ON-CHIP DETERMINISTIC COUNTER-BASED TPG WITH LOW HEAT DISSIPATION *

X. Kavousianos D. Nikolos
The University of Patras
Computer Engineering and Informatics Dept.
Patras, Greece 26500
kabousia@ceid.upatras.gr/nikolos@cti.gr

S. Tragoudas
The University of Arizona
Electrical and Computer Engineering Dept.
Tucson, AZ 85721
spyros@ece.arizona.edu

ABSTRACT

An on-chip test pattern generation (TPG) scheme for the digital components of a mixed signal system is presented. The TPG is a counter. We propose CAD tools that automate its design so that the heat dissipation during test application is low. Experimental results on the ISCAS'85 benchmarks show the impact of the proposed methods.

I. INTRODUCTION

A popular method of testing systems containing mixed-signal devices uses a dual-mode Automatic Test Equipment (ATE) which consists of a System Controller, the Analog Measurement System and a Switch [5]. Each device is accessed and tested through a mixed-signal Test Bus. Each device may contain digital and analog components which are tested in different testing modes. This process can be significantly simplified and accelerated if Built-In Self-Test (BIST) mechanisms are incorporated into all, or most, of the digital components in each device. The latter components may be self-tested while other components in the same or other devices are tested using the ATE.

The main objective of most BIST techniques has been the design of on-chip Test Pattern Generators (TPGs) that achieve high fault coverage at acceptable test lengths. Pseudorandomly generated patterns can detect the easy to detect faults.

A methodology for on-chip TPG suggests that the test patterns for the hard to detect faults be stored in a ROM or generated on-chip by less hardware intensive mechanisms such as a Linear Feedback Shift Register (LFSR) or a binary counter. The goal is to reproduce on-chip a set T of patterns, referred to as the test matrix, that an Automatic Test Pattern Generation Tool (ATPG) has already generated for detecting the hard to detect faults. This is also called the test set embedding problem, and is different than the pseudoexhaustive/pseudorandom TPG problem which is a fault-independent TPG problem.

Several on-chip test set embedding TPG schemes for fully scanned digital systems have been proposed. They are based on Weighted Random LFSRs (WRLFSRs), counters, and cellular automata. Their test application time and hardware overhead is low.

However, WRLFSR-based TPGs require long test sequences to attain high fault coverage for circuits that have a large number of random pattern resistant faults. In addition, the correlation between consecutive patterns generated by LFSRs is much lower than the patterns which are applied on the circuit during operation mode. It has been observed that this type of on-chip TPG may result to switching activity in the circuit that can be significantly higher during BIST than during its normal operation [6].

Excessive switching activity during test increases the heat dissipation in a CMOS circuit since the latter quantity is proportional to switching activity. This may cause permanent damage of the circuit. Heat dissipating in the test mode is already affecting test methodologies and test scheduling [6]. The problem becomes alarming as advances in high performance allow smaller chips to be placed closer in order to decrease interconnect delays. Many neighboring chips may be simultaneously self-tested or tested using ATE that supports mixed-mode devices.

The use of special cooling equipment to remove excessive heat dissipated during test application becomes increasingly difficult, especially on mixed-signal BIST at board applications. The other alternative is to design on-chip TPGs which apply test patterns that cause switching activity which is comparable to that generated during normal operation.

An LFSR-based on-chip TPG was proposed recently in [7] that guarantees reduced switching activity when compared to traditional LFSR-based TPGs.

*Partially supported by NSF grant CCR-9815229

0-7803-5510-0-5/99 $10.00 © 1999 IEEE

However, [7] considers pseudoexhaustive TPG. This approach will generate unnecessarily high heat dissipation (due to the very large application time) when applied for deterministic on-chip TPG.

This paper proposes a low heat dissipation deterministic on-chip TPG based on a binary counter. The counter will reproduce on-chip an input test matrix T consisting of p test patterns. Counter-based schemes with low test application time and hardware overhead were recently proposed in [3, 4]. The lack of randomness between consecutive patterns indicates a promising framework for low switching activity and thus heat dissipation. The latter objective was not considered in [3, 4].

