Energy Minimization for Hybrid BIST in a System-on-Chip Test Environment

Raimund Ubar, Tatjana Shchenova
Department of Computer Engineering
Tallinn University of Technology, Estonia
raiub@pld.ttu.ee, tatjana.shchenova@kreenholm.ee

Gert Jervan, Zebo Peng
Embedded Systems Laboratory (ESLAB)
Linköping University, Sweden
{gerje, zebpe}@ida.liu.se

Abstract[1]

This paper addresses the energy minimization problem for system-on-chip testing. We assume a hybrid BIST test architecture where a combination of deterministic and pseudorandom test sequences is used. The objective of our proposed technique is to find the best ratio of these sequences so that the total energy is minimized and the memory requirements for the deterministic test set are met without sacrificing test quality. We propose two different heuristic algorithms and a fast estimation method that enables considerable reduction of the computation time. Experimental results have shown the efficiency of the approach for finding reduced energy solutions with low computational overhead.

1. Introduction

The latest advance in microelectronics technology has enabled the integration of an increasingly large number of transistors into a single die. This has imposed a major production challenge, due to the increased density of such chips and the increased power dissipation. At the same time the number of portable, battery operated devices (such as laptops, PDA-s, and mobile phones) is rapidly increasing. These devices require advanced methods for reducing power consumption in order to prolong the life of the batteries and thus increase the length of the operating periods of the system. There are several well investigated techniques for handling power dissipation during the normal operation. And various research has shown that the switching activity, and consequently the power dissipation, during the test mode may be several times higher than during the functional mode [1], [2]. The self-tests, regularly executed in portable devices, can hence consume significant amounts of energy and consequently reduce the lifetime of the batteries [3]. Excessive switching activity during the test mode can also cause problems with circuit reliability [4]. And the increased current levels can lead to serious silicon failure mechanisms (such as electromigration [5]) and may need expensive packages for removal of the excessive heat. Therefore, it is important to find ways for reducing circuit power dissipation during the testing process.

There are several components contributing to the power consumption of standard CMOS technology: dynamic power dissipation caused by the switching activity, and static power dissipation caused mainly by leakage. The leaks contribute usually only marginally to the total power consumption and can therefore be neglected. The main contributing factor remains to be dynamic power dissipation caused by switching of the gate outputs. This activity accounts for more than 90% of the total power dissipation for current technology, even though the importance of static power dissipation will increase with the scaling down of feature sizes [6]. For every gate the dynamic power, P_d, required to charge and discharge the circuit nodes, can be calculated as follows:

$$P_d = 0.5 \times C_{load} \times (V_{DD}^2 / T_{cyc}) \times N_G \qquad (1)$$

where C_{load} is the load capacitance, V_{DD} is the supply voltage, T_{cyc} is the global clock period, and N_G is the switching activity, i.e., the number of gate output transitions per clock cycle.

While assuming that the V_{DD} as well as T_{cyc} remain constant during the test and that the load capacitance for each gate is equal to the number of fan-outs of this gate, we can define switching activity as a quantitative measure for power dissipation. Therefore, the most straightforward way to reduce the dynamic power dissipation of the circuit during test is to minimize the circuit's switching activity.

Several approaches have been proposed to handle the power issues during test application. They can be divided into three categories: energy, average power and peak power reduction techniques. Energy reduction techniques aim at the reduction of the total switching activity generated in the circuit during the test application and have thus impact on the battery lifetime [7]-[10]. Average power dissipation is the amount of dissipated energy divided over the test time. The reduction of average power dissipation can improve the circuit's reliability by reducing temperature and current density. Some of the methods to reduce average power dissipation have been proposed in [11] and [12]. The peak power corresponds to the maximum sustained power in a circuit. The peak power determines the thermal and electrical limits of the components and the system packaging requirements. If the peak power exceeds certain limits, the correct functioning of the entire circuit is no longer guaranteed. Methods for peak power reduction include those described in [13]-[17].

In a System-on-Chip testing environment, several test power related problems are handled at the core level, with the methods described above. However, the high degree of parallelism in SoCs facilitates parallel testing to reduce the test application time. Consequently, this might also lead to excessive power dissipation. In such cases the system-wide peak power values can be controlled with intelligent test scheduling and this problem has been studied by many authors [18]-[21].

