

Power-Aware Optimization of SoC Test Schedules Using Voltage and Frequency Scaling

Vijay Sheshadri^{1,2} · Vishwani D. Agrawal¹  · Prathima Agrawal¹

Received: 15 September 2016 / Accepted: 17 February 2017 / Published online: 15 March 2017
© Springer Science+Business Media New York 2017

Abstract With shrinking device sizes, System-on-Chip (SoC) cores are growing in number and complexity. This has led to high volumes of test data and long test times. Therefore, reducing test cost by minimizing the overall test time is one of the main goals of SoC testing. To efficiently manage test resources and power dissipation, tests for the SoC cores are arranged into test schedules. Traditional SoC test methods assume a constant test frequency and supply voltage (V_{DD}) for the entire test schedule. However, test power and test time can be regulated by varying V_{DD} and test clock frequency to optimize SoC test schedules for a given power budget. The research presented in this paper focuses on power-aware optimization of SoC test schedules to minimize test time by scaling the supply voltage and test clock rate. This scaling can be on a per session basis (in case of session-based test schedules) or dynamically (in case of sessionless test schedules). Exact and heuristic algorithms for solving the optimization problem are discussed.

These algorithms are implemented and applied to several SoC benchmarks. Results show a significant reduction in SoC test time over the conventional test schedules where V_{DD} and clock are fixed at given nominal values.

Keywords System-on-chip (SoC) test · Integer linear program · Power constrained test · Test scheduling · Test time reduction

1 Introduction

Advances in semiconductor technology have made it possible to integrate an entire system onto a single chip. Owing to their modularity, small area and low power consumption, System-on-Chip (SoC) devices are becoming increasingly popular. An SoC is often designed by embedding reusable blocks called cores. These cores or blocks include multiple types of design blocks and intellectual property blocks, such as digital logic, processors, memories, and analog and mixed signal circuits. The number of cores being embedded in SoC devices is increasing with further advancements in technology [14]. The resulting increased complexity in SoC devices has led to high volumes of test data and longer test times. Thus, test time reduction has become a major objective in the field of SoC testing.

While testing multiple cores simultaneously can reduce the test time significantly, such concurrent execution is limited by high power consumption due to increased switching activity. The power consumption of a circuit during test mode is often higher than that in the functional mode [9, 34, 37]. Elevated power levels and heat dissipation by neighboring cores can lead to the formation of thermal hotspots and undesirable power droops. Thermal hotspots may eventually cause irreversible damages to the chip whereas power

Responsible Editor: K. K. Saluja

✉ Vishwani D. Agrawal
agravvd@auburn.edu

Vijay Sheshadri
vijay.sheshadri@intel.com

Prathima Agrawal
agravpr@auburn.edu

¹ Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849, USA

² Present address: Intel Corporation, Hillsboro, OR 97124, USA

droops induce clock stretching which may lead to a good chip incorrectly failing a timing test [9]. Therefore, power-aware test strategies are needed for efficient test power management. Power management here refers to both peak and average power. While peak power is caused by instantaneous switching and creates power droop, average power is a result of high activity over a time period and leads to thermal issues. Recently, reduced voltage testing has been shown to significantly reduce test time under specified power constraints [7, 42, 47]. By scaling the supply voltage during testing, the test power can be lowered drastically, thereby allowing an increase in the test clock rate without violating the power limit. The concept of varying V_{DD} and clock frequency for SoCs has been proposed in the past. Termed as dynamic voltage and frequency scaling (DVFS) techniques, they have been adopted to optimize energy efficiency, leakage power management, etc., in multi-core SoCs [1, 45].

