Power Profile Manipulation: A New Approach for Reducing Test Application Time Under Power Constraints

Paul M. Rosinger,* Bashir M. Al-Hashimi and Nicola Nicolici

Paul M. Rosinger and Bashir M. Al-Hashimi

Electronic Systems Design Group

Department of Electronics and Computer Science

University of Southampton

Southampton SO17 1BJ, U.K.

Tel: +44-23-8059-6665 / +44-23-8059-3249   Fax: +44-23-8059-2901

Email: {p.rosinger,bmah}@ecs.soton.ac.uk

Nicola Nicolici

Computer-Aided Design and Test Research Group

Department of Electrical and Computer Engineering

McMaster University

1280 Main St. W., Hamilton, ON L8S 4K1, Canada

Tel: +1-905-525-9140 ext. 27598   Fax: +1-905-521-2922

Email: nicola@ece.mcmaster.ca

---

*Corresponding author

Power Profile Manipulation: A New Approach for Reducing Test Application Time Under Power Constraints

Abstract

This paper proposes a power profile manipulation approach which merges two distinct research directions in low power testing: minimization of test power dissipation and test application time reduction under power constraints. It is shown how *complementary techniques* can be easily combined through this approach to significantly increase test concurrency under power constraints. This is achieved in two steps: in the first step power dissipation is considered a design objective and consequently it is minimized, result further exploited in the second step, when power becomes a design constraint under which the test application time is reduced. A distinctive feature of the proposed power profile manipulation approach is that it can be included in, and consequently improve, *any* existing power constrained test scheduling algorithm. Extensive experimental results using benchmark circuits, considering test-per-clock as well as test-per-scan schemes, show that *by integrating* the proposed power profile manipulation approach *into any existing power constrained test scheduling algorithm*, savings up to 41% in test application time are achieved.

1 Introduction

Low power design of complementary metal-oxide semiconductor (CMOS) integrated circuits (ICs) and systems is a well researched area [16, 18]. Testing low power ICs and systems is a relatively new research topic, which is receiving increasing attention lately [2, 3, 6–15, 17, 22, 23]. It was reported in [23] that, due to an increased activity on chip, the power dissipation during test is significantly higher than during the functional operation which constitutes an emerging yield and reliability problem [22, 23].

Another important issue in testing modern systems is the test time. With the rapidly increasing complexity of modern systems, testing becomes more time consuming which has a serious impact on the final cost of the design. Test scheduling is a commonly adopted strategy for reducing test application time by augmenting the concurrency of test activities in the system. However, increased test concurrency leads to higher power dissipation which is not tolerated by low power systems. Consequently, several solutions to the problem of power dissipation during test have been recently proposed [2, 3, 6–15, 17, 22, 23]. Within these recent solutions, two main research directions can be identified: one considering power dissipation during test an optimization objective, and the other which considers power as a design constraint under which other test parameters, such as test time, are reduced. The first category includes, although it is not limited to, test vector reordering methods [4, 6, 7, 9] which aim to increase the correlation between successive test patterns and thus reduce the power dissipation during test application. The test vector ordering is done in a post-ATPG phase, thus it does not increase the test application time. Since both power dissipation and test application time represent major issues of test, several power constrained test scheduling (PCTS) algorithms, belonging to the second research direction, were reported recently. PCTS algorithms [2, 3, 10–12, 14, 17] aim to minimize test application time under a given power constraint imposed by the package type and energy limitations. The basic idea of PCTS algorithms is to maximize the test concurrency, without exceeding the power constraint. Test scheduling is performed during system integration. Usually the embedded cores are delivered as IP blocks accompanied with test data, thus the system integrator might not have access to their internal structure. Therefore, unless the cores are pre-designed with special scan architectures, the system integrator can control the power dissipation during test *only* by means of test data transformations, such as test vector reordering and/or test sequence expansion.

This paper introduces a new approach for reducing test application time by manipulating (changing) power profiles of test sets that are applied to every block in the system. This power profile manipulation consists of lowering, reshaping and then rotating the power profiles such that maximum power dissipation of every test is minimized at the block level and test concurrency is maximized at

the system level. The proposed methodology represents a general solution for the system integrator without putting any constraints on the scan architectures of the cores. While several power constrained test scheduling methods have been reported [2, 3, 10–12, 14, 17] to the best of our knowledge this is the first solution where: a) not only the average and/or peak values of power dissipation are considered, *but also the shape* of the power profile, and b) the *test sequences' slack time* is exploited via test sequence expansion for further lowering of the power profiles. The possibility of controlling the position and size of higher and lower power parts in the power profiles of system's block would allow *any PCTS algorithm* [2, 3, 10–12, 14, 17], to easily increase test concurrency. Moreover, a test session from a test schedule usually consists in a number of unequal length tests. The test length differences can be used to extend the shorter test sets in the test session with additional vectors. Careful selection of the additional test vectors can be used to reduce the peak power thus making room under the same power constraint for more tests. By manipulating the power profile during test scheduling, the proposed solution is a fusion between the two existing research directions in low power testing: minimizing test power dissipation [4, 6, 7, 9, 22] and minimizing test application time under power constraints [2, 3, 10–12, 14, 17]. Hence, this paper shows how complementary techniques can be easily combined to significantly increase test concurrency under given power constraints. *The proposed power profile manipulation approach is not a test scheduling algorithm, rather it represents a complementary technique meant to enhance the performance of existing power constrained test scheduling algorithms.* The distinctive benefit of the power profile manipulation approach is that it does not depend on the initial test sets, as well as it is independent on the test scheduling policy. Consequently, it can be *equally* embedded into *any* existing PCTS algorithm to leverage its performance. However, to be noted that this methodology is addressed only to the cases when testing is performed exclusively using ATPG-generated test vectors and where the order of the test vectors can be changed, such as testing for stuck-at faults in combinational or full-scan sequential circuits, which is still one of the most popular fault models [21].