In this paper we focus on total test energy minimization for SoC testing. We assume a hybrid built-in self-test (BIST) test

[1] This work has been supported by the EU projects REASON (IST-2000-30193) and e-Vikings II (IST-2001-37592), Estonian Science Foundation grants G5649 and G5910, and the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program.

architecture, where the test set is composed of core-level locally generated pseudorandom patterns and additional deterministic test patterns that are generated offline and stored in the system. The exact composition of these patterns defines not only the test length and test memory requirements but also the energy consumption. In general, since a deterministic test pattern is more effective in detecting faults than a pseudorandom pattern, using more deterministic test patterns for a core will lead to a short test sequence, and consequently less energy on the average case. However, the total number of deterministic test patterns is constrained by the test memory requirements, and at the same time, the deterministic test patterns of different cores of a SoC have different energy and fault detection characteristics. A careful trade-off between the deterministic pattern lengths of the cores must therefore be made in order to produce a globally optimal solution. In this paper we propose two heuristics that try to minimize the total switching energy while taking into account the assumed test memory constraint. The solutions are obtained by modifying the ratio of pseudorandom and deterministic test patterns for every individual core such that the total energy dissipation is minimized.

The rest of this paper is organized as follows. In the next section an overview of the hybrid BIST technique and the proposed approach is given. Section 3 is devoted to basic definitions and Section 4 describes the fast estimation mechanism used in hybrid BIST calculations. Section 5 describes the proposed heuristics for energy minimization and in Section 6 the experimental results are presented. The paper is concluded in Section 7.

2. Hybrid BIST and Energy Reduction

Several BIST solutions have been proposed in the literature for performing test pattern generation and output response compaction on the chip, by using pseudorandom patterns. Due to several reasons, like very long test sequences, and random pattern resistant faults, these solutions may not always be efficient. Therefore different hybrid approaches have been proposed, where pseudorandom test patterns are complemented with a set of deterministic test patterns. These approaches are generally referred to as hybrid BIST [22]-[26]. Such a hybrid approach reduces the memory requirements compared to the pure deterministic testing, while providing higher fault coverage and reduced test times compared to the stand-alone BIST solution.

In the current work we have assumed a hybrid BIST test architecture where all cores have their own dedicated BIST logic that is capable to produce a set of independent pseudorandom test patterns, i.e. the pseudorandom tests for all cores can be carried out concurrently. The deterministic tests, on the other hand, are applied from an on-chip memory, one core at a time. And we have also assumed for test data transportation an AMBA-like test bus [27]. AMBA (Advanced Microcontroller Bus Architecture) integrates an on-chip test access technique that reuses the basic bus infrastructure [28]. An example of a multi-core system with such test architecture is given in Figure 1.

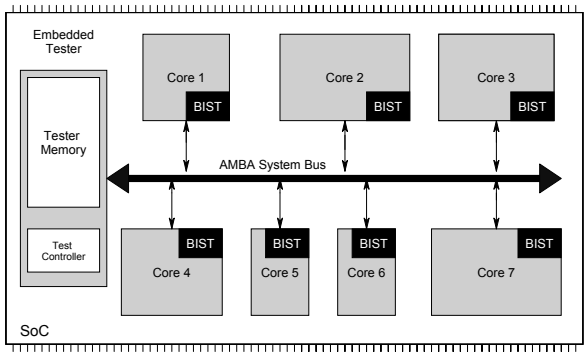


Figure 1. AMBA bus-based hybrid BIST architecture

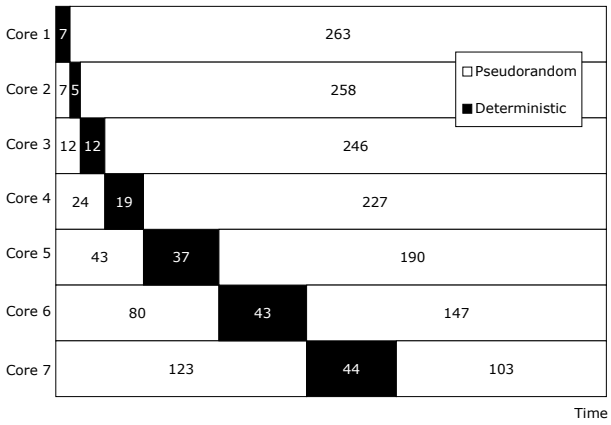


Figure 2. Time minimized test schedule

For portable systems with such a test architecture, one of the most important test constraints is the total amount of on-chip test memory. In [23] and [29] we have proposed methods for test time minimization under given test memory constraint for test-per-clock and test-per-scan schemes. If the objective is only test time minimization and power/energy is not taken into account then the shortest test schedule for such a test architecture (Figure 1), is the one where all cores are tested concurrently and have the same tests lengths, as depicted in Figure 2.