The paper is organized as follows. Section II describes techniques that we use to effectively synthesize counters as deterministic on-chip TPGs. These methods amount to operations on the test matrix T. Section III proposes metrics that can be used in order to synthesize counters with low heat dissipation. It also presents our proposed methodology. Section IV gives experimental results on ISCAS'85 benchmarks, and Section V concludes.

II. PRELIMINARIES

The work in [3, 4] proposes synthesis methods that may have a significant impact in the performance of the designed counter TPG. They are described as operations on the test matrix T. However, it is explained that any modification of T amounts to a well defined resynthesis (redesign) process for the counter TPG. [3, 4] show that these operations have a great impact on the the test matrix reproduction time. This paper shows that the heat dissipation is drastically reduced when these operations are used appropriately in the design of the on-chip counter TPG. The matrix operations are briefly reviewed in this section.

Let us first assume that the matrix T is binary, i.e., no test pattern in T has don't cares. Observe that up to f identical columns of the binary matrix T can be collapsed (merged) into a single one. This column will then be generated by a single counter cell. Quantity f is a precomputed upper bound for each circuit so that the fan-out stems of the counter cells do not cause timing violations. This is particularly important for at-speed testing, which is used to detect slow chips. The number of counter cells w is reduced as the value of f increases. Therefore the test reproduction time t is reduced as well.

Observe that complimentary columns can be merged into a single column since they can be reproduced by the same counter cell. Also any column can be substituted by its binary compliment. Furthermore, all columns of T that are either constant at 0 or 1 can be eliminated; they can connected to the power or ground wires. These operations may reduce w significantly and thus the test application time t.

In addition, the columns of T can be permuted at any order as long as the wires that connect the respective counter cells to the circuit inputs are not excessively long. This restriction, however, is rarely an issue because the operations described earlier reduce w significantly.

Finally, observe that if one is willing to use more than one counters, the vectors of T can be partitioned into submatrices. The operations described earlier can then be applied in each submatrix. [4] explain that matrix partitioning may significantly reduce time t. It presents several counter-based schemes whose hardware overhead is bounded by that of two counters. The described operations may applied on the submatrices in different ways which define alternative synthesis schemes with clear trade-offs between the test reproduction time t and the hardware overhead [4].

The work in [3, 4] shows that it is computationally intensive to determine the optimal way of applying the above operations on T so that the matrix reproduction time t is minimum. However, sophisticated CAD tools are proposed that apply them effectively. The example below illustrates the impact of these operations on the test reproduction time t when they are appropriately applied on a binary matrix T.

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
p_1	1	1	0	1	1	1	0	1
p_2	0	1	0	1	1	1	0	1
p_3	1	1	0	1	1	1	1	1
p_4	1	0	1	0	0	0	0	0

(a)

	c_3	$\bar{c}_1$	c_7
v_1	0	0	0
v_2	0	1	0
v_3	0	0	1
v_4	1	0	0

(b)

Fig. 1(a): A test matrix with reproduction time 130 (leftmost column is most significant bit). Four patterns are applied on a 8-bit circuit.

Fig. 1(b): After identical column merging (c_2, c_4, c_5, c_6, c_8), complementary column merging (c_2, c_3), complementary column generation (c_1), and permutation, the reproduction time becomes $t = 4$.

Example: Consider the test matrix T in Fig. 1(a) which consists of four test patterns, i.e., p=4. The combinational logic circuit that has eight inputs. A

simple binary counter requires 130 clock cycles to apply the patterns of Fig. 1(a).

However, Fig. 1(b) shows how T was modified by the CAD tool in [3] which considers a single counter TPG. Observe that the reproduction time t is now optimal and equal to 4. In general, matrix T is ternary and contains test patterns with don't cares. An appropriate assignment of the don't care values may result into drastic reductions on the reproduction time t. The CAD tools in [3, 4] utilize don't care assignment effectively.