In the work presented in this paper, a power-aware test scheme is proposed that optimizes the overall test time of an SoC by exploiting the influence of V_{DD} and test clock rate over test power and test time of individual cores. In this work, both exact and heuristic approaches for test optimization are provided; while the exact method, using Mixed-Integer Linear Programming (MILP) method, provides the most optimal result, the heuristic method achieves near-optimal results but addresses the problem of scalability. The remainder of the manuscript is organized as follows: Prior work in the area of SoC test scheduling is outlined in Section 2. Section 3 discusses the scaling of frequency and voltage for power management. Section IV discusses the proposed MILP formulation. Section 5 discusses the proposed MILP formulation and simulated annealing based heuristic algorithm for session-based test scheduling, including experimental results and the experimental setup. Optimization of sessionless test schedules,

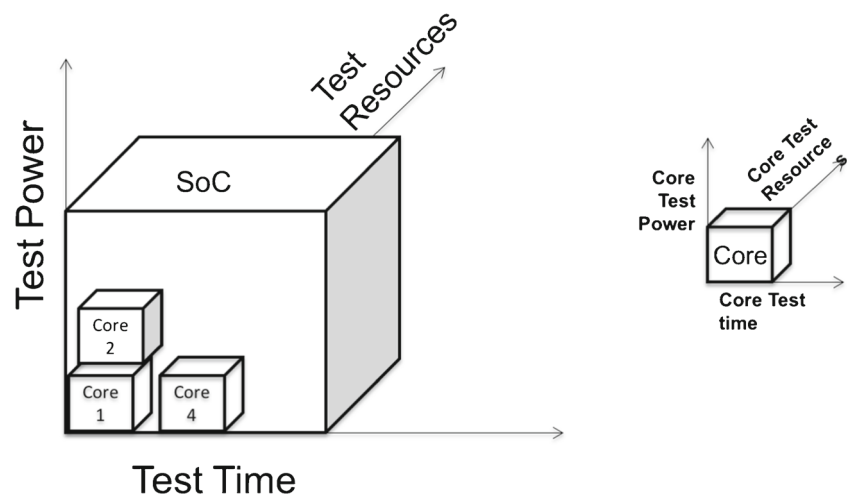
along with the optimal results, is presented in Section 5. A post-result discussion is presented in Section 6 and finally, Section 7 concludes the paper.

This paper documents doctoral research [40] supported by National Science Foundation Grants CCF-1116213, IIP-0738088 and IIP-1266036, and the Wireless Engineering Research and Education Center (WEREC) at Auburn University. Parts of this work have appeared in recent articles [41–44].

2 Related Work

The SoC testing problem can be modeled as a 3-dimensional optimization problem, where the SoC's power limit, test time and resources (such as pin count, etc.) form the three axes. The power limit is fixed for the SoC and the resources have a limited availability. The objective of the 3-D optimization would be to minimize the test time by effective allocation of resources such that the power limit is not exceeded. This optimization problem has been modeled as a 3-D bin packing problem [13] as shown in Fig. 1. Each core in the SoC can be modeled as a cuboid, where the core's test power, test time and test resources, such as Built-In Self-Test (BIST) resources, wrapper width, etc., constitute the three dimensions. The idea here is to place the cores in the cuboid representing the SoC in such a way that the test time is minimized while satisfying the power and resource constraints. This bin packing problem differs from the general bin packing problem in that if two cores are tested simultaneously, they overlap only on the time axis and not on the other two axes. This is because when multiple tests are scheduled to run at the same time instant, their powers add up and also, the same hardware resources cannot be shared among simultaneously running tests. For example: in Fig. 1, cores 1 and 2 are tested at the same time (concurrently),

Fig. 1 SoC Test scheduling modeled as 3D optimization problem



stacked along the power axis, but are misaligned on the test resource axis (resource conflict).

The complexity of SoC testing is mitigated to an extent by adopting modular testing, where individual blocks (cores) of the SoC can be tested almost independent of one another. Concurrent testing of these cores involves a trade-off between test time and test power. An optimal arrangement of core tests can be found so as to yield a minimal test time while maintaining the test power under given safe limit. Such an arrangement is termed a *test schedule*.