The rest of the paper is organized as follows. Section 2 provides the background on test scheduling, test power modeling, and motivates power profile manipulation. The new approach, including a new test power approximation model, is detailed in Section 3. Section 4 shows through an example how the proposed can be integrated into existing power constrained test scheduling algorithms. To validate the proposed approach, extensive experimental data is given and interpreted in Section 5, while Section 6 concludes the paper by outlining its contributions.

2 Background information

This section gives the definitions of the basic terms which will be used in the rest of the paper.

2.1 Test scheduling

Test scheduling algorithms aim to reduce test application time by increasing the parallelism of the testing activities in the system. When ignoring power issues, the maximum test concurrency is limited by the resource sharing conflicts. The blocks of a system which can be tested simultaneously without generating any resource conflicts are said to be *test compatible*. The tests which are executed at the same time form a *test session*. The tests corresponding to test compatible blocks are said to be *resource compatible tests*. The test compatibility relations among the blocks of a system are represented using the *test compatibility graph* (TCG). Each block and its corresponding test are associated to a node in the TCG. An arc between two tests in the TCG signifies that the two corresponding tests are resource compatible.

2.2 Test power modeling

In order to consider power during test scheduling, the power dissipated by the block under test needs to be modeled. *The power profiles* capture the power dissipation of a block over time when applying a sequence of test vectors to the inputs and/or pseudo-inputs of the block. The power profiles give cycle-accurate descriptions of power dissipation which makes them too complex to be considered in the test scheduling process. Therefore simple and reliable approximate power models are needed. The following section analyzes a commonly used power approximation model and justifies the need for a new power approximation model for power constrained test scheduling.

2.3 The global peak power approximation model

The power approximation model currently used by most of the existing PCTS algorithms [2, 3, 10–12, 14, 17] is the global peak power approximation model (GP-PAM). As shown in Figure 1, the GP-PAM basically flattens the power profile of a block to the worst case power dissipation value, i.e. its peak value. According to this model, the power profile of a block is described by the pair (P_{hi}, L), where P_{hi} is the global peak value of the power profile, and L is the sequence length. This simple approximation model, although it guarantees that power dissipation is not under estimated for any time instance, it introduces a high approximation error, indicated by the *false power* from Figure 1.

The false power component introduced by the power approximation model leads to under-optimal test concurrency, and hence longer test application time.

Section 3 will show how test concurrency can be increased by reducing the false power component of the power profile. The false power is minimized by changing the test power profiles and describing it using a suitable power approximation model.

3   The new power profile manipulation approach

The previous section has shown that, regardless of its simplicity and reliability, global peak power approximation model leads to large approximation errors and consequently to low test concurrency. This can be avoided if the shape of the power profile can be manipulated such that it allows a more accurate approximation. This section proposes a power profile manipulation technique for increasing test concurrency under power constraints. This approach consists of the following components:

- **test vector reordering (Section 3.1)** - initially, power profiles for the block tests are lowered by increasing the correlation between successive vectors in the test sequence; test vector reordering is used for peak power reduction, as well as for power profile reshaping;

- **test sequence expansion (Section 3.2)** - additional test vectors are added to a test sequence in order to further reduce the peak of its power profile. Only the test sequences which do not influence the test session length are extended, in order to preserve the total test application time;

- **new power approximation model (Section 3.3)** - the proposed test vector reordering produces a power profile consists of an initial low power part of the profile followed by a high power part; hence a simple and reliable approximation model exploiting these two parts will provide more accurate descriptions of the power profile than the GP-PAM;

- **test sequence rotation (Section 3.4)** - finally, the low power profiles are rotated and piled up together such that the high power parts do not overlap in order to obtain improved usage of the power constraint;

As it is shown in the following sections, the proposed methodology performs low complexity operations with on simple data representations, thus even for large amounts of test data corresponding to real-life circuits the required computational effort is achievable on typical workstations in reasonable execution times, as demonstrated by the experimental results in section 5.

3.1 Test vector reordering

In this section, *power dissipation is seen as a design objective and consequently it is minimized*. Dynamic power represents one of the main components of power dissipation in CMOS circuits. Its source is the capacitance current flowing to charge/discharge the capacitive loads during logic changes [16]. The dynamic power dissipation is dependent on the switching activity, i.e. the average number of gate transitions per clock period [18]. The number of gate transitions depends on the switching activities at the inputs of the gate as well as on the spatio-temporal correlations among gate's inputs. Thus, the order in which the patterns are applied to the primary and pseudo-inputs influences the power dissipation in the circuit. Reordered test sequences can be applied to the circuit with automatic testing equipment (ATE) when using external testing, or they can be generated on-chip using embedded deterministic tests when built-in test is adopted. It should be noted that in the case of external testing, the resource conflicts (used to generate the test compatibility graph of a system - section 2.1) are caused by sharing the limited number of channels between the ATE and the chip under test, while for embedded deterministic tests the resource conflicts are caused by sharing the test sources, sinks and access mechanisms [24].

The test vector reordering algorithm described below aims to achieve the following two objectives: *minimize the peak power dissipation values and produce a power profile suitable for simple, reliable and accurate test power modeling*. Test sequences with lower power allow higher test concurrency under a given power constraint. The use of accurate descriptions of power profiles can also increase the test concurrency under power constraints as it eliminates the "false power" component, which, from the test scheduling perspective, is equivalent with having test sequences with lower power profiles.