Once the CAD tools in [3, 4] modify T using the above operations, it is easy to synthesize the counter TPG so that it maps the reduced matrix with the minimal reproduction time t to the original test matrix T. This step can be performed very fast. It only requires time that is linear to the number of the circuit inputs (number of columns in T).

III. SYNTHESIS SCHEMES

This section proposes methods for synthesizing counter TPGs with low heat dissipation using algorithms that benefit from the operations in Section II. We assume a single counter TPG.

We consider power dissipation due to the consumption of dynamic switching components, resulting from the charging and discharging of capacitors. It is known that this dominates the power consumption.

Let V_{dd} denote the power supply voltage, C_l the load capacitance at line l, and tr_l the total number of transitions on line l during the test reproduction time. The total heat dissipation is:

$$H = 1/2 \cdot V_{dd}^2 \sum_l C_l \cdot \mathrm{tr}_l. \quad (1)$$

Observe that average heat dissipation cannot consider the applied test patterns as accurately as H. Clearly, average heat dissipation is an important measure when designing chips whose operation time is not known in advance. However, in TPG (with the exception of pseudorandom TPG) it is preferable to consider instead the total heat dissipation, especially when the order of the applied patterns is known.

The goal of this paper is to provide a methodology for applying the operations of Section II on matrix T and design the counter so that H is minimized. Observe that Equation (1) implicitly takes into consideration the test application time t as well as the set of vectors that that synthesized counter will generate. If the test application time t is very high it is very unlikely that H will be low.

We cannot however compute H before we explicitly apply the patterns in the order generated by the designed counter. Furthermore, it is impossible to compute for every line l the quantity tr_l without explicitly simulating each generated pattern on the circuit under test.

The methods in [3, 4] were able to benefit from the operations of Section II by being able to identify *circuit independent properties* on the columns of T that have a significant impact on the reproduction time t. Clearly, the problem studied here is significantly more difficult because *not all properties of the columns of T can be circuit independent.*

This section proposes two metrics. The first metric, described in subsection A, identifies a circuit independent property that is very useful for minimizing H. The metric of subsection A is the starting point of our proposed method. Subsection B then proposes a second metric which takes into consideration the circuit under test. Finally, subsection C shows how our method benefits from the two presented metrics.

A. The First Metric

It is expected that when two consecutive patterns p_i and p_j that the counter generates induce transitions on many inputs in the circuit then many lines in the circuit will also have transitions and thus their application order will contribute significantly on H. The metric of this subsection is based on this assertion and proposes to apply the operations of subsection II on T so that the synthesized counter generates a sequence of test patterns which minimize the total number of transitions on the inputs of the circuit. The latter quantity is precisely

$$\mathrm{tr} = \sum_{p_i, p_j} b_{p_i, p_j}. \quad (2)$$

The metric of this section proposes that the total heat dissipation H is estimated by the quantity

$$H_1 = 1/2 \cdot V_{dd}^2 \cdot \sum_{p_i, p_j} b_{p_i, p_j}. \quad (2')$$

Thus, the goal of the respective counter synthesis CAD tool is to apply the operations of Section II on T so that Equation (2') or, equivalently, (2) is minimized. In the following, we show that the problem of minimizing tr is related to the problem minimizing the test reproduction time t. Thus, the design of the new CAD tools can benefit from the algorithms and ideas in [3].

Definition We call a test matrix T basic if it is binary (does not have don't cares), no column is constant

at 0 or 1, and not any two columns are identical or complimentary to each other.

The only two operations that may apply on a basic matrix T so that the number of input transitions tr is minimized are the column permutation and the complimentary column generation. We call these operations basic. Note that the number of columns remains invariant, under any application of the basic operations, and equal to w.