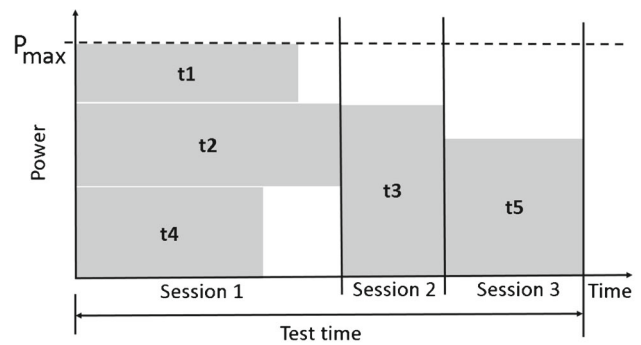
A test schedule consists of one or more *test sessions*, each containing a set of core tests to be run concurrently [4–6, 24, 55]. Existing test scheduling strategies may be broadly classified as:

- Session-based test scheduling, where no new test is allowed to start until all tests of a previous session are completed [4, 5, 24, 41, 42, 55]. The illustration in Fig. 2a shows five tests, t1 through t5, arranged in three sessions conforming to a power budget P_{max} . No test is partitioned or split to run in multiple sessions separated by vertical lines.
- Sessionless test scheduling, where test session boundaries are ignored and a test may be scheduled to start as soon as possible [12, 30, 38, 43, 44, 56]. The sessionless or partitioned test scheduling can be further divided into preemptive and non-preemptive scheduling illustrated in Fig. 2b and c, respectively. In the preemptive strategy, tests can be interrupted and restarted at any time [16, 23]. An example is the splitting of test t2 as t2a and t2b in Fig. 2c. The non-preemptive strategy does not allow such interruptions, i.e., a test once initiated must be completed. Only non-preemptive strategy is considered in this work.

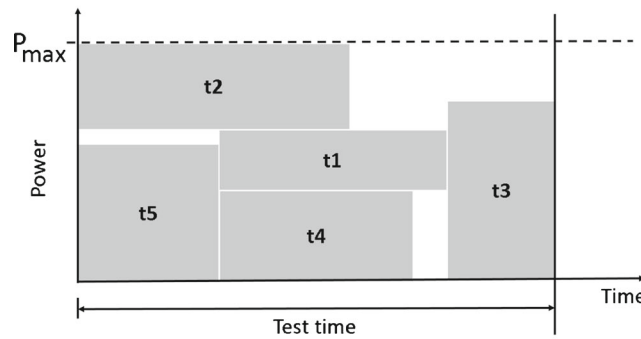
The three test scheduling strategies shown in Fig. 2 were originally proposed by Craig et al. [6], who referred to them as nonpartitioned, partitioned with run to completion, and partitioned testing, respectively.

Power-constrained test scheduling, where a test power budget is defined for the SoC, has been studied in the past [4, 5, 9, 16, 23, 39, 41, 42, 53]. In general, functional activity on the SoC causes power consumption. Instantaneous peak power determines the maximum supply current and average power determines the heat generation and temperature. Proper physical design of power buses, heat sinks, etc., is required to ensure performance and reliability. Thus, specifications of an SoC, which take the functional mode peak power and average power into account, must not be exceeded during test.

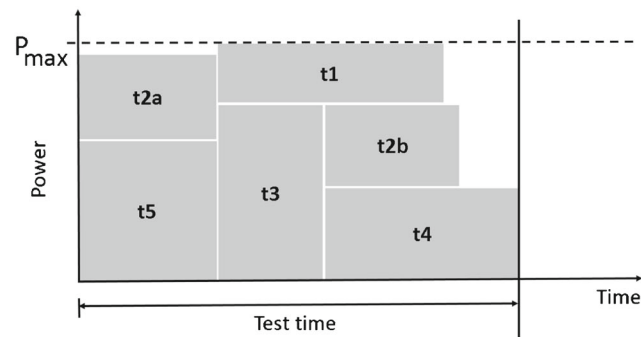
The concept of fixing a single power budget for the SoC is known as the global peak power model and has been widely used in the literature. However, this model is regarded as a pessimistic approach since the single power



(a) Session based test schedule.