The input to the test vector reordering algorithm is a transition graph described below. Given a test sequence TS with N test vectors, the *input transition graph $ITG = (\Psi, E)$* can be computed. ITG is a complete directed graph with $|\Psi| = N$ nodes and $|E| = N(N-1)$ edges, where each node $V_i \in \Psi$ represents a vector in TS and each edge $(V_i, V_j) \in E$ represents a transition at the primary inputs from V_i to V_j. The ITG edges are labeled with an estimation of the power dissipated in the circuit by the corresponding input transition. The edge weights are computed differently depending on the adopted testing scheme: test-per-clock or test-per-scan [1]:

- In a test-per-clock testing scheme, the test vectors are applied to the primary inputs one vector at each clock cycle. Each edge (V_i, V_j) in ITG is weighted with the power P consumed in the circuit during the transition of the primary inputs from V_i to V_j: $Weight(V_i, V_j) = P(V_i, V_j)$. All the possible ordered pairs $(V_i, V_j), i \neq j, V_i, V_j \in \Psi$ have to be simulated using a power estimation

tool in order to compute the ITG edge weights.

- In a test-per-scan testing scheme, a test vector is first scanned-in during m clock cycles, where m is the scan chain length, then it is applied to the block during the clock cycle $m + 1$, and the circuit response is scanned-out during the next m clock cycles, simultaneously with the scanning-in of the next test vector. Edge (V_i, V_j) in ITG is weighted with the power consumed by the simultaneous scan-out of V_i and scan-in of V_j. It was shown in [19] that the *weighted transition count* (WTC) is very well correlated with the real power dissipation. The WTC values corresponding to V_i scan-in and scan-out respectively are given by:

$$WTC_{scanin}(V_i) = \sum_{j=1}^{m-1} (V_i(j) \oplus V_i(j+1))(m-j) \tag{1}$$

$$WTC_{scanout}(V_i) = \sum_{j=1}^{m-1} (V_i(j) \oplus V_i(j+1))j \tag{2}$$

where $V_i(j)$ represents the jth bit from vector V_i. Finally, ITG edge weights for test-per-scan are computed using:

$$Weight(V_i, V_j) = WTC_{scanout}(V_i) + WTC_{scanin}(V_j)$$

It should be noted that although the switching activity during the capture cycle is not considered in the above equation, the edge weight formulation can easily be extended to account for it.

Having computed the ITG edge weights, reordering the test sequence for low power reduces to the problem of finding in ITG a low cost Hamiltonian tour. As ITG is a complete directed graph, finding a low cost Hamiltonian cycle in it represents an instance of the asymmetric traveling salesman problem which is known to be NP-hard. Therefore, a greedy depth-first search heuristic was implemented to determine a good solution to this problem. The algorithm starts from a randomly selected vector in the sequence and at each iteration selects the neighboring node which generates the lowest power dissipation, i.e. the outgoing edge with the smallest weight. *Due to the greedy nature of the method adopted for traversing the ITG the power profile corresponding to resulting path will exhibit an initial long low power part followed by a short high power part towards the end of the sequence.* This is because the edges with lower weights are added to the path in early iterations, leaving the edges with higher weights to the end of the profile. This particular shape of the power profile has the following advantages:

- it has lower peak power than a random path in the graph (such as the initial unordered test sequence);

- in conjunction with the new power approximation model introduced later it brings significant approximation accuracy improvement over its GP-PAM representation as shown later in Table 1 in Section 5;

3.2 Test sequence expansion

It was shown that peak power can be reduced by increasing the correlation, i.e. reducing the number of transitions between consecutive test vectors. The previous section achieved this by reordering the test sequence. Another way of increasing the correlation between consecutive vectors is to insert additional test vectors in the test set between vector pairs causing high power dissipation. The following example illustrates this method.

Example 1 Consider a test-per-clock scheme the the following test vector pair: $V_1 = 00001111$ and $V_2 = 11110000$. The sequence (V1,V2) will produce 8 transitions at the inputs of the circuit. However, by inserting $V^* = 00111100$ between V_1 and V_2 will produce only 4 transitions for each of the two clock cycles. Consider now a test-per-scan scheme, and the following test vectors: $V_1 = 1010$ and $V_2 = 0101$, where V_1 is the test response which needs to be shifted-out from the scan chain and V_2 is the next test vector to be loaded into the scan-chain. Shifting the sequence (V1,V2) will produce 12 (6+6) transitions at the inputs of the circuit (see Equations 1 and 2 from Section 3.1). However, by inserting between V_1 and V_2 $V_{in}^* = 0001$, $V_{out}^* = 1000$, where V_{out}^* is the circuit test response for V_{in}^* will produce only in 7 transitions on circuits inputs during each of the two shifting cycles.

The previous example has shown how switching activity at circuit's inputs can be reduced by inserting carefully selected test vectors in the test sequence. Test sequence expansion is suitable for nonpartitioning testing schemes for unequal test lengths [5]. The test length differences can be used to extend the shorter tests in order to lower their power profiles without increasing the test session length. Thus, higher test concurrency can be achieved under given power constraints, which leads to shorter test times. To be noted that the amount of additional vectors has to be kept low as it affects the test data storage requirements.

3.3 New power approximation model

Section 3.1 has shown how, *by considering power as a design objective*, test vector reordering can generate a test sequence with a regular power profile which has an initial long low power part followed by a short high power part towards the end of the sequence. This regular shaped power profiles can be accurately described using simple approximation models as shown in the example from Figure

7

2. By modeling low and high power parts of the profile using their local power peaks and lengths $((P_{lo}, L_{lo})$ and $(P_{hi}, L_{hi}))$, then the value, position and size of each part of the profile are available as inputs for power constrained test scheduling algorithms. The improvement in approximation accuracy compared to the GP-PAM, which is represented by the dashed rectangle in Figure 2, is given by $\Delta_{approx.improv} = (P_{hi} - P_{lo})L_{lo}$. The new power approximation model will be further referred to as the *two local peak power approximation model* (2LP-PAM) and will be represented by the 4-tuple $(P_{lo}, L_{lo}, P_{hi}, L_{hi})$. While P_{hi} is fixed to the global peak value, P_{lo} can be derived based on the L_{lo} and L_{hi} ratio. Thus several 4-tuple descriptions are possible for the same power profile, however the optimum is the one with the highest $\Delta_{approx.improv}$. This optimum 4-tuple approximation can be computed in linear time with the length of the test sequence.