In a hybrid BIST approach the test set is composed of pseudorandom and deterministic test patterns, where the ratio of these patterns is defined by different design constraints, such as test memory and test time. In general, a shorter pseudorandom test set implies a larger deterministic test set. This requires additional memory space, but at the same time, shortens the overall test process, since deterministic test vectors are more effective in covering faults than the pseudorandom ones. A longer pseudorandom test, on the other hand, will lead to longer test application time with reduced memory requirements [22]. From an energy perspective, different cores have different energy dissipation while applying the same amount of test patterns. Furthermore, the pseudorandom and deterministic test sequences for the same core have different power characteristics. Therefore for total energy minimization it is important to find, for every individual core, such ratio of the pseudorandom and deterministic test patterns that leads to the overall reduction of switching energy. At the same time the basic design constraints, such as test memory, should not be violated. Once a low-energy hybrid BIST solution has been found, appropriate test scheduling methods can be used for managing peak power related problems. In the following some basic definitions together with the problem formulation are given.

3. Basic Definitions and Problem Formulation

Let us assume that a system S consists of n cores $C_1, C_2, \ldots, C_n$. For every core $C_k \in S$ a complete sequence of deterministic test patterns TD^F_k and a complete sequence of pseudorandom test

patterns TP^F_k will be generated. It is assumed that both test sets can obtain by itself maximum achievable fault coverage F_{max}.

Definition 1: A *hybrid BIST set* $TH_k = \{TP_k, TD_k\}$ for a core C_k is a sequence of tests, constructed from a subset of the complete pseudorandom test sequence $TP_k \subseteq TP^F_k$, and a subset of the deterministic test sequence $TD_k \subseteq TD^F_k$. The test sequences TP_k and TD_k complement each other to achieve the maximum achievable fault coverage, and define the hybrid test set TH_k.

By knowing the length of the pseudorandom test sequence TP_k we can always find the amount of additional deterministic patterns TD_k [22]. Therefore we can say that the pseudorandom test sequence TP_k uniquely defines the structure of the entire hybrid test set.

Definition 2: $J = (j_1, j_2, \ldots, j_n)$ is called the *characteristic vector* of a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$, where $j_k \in J$ is the length of the pseudorandom test sequence $TP_k \subseteq TH_k$.

According to the above definitions, for each j_k corresponds a pseudorandom subsequence $TP_k(j_k) \subseteq TP^F_k$. In order to form a hybrid test sequence TH_k this subsequence is complemented with a deterministic test sequence, denoted with $TD_k(j_k)$, that is generated such that the hybrid sequence TH_k reaches to the maximal achievable fault coverage. Based on this we can conclude that the characteristic vector J determines entirely the structure of the hybrid test set TH_k for all cores $C_k \in S$.

***Definition 3*:** Let us denote with $M_k(j_k)$ and $E_k(j_k)$ respectively the memory cost and energy cost of the hybrid BIST set $TH_k = \{TP_k, TD_k\}$ of the core $C_k \in S$ as functions of its pseudorandom test sequence with length j_k.

Note that it is very time consuming to calculate the exact values of $M_k(j_k)$ and $E_k(j_k)$ for any arbitrary hybrid BIST set TH_k, since it requires exact calculation of the corresponding hybrid test set which is an expensive procedure [23]. To overcome the problem we propose in this paper to use an energy estimation method that is based only on a few critical point calculations.

Definition 4: Let us denote with $M(J)$ and $E(J)$ respectively the memory cost and energy cost of the corresponding hybrid BIST set TH with characteristic vector J. These costs can be calculated using the following formulas:

$$M(J) = \sum_{k=1}^{n} M_k(j_k) \qquad E(J) = \sum_{k=1}^{n} E_k(j_k) \qquad (2)$$

A hybrid BIST set $TH = \{TH_1, TH_2, \ldots, TH_n\}$ for a system $S = \{C_1, C_2, \ldots, C_n\}$ consists of hybrid BIST sets TH_k for each individual core C_k. In our approach the pseudorandom components of the TH are going to be scheduled in parallel, while the deterministic components of the TH, based on the given test architecture (Figure 1), have to be scheduled in sequence.

The objective of this paper can be thus formulated as to find a hybrid test set TH with a characteristic vector J for the system S, such that $E(J)$ is smallest possible and the memory constraint $M(J) \le M_{LIMIT}$ is satisfied.

Next, we are going to describe the estimation method that is going to be used for calculating the hybrid BIST structure.

4. Parameter Estimation

For hybrid BIST energy minimization at a given memory constraint we have to obtain values of $M_k(j_k)$ and $E_k(j_k)$ for every core $C_k \in S$ and for any possible j_k value. This would give us a possibility to compare memory and energy values of the different alternatives. However, the procedure to calculate the cost functions $M(J)$ and $E(J)$ is very time consuming, since it assumes that the deterministic test sets TD_k for all possible values of the characteristic vector J are available. This means that for each possible pseudorandom test TP_k, a set of not yet detected faults $F_{NOT}(TP_k)$ should be calculated, and the needed deterministic test set TD_k has to be found. This can be done either by repetitive use of the automatic test pattern generator or by systematically analyzing and compressing the fault tables for each TP_k [30]. Both procedures are time-consuming and therefore not feasible for larger designs.

