Hybrid BIST energy minimisation technique for system-on-chip testing

G. Jervan, Z. Peng, T. Shchenova and R. Ubar

Abstract: The energy minimisation problem for system-on-chip testing is addressed. A hybrid built-in self-test architecture is assumed where a combination of deterministic and pseudorandom test sequences are used. The objective of the proposed technique is to find the best ratio of these sequences so that the total energy is minimised and the memory requirements for the deterministic test set are met without sacrificing test quality. Unfortunately, exact algorithms for finding the best solutions to the above problem are computationally very expensive. Therefore, an estimation methodology for fast calculation of the hybrid test set and two different heuristic algorithms for energy minimisation were proposed. Experimental results have shown the efficiency of the proposed 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, PDAs 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 their operating periods. There are several well-investigated techniques for handling power dissipation during the normal operation of such devices. And various researches have 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]. Increased current levels can lead to serious silicon failure mechanisms (such as electromigration [5]) and may need expensive packages for the 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 that contribute to the power consumption of standard CMOS technology, including 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 is the 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

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

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

While assuming that the V_{DD} as well as T_{cyc} remain constant during testing and that the load capacitance for each gate is proportional to the number of its fan-outs, 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 testing is to minimise 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 whole test application process and have thus impact on the battery lifetime [7–9]. 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 by Wang and Gupta [10] and Chakravarty and Dabholkar [11]. 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 are described in various studies [12–16].

IEE Proceedings online no. 20050064

doi:10.1049/ip-cdt:20050064

Paper first received 1st April and in revised form 31st October 2005

G. Jervan, T. Shchenova and R. Ubar are with the Department of Computer Engineering, Tallinn University of Technology, Estonia

Z. Peng is with the Embedded Systems Laboratory (ESLAB), Linköping University, Sweden

E-mail: gerje@pld.ttu.ee

In a system-on-chip (SoC) testing environment, several test-power related problems are usually handled at the core level. However, the high degree of parallelism in SoCs facilitates parallel testing to reduce the test application time, and might also lead to excessive power dissipation. In such cases, the system-wide peak power values can be controlled with intelligent test scheduling [17–20].

We focus on total test energy minimisation for SoC testing. We assume a hybrid built-in self-test (BIST) architecture, where the test set for each core is composed of core-level locally generated pseudorandom patterns and additional deterministic test patterns that are generated offline and stored in the system. Here, the lengths of the pseudorandom tests are an important design parameter that determines the behaviour of the whole test process. For a given core, a shorter pseudorandom test sequence implies a larger deterministic test set. This requires additional memory space, but at the same time, shortens the overall test time. A longer pseudorandom test, in contrast, will lead to a larger test application time with reduced memory requirement.

The exact composition of these test patterns defines not only the test length and test memory requirements but also the energy consumption. In general, as 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 an 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. We propose two heuristics that try to minimise 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 minimised [21].

2 Hybrid BIST and energy reduction

A classical BIST architecture consists of a test pattern generator (TPG), a test response analyser and a BIST control unit (BCU), all implemented on the chip. Different implementations of such a BIST architecture have been available, and some of them have wide acceptance. One of the major problems of the classical BIST implementations is related to the TPG design. Typically, such a TPG is implemented by linear feedback shift registers (LFSR) [22, 23]. As the test patterns generated by an LFSR are pseudorandom by nature and have linear dependencies [24], the LFSR-based approach often does not guarantee a sufficiently high fault coverage (especially in the case of large and complex designs), and demands very long test application times in addition to high area overheads. Therefore different hybrid approaches have been proposed, where pseudorandom test patterns, generated by LFSRs, are complemented with a set of deterministic test patterns. These approaches are generally referred to as hybrid BIST [25–32]. 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

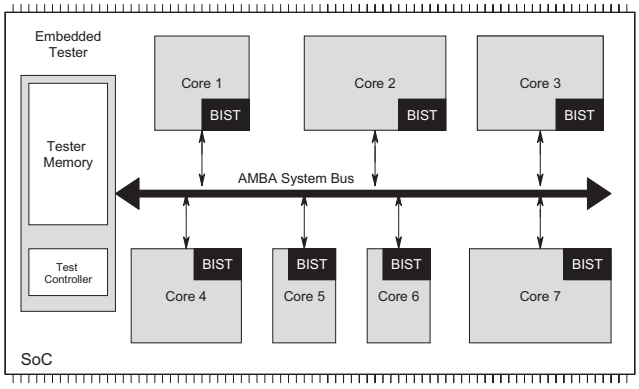


Fig. 1 *AMBA bus-based hybrid BIST architecture*

pseudorandom test patterns, that is, the pseudorandom tests for all cores can be carried out concurrently. The deterministic tests, in contrast, are applied from an on-chip memory, one core at a time. And we have also assumed for test data transportation an AMBA (advanced microcontroller bus architecture)-like test bus [33]. AMBA integrates an on-chip test access technique that re-uses the basic bus infrastructure [34]. An example of a multi-core system with such test architecture is given in Fig. 1.