3.4 Test sequence rotation

Having lowered and reshaped the test power profiles, using test vector reordering and modeled using the 2LP-PAM, this section explains how more compatible tests can be combined into a test session through the use of test sequence rotation. Since 2LP-PAM offers information on the position and size of both low and high power parts, the power profiles can be rotated such that when added to a test session their high power parts do not overlap with the high power parts of the profiles already in the test session. *This leads to a higher test concurrency under power constraints* as explained in the following example.

Example 2 Consider the power profiles shown in Figure 3 that belong to two compatible test sequences TS1 and TS2 that can be merged in the same test session. Figures 3(a) and 3(b) show the 2LP-PAM power profile approximations corresponding to TS1 and TS2 with reordered test vectors. First TS2 is added to the empty test session. Then, TS1 is rotated left by L_{hi2} vectors, as illustrated in Figure 3(c). The joint power profile obtained by adding the rotated TS1 to the test session is shown in Figure 3(d). Unlike the GP-PAM based approach where the maximum power dissipation for the test session composed of the two tests would be given by

$$P_{Session}(GP - PAM) \quad = \quad \sum_{t_i \in Session} P_{hi}(t_i) = P_{hi_1} + P_{hi_2}$$

by using the 2LP-PAM, the maximum test session power dissipation becomes

$$P_{Session}(2LP - PAM) \quad = \quad \max_{t_i \in Session} \left(P_{hi}(t_i) + \sum_{t_j \in Session, t_j \neq t_i} P_{lo}(t_j)\right) =$$
$$= \quad max(P_{hi_1} + P_{lo_2}, P_{hi_2} + P_{lo_1}) < P_{hi_1} + P_{hi_2}.$$

8

Thus, $P_{Session}(2LP_PAM) < P_{Session}(GP_PAM)$. This examples has shown how by controlling the rotation of the test sequences before adding them to a test session, the high power parts of their power profiles are uniformly spread over the entire test session length, rather than being piled up on top of each other, as in the case of the GP-PAM approach. Therefore, joint power profile of the test session when using the 2LP-PAM, becomes more flat and can fit more tests under the same power constraint. Test sequence rotation does not influence the peak or average power of a test sequence; rather, it helps the test scheduling algorithm to perform better allocation of the power under the power constraint, and consequently produce shorter test schedules.

It should be noted, that *cyclic power profiles* are needed for test sequence rotation. A cyclic power profile has to contain the transition between the last test vector in the test sequence and the first one. Test sequence rotation consists only in assigning a value to an *offset* parameter which specifies from which position in the initial test sequence starts the rotated one. Therefore, the computational effort involved is virtually inexistent.

4 Power constrained test scheduling using the proposed power profile manipulation

So far, the new power profile manipulation approach was introduced using the following components: test vector reordering, test sequence expansion, two local peak power approximation model, and test sequence rotation. This section shows how power profile manipulation can be integrated into existing power constrained test scheduling algorithms.

The non-partitioning test scheduling algorithm for unequal test lengths proposed in [3] will be extended for use in conjunction with power profile manipulation. We maintain the assumption made in [3] that a new test session cannot start before all tests in the current test session are completed, even if the resources required for the new test session are already available. A practical reason for this restriction is that interruption of a test session to start new tests increases the complexity of the test controller which will add area overhead. The extended PCTS algorithm starts with a preprocessing step (lines 1 to 5 in Figure 4), where all the test sequences are *reordered* (Section 3.1) and *extended* (Section 3.2) by a small, fixed number (up to 5 in our experiments) of additional test vectors. The resulting power profiles for both the reordered and the reordered-and-extended test sequences are modeled using the 2LP-PAM. The high power parts of the power profiles are then moved to the beginning of the test sequence by rotating them to the left by L_{lo} vectors (line 4).

Next, the algorithm shown in Figure 4 determines all the cliques (i.e. completely connected

subgraphs) of the TCG (line 6). The TCG cliques represent the maximal groups of test compatible blocks. For each TCG clique, the algorithm computes all the maximal ordered subsets which comply with the given power constraint, referred to as the *power compatible lists* (PCLs) (lines 7 and 8). The following example explains PCLs.

Example 3 Consider the system with the TCG and 2LP-PAM power profiles (Section 3.3)shown in Figure 5. The cliques in this case are $(T_2, T_5), (T_1, T_4, T_5), (T_3, T_4, T_5)$. The maximality requirement of the PCLs means that no other test can be added to them without exceeding the power constraint. The tests in a PCL are arranged in the descending order of their length. Consider the test compatible clique composed of tests (T_3, T_4, T_5). Figures 6(a) and 6(b) show the PCLs corresponding to the clique and to a power constraint of 10 using 2LP-PAM and respectively GP-PAM approximations. The test list sorted in descending order of their length is (T_5, eT_3, eT_4), where eT_3 and eT_4 represent the *reordered-and-extended* test sequences T_3 and T_4. T_5 determines the test session length is the longest test sequence in the clique, thus the power profile for non-extended test sequence T_5 is used. However, T_3 and T_4 can be extended as their length is smaller then the length of the session to which they are about to be assigned to. T_5 added unrotated to the empty test session. The next test in the ordered list, eT_3, is rotated right and added to the test session such that its high power part does not overlap with the high power part of T_5 which is already in the test session. Finally, test eT_4 is rotated right such that its high power part does not overlap with the high power parts of the two test sequences already in the test session. The maximum value of the resulted power profile is $P_{max} = P_{hi5} + P_{lo3} + P_{lo4} = 9$, which is less than the power constraint. This means that by using the 2LP-PAM on reordered (and some extended) test sequences all tests in the clique can be scheduled in the same 100 clock cycles long test session under the given power constraint, while by using the GP-PAM on the original test sequences, two test sessions, summing up 160 clock cycles, are required to cover all tests in the clique.