Let t be the matrix reproduction time after an arbitrary application of the basic operations. Assume that the columns are numbered in increasing order from the rightmost column (least significant counter cell) to the leftmost column (most significant counter cell) of the resulting test matrix. Then the number of the input bit transitions on column k is

$$\lceil \frac{t}{k} \rceil - 1.$$

Therefore the total number of bit transitions on the inputs is

$$\sum_{k=1}^{w} (\lceil \frac{t}{k} \rceil - 1).$$

Since w is invariant of the order of applying basic operations, we have been able to show the following theorem.

Theorem 0.1 *The design of a counter that minimizes the test reproduction time t also minimizes H_1 for any basic matrix T.*

Theorem 0.1 allows for a direct application of the CAD tools that were developed in [3, 4] to this new problem formulation. However, basic test matrices are only of theoretical importance. In practice, many columns that are identical or complimentary to each other, and are represented by a single column, which we call a column representative. Every column representative corresponds to a a single counter cell.

The proposed CAD tool sorts the column representatives in decreasing order according to the number of the original columns (number of circuit inputs) that each representative column contains. Let x_i be the number of columns represented by the ith column representative, i.e., $x_{i+1} \geq x_i$.

The representative columns are then assigned in their sorted order from the most significant counter cell to the least significant counter cell. According to our previous analysis, this minimizes the total number of transitions on the inputs for a given reproduction time t.

More precisely, the total number of input bit transitions for reproduction time t is

$$\sum_{k=1}^{w} x_k \cdot (\lceil \frac{t}{k} \rceil - 1).$$

Since $x_{k+1} \geq x_k \geq 1$ and $1 \leq \lceil \frac{t}{k+1} \rceil - 1 \leq \lceil \frac{t}{k} \rceil - 1$, $\forall k$, the above approach minimizes the number of input transitions for a given value of t. In addition, the described method resembles the one used in [3, 4] in order to minimizing t heuristically. Therefore the approach tends to minimize t as well as the total number of input transitions that may occur while T is embedded using a counter.

It is a difficult task to determine the column representatives in the presence of don't cares [3]. The approach we have followed works as follows. We form a graph G where each column of T corresponds to a node in G, and two nodes are connected with an edge if the respective columns are either identical or complimentary. Then the CAD tool heuristically selects the clique (subgraph with all possible induced edges) that has the maximum number of nodes, and assigns all the respective columns under a single representative column. The respective nodes and induced edges are then removed from G and the process is repeated until no more representative columns can be generated.

We observe that this approach does not necessarily provide a good heuristic for the problem of minimizing the number of representative columns. This is an important parameter because the smaller the number of representative columns, the smaller the reproduction time t tends to be. Such modifications of the heuristic are currently under investigation.

B. The Second Metric

Our second metric takes into consideration the circuit under test so that quantity tr_l is taken into consideration while applying the operations of Section II. Let $f(l)$ denote the function of line l, and $\frac{\theta f(l)}{\theta \text{in}_i}$ denote the boolean difference of $f(l)$ with respect to input in_i. This boolean function indicates whether $f(l)$ is sensitive to changes on input in_i.

Let $f(l)_{\text{in}_i}$ (resp., $f(l)_{\text{in}_{i'}}$) denote the cofactor of $f(l)$ with respect to input variable in_i (resp., $\text{in}_{i'}$) and $\oplus$ be the XOR operator. The boolean difference is precisely

$$\frac{\theta f(l)}{\theta \text{in}_i} = f(l)_{\text{in}_i} \oplus f(l)_{\text{in}_{i'}}. \quad (3)$$

Let $P(\frac{\theta f(l)}{\theta \text{in}_i})$ denote the probability that function $\frac{\theta f(l)}{\theta \text{in}_i}$ evaluates to 1. The estimated heat dissipation is

$$H_2 = 1/2 \cdot V_{dd}^2 \cdot \sum_l C_l \cdot P(\frac{\theta f(l)}{\theta \text{in}_i}) \cdot \text{tr}_{\text{in}_i}. \quad (4)$$

Once the probability $P(\frac{\theta f(l)}{\theta \text{in}_i})$ is computed, each input in_i is assigned a weight

$$w(\text{in}_i) = \sum_l P(\frac{\theta f(l)}{\theta \text{in}_i}).$$

Weights $w(\text{in}_i)$ are used to guide a CAD tool which applies the operations of Section II on T so that metric H_2 is minimized heuristically.