For portable systems with such a test architecture, one of the most important test constraints is the total amount of on-chip test memory. Methods for test time minimisation under given test memory constraint for test-per-clock and test-per-scan schemes are proposed [27, 35]. If the objective is only test time minimisation and power/energy is not taken into account, then the shortest test schedule for such a test architecture (Fig. 1) is the one where all cores are tested concurrently and have the same tests lengths, as depicted in Fig. 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, as deterministic test vectors are more effective in covering faults than the pseudorandom ones. A longer pseudorandom test, in contrast, will lead to longer test application time with reduced memory requirements [26]. From an energy perspective, different cores have different energy dissipation while applying the same amount of test patterns. Furthermore, the pseudorandom and deterministic test

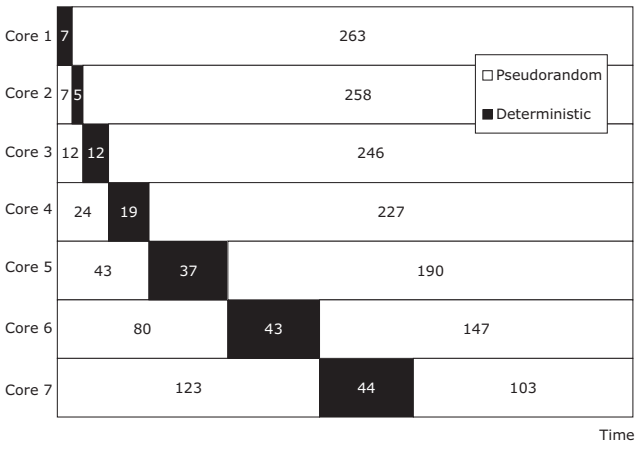


Fig. 2 *Time minimised test schedule*

sequences for the same core have different power characteristics. Therefore for total energy minimisation, 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.

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_k^F and a complete sequence of pseudorandom test patterns TP_k^F will be generated. It is assumed that both test sets can obtain by itself the 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 $TP_k \subseteq TP_k^F$, and a subset of the deterministic test sequence $TD_k \subseteq TD_k^F$. The test sequences TP_k and TD_k complement each other to achieve the maximum achievable fault coverage $F_{\max}$, and define the hybrid test set TH_k.

By knowing the length $|TP_k|$ and the fault coverage value of the pseudorandom test sequence TP_k, we can find the amount of additional deterministic patterns TD_k [26]. Therefore we can say that the pseudorandom test sequence TP_k uniquely defines the structure of the entire hybrid test set.

Definition 2: When assuming the test architecture described above, a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$ for a system $S = \{C_1, C_2, \ldots, C_n\}$ consists of hybrid tests TH_k for each individual core C_k, where the pseudorandom components of the TH can be scheduled in parallel, whereas the deterministic components of TH must be scheduled in sequence due to the shared test resources.

Definition 3: $\boldsymbol{J} = (\boldsymbol{j}_1, \boldsymbol{j}_2, \ldots, \boldsymbol{j}_n)$, where $\boldsymbol{j}_k = |TP_k|$ $(0 \leq \boldsymbol{j}_k \leq |TP_k^F|)$, is called the characteristic vector of a hybrid test set $TH = \{TH_1, TH_2, \ldots, TH_n\}$. $\boldsymbol{j}_k$ denotes the length of the pseudorandom test sequence $TP_k \subseteq TH_k$.

According to these definitions, for a given core C_k, different values of $\boldsymbol{j}_k$ correspond to pseudorandom subsequences of different lengths. Given a value of $\boldsymbol{j}_k$, we have a pseudorandom subsequence of length $\boldsymbol{j}_k$ for the core C_k. In order to form a hybrid test sequence TH_k, this subsequence should be complemented with a deterministic test sequence, which is generated such that the hybrid sequence TH_k reaches to the maximal achievable fault coverage. On this basis, we can conclude that the characteristic vector $\boldsymbol{J}$ determines entirely the structure of the hybrid test set TH_k for all cores $C_k \in S$.

Definition 4: Let us denote with $M_k(\boldsymbol{j}_k)$ and $E_k(\boldsymbol{j}_k)$, 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 the length $\boldsymbol{j}_k$ of its pseudorandom test sequence.

Note that it is very time-consuming to calculate the exact values of $M_k(\boldsymbol{j}_k)$ and $E_k(\boldsymbol{j}_k)$ for any arbitrary hybrid BIST set TH_k, as it requires exact calculation of the corresponding hybrid test set which is an expensive procedure [27]. To overcome the problem, we propose to use an estimation method for memory and energy calculation that is based only on a few critical point calculations.

Definition 5: Let us denote with $M(\boldsymbol{J})$ and $E(\boldsymbol{J})$, respectively, the total memory cost and energy cost of the corresponding hybrid BIST set TH with characteristic vector $\boldsymbol{J}$ for a given system with n cores. These costs can be calculated using the following formulas

$$M(\boldsymbol{J}) = \sum_{k=1}^{n} M_k(\boldsymbol{j}_k) \quad E(\boldsymbol{J}) = \sum_{k=1}^{n} E_k(\boldsymbol{j}_k) \tag{2}$$

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 set TH_k for each individual core C_k $(k = 1, 2, \ldots, n)$. In our approach, the pseudorandom components of the TH are going to be scheduled in parallel, whereas the deterministic components of the TH, based on the given test architecture (Fig. 1), have to be scheduled in sequence.

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

Next, we are going to describe the estimation method that is going to be used for fast calculations of the hybrid BIST structure, that is, the amount of pseudorandom and deterministic test patterns in the hybrid test set.