To overcome the complexity explosion problem we propose an iterative algorithm, where the costs $M(J)$ and $E(J)$ for the deterministic test sets TD_k are calculated based on estimates in a similar way as described in [23]. The estimation method is based on fault coverage figures and does not require accurate calculations of the deterministic test sets for not yet detected faults $F_{NOT}(TP_k)$.

The estimation method is described in Algorithm 1 given below. We will use $FD_k(i)$ and $FP_k(i)$ to denote the fault coverage figures of the test sequences $TD_k(i)$ and $TP_k(i)$, correspondingly, where i is the length of the test sequence. Also a pattern in a pseudorandom test sequence will be called *effective pattern* if it detects at least one new fault that is not detected by the previous test patterns in the same sequence.

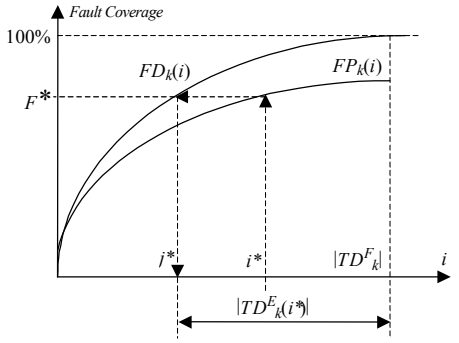
Algorithm 1: Estimation of the length of the deterministic test set TD_k.

1. Calculate, by fault simulation, the fault coverage functions $FD_k(i)$, $i = 1, 2, \ldots, |TD^F_k|$, and $FP_k(i)$, $i = 1, 2, \ldots, |TP^F_k|$. The patterns in TD^F_k are ordered in such a way that each pattern put into the sequence contribute with maximum increase in fault coverage.
2. For each $i^* \le |TP^F_k|$, find the fault coverage value F^* that can be reached by the pseudorandom test sequence with length i^* (see Figure 3a).
3. By solving the equation $FD_k(i) = F^*$, find the maximum integer value j^* that satisfies the condition $FD_k(j^*) \le F^*$. The value of j^* is the length of the deterministic sequence TD_k that can achieve the same fault coverage F^*.
4. Calculate the value of $|TD^E_k(i^*)| = |TD^F_k| - j^*$ which is the number of test patterns needed in addition to the pseudorandom patterns to reach to the maximum achievable fault coverage.

The value $|TD^E_k(i^*)| = |TD^F_k| - j^*$, calculated by Algorithm 1, can be used to estimate the length of the deterministic test sequence TD_k in the hybrid test set $TH_k = \{TP_k, TD_k\}$ with i^* efficient test patterns in TP_k, ($|TP_k| = i^*$).

The algorithm is illustrated with the example given in Figures 3b and 3c. In the given example, if $i^* = 524$ then the fault coverage is $F^* = 97.5\%$. From the fault simulation table (Figure 3b) we can find that similar fault coverage can be achieved by 60 deterministic test patterns ($j^* = 60$). Based on the fourth step of Algorithm 1 we can say that it takes approximately 90-60=30 deterministic test patterns in addition to the 524 pseudorandom test patterns in order to reach the maximum achievable fault coverage.

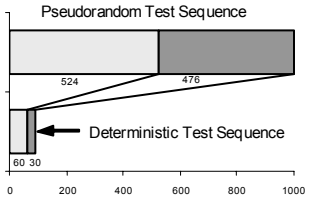
Based on the created relationships between the pseudorandom test sequence length $|TP_k|$ and estimated deterministic test length $|TD^E_k|$ we can easily solve the equation $|TD^E_k| = f(|TP_k|)$ also in the opposite way when TD^E_k is given and $|TP_k|$ has to be found.



a)

|TP|	FC%	|TD|	FC%
1	21.9	1	43.3
2	34.7	2	45.6
…			
524	97.5	60	97.5
…			
1000	98.9	90	100

b)



c)

Figure 3. Estimation of the length of the deterministic test sequence

These calculations will be used in the next section, where the energy minimization heuristics will be described.

5. Heuristic Algorithms for Hybrid BIST Energy Minimization

To minimize the energy consumption at the given memory constraint we have to create a hybrid test *TH* with characteristic vector *J* for the system *S*, so that $E(J)$ is minimal at the constraint $M(J) \le M_{LIMIT}$.

To solve this complex combinatorial task we propose two fast heuristic algorithms: Local Gain Algorithm and Average Gain Algorithm. Both are based on the estimation methodology described in Algorithm 1.