This CAD tool is similar to the one described earlier for the first metric. The representative columns are generated and assigned to counter cells as follows. Graph G is now a weighted graph; every node of G is assigned a weight equal to the $w(\text{in}_i)$ weight of the column it represents. The algorithm selects each time the largest weighted clique which corresponds to a column representative. This column representative is then assigned to the most significant counter cell that has not been assigned a representative column.

Next, we describe how the probabilities $P(\frac{\theta f(l)}{\theta \text{in}_i})$ are computed. Their computation reduces to computing the signal probabilities for all lines l using the cutting algorithm in [2]. We have currently implemented the Full Range Cutting algorithm but we intend to implement and use the Partial Range cutting algorithm which tends to be more accurate.

The cutting algorithm gives for some circuit lines the exact probabilities but, in general, for each probability it gives a range that the probability belongs to. We compute the signal probability by considering the signal probability as the median of the returned range.

We execute the Full Range Cutting algorithm a total of $2 \cdot w$ times. Each time the probability values of the input lines change. Each input in_i is considered twice, the first time its probability is set to 0 and the second time is set to 1. In both cases, the probability of the other inputs is set to 1/2. That way, for each internal line l we find two probabilities P_0^i and P_1^i with respect to the main input in_i.

Subsequently, the XOR of P_0^i and P_1^i is computed as follows: If line l does not depend on in_i then $P(\frac{\theta f(l)}{\theta \text{in}_i})$ is 0. Otherwise, we consider an XOR gate with two inputs with probabilities P_0^i and P_1^i and $P(\frac{\theta f(l)}{\theta \text{in}_i})$ is set to the probability of the output of the gate.

Finally, we have estimated the capacitance at each line of the circuit as follows: For each line we add the capacitance of the output of the gate that drives this line, and the capacitances of the inputs of the gates driven by this line. The capacitances of the inputs/outputs of each type of gate (And, Nand, Or etc) have been estimated taking into account an 1-micron Technology implementation and normalized to integer values. For all types of gates, the input/output capacitance is considered to be equal to 4, except the XOR and XNOR input capacitance which is equal to 6.

C. The Proposed Method

The first metric tends to minimize the number of bit transitions on the inputs and also the test application time. Both factors are critical in the heat dissipation. Its disadvantage is that it completely ignores the circuit under test.

The second metric takes into consideration the circuit under test but only implicitly considers the number of the transitions on the inputs and the test application time. For example, the CAD tool in the previous subsection may assign to the most significant counter bit a representative column that may contain much less number of circuit inputs. Furthermore, since the don't care assignment is also driven by the input weights $w(\text{in}_i)$, the CAD tool may result into a higher number of representative columns than the CAD tool for the first metric. That way, the test application time may be increased significantly. We would like to assign higher priority to these two factors.

Recall that the probabilities $P(\frac{\theta f(l)}{\theta \text{in}_i})$ that the second metric uses in order to consider the circuit under test are computed approximately and often this computation is not very accurate. On the other hand, we have shown that the two circuit independent quantities can be tackled more efficiently.

We thus propose to combine the ideas of subsections A and B by giving higher priority to the quantities considered by the first metric. We modify the CAD tool for the first metric so that it also considers the weights $w(\text{in}_i)$ on the inputs.

More precisely, the CAD tool consists of two major steps. First it constructs the representative columns. Since the number of these columns has a direct impact on the test application time (is equal to the number of cells in the counter) and implicitly on the number of transitions on the circuit inputs, we construct them as in subsection A, i.e., based on the cardinalities of the cliques in the intermediate graph G.

The second step assigns one representative column per counter cell. This assignment is now driven by the weights on the representative columns. The weight $w(I)$ of a representative column I is

$$w(I) = \sum_{\text{in}_i \in I} w(\text{in}_i).$$

The column representatives are sorted in descending order according to their $w()$ weights and they are assigned in that order from the most significant counter cell to the least significant counter cell.