4 Hybrid BIST structure estimation

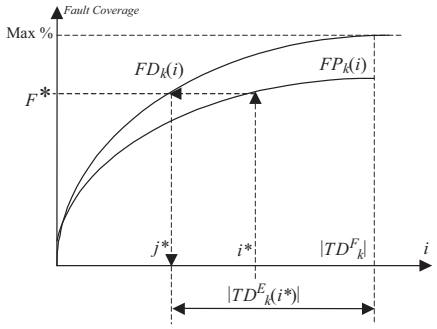
For hybrid BIST energy minimisation at a given memory constraint, we should calculate for every core $C_k \in S$ for any possible pseudorandom test sequence length $\boldsymbol{j}_k$, the size of the memory $M_k(\boldsymbol{j}_k)$ and the amount of dissipated energy $E_k(\boldsymbol{j}_k)$. This would give us a possibility to compare memory and energy values of different alternatives and to find the optimal solution. However, the procedure to calculate the cost functions $M(\boldsymbol{J})$ and $E(\boldsymbol{J})$ exactly is very time-consuming, as it assumes that the deterministic test sets TD_k for all possible values of the characteristic vector $\boldsymbol{J}$ are available. This means that for each possible pseudorandom test TP_k, a set of not-yet-detected faults $F_{\text{NOT}}(TP_k)$ should be calculated, and the needed deterministic test set TD_k has to be found. This can be done either by repetitive use of an automatic TPG or by systematically analysing and compressing the fault tables for each TP_k [36]. These two procedures are both time-consuming and therefore not feasible for larger designs.

To overcome the complexity explosion problem, we propose an iterative algorithm, where the memory and energy costs for the deterministic test sets TD_k are calculated based on estimates in a similar way described by Jervan *et al.* [37]. 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 $F_{\text{NOT}}(TP_k)$.

The estimation method is described in Algorithm 1 given below. We analyse first the fault detection capabilities of both sequences, pseudorandom and deterministic, in isolation and use this information for estimating the composition of the combined hybrid BIST test set. In the following, we will use $FD_k(\boldsymbol{i})$ and $FP_k(\boldsymbol{j})$ to denote the fault coverage figures of the test sequences TD_k with length $\boldsymbol{i}$ and TP_k with length $\boldsymbol{j}$, respectively.

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

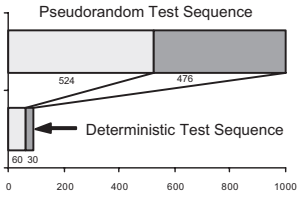
1. Calculate, by fault simulation, the fault coverage functions $FD_k(\boldsymbol{i})$, $\boldsymbol{i} = 1, 2, \ldots, |TD_k^F|$, and $FP_k(\boldsymbol{j})$, $\boldsymbol{j} = 1, 2, \ldots, |TP_k^F|$. The patterns in TD_k^F are ordered in such a



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	98.9

b



c

Fig. 3 *Estimation of the length of the deterministic test sequence*

way that each pattern put into the sequence contribute with maximum increase in fault coverage.

2. For each $\boldsymbol{i}^* \leq |TP_k^F|$, find the fault coverage value F^* that can be reached by the pseudorandom test sequence with length $\boldsymbol{i}^*$ (see Fig. 3*a*).

3. By solving the equation $FD_k(\boldsymbol{j}) = F^*$, find the maximum integer value $\boldsymbol{j}^*$ that satisfies the condition $FD_k(\boldsymbol{j}^*) \leq F^*$. The value of $\boldsymbol{j}^*$ is the length of the deterministic sequence TD_k that can achieve the same fault coverage F^*.

4. Calculate the value of $|TD_k^E(\boldsymbol{i}^*)| = |TD_k^F| - \boldsymbol{j}^*$, which is the number of test patterns needed in addition to the pseudorandom patterns to reach to the maximum achievable fault coverage.

The value $|TD_k^E(\boldsymbol{i}^*)| = |TD_k^F| - \boldsymbol{j}^*$ 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 $\boldsymbol{i}^*$ test patterns in TP_k, ($|TP_k| = \boldsymbol{i}^*$).

The algorithm is illustrated with the example given in Figs 3*b* and *c*. In the given example, if $\boldsymbol{i}^* = 524$ then the fault coverage is $F^* = 97.5\%$. From the fault simulation table (Fig. 3*b*) we can find that similar fault coverage can be achieved by 60 deterministic test patterns ($\boldsymbol{j}^* = 60$). On the basis of 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. Obviously, the algorithm produces only an estimate and therefore it is important to consider this fact while generating the final result.

On the basis of the created relationships between the pseudorandom test sequence length $|TP_k|$ and estimated deterministic test length $|TD_k^E|$, we can easily solve the equation $|TD_k^E| = f(|TP_k|)$ also in the opposite direction when TD_k^E is given and $|TP_k|$ has to be found. These calculations will be used in the next section, where the energy minimisation heuristics will be described.

