.03	99.55
8 × 8	Cell 2	98.98	99.70	97.51	99.61
	Cell 3	99.13	100.0	97.36	99.61
	Cell 1	99.61	99.97	98.90	99.74
16 × 16	Cell 2	99.72	99.88	99.13	99.77
	Cell 3	99.71	100.0	99.05	99.81
	Cell 1	99.78	99.97	99.45	99.86
32 × 32	Cell 2	99.88	99.93	99.61	99.87
	Cell 3	99.87	100.0	99.59	99.92

The hardware required for the implementation of the proposed BIST scheme consists of 2n multiplexers and the TPG circuit consisting of 8 flip-flops and the combinational circuit that drives them, approximating to a total of $6n + 111$ gates.

Hence, the hardware overhead (HO), for the cases of CPAMs and CSAMS, is given in the following relation:

$$HO = \frac{6n + 111}{25n + 12n^2} \qquad (5)$$

From relation (5), for n = 16, 32 and 64, we get respectively HO = 5.96%, 2.31% and 0.97%. The hardware overhead for the reference BIST scheme proposed in [15], for n = 16, 32 and 64, is equal to 5.70%, 2.25% and 0.96% respectively. We can see that the additional hardware overhead of the proposed BIST scheme is negligible.

The design of the MBM consists of n/2 r-cells, $(n-1)(n/2)$ pp-cells, n/2 1_pp-cells, n/2 r_pp-cells, $(n-1)[(n/2)-2]+1$ full adders, $n+(n/2)-3$ half adders and n/2 2-input OR gates. We add 2n full adders for the RC-MBM or n/2 carry look-ahead adder cells for the CL-MBM. We assume that an accumulator is already part of the circuit so we add 2n full adders and an equal number of flip-flops for the Rotate Carry Adder. Thus the total number of gates is: $(23n^2 + 110n + 30)/2$ in the case of RC-MBMs and $(23n^2 + 150n + 30)/2$ in the case of CL-MBMs.

The hardware required for the implementation of the proposed BIST scheme consists of 2n multiplexers, 8 flip-flops and the combinational circuit of the TPG circuit, giving a total number of $6n + 136$ gate equivalents. Hence the hardware overhead for the RC-MBM and the CL-MBM is given by the relations:

$$HO_{RC-MBM} = \frac{2(6n + 136)}{23n^2 + 110n + 30}, \qquad (6)$$

$$HO_{CL-MBM} = \frac{2(6n + 136)}{23n^2 + 150n + 30} \qquad (7)$$

Relations (6) and (7), for n = 16, 32 and 64, result in a hardware overhead less than 6.1%, 2.5% and 1.1% respectively, that is, very small. Note that the power reduction results presented earlier include the power dissipation of both the

multiplier and the BIST circuitry. Therefore we do not present power dissipation comparisons among the different TPG circuits in this section.

VI. CONCLUSION

BIST schemes that achieve low power dissipation during testing attain an increasing interest due to cost, reliability and technology related issues. In this paper we have proposed low power BIST schemes for carry-propagate, carry-save and modified-Booth multipliers. Starting off by the already proposed BIST schemes, we have shown that the low power objective can be achieved by: (a) proper assignment of the TPG outputs to the multiplier inputs, (b) the use of Gray encoding and (c) reducing the test set length without affecting the fault coverage. The combination of these three techniques resulted to BIST schemes with total power dissipation reduction up to 54.9%, average power dissipation reduction up to 36.5% and peak power dissipation reduction up to 50.2%. The introduced BIST schemes have small hardware overhead and also achieve a significant test application time reduction with respect to the already proposed BIST schemes.

References

[1] Zorian Y., "A Distributed BIST control scheme for Complex VLSI devices", *Proc. of VLSI Test Symposium*, pp. 4–9, Atlantic City, USA, April, 1993.

[2] Chou, R., Saluja, K. and Agrawal, V., "Scheduling Tests for VLSI Systems Under Power Constraints", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2), 175–185, June, 1997.

[3] Girard, P., "Low Power Testing of VLSI Circuits: Problems & Solutions", *Proc. of the 1st IEEE International Symposium on Quality Electronic Design*, pp. 173–179, San Jose, USA, March, 2000.

[4] Wang, S. and Gupta, S. K., "DS-LFSR: A New BIST TPG for Low Heat Dissipation", *Proc. of International Test Conference*, pp. 848–857, Washington, USA, November, 1997.

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

[6] Manich, S., Gabarro, A., Lopez, M., Figueras, J., Girard, P., Guiller, L., Landrault, C., Pravossoudovitch, S.,

[ ] Texeira, P. and Santos, M. (2000). "Low Power BIST by Filtering Non-Detecting Vectors", *Journal of Electronic Testing: Theory & Applications*, 16(3), 193–202, June, 2000.

[7] Roy, R., "Scaling Towards Nanometer Technologies: Design for Test Challenges", Panel, *Design Automation and Test in Europe*, Munich, Germany, March, 1999.

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

[9] Parkar, R., "Bare Die Test", *Proc. of IEEE Multi-Chip Module Conference*, pp. 24–27, Santa Cruz, USA, March, 1992.

[10] Wang, S. and Gupta, S. K. (1998). "ATPG for Heat Dissipation Minimization During Test Application", *IEEE Transactions on Computers*, 47(2), 256–262, February, 1998.

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

[12] Rabaey, J. and Pedram, M., *Low Power Design Methodologies*, Kluwer Academic Publishers, 1996.