VI. EXPERIMENTAL RESULTS

Table 1 provides experimental comparisons on the described metrics. We consider the ISCAS'85 benchmarks. In order to further evaluate the impact of our proposed method in Section III.C we implemented another approach, we call it Random Assignment Method (RAM). This approach that determines the representative columns as in the proposed method but then assigns them on the cells of the counter in a random order. Comparing our method with this method allows us to evaluate the impact of the $w(\text{in}_i)$ weights on quality of the solution. Note that the RAM approach already provides significant savings over the brute–force approach that does not consider the operations of Section II. Such savings are not reported here simply because it is very time consuming to simulate the power consumption with the brute–force approach. Due to the huge number of patterns required by the latter method, the power consumption will also be huge.

The first column of Table 1 gives the name of each benchmark. Column 2, labeled p, gives the number of patterns p that were embedded for each benchmark. These patterns were provided by Sunrise Inc. for all the hard to detect faults.

Column 3, labeled PM (proposed metric), gives information on the total power when the counter TPG was designed with the method proposed in this paper. It reports the % savings on the heat dissipation obtained by our method over the RAM approach. The results clearly show the impact of the $w(\text{in}_i)$ weights.

Columns 4 and 5 are labeled FM (first metric) and SM (second metric) and they list the % savings on the heat dissipation obtained by these two methods over the RAM approach.

The results clearly show the superiority and impact of the proposed method. None of the other methods is consistently good. There are instances where each of these methods may produce 1/3 more heat dissipation than our method.

Although space limitations do not allow us to list and analyze our experiments in more details, we note that we have observed several instances where the test reproduction time as well as the number of transitions on the inputs obtained by some metric is much higher than the respective values of another and at the same time the total heat dissipation is reduced. For example, metric FM requires 209 cycles to embed the patterns for c3540 and its heat dissipation is 209,771. In contrast, our metric PM requires 216 cycles but the heat dissipation is reduced to 206,342. These observations show that it is dangerous to use the methods in [3] when heat dissipation is a concern.

circuit	p	PM	FM	SM
c432	6	46,3 %	22,7 %	40,2 %
c499	14	19,2 %	6,1 %	19,2 %
c880	17	62,9 %	57,2 %	54,1 %
c1908	22	51,1 %	29,9 %	49,1 %
c3540	22	29,7 %	28,6 %	20,4 %
c5315	7	35,9 %	41,2 %	16 %

Table 1. Experimental comparisons

In benchmark c5315, metric FM performed slightly better than ours. We observed that this is due to reduced test embedding time as well as reduced number of transitions on the inputs.

V. CONCLUSIONS

We have presented a method for synthesizing a counter in order to reproduce on chip a set of precomputed test patterns so that the total heat dissipation is minimized. The listed results show that the presented approach is promising.

References

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

[2] P.H. Bardell, W.H. McAnney, and J. Savir, Built-In Test for VLSI: Pseudorandom Techniques, John Wiley and Sons, 1987.

[3] D. Kagaris, S. Tragoudas and A. Majumdar, "On the Use of Counters for Reproducing Deterministic Test Sets", IEEE Transactions on Computers, vol. 45, no. 12, pp.1405–1419, December 1996.

[4] D. Kagaris and S. Tragoudas, "On the Design of Optimal Counter–based Schemes for Test Set Embedding", IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems (IEEE-TCAD), to appear.

[5] R.J. Russell, "A Method of Extending an 1149.1 Bus for Mixed Signal Testing", Proceedings of the International Test Conference, pp. 410–416, 1996.

[6] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices", Proceedings VLSI Test Symposium, pp. 4–9, 1993.

[7] S. Wang and S.K. Gupta, "DS-LFSR: A New BIST TPG for Low Heat Dissipation", Proceedings International Test Conference, pp. 848–857, 1997.