5 Heuristic algorithms for hybrid BIST energy minimisation

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

To solve this complex combinatorial task, we propose two fast heuristic algorithms: local gain algorithm and average gain algorithm [21]. These algorithms are alternatives to each other and are based on the estimation methodology described in Algorithm 1. The general concept of our proposed technique is depicted in Fig. 4. The main idea of both algorithms is to start with the pure deterministic test set $TH_k = \{TP_k = \varnothing, TD_k^F\}$ for each core $C_k \in S$. This is a solution where the maximum achievable fault coverage is guaranteed but the memory constraint is usually not satisfied. Next, the deterministic test patterns are gradually substituted by corresponding sequences of pseudorandom patterns $PR_i \subseteq TP_k^F$ 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. The idea is that with every iteration, we try to reduce the memory requirements such that the energy increase is the smallest.

5.1 Local gain algorithm

In this algorithm, $\Delta M_{k,i}$ corresponds to the estimated memory gain when deterministic test pattern $DP_i \in TD_k^F$ is removed from the memory, and $\Delta E_{k,i}$ corresponds to the estimated increase in energy dissipation by the sequence of pseudorandom patterns $PR_i \subseteq TP_k^F$ that are substituting the deterministic test pattern $DP_i \in TD_k^F$. 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 estimated so that the fault coverage of the core is not reduced and thus the maximum achievable fault coverage is always guaranteed.

1. Calculate the complete deterministic and pseudorandom test sets for every core $C_k \in S$
2. Choose a core, based on selected strategy

   //Either Local Gain or Average Gain strategy
3. Substitute a certain number of deterministic test patterns from the selected core's test set with pseudorandom test patterns, such that the fault coverage is not compromised (based on estimates)
4. IF Memory constraint is not satisfied THEN

   GOTO 2

   ELSE

   Use the estimated solution for calculating the final solution. If memory constraint is not satisfied or fault coverage is below the highest possible, improve the result.

Fig. 4 *Energy minimisation technique*

Let us introduce the following additional notations: M, the current memory cost; L, the current pseudorandom test length. The algorithm starts with initial values: $L = 0$ and $M = M(TD_1^F) + M(TD_2^F) + \cdots + M(TD_n^F)$ where $M(TD_k^F)$ is the memory cost of the complete deterministic test set of core $C_k \in S$. Initially: $TH_k = \{TP_k = \varnothing,\ TD_k^F\}$, as mentioned before, and the local index in each core, i, is set to be 1.

Algorithm 2: Local gain algorithm:

1. Select core $C_k \in S$ where $\Delta M_{k,i}/\Delta E_{k,i}$ has the highest value.
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. Update the local index of the selected core $i = i + 1$.
5. If $M > M_{\text{LIMIT}}$, then go to Step 1.
6. END.

The algorithm is illustrated with the example given in Fig. 5. At every improvement iteration, we calculate the memory–energy ratio for all cores assuming that one deterministic test pattern for each core (denoted as white boxes in Fig. 5*a*) would be replaced with pseudorandom patterns. The core with the 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 this example, Core 3 was selected and the deterministic pattern denoted as a white box, belonging to Core 3, has been replaced by several pseudorandom patterns, as illustrated in Fig. 5*b*. At the end of every iteration, we calculate the new memory (M) and energy (E) values for the entire system. This procedure is then repeated until $M \leq M_{\text{LIMIT}}$.

5.2 Average gain algorithm

The alternative heuristic is called the 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.

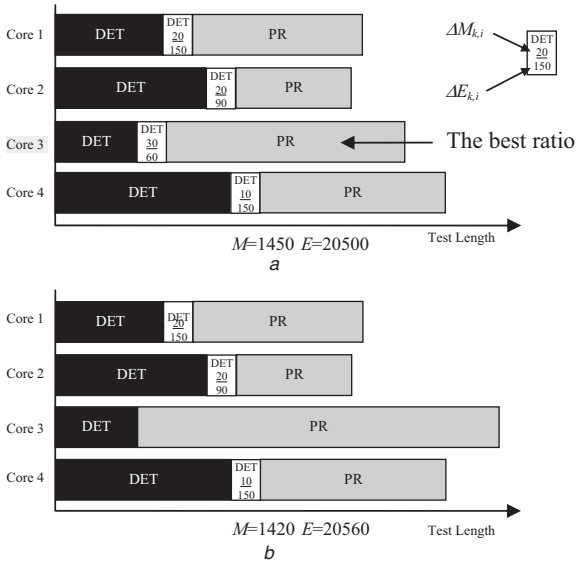


Fig. 5 *Illustration of the local gain algorithm*

The algorithm starts also with initial values: $L = 0$ and $M = M(TD_1^F) + M(TD_2^F) + \cdots + M(TD_n^F)$ where $M(TD_k^F)$ is the memory cost of the complete deterministic test set of core $C_k \in S$. Initially: $TH_k = \{TP_k = \varnothing,\ TD_k^F\}$, as in the local gain algorithm, and for all cores $\Delta M_k = \Delta M_{k,1}$, $\Delta E_k = \Delta E_{k,1}$, and the local index $i = 1$.

Algorithm 3: Average gain algorithm:

1. Select core $C_k \in S$ where $\Delta M_k/\Delta E_i$ has the highest value.
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 $M > M_{\text{LIMIT}}$, then go to Step 1.
8. END.

The main difference between the above two heuristics is that Algorithm 2 takes into account only the immediate effect of the test pattern substitution. Algorithm 3, in contrast, takes into account the entire history of pattern substitutions. In Section 6, the results of both algorithms will be presented, and compared to each other.