The PCLs are determined for each clique C from TCG and the given power constraint using the algorithm shown in Figure 7. For each subset of C the algorithm computes the optimum arrangement of its tests using the test sequence rotation described earlier in Section 3.4. The Offset variable guides the rotation of the test sequences to be inserted into the current test session. For the longest test(s) in the test subset, *its reordered-non-extended* test sequence is used in order to preserve the length of the test session, while for shorter tests, the *reordered-and-extended* test sequences are used as they exhibit lower power profiles and do not increase the test session length.The maximal power compatible subsets are then added to the set of PCLs.

Finally, finding the optimum test schedule under the given power constraint is reduced to the problem of finding a minimum cost cover for the PCLs set (line 9 in Figure 4), where the cost associated to

each PCL is the length of the longest test in the PCL, i.e. the test session length. In the proposed implementation, the minimum cost covering problem was formulated as an integer linear programming (ILP) problem and solved using *lp_solve* [20].

This section has shown how the proposed power profile manipulation can be integrated into power constrained test scheduling algorithms. Although the integration was detailed for the algorithm presented in [3], the proposed approach can be included into *any* other existing power constrained test scheduling algorithm [2, 10–12, 14, 17], to leverage its performance. *This is possible because the proposed power profile manipulation approach is a complementary technique to, and hence it is independent on, the test scheduling policy.*

5   Experimental results

This section describes the experiments performed to asses the efficiency of the power profile manipulation technique. The algorithms were implemented in C++ and run on an AMD Athlon 1.2Ghz workstation with 384Mb RAM running Linux. Due to the simplicity of the WTC model, determining the ITG edge weights does not require a high computational effort. For example, the ITG weights for a test set of 7000 vectors, 1000-bit wide for a test-per-scan scheme were computed in less than 120 seconds. Reordering the vectors of the same sequence was computed in 71 seconds, while expanding the sequences by 5 test vectors as well as computing the two local peak approximation were each performed in less than one second.

The first set of experiments compares the proposed 2LP-PAM with the GP-PAM in terms of approximation accuracy improvement, which basically shows how much "false power" is saved by using the 2LP-PAM. Test-per-clock and, where suitable, test-per-scan test sequences for the largest ISCAS85 and ISCAS89 benchmark circuits were reordered using the algorithm described in section 3.1. The power profiles of the reordered sequences were approximated using the 2LP-PAM and the GP-PAM. Columns 2 and 3 in Table 1 shows the improvement in approximation accuracy of the proposed 2LP-PAM over the traditional GP-PAM. Next, the efficiency of the test sequence expanding method is evaluated. The reordered test sequences were extended with 5 test vectors which led to the peak power reductions shown in column 4 in Table 1.

The next set of experiments evaluates the performance improvement which can be achieved by integrating the proposed power profile manipulation approach into an existing power constrained test scheduling algorithm. We have integrated our method into the power constrained test scheduling algorithm presented in [3]. The modified and the original algorithms were applied on hypothetical systems. Each system was represented as a set of embedded blocks and a randomly generated test

compatibility graph which shows the dependencies between the embedded blocks. The embedded blocks were selected randomly from the ISCAS85 and ISCAS89 benchmarks. Systems with 8 to 16 blocks were considered in our experiments. The original PCTS algorithm from [3] was applied on unordered test sequences and used the GP-PAM to model power. The resulting test application times for a wide range of power constraints are reported in column 3 of Tables 2 and 3 respectively. By integrating the proposed power profile manipulation approach, that is test vector reordering, test sequence expansion, the 2LP-PAM and test sequence rotation, into the PCTS algorithm from [3] reduced the previous test application times to the values reported in column 4 of Tables 2 and 3 respectively. The percent reduction of the test application time through the use of the proposed power profile manipulation approach in conjunction with the original algorithm is reported in column 5.

The experimental results show how by using the proposed power profile manipulation in conjuction with an existing Power constrained test scheduling algorithm, test application time can be reduced with up to up to 41%.

6 Concluding remarks

Currently, two main research directions can be identified in low power testing. The first research direction considers power dissipation during test as a minimization objective. The second direction considers power dissipation as a design constraint, while test application time becomes the minimization objective. This paper presented a link between the two separate research directions, using a new power profile manipulation approach based on the following components: test vector reordering, test sequence expansion, two local peak power approximation model and test sequence rotation. Test vector reordering is used to lower and reshape the test power profiles. Test sequence expansion further lowers the power profiles of shorter tests which do not affect the total test application time. Then the proposed two local peak power approximation model translates the power profile it into a simple, reliable and accurate test power representation, which can be exploited by test sequence rotation in order to increase test concurrency under a power constraint. Since the proposed power profile manipulation is orthogonal to the test scheduling policy and the test set values, the distinctive feature of the proposed solution is that it can be *equally* included in, and consequently leverage the performance of, *any* existing power constrained test scheduling algorithm.

References

[1] V.D. Agrawal, C.R. Kime, and K.K. Saluja. A tutorial on built-in self test - part 2: Applications. *IEEE Design and Test of Computers*, 10(1):69–77, June 1993.

[2] K. Chakrabarty. Design of system-on-a-chip test access architectures under place-and-route and power constraints. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pages 432–437, 2000.

[3] R.M. Chou, K.K. Saluja, and V.D. Agrawal. Scheduling tests for VLSI systems under power constraints. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2):175–184, June 1997.