5.1. Local Gain Algorithm

The main idea of this algorithm is to start with pure deterministic test sets $TH_k = \{TP_k = \varnothing, TD^F_k\}$ for all cores $C_k \in S$. Next, the deterministic test patterns are gradually substituted by corresponding sequences of pseudorandom patterns $PR_i \subseteq TP^F_k$ until the memory constraint is satisfied. For every deterministic test pattern substitution a core $C_k \in S$ with the maximum memory-energy ratio ($\Delta M_{k,i}/\Delta E_{k,i}$) is selected. Here $\Delta M_{k,i}$ corresponds to the estimated memory gain when deterministic test pattern $DP_i \in TD^F_k$ is removed from the memory, and $\Delta E_{k,i}$ corresponds to the estimated increase in energy dissipation by the sequence of pseudorandom patterns $PR_i \subseteq TP^F_k$ that are substituting the deterministic test pattern $DP_i \in TD^F_k$. In other words, at any iteration of the algorithm we always select a core that provides the best local gain in terms of $\Delta M_{k,i}/\Delta E_{k,i}$ and substitute, in the hybrid test set of this core, one deterministic test pattern with appropriate number of pseudorandom patterns. The number of inserted pseudorandom test patterns is calculated so that the fault coverage of the core is not reduced and thus the maximum achievable fault coverage is always guaranteed.

Let us introduce the following additional notations: M – current memory cost, L – current pseudorandom test length, and M_{LIMIT} – memory constraint. The algorithm starts with initial values: $L = 0$, and $M = M(TD^F_1) + M(TD^F_2) + \ldots + M(TD^F_n)$ where $M(TD^F_k)$ is memory cost of the complete deterministic test set of core $C_k \in S$. Initially: $TH_k = \{TP_k = \varnothing, TD^F_k\}$.

Algorithm 2: Local Gain Algorithm.

1. Select core $C_k \in S$ where $\Delta M_{k,i}/\Delta E_{k,i} = \max$;
2. Remove $DP_{k,i} \in TD_k$ from TD_k, estimate the needed PR_i and include PR_i into TP_k.
3. Update the current memory cost: $M = M - \Delta M_{k,i}$
4. If $M > M_{LIMIT}$ then go to 1
5. *END.*

The algorithm is illustrated with the example given in Figure 4. We start from an all-deterministic solution. At every step we calculate the memory-energy ratio for all cores if one deterministic test pattern (denoted as white boxes in Figure 4a) would be replaced with pseudorandom patterns. Thereafter the core with highest $\Delta M_{k,i}/\Delta E_{k,i}$ value is selected and a deterministic test pattern in this core's test set is replaced with a set of pseudorandom patterns. In our example the Core 3 was selected (Figure 4b). At the end of every step we can calculate new memory (M) and energy (E) values for the entire system. This procedure is repeated until $M \le M_{LIMIT}$.

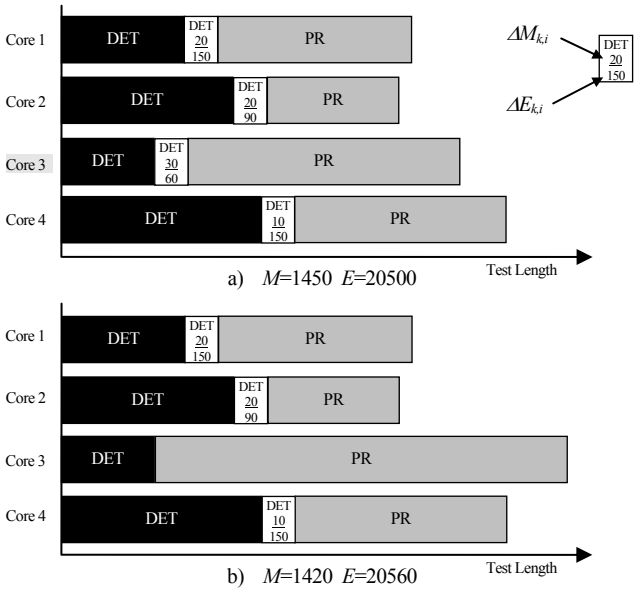


Figure 4. Local Gain Algorithm

5.2. Average Gain Algorithm

Another heuristic is called Average Gain Algorithm. The main idea of the Average Gain Algorithm is to guide the selection of cores based on the highest average ratio of $\Delta M_k/\Delta E_k$ over all iterations of the algorithm. Here ΔM_k denotes the estimated memory gain from the beginning of the algorithm, including the selected substitution, for the core $C_k \in S$, and ΔE_k denotes the estimated increase of energy dissipation for the same core from the beginning of the algorithm, including the current selected substitution.