5.3 Generation of the final solution

Both Algorithms 2 and 3 create an energy-optimised hybrid BIST solution $TH_k = \{TP_k,\ TD_k\}$ where energy consumption is minimised with respect to the given memory constraint. However, the algorithms are based on estimated memory and energy values for TD_k, and therefore the final results usually do not correspond to the optimal solutions. 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 generated. The algorithm for this improvement is similar to the one proposed by Jervan *et al.* [27].

Both Algorithms 2 and 3 substitute deterministic test patterns with pseudorandom ones. As discussed earlier, many faults cannot be detected by pseudorandom patterns and thus also the final hybrid test set might not cover all faults if this aspect is not considered explicitly. Therefore the final solution, calculated based on the estimated pseudorandom test set, will be improved with deterministic test patterns until the fault coverage is the maximum achievable one. The algorithm uses as a starting point the result (the estimated amount of pseudorandom test vectors) obtained with Algorithm 2 or 3.

Algorithm 4: Calculation of final test sets:

1. Let $\boldsymbol{J}^* = (\boldsymbol{j}_1^*, \boldsymbol{j}_2^*, \ldots, \boldsymbol{j}_n^*)$ denote the characteristic vector of the hybrid test solution generated by either Algorithm 2 or Algorithm 3, M_k^* denote the estimated memory cost of each core $C_k \in S$, and $M^* = \sum_{k=1}^{n} M_k^*$.
2. To calculate the real cost M_{REAL} for the candidate solution $\boldsymbol{J}^*$, find the set $F_{\text{NOT},k}(\boldsymbol{j}_k^*)$ of faults not-yet-detected by the pseudorandom test for each core $C_k \in S$ and generate the corresponding deterministic test set TD_k^* by using an automatic test pattern generation (ATPG) algorithm.
3. If $M_{\text{REAL}} = M_{\text{LIMIT}}$, go to Step 6.

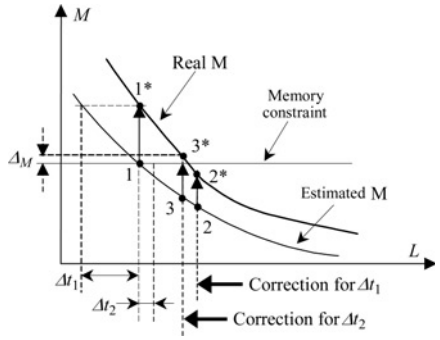


Fig. 6 *Calculation of the final result*

4. Let $\Delta t = M_{\mathrm{LIMIT}} - M_{\mathrm{REAL}}$; if $|\Delta t|$ is bigger than that in the previous iteration, let $\Delta t = \Delta t/2$.
5. Calculate a new estimated memory cost $M^* = M^* + \Delta t$ and find a new characteristic vector $\boldsymbol{J}^*$, go to Step 2.
6. Return $\boldsymbol{J}^* = (\boldsymbol{j}_1^*, \boldsymbol{j}_2^*, \ldots, \boldsymbol{j}_n^*)$ as the final solution.
7. END.

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 Fig. 6. 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