[4] F. Corno, M. Rebaudengo, M. S. Reorda, and M. Violante. Optimal vector selection for low power BIST. In *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 219–226, 1999.

[5] G.L. Craig, C.R. Kime, and K.K. Saluja. Test scheduling and control for VLSI built-in self-test. *IEEE Transactions on Computers*, 37(9):1099–1109, September 1988.

[6] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S.M. Reddy. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1325–1333, December 1998.

[7] P. Flores, J. Costa, H. Neto, J. Monteiro, and J. Marques-Silva. Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation. In *12th International Conference on VLSI Design*, pages 37–41, 1999.

[8] P. Girard. Low power testing of VLSI circuits: Problems and solutions. In *First International Symposium on Quality of Electronic Design (ISQED)*, pages 173–180, 2000.

[9] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac. Reduction of power consumption during test application by test vector ordering. *IEE Electronics Letters*, 33(21):1752–1754, 1997.

[10] V. Iyengar and K.Chakrabarty. Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip. In *VLSI Test Symposium (VTS)*, pages 368–374, 2001.

13

[11] Erik Larsson and Zebo Peng. An integrated system-on-a-chip test framework. In *Design Automation and Test conference in Europe (DATE)*, pages 138–144, 2001.

[12] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu. A comparison of classical scheduling approaches in power-constrained block-test scheduling. In *Proc. IEEE International Test Conference (ITC 2000)*, pages 882–891, 2000.

[13] N. Nicolici. *Power Minimisation Techniques for Testing Low Power VLSI Circuits*. PhD thesis, University of Southampton, UK, http://www.bib.ecs.soton.ac.uk/records/4937/, October 2000.

[14] N. Nicolici and B.M. Al-Hashimi. Power conscious test synthesis and scheduling for BIST RTL data paths. In *Proc. IEEE International Test Conference (ITC 2000)*, pages 662–671, 2000.

[15] N. Nicolici, B.M. Al-Hashimi, and A.C. Williams. Minimisation of power dissipation during test application in full scan sequential circuits using primary input freezing. *IEE Proceedings - Computers and Digital Techniques*, 147(5):313–322, September 2000.

[16] M. Pedram. Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(1):3–56, January 1996.

[17] C.P. Ravikumar, G. Chandra, and A. Verma. Simultaneous module selection and scheduling for power-constrained testing of core based systems. In *13th International Conference on VLSI Design*, pages 462–467, 2000.

[18] K. Roy and S. Prasad. *Low-Power CMOS VLSI Circuit Design*. John Wiley & Sons, 2000.

[19] R.R.Oruganti R.Sankaralingam and N.A.Touba. Static compaction to control scan vector power dissipation. In *VLSI Test Symposium*, pages 35–40, 2000.

[20] H. Schwab. LP Solve. In *http://elib.zib.de/pub/Packages/mathprog/linprog/lp-solve*, 1997.

[21] Test and Measurement World, October 2001. Rethink fault models for submicron-ic test.

[22] S. Wang and S.K. Gupta. ATPG for heat dissipation minimization during test application. *IEEE Transactions on Computers*, 47(2):256–262, February 1998.

[23] Y. Zorian. A distributed BIST control scheme for complex VLSI devices. In *Proc. 11th IEEE VLSI Test Symposium*, pages 4–9, 1993.

[24] Y. Zorian, E.J. Marinissen, and S. Dey. Testing embedded core-based system chips. *Computer*, 32(6):52–60, June 1999.
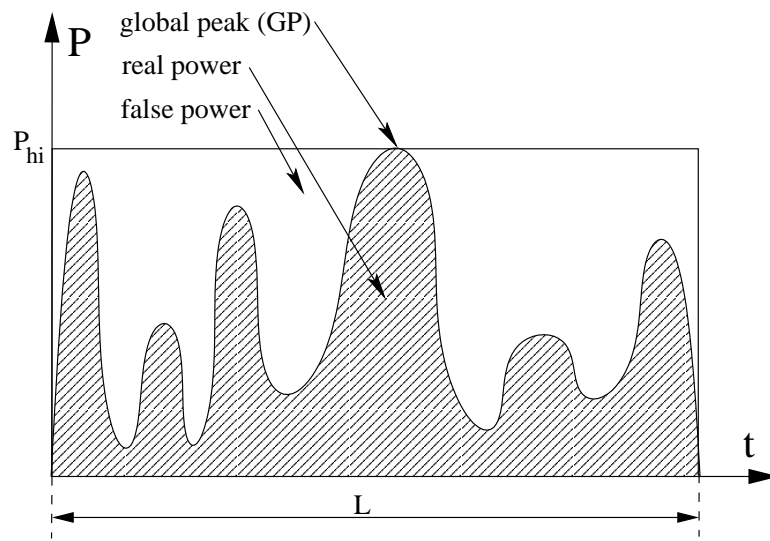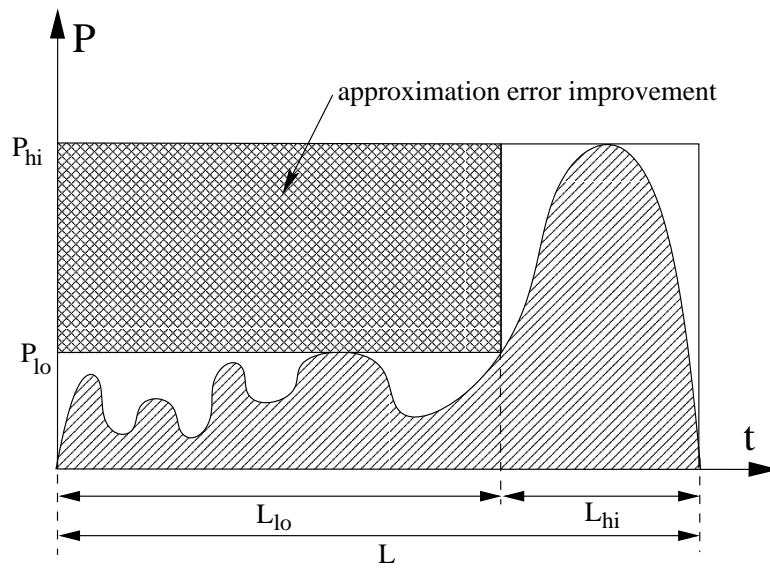
Figure 1: Global peak power approximation model