(b) Session-less test schedule with non-preemptive policy.



(c) Session-less test schedule with preemptive policy.

Fig. 2 Session based and session-less test scheduling strategies [6, 22, 24]

limit value for the entire SoC is based on the peak power consumption of the circuit. Samii et al. [37] proposed a cycle-accurate power model where there is a power value for every clock cycle. While this model is more accurate than the global peak power model, it is more complex and computationally expensive and hence, in this research, the global peak power model is adopted. Several papers [17, 18, 21, 33, 53] have considered test time reduction through optimizing the test access mechanism (TAM) and wrapper chains of the core. In our work, we make use of the test clock frequency to reduce test time and it is assumed that the design for testability (DFT) infrastructure is already in place for the

SoC and that the TAM assignment and the wrapper design have been optimized.

The idea of scaling voltage and frequency has been prevalent in the field of microprocessors and SoCs. In [45], a locally placed configurable dynamic voltage and frequency scaling (DVFS) controller enables a large number of on-chip processors to switch V_{DD} by selecting from two power grids and also independently controls their clock rates, in order to improve the energy efficiency of the multi-processor SoC (MPSoC). Voltage and frequency scaling techniques have also been employed in testing of SoCs.

Recently, multi-frequency SoC testing has been investigated. Sehgal et al. [39] have focused on the use of a multi-channel ATE capable of providing multiple frequencies. Zhao et al. [52] discuss test optimization through wrapper design in order to perform bandwidth matching between the ATE's clock input and the core's frequency. The PMScan system, introduced by Devanathan et al. [7], utilizes an adaptive supply voltage regulation scheme that lowers the V_{DD} to balance the power dissipation and the frequency during the shift operation, while satisfying the timing requirements. Scheduling with multiple voltage islands and testing of cores at multiple voltages has also been considered [19]. These authors schedule core tests at multiple voltage levels and clock domains and reduce the clock frequency during low voltage testing to enable a time division multiplexing scheme for concurrent testing of cores.

Venkataramani et al. [46–51] discuss two constraints on test speed, namely, a power constraint when the test clock speed is limited by the rated power and a structure constraint when the test clock speed is limited by the critical path or other timing considerations. It has been reported [34] that at the nominal voltage test power can be twice or even higher than the functional power. However, the circuit may be designed to consume no more than the maximum functional mode power. The test is thus power constrained and must run at a slower clock rate to keep the test mode power dissipation under the design limit. This test clock may be significantly slower than the fastest permitted by the critical path delay. Lowering the operating voltage relaxes the power constraint on the clock rate, allowing the clock rate to increase as long as it still satisfies the constraint of the somewhat increased critical path delay. As we reduce V_{DD} , on one hand, the reduced power consumption allows higher clock rates thereby shrinking the test time while, on the other hand, the increased circuit delay enforces slower clock rates and longer test time.

Figure 3 [46–51] shows this phenomenon. The left curve is the structure constrained lower bound on test time $N \times T_{structure}$ as a function of supply voltage, where the test contains N clock cycles and $T_{structure}$ is critical path delay. Note that the clock rate is assumed fixed throughout the