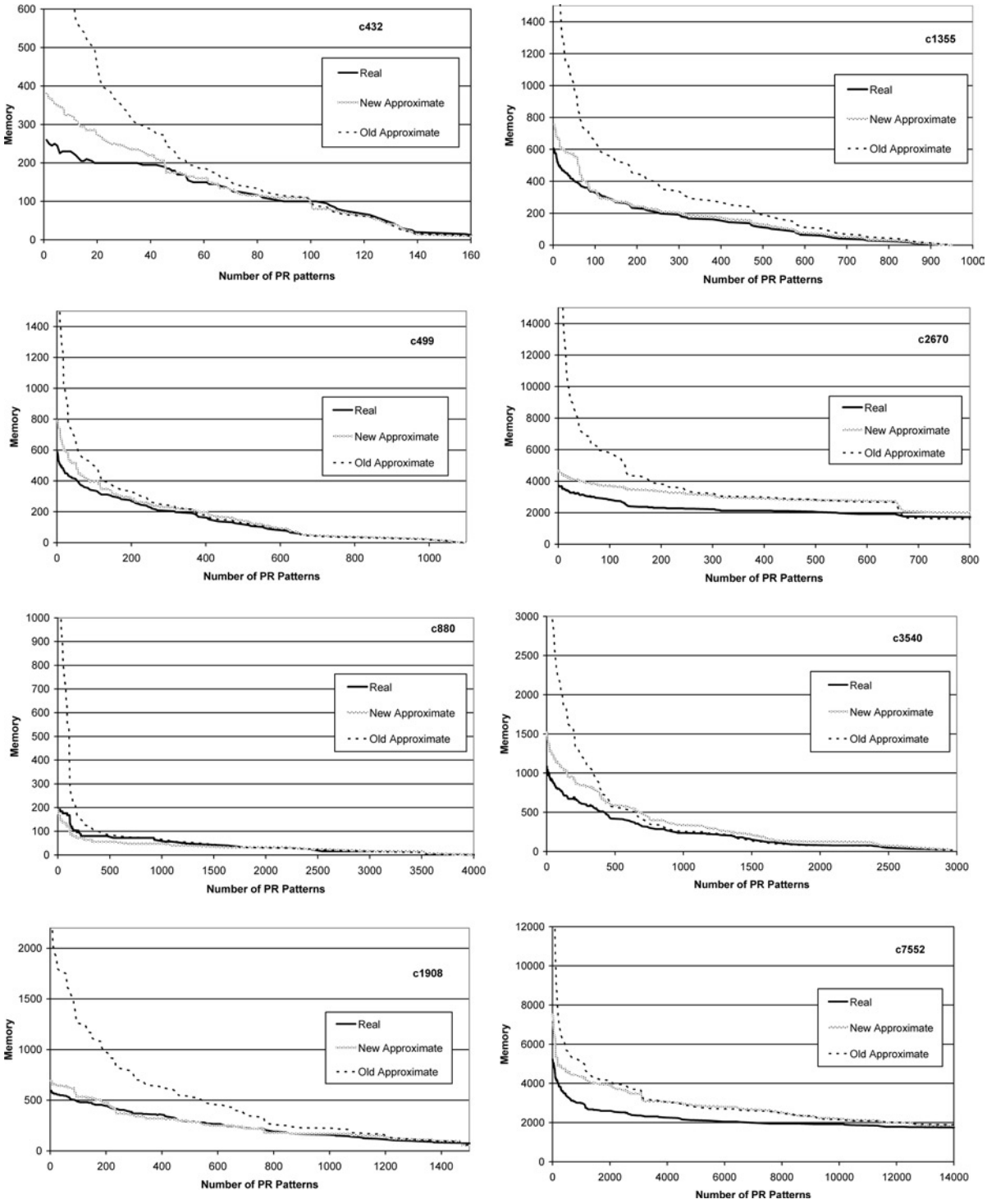


Fig. 7 *Experimental results with ISCAS 85 benchmark designs*

given constraint. The outcome of this algorithm is the final solution.

6 Experimental results

At first, we analysed the quality of the proposed estimation algorithm and performed therefore experiments with all designs from the ISCAS85 benchmark family. Some of the results are illustrated in Fig. 7. In those charts, we have depicted the memory requirement (the size of the deterministic test set) for every pseudorandom test length. Obviously, the longer the pseudorandom test sequence, the smaller the memory requirement. We have compared our earlier estimation methodology [26], labelled as 'old approximate,' against the estimation methodology proposed in this paper, labelled as 'new approximate.' In addition, we have also depicted the real memory cost in Fig. 7. This has been obtained by the repetitive use of an ATPG algorithm [26]. As it can be seen from the results, the improved estimation methodology gives better estimates than the previous one (the curve 'new approximate' is much closer to the 'real' than the 'old approximate'), mainly in the situations when the hybrid test set contains smaller amount of pseudorandom test patterns.

Table 1: Experimental results

Algorithm	M_{LIMIT}	Energy (switches)	Compared to Jervan *et al.* [27] (%)	Test length (clocks)	Compared to Jervan *et al.* [27] (%)	CPU time (s)
System 1–6 cores						
[23]	1500	2 588 822	100.00	24 689	100.00	8.41
A2		1 281 690	49.51	31 619	128.07	11.09
A3		1 281 690	49.51	31 619	128.07	6.64
SA		1 240 123	47.90	31 619	128.07	5326.24
[23]	2500	635 682	100.00	6726	100.00	24.61
A2		426 617	67.11	10 559	156.99	14.23
A3		446 944	70.31	10 679	158.77	4.84
SA		409 576	64.43	10 529	156.54	2944.26
[23]	3000	717 026	100.00	7522	100.00	26.51
A2		265 282	37.00	8126	108.03	36.31
A3		286 883	40.01	8129	108.07	26.96
SA		241 123	33.63	8153	108.39	1095.21
System 2–6 cores						
[23]	1700	6 548 659	100.00	52 145	100.00	12.05
A2		5 502 763	84.03	70 331	134.88	12.49
A3		5 318 781	81.22	70 331	134.88	4.28
SA		4 747 498	72.50	83 865	160.83	3805.23
[23]	3000	2 315 958	100.00	19 208	100.00	20.21
A2		1 998 390	86.29	23 774	123.77	7.66
A3		1 861 844	80.39	24 317	126.60	18.79
SA		1 845 022	79.67	28 134	146.47	5032.05
[23]	4700	893 184	100.00	8815	100.00	21.47
A2		742 462	83.13	9537	108.19	26.45
A3		746 479	83.58	9537	108.19	55.09
SA		723 817	81.04	12 596	142.89	3654.02
System 3–20 cores						
[23]	5000	12 830 419	100.00	40 941	100.00	47.49
A2		9 242 890	72.04	70 331	171.79	51.43
A3		9 839 005	76.68	70 331	171.79	40.49
SA		7 367 201	57.42	60 495	147.76	29 201.96
[23]	7000	6 237 211	100.00	20 253	100.00	53.39
A2		4 039 622	64.77	31 619	156.12	73.58
A3		4 223 263	67.71	32 145	158.71	14.36
SA		3 500 894	56.13	31 919	157.60	20 750.03
[23]	10 000	4 686 729	100.00	15 483	100.00	45.37
A2		1 719 726	36.69	17 499	113.02	115.53
A3		1 815 129	38.73	17 554	113.38	90.52
SA		1 606 499	34.28	17 992	116.20	14 572.33

Thereafter, we have used the improved estimation method in the proposed algorithms for hybrid BIST energy minimisation. We have performed experiments with different designs containing the ISCAS'89 benchmarks as cores. It was impossible to use ITC'02 SoC test benchmarks as the proposed approach relies on structural information about each core for test generation and fault simulation. Such detailed information about ITC'02 benchmarks is unavailable and therefore the ISCAS benchmarks were chosen. The complexity of these designs ranges from system with six cores to 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. This assumption simplifies only the test controlling process and does not have any impact to the proposed algorithms. For the BIST part, STUMPS architecture was used.