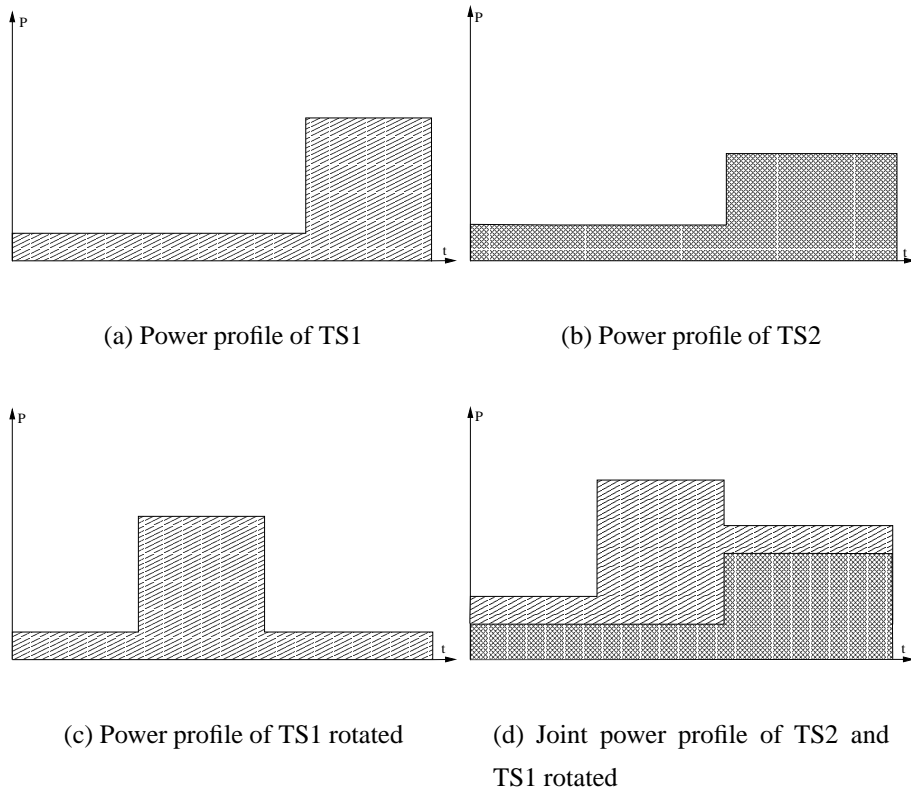Figure 2: Approximation of regular power profiles

(a) Power profile of TS1

(b) Power profile of TS2

(c) Power profile of TS1 rotated

(d) Joint power profile of TS2 and TS1 rotated

Figure 3: Test sequence rotation

ALGORITHM: **PCTS Using the Proposed Power Profile Manipulation**

INPUT: test compatibility graph (TCG), power constraint P_{constr} and additional vector count AVC

OUTPUT:power constrained test schedule

1 **for every** test sequence TS_i {

2   **reorder** TS_i and **extend** by AVC vectors the reordered test sequence

3   **compute** the 2LP-PAM approximations

            for both the reordered and the reordered-and-extended test sequences

4   **rotate left** the power profile by L_{lo} vectors

5 }

6 **compute** clique set Ω for TCG

7 **for every** clique $C_i \in \Omega$

8   **compute** power compatible lists PCL_i for C_i P_{constr} using the algorithm from Figure 7

9 **compute** power constrained test schedule as a minimum cost cover of the PCLs set

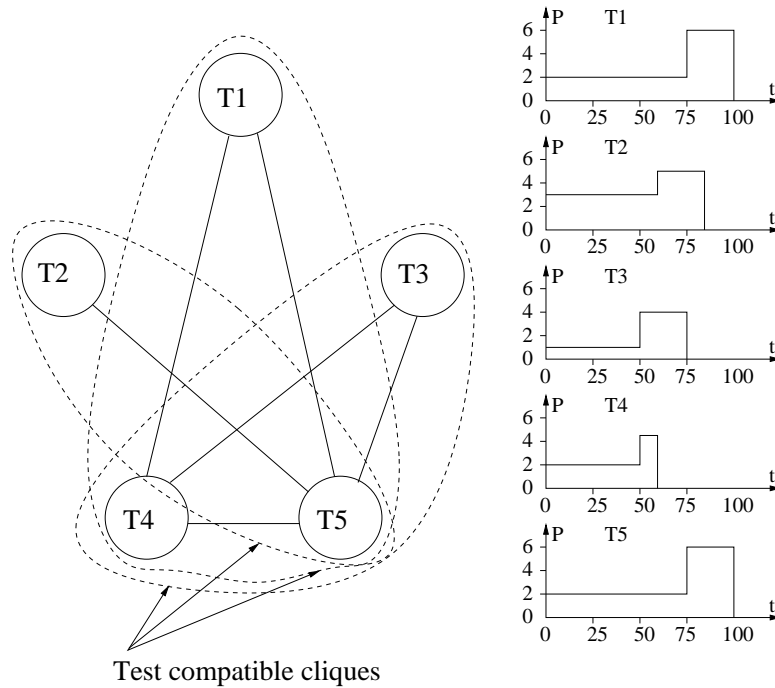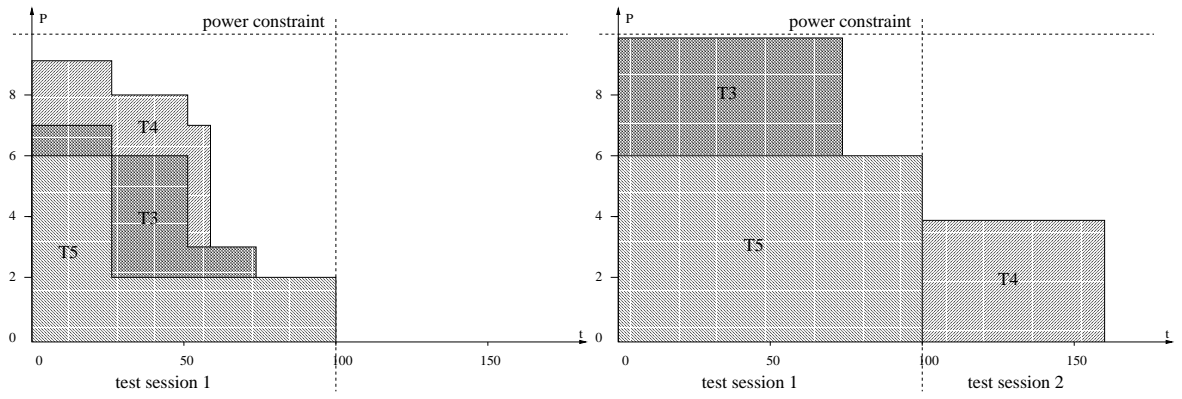Figure 4: PCTS using the proposed power profile manipulation