The algorithm starts again with initial values: $L = 0$, and $M = M(TD^F_1) + M(TD^F_2) + \ldots + M(TD^F_n)$ where $M(TD^F_k)$ is the memory cost of the complete deterministic test set of the core

$C_k \in S$. Initially: $TH_k = \{TP_k = \varnothing, TD^F_k\}$. For all cores $\Delta M_k = \Delta M_{k,1}$, and $\Delta E_k = \Delta E_{k,1}$, and for all cores $i = 1$.

Algorithm 3: Average Gain Algorithm.

1. Select the core $C_k \in S$ where $\Delta M_{k,i}/\Delta E_{k,i} = \max$;
2. Remove $DP_{k,i} \in TD_k$ from TD_k, and include PR_i into TP_k.
3. Update the current memory cost: $M = M - \Delta M_{k,i}$.
4. Update the total memory cost for the selected core: $\Delta M_k = \Delta M_k - \Delta M_{k,i+1}$.
5. Update the total energy dissipation for the selected core: $\Delta E_k = \Delta E_k + \Delta E_{k,i+1}$.
6. Update for the selected core $i = i + 1$.
7. If $M > M_{LIMIT}$ then go to 1
8. *END*

The main difference between Algorithms 2 and 3 is that Algorithm 2 takes into account only the immediate effect of the test pattern substitution. Algorithm 3, on the other hand, takes into account the entire history of pattern substitutions.

5.3. The iterative procedure

Both Algorithms 2 and 3 create an energy-optimized hybrid BIST solution $TH_k = \{TP_k , TD_k\}$ where energy consumption is minimized with respect to the given memory constraint. However, the algorithms are based on estimated memory and energy values for TD_k, and therefore also the final result is only an estimate. We refer them as quasi-optimal solutions. After obtaining a quasi-optimal solution the cost estimates should be improved and another, better, quasi-optimal solution can be calculated which leads us to the following iterative procedure [23].

Algorithm 4: Iterative procedure.

The algorithm uses as a starting point the result obtained with Algorithm 2 or 3. Let us denote the estimated memory cost with M^*.

1. Based on M^*, find a candidate solution $J^* = (j^*_1, j^*_2, \ldots, j^*_n)$.
2. To calculate the real cost M_{REAL} for the candidate solution J^*, find the set $F_{NOT,k}(j^*_k)$ of faults not yet detected by pseudorandom test for each core $C_k \in S$ and generate the corresponding deterministic test set TD^*_k by using an ATPG algorithm.
3. If $M_{REAL} = M_{LIMIT}$, go to the Step 6.
4. If the difference $|\Delta t| = |M_{LIMIT} - M_{REAL}|$ is bigger than that in the earlier iteration make a correction $\Delta t = \Delta t/2$.
5. Calculate a new constraint $M' = M' + \Delta t$ for the next iteration, go to Step 2
6. *END:* The vector $J^* = (j^*_1, j^*_2, \ldots, j^*_n)$ is the solution.

Algorithm 4 transfers a solution that was generated based on estimates to the solution that is calculated based on real test sets. This iterative algorithm is illustrated in Figure 5. It is easy to see that Algorithm 4 always converges. By each iteration we get closer to the memory constraint level, and also closer to the minimal energy consumption at a given constraint. The outcome of this algorithm is the final solution: amount of pseudorandom and deterministic test patterns for every individual core such that the system memory constraint is satisfied.

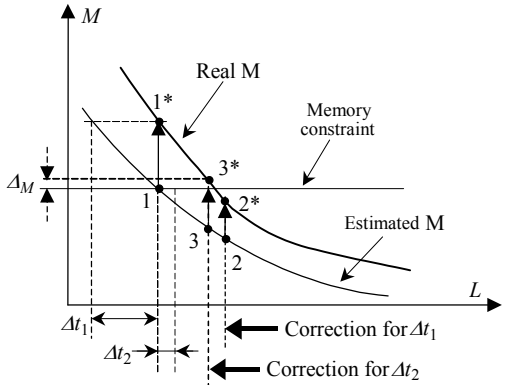


Figure 5. Calculation of the final result

6. Experimental results

We have performed experiments with different designs containing the ISCAS'89 benchmarks as cores. The complexity of these designs ranges from system with 6 cores till system with 20 cores. All cores were redesigned in order to include a scan chain. For simplicity we assumed that all flip-flops are connected into one single scan chain. For the BIST part a STUMPS architecture was used.

In Table 1 we have listed the results for every system with three different memory constraints. We have listed results from [23], which provide shortest possible test length without considering energy consumption, our two algorithms (Local Gain Algorithm is denoted with A2 and Average Gain Algorithm with A3, both improved with Algorithm 4) and simulated annealing (SA). In every solution the minimized test time solution from [23] has taken as a baseline (100%) and every solution is compared against this result.