In Table 1, we have listed the results for every system with three different memory constraints. We have listed results generated by the approach discussed by Jervan *et al.* [27], which provide the shortest possible test length without considering energy consumption, by our two algorithms (the local gain algorithm is denoted by A2, and the average gain algorithm by A3, both improved with Algorithm 4), and by simulated annealing (SA). In every experiment, the minimised test time produced by Jervan *et al.* [27] is taken as a baseline (100%) and all other solutions are compared against this result.

As shown in Table 1 (columns 3 and 4) both proposed algorithms lead to reduced energy solutions (in some cases up to 63% reduction of the total switching activity). When compared to the SA 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 data and reported them in Table 1 (columns 5 and 6). As can be expected in all these solutions generated by our techniques, the test time has increased compared to the technique that targets towards test length minimisation [27]. 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

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 to guide the modification of these ratios together with an iterative procedure for transforming the estimated results to the final results. Experimental results have shown the efficiency of these heuristics for energy reduction under test memory constraints.

8 Acknowledgments

This work has been supported by the Estonian Science Foundation grants G6829 and G5910, Enterprise Estonia Project Technology Development Centre ELIKO, the Swedish Foundation for Strategic Research (SSF) under the Strategic Integrated Electronic Systems Research (STRINGENT) program, and the Swedish National Program on Socware (System-on-Chip).

9 References

1 Zorian, Y.: 'A distributed BIST control scheme for complex VLSI devices'. Proc. IEEE VLSI Test Symp., Atlantic City, NJ, USA, 6–8 April 1993, pp. 4–9
2 Devadas, S., and Malik, M.: 'A survey of optimization techniques targeting low power VLSI circuits'. Proc. 32nd ACM/IEEE Design Automation Conf., San Francisco, CA, USA, 12–16 June 1995, pp. 242–247
3 Girard, P., Guiller, L., Landrault, C., and Pravossoudovitch, S.: 'A test vector inhibiting technique for low energy BIST design'. Proc. VLSI Test Symp., Dana Point, CA, USA, 25–29 April 1999, pp. 407–412
4 Girard, P.: 'Low power testing of VLSI circuits: problems and solutions'. Proc. IEEE Int. Symp. on Quality Electronic Design, San Jose, CA, USA, 20–22 March 2000, pp. 173–179
5 Li, P.C., and Young, T.K.: 'Electromigrations: the time bomb in deep-submicron ICs', *IEEE Spectrum*, 1996, **33**, (9), pp. 75–78
6 Chandrakasan, A., Sheng, T., and Brodersen, R.W.: 'Low power CMOS digital design', *J. Solid State Circuits*, 1992, **27**, (4), pp. 473–484
7 Girard, P., Guiller, L., Landrault, C., Pravossoudovitch, S., and Wunderlich, H.J.: 'A modified clock scheme for a low power BIST test pattern generator'. Proc. VLSI Test Symp., Marina Del Rey, CA, USA, 29 April–3 May 2001, pp. 306–311
8 Girard, P., Guiller, L., Landrault, C., Pravossoudovitch, S., Figueras, J., Manich, S., Teixeira, P., and Santos, M.: 'Low-energy BIST design: impact of the LFSR TPG parameters on the weighted switching activity'. Proc. Int. Symp. on Circuits and Systems, Orlando, FL, USA, 30 May–2 June 1999, pp. 110–113
9 Gizopoulos, D., Kranitis, N., Paschalis, A., Psarakis, M., and Zorian, Y.: 'Low power/energy BIST scheme for datapaths'. Proc. VLSI Test Symp., Montreal, Quebec, Canada, 30 April–4 May 2000, pp. 23–28
10 Wang, S., and Gupta, S.: 'ATPG for heat dissipation minimization during test application', *IEEE Trans. Comput.*, 1998, **46**, (2), pp. 256–262
11 Chakravarty, S., and Dabholkar, V.: 'Minimizing power dissipation in scan circuits during test application'. Proc. IEEE Int. Workshop on Low Power Design, Napa Valley, CA, USA, 24–27 April 1994, pp. 51–56
12 Gerstendorfer, S., and Wunderlich, H.J.: 'Minimized power consumption for scan-based BIST'. Proc. IEEE Int. Test Conf., Atlantic City, NJ, USA, 28–30 September 1999, pp. 77–84
13 Whetsel, L.: 'Adapting scan architectures for low power operation'. Proc. IEEE Int. Test Conf., Atlantic City, NJ, USA, 3–5 October 2000, pp. 863–872
14 Sankaralingam, R., Pouya, B., and Touba, N.A.: 'Reducing power dissipation during test using scan chain disable'. Proc. VLSI Test Symp., Marina Del Rey, CA, USA, 29 April–3 May 2001, pp. 319–324
15 Rosinger, P.M., Al-Hashimi, B.M., and Nicolici, N.: 'Scan architecture for shift and capture cycle power reductions'. Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems, Vancouver, BC, Canada, 6–8 November 2002, pp. 129–137
16 Basturkmen, N.Z., Reddy, S.M., and Pomeranz, I.: 'A low power pseudo-random BIST technique'. Proc. IEEE Int. On-Line Testing Workshop, Isle of Bendor, France, 8–10 July 2002, pp. 140–144
17 Cota, E., Carro, L., Lubaszewski, M., and Orailoglu, A.: 'Test planning and design space exploration in a core-based environment'. Proc. Design, Automation and Test in Europe Conf., Paris, France, 4–8 March 2002, pp. 478–485
18 Huang, Y., Cheng, W.-T., Tsai, C.-C., Mukherjee, N., Samman, O., Zaidan, Y., and Reddy, S.M.: 'Resource allocation and test scheduling for concurrent test of core-based SOC design'. Proc. IEEE Asian Test Symp., Kyoto, Japan, 19–21 November 2001, pp. 265–270
19 Iyengar, V., Chakrabarty, K., and Marinissen, E.J.: 'Test wrapper and test access mechanism co-optimization for system-on-chip'. Proc. IEEE Int. Test Conf., Baltimore, MD, USA, 30 October–1 November 2001, pp. 1023–1032
20 Larsson, E., and Peng, Z.: 'An integrated framework for the design and optimization of SOC test solutions', *J. Electron. Test. Theory Appl.*, 2002, **18**, (4/5), pp. 385–400
21 Ubar, R., Shchenova, T., Jervan, G., and Peng, Z.: 'Energy minimization for hybrid BIST in a system-on-chip test environment'. Proc. IEEE European Test Symp., Tallinn, Estonia, 22–25 May 2005, pp. 2–7
22 Agrawal, V.D., Kime, C.R., and Saluja, K.K.: 'A tutorial on built-in self-test.', *IEEE Des. Test Comput.*, 1993, **10**, (1), pp. 73–82; 1993, **10**, (2), pp. 69–97