[13] Shen, J. and Ferguson, F., "The design of easily testable VLSI array multipliers", *IEEE Transactions on Computers*, 33(6), 554–560, June, 1984.

[14] Hong, S. (1988). "An easily testable parallel multiplier", *Proc. of the International Symposium on Fault-Tolerant Computing*, pp. 214–219, Tokyo, Japan, June, 1988.

[15] Gizopoulos, D., Paschalis, A. and Zorian, Y., "An Effective Built-In Self-Test Scheme for Parallel Multipliers", *IEEE Transactions on Computers*, 48(9), 936–950, September, 1999.

[16] Annaratone, M., *Digital CMOS Circuit Design*, Kluwer Academic Publishers, 1986.

[17] Gizopoulos, D., Nikolos, D., Paschalis, A. and Halatsis, C., "C-Testable Modified Booth Multipliers", *Journal of Electronic Testing: Theory & Applications*, 8(3), 241–259, June, 1996.

[18] Van Sas, J., Nowe, C., Pollet, D., Catthoor, F., Vanoostende, P. and De Man, H., "Design of a C-Testable Booth Multiplier using a Realistic Fault Model", *Journal of Electronic Testing: Theory & Applications*, 5(1), 29–41, February, 1994.

[19] Gizopoulos, D., Paschalis, A. and Zorian, Y., "Effective Built-In Self-Test for Booth Multipliers", *IEEE Design and Test of Computers*, pp. 105–111, July–September, 1998.

[20] Rajski, J. and Tyszer, J., "Test Responses Compaction in Accumulators with Rotate Carry Adders", *IEEE Transactions on Computer Aided Design*, 12(4), 531–539, April, 1993.

[21] Gizopoulos, D., Paschalis, A. and Zorian, Y., "An Effective BIST Scheme for Datapaths", *Proc. of the International Test Conference*, pp. 76–85, Washington, USA, October, 1996.

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

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

[24] Kalligeros, E., Vergos, H. T., Nikolos, D., Tsiatouhas, Y. and Haniotakis, T., "Path Delay Fault Testable Modified Booth Multipliers", *Proc. of XIV Design of Circuits and Integrated Systems Conference*, pp. 301–306, Palma de Mallorca, Spain, November, 1998.

[25] Haniotakis, T., Tsiatouhas, Y. and Nikolos, D., "C-Testable One-Dimensional ILAs with Respect to Path Delay Faults: Theory and Applications", *Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 155–163., Austin, USA, November, 1998.

[26] Shen, A., Ghosh, A., Devadas, S. and Keutzer, K., "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks", *Proc. of ACM/IEEE International Conference on CAD*, pp. 402–407, Santa Clara, USA, November, 1992.

[27] Bakalis, D., Nikolos, D. and Kavousianos, X., "Test Response Compaction by an Accumulator Behaving as a Multiple-Input Non-Linear Feedback Shift Register", *Proc. of International Test Conference*, pp. 804–811, Atlantic City, USA, October, 2000.

[28] *AT&T HS500C CMOS Standard Cell Library Catalogue*, Allentown, Pa.: AT&T Microelectronics, 1995.

Authors' Biographies

Dimitris N. Bakalis received a Diploma degree in 1995 and a M.Sc. degree in 2000 in Computer Engineering, both from the Department of Computer Engineering and Informatics of the University of Patras in Greece. He is currently pursuing his Ph.D. degree at the same department in the area of Low Power Testing. His other research interests include VLSI design and test. He is a student member of the IEEE.

Xrysovalantis I. Kavousianos is currently pursuing his Ph.D. at the Department of Computer Engineering and Informatics at the University of Patras, Greece in the area of On-Line Testing. He holds a Diploma degree in Computer Engineering from the same department. In 1997 he was awarded a scholarship of the Technical Chamber of Greece due to his excellent student records. His other research interests include VLSI design and Low Power Testing. He is also a student member of the IEEE.

Haridimos T. Vergos was born in Patras, Greece in 1968. He received the Diploma in Computer Engineering and his Ph.D. degree from the University of Patras, Greece in 1991 and 1996 respectively. In 1998, he worked as an ASIC designer in the Multimedia and Networking Group of Atmel Inc. Currently, he holds a Lecturer position in the Computer Engineering and Informatics Dept., of the University of Patras, Greece. His main research interests include Fault Tolerant Digital Systems and Reliable Design and Test of computer systems. He is a member of the Greek Computer Engineering Society and the Technical Chamber of Greece.

Dimitris Nikolos received the B.Sc. degree in Physics, the M.Sc. degree in Electronics and the Ph.D. degree in Computer Science, all from the University of Athens. He is currently professor in the Computer Engineering and Informatics Dept. of the University of Patras and head of the Technology and Computer Architecture Laboratory. He has served as program cochairman of the last four IEEE On-Line Testing Workshops. He also served on the program committees for the IEEE International On-Line Testing Workshop in 1995 and 1996, for the 1997, 1998 and 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI systems, for the Third European Dependable Computing Conference and for the DATE-2000 Conference. His main research interests are fault-tolerant computing, computer architecture, VLSI design, test and design for testability. Prof. Nikolos has authored or co-authored more than 100 scientific papers. He is a member of the IEEE.

George Ph. Alexiou received a B.Sc. in Physics in 1976 and a Ph.D. in Electronics in 1980, both from the University of Patras, Greece, where he is now a professor in the Department of Computer Engineering and Informatics. He is publishing papers in a number of international journals and conference proceedings. His research interests include VLSI design, VLSI CAD tools, signal processing, digital systems and data communications. He is a member of the IEEE.