Test compatible cliques

Figure 5: Example: Test compatibility graph

(a) PCL for (T_3, T_4, T_5) using the 2LP-PAM

(b) PCL for (T_3, T_4, T_5) using the GP-PAM

Figure 6: Example: Power compatible lists under the 2LP-PAM and GP-PAM

ALGORITHM: **Power Compatible Lists**

INPUT: test compatible clique C and the power constraint P_{constr}

OUTPUT: power compatible lists PCL for C and P_{constr}

1 PCL = ϕ

2   **for every** subset $S_i \subset$ C {

3   Offset = 0; Session = ϕ

4   **compute** pcl_i by sorting tests in S_i in descending order of their lengths

5   **for every** test $T_j \in pcl_i$ {

6   **if** ($L_{lo\,j} \geq Offset$) **rotate right** T_j by Offset vectors **else** Offset = 0

7   Session = Session $\cup T_j$

8   }

9   **compute** maximum power dissipation P_{max} for Session

10   **if** ($P_{max} < P_{constr}$) {

11   MaximalSet = TRUE

12   **for every** $T_j \in C$ **and** $T_j \notin pcl_i$ **with** length less than the longest test sequence in pcl_i {

13   **if** ($L_{lo\,j} \geq Offset$) **rotate right** T_j by Offset vectors

14   Session' = Session $\cup T_j$

15   **compute** maximum power dissipation P'_{max} for Session'

16   **if** ($P'_{max} \leq P_{constr}$) MaximalSet = FALSE;

17   }

18   **if** (MaximalSet = TRUE) PCL = PCL $\cup pcl_j$

19 }

Figure 7: Power compatible lists

List of Figures

Circuit	Approximation accuracy improvement(%)		Peak pow. reduction (%)
	test-per-clock	test-per-scan	by extending with 5 vec.
c1355	3.97	10.93	13.89
c1908	9.59	7.52	8.3
c432	14.04	23.23	13.7
c499	7.36	8.83	5.14
c6288	2.76	13.53	7.24
c7552	21.94	4.35	11.51
s5378	N/A	8.4	9.02
s9234	N/A	6.75	3.53
s13207	N/A	14.41	4.75
s15850	N/A	11.77	7.5
s35932	N/A	31.51	38.16
s38417	N/A	0	7.87
s38584	N/A	0	2.75

Table 1: Approximation accuracy improvement for test-per-clock and test-per-scan testing schemes

Block count	Power constr.(mW)	$T_{orig.PCTS}$ (clks) [3]	$T_{proposedPCTS}$ (clks)	Test time reduction (%)
8	330	594	477	19.7
8	346.5	594	477	19.7
8	363	594	477	19.7
10	412.5	720	604	16.11
10	429	720	604	16.11
10	445.5	720	604	16.11
12	445.5	1069	714	33.21
12	561	952	598	37.18
12	577.5	831	598	28.04
14	363	1070	836	21.87
14	445.5	948	715	24.58
14	544.5	831	598	28.04
16	412.5	1545	1074	30.49
16	511.5	1428	952	33.33
16	544.5	1428	836	41.46

Table 2: Experimental results for the test-per-clock testing scheme (ISCAS85 benchmarks)

Block count	Power constr.(WTC/1000)	$T_{orig.PCTS}$ (clks) [3]	$T_{proposedPCTS}$ (clks)	Test time reduction(%)
8	450	870514	545696	37.31
8	550	718986	521173	27.51
8	600	718986	466877	35.06
8	650	533826	406419	23.86
10	500	958550	624515	34.84
10	650	642928	545696	15.12
10	750	588524	460715	21.71
10	900	533826	381896	28.46
12	500	991888	649038	34.56
12	600	916124	649038	29.15
12	800	642928	545696	15.12
12	850	567164	491400	13.35
14	650	903852	664288	26.50
14	700	794750	612645	22.91
14	750	794750	579307	27.10
14	850	664288	555186	16.42
16	700	903852	700681	22.47
16	800	903852	667343	26.16
16	850	828088	649038	21.62
16	900	828088	570219	31.14

Table 3: Experimental results for the test-per-scan testing scheme (ISCAS89 benchmarks)

List of Tables