23 Bardell, P.H., and McAnney, W.H.: 'Self-testing of multichip logic modules'. Proc. IEEE Int. Test Conf., Philadelphia, PA, USA, November 1982, pp. 200–204
24 Golomb, S.W.: 'Shift register sequences' (Aegan Park Press, 1982)
25 Chatterjee, M., and Pradhan, D.K.: 'A novel pattern generator for near-perfect fault-coverage'. Proc. IEEE VLSI Test Symp., Princeton, NJ, USA, 30 April–3 May 1995, pp. 417–425
26 Jervan, G., Peng, Z., and Ubar, R.: 'Test cost minimization for hybrid BIST'. Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems, Yamanashi, Japan, 25–27 October 2000, pp. 283–291
27 Jervan, G., Eles, P., Peng, Z., Ubar, R., and Jenihhin, M.: 'Test time minimization for hybrid BIST of core-based systems'. Proc. IEEE Asian Test Symp., Xian, China, 17–19 November 2003, pp. 318–323
28 Sugihara, M., Date, H., and Yasuura, H.: 'Analysis and minimization of test time in a combined BIST and external test approach'. Proc. Design, Automation and Test in Europe Conf., Paris, France, 27–30 March 2000, pp. 134–140
29 Hellebrand, S., Tarnick, S., Rajski, J., and Courtois, B.: 'Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers'. Proc. IEEE Int. Test Conf., Baltimore, MD, USA, 20–24 September 1992, pp. 120–129
30 Hellebrand, S., Wunderlich, H.-J., and Hertwig, A.: 'Mixed-mode BIST using embedded processors', *J. Electron. Test. Theory Appl.*, 1998, **12**, pp. 127–138
31 Touba, N.A., and McCluskey, E.J.: 'Synthesis of mapping logic for generating transformed pseudo-random patterns for BIST'. Proc. IEEE Int. Test Conf., Washington, DC, USA, 21–25 October 1995, pp. 674–682
32 Zacharia, N., Rajski, J., and Tyzer, J.: 'Decompression of test data using variable-length seed LFSRs'. Proc. IEEE VLSI Test Symp., Princeton, NJ, USA, 28 April–1 May 1996, pp. 426–433
33 Flynn, D.: 'AMBA: enabling reusable on-chip designs.', *IEEE Micro*, 1997, **17**, (4), pp. 20–27
34 Harrod, P.: 'Testing reusable IP—a case study'. Proc. IEEE Int. Test Conf., Atlantic City, NJ, USA, 27–30 September 1999, pp. 493–498
35 Jervan, G., Eles, P., Peng, Z., Ubar, R., and Jenihhin, M.: 'Hybrid BIST time minimization for core-based systems with STUMPS architecture'. Proc. IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems, Cambridge, MA, USA, 3–5 November 2003, pp. 225–232
36 Jervan, G., Peng, Z., Ubar, R., and Kruus, H.: 'A hybrid BIST architecture and its optimization for SoC testing'. Proc. IEEE Int. Symp. on Quality Electronic Design, San Jose, California, USA, 18–20 March 2002, pp. 273–279
37 Jervan, G., Peng, Z., Ubar, R., and Korelina, O.: 'An improved estimation methodology for hybrid BIST cost calculation'. Proc. IEEE Norchip Conf., Oslo, Norway, 8–9 November 2004, pp. 297–300