Alg.	M_{LIMIT}	Energy (switches)	Comp. to [23] (%)	Test Length (clocks)	Comp. to [23] (%)	CPU Time (sec)
System 1 – 6 cores						
[23]	1500	2588822	100.00%	24689	100.00%	8.41
A2		1281690	49.51%	31619	128.07%	11.09
A3		1281690	49.51%	31619	128.07%	6.64
SA		1240123	47.90%	31619	128.07%	5326.24
[23]	2500	635682	100.00%	6726	100.00%	24.61
A2		426617	67.11%	10559	156.99%	14.23
A3		446944	70.31%	10679	158.77%	4.84
SA		409576	64.43%	10529	156.54%	2944.26
[23]	3000	717026	100.00%	7522	100.00%	26.51
A2		265282	37.00%	8126	108.03%	36.31
A3		286883	40.01%	8129	108.07%	26.96
SA		241123	33.63%	8153	108.39%	1095.21
System 2 – 6 cores						
[23]	1700	6548659	100.00%	52145	100.00%	12.05
A2		5502763	84.03%	70331	134.88%	12.49
A3		5318781	81.22%	70331	134.88%	4.28
SA		4747498	72.50%	83865	160.83%	3805.23
[23]	3000	2315958	100.00%	19208	100.00%	20.21
A2		1998390	86.29%	23774	123.77%	7.66
A3		1861844	80.39%	24317	126.60%	18.79
SA		1845022	79.67%	28134	146.47%	5032.05
[23]	4700	893184	100.00%	8815	100.00%	21.47
A2		742462	83.13%	9537	108.19%	26.45
A3		746479	83.58%	9537	108.19%	55.09
SA		723817	81.04%	12596	142.89%	3654.02
System 3 – 20 cores						
[23]	5000	12830419	100.00%	40941	100.00%	47.49
A2		9242890	72.04%	70331	171.79%	51.43
A3		9839005	76.68%	70331	171.79%	40.49
SA		7367201	57.42%	60495	147.76%	29201.96
[23]	7000	6237211	100.00%	20253	100.00%	53.39
A2		4039622	64.77%	31619	156.12%	73.58
A3		4223263	67.71%	32145	158.71%	14.36
SA		3500894	56.13%	31919	157.60%	20750.03
[23]	10000	4686729	100.00%	15483	100.00%	45.37
A2		1719726	36.69%	17499	113.02%	115.53
A3		1815129	38.73%	17554	113.38%	90.52
SA		1606499	34.28%	17992	116.20%	14572.33

Table 1. Experimental results

As shown in Table 1 both proposed algorithms lead to reduced energy solutions (in some cases up to with 63% reduction of the total switching activity). When compared to the simulated annealing algorithm our heuristics have significantly lower execution time, while maintaining acceptable accuracy.

To understand the impact of our algorithms on the test length we have also collected these figures and reported them in Table 1. As can be expected in all these solutions generated by our techniques the test time has increased compared to the technique which targets towards test length minimization [23]. Nevertheless if the main objective is to reduce energy dissipation during the test mode (for example in portable devices) the increase of the test length is tolerable. A future work is to design a technique to make trade-offs of all parameters, including energy, power, test time, and test memory requirements.

The experiments show also that the number of iterations made by Algorithm 4 was in all cases below 10, illustrating the efficiency of the proposed estimation method.

7. Conclusions

In this paper we have proposed two heuristics for test energy reduction for hybrid BIST. Both algorithms modify the ratios between pseudorandom and deterministic test patterns. We have also proposed a fast estimation mechanism for the modification of these rations together with an iterative procedure for transforming the estimated results to the real results. Experimental results have shown the efficiency of these heuristics for energy reduction under test memory constraints.

References

[1] Y. Zorian. A Distributed BIST Control Scheme for Complex VLSI Devices. *IEEE VLSI Test Symposium*, pp. 4-9, 1993.

[2] S. Devadas, M. Malik. A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. *ACM/IEEE Design Automation Conference,* pp. 242-247, 1995.

[3] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch. A Test Vector Inhibiting Technique for Low Energy BIST Design. *VLSI Test Symposium*, pp. 407 – 412, 1999.

[4] P. Girard. Low Power Testing of VLSI Circuits: Problems and Solutions. *IEEE International Symposium on Quality Electronic Design*, pp. 173 – 179, 2000.

[5] P.C. Li, T.K. Young. Electromigrations: The Time Bomb in Deep-Submicron ICs. *IEEE Spectrum*, vol. 33, no. 9, pp. 75-78, 1996.

[6] A. Chandrakasan, T. Sheng, R. W. Brodersen. Low Power CMOS Digital Design. *Journal of Solid State Circuits,* Vol. 27, No. 4, pp. 473–484, 1992.

[7] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch. A Test Vector Inhibiting Technique for Low Energy BIST Design. *VLSI Test Symposium*, pp. 407-412, 1999.

[8] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, H.J. Wunderlich, H.J. A Modified Clock Scheme for a Low Power BIST Test Pattern Generator. *VLSI Test Symposium*, pp. 306 – 311, 2001.

[9] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, J. Figueras, S. Manich, P. Teixeira, M. Santos. Low-Energy BIST Design: Impact of the LFSR TPG Parameters on the Weighted Switching Activity. *International Symposium on Circuits and Systems*, pp 110-113, 1999.

[10] D. Gizopoulos, N. Kranitis, A. Paschalis, M. Psarakis, Y. Zorian. Low Power/Energy BIST Scheme for Datapaths. *VLSI Test Symposium*, pp. 23-28, 2000.

[11] S. Wang, S. Gupta. ATPG for Heat Dissipation Minimization during Test Application. *IEEE Transactions on Computers*, Vol. 46, No. 2, pp. 256-262, 1998.

[12] S. Chakravarty, V. Dabholkar. Minimizing Power Dissipation in Scan Circuits During Test Application. *IEEE International Workshop on Low Power Design*, pp. 51-56, 1994.

[13] S. Gerstendorfer, H.J. Wunderlich. Minimized Power Consumption for Scan-based BIST. *IEEE International Test Conference*, pp. 77-84, 1999.

[14] L. Whetsel. Adapting Scan Architectures for Low Power Operation. *International Test Conference*, pp. 863-872, 2000.

[15] R. Sankaralingam, B. Pouya, N. A. Touba. Reducing Power Dissipation During Test Using Scan Chain Disable. *VLSI Test Symposium*, pp. 319-324, 2001.

[16] P. M. Rosinger, B. M. Al-Hashimi, N. Nicolici. Scan Architecture for Shift and Capture Cycle Power Reductions, *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 129-137, 2002.

[17] N. Z. Basturkmen, S. M. Reddy, I. Pomeranz. A Low Power Pseudo-Random BIST Technique, *IEEE International On Line Testing Workshop*, pp. 140-144, 2002.

[18] E. Cota, L. Carro, M. Lubaszewski, A. Orailoglu. Test Planning and Design Space Exploration in a Core-based Environment. *Design, Automation and Test in Europe Conference*, pp. 478-485, 2002.

[19] Y. Huang, W.-T. Cheng, C.-C. Tsai, N. Mukherjee, O. Samman, Y. Zaidan, S. M. Reddy. Resource Allocation and Test Scheduling for Concurrent Test of Core-based SOC Design. *IEEE Asian Test Symposium*, pp. 265-270, 2001.

[20] V. Iyengar, K. Chakrabarty, E. J. Marinissen. Test Wrapper and Test Access Mechanism Co-optimization for System-on-Chip. *IEEE International Test Conference*, pp. 1023-1032, 2001.

[21] E. Larsson, Z. Peng. An Integrated Framework for the Design and Optimization of SOC Test Solutions. *Journal of Electronic Testing; Theory and Applications*, Vol. 18, No. 4/5, pp. 385-400, 2002.

[22] G. Jervan, Z. Peng, R. Ubar. Test Cost Minimization for Hybrid BIST. *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 283-291, 2000.

[23] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin. Test Time Minimization for Hybrid BIST of Core-Based Systems. *IEEE Asian Test Symposium*, pp. 318-323, 2003.

[24] M. Sugihara, H. Date, H. Yasuura. Analysis and Minimization of Test Time in a Combined BIST and External Test Approach. *Design, Automation and Test in Europe Conference*, pp, 134-140, 2000.

[25] S. Hellebrand, S. Tarnick, J. Rajski, B. Courtois. Generation of Vector Patterns through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers. *IEEE International Test Conference*, pp. 120-129, 1992.

[26] N. A. Touba, E. J. McCluskey. Synthesis of Mapping Logic for Generating Transformed Pseudo-random Patterns for BIST. *IEEE International Test Conference*, pp. 674-682, 1995.

[27] D. Flynn. AMBA: Enabling Resuable On-Chip Designs. *IEEE Micro*, Vol. 17, No. 4, pp. 20-27, 1997.

[28] P. Harrod. Testing Reusable IP – A Case Study. *IEEE International Test Conference*, pp. 493-498, 1999.

[29] G. Jervan, P. Eles, Z. Peng, R. Ubar, M. Jenihhin. Hybrid BIST Time Minimization for Core-Based Systems with STUMPS Architecture. *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 225-232, 2003.

[30] G. Jervan, Z. Peng, R. Ubar, H. Kruus. A Hybrid BIST Architecture and its Optimization for SoC Testing. *IEEE International Symposium on Quality Electronic Design*, pp. 273-279, 2002.