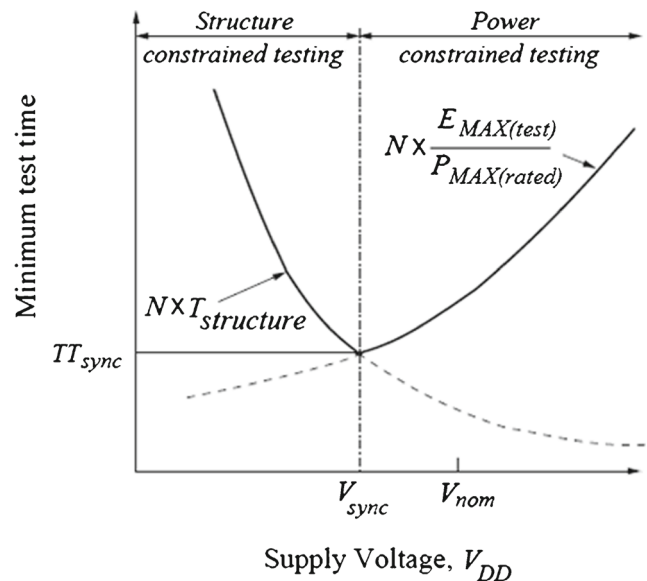


Fig. 3 Test time as a function of V_{DD} [47, 48]. The nominal and the optimal V_{DD} are denoted by V_{nom} and V_{sync} , respectively. The term “sync” means that the clock rate is kept constant throughout the test, referring to synchronous clock testing. In contrast, asynchronous clock testing, where the rate may change during test, has also been considered in the literature [46–51]

entire test. This is called *synchronous clock test*. In contrast the clock rate can be varied during testing and that has been referred to as *asynchronous clock test* [46–51]. Also shown in the figure is a power constrained lower bound (right curve) for which clock period is the ratio of maximum energy $E_{MAX(test)}$ consumed by a test cycle to the power budget $P_{MAX(rated)}$. For a feasible clock period the test time cannot be below either of the two constraint curves. Notice that at the rated voltage V_{nom} test is power constrained. For synchronous clock test as voltage reduces to V_{sync} , clock period drops to give the lowest test time TT_{sync} . Experiments on ISCAS benchmark circuits by those authors show test time reductions of up to 62 % at optimal values of V_{DD} [47].

More recently, Millican et al. [29] have proposed exact methods to provide optimal solutions involving DVFS for session-based and sessionless test scheduling of SoCs. This work assumes that the test frequency and voltage of tests in a session can be different from each other and their MILP formulation selects the best frequency/voltage pair per test to achieve optimization. Therefore, the MILP model presented in this work could be regarded as a special case of [29], wherein an added constraint that the clock rates of all tests in a session should be same, is placed. In reality, the test equipment/infrastructure may impose constraints on the clock rate and/or voltage of tests running concurrently based on feasibility of clock and supply pins. Therefore, while the MILP model in this work may be too restrictive

and the MILP formulation in [29] may come across as less constrained, the real-world scenario perhaps lies between the two methods.

3 Frequency and Voltage Scaling

The general test scheduling problem can be defined as follows: Given an SoC with n cores named C_1, \dots, C_n and a peak power budget P_{max} not to be exceeded during the entire testing of SoC, find a test schedule to:

- a. Test all cores.
- b. Reduce the overall test time.
- c. Conform to the SoC power budget.

P_{max} is chosen so as to account for voltage droops and thermal hotspots that may occur due to peak activity during testing. Core C_i has a test named t_i with associated test time T_{t_i} and test power P_{t_i} , which have been characterized at nominal operating conditions (nominal voltage V_{nom} and clock rate f_{nom}). The test time and power can be influenced by the test clock. A faster clock reduces the test time but increases power consumption. Lowering the clock frequency, on the other hand, reduces the test power but leads to longer test time. Thus, there exists a trade-off between the test time and test power. However, the energy spent during testing remains constant for a fixed voltage. Energy spent per core test, $E_{t_i} = P_{t_i} \times T_{t_i}$. The total energy spent for testing the entire SoC can be expressed as, $E_{total} = \sum_{i=1}^n P_{t_i} \times T_{t_i}$.

In [46–48, 50, 51], the authors have proved that a lower bound on the total test time is given by the ratio of the total energy spent during the test and the power budget,

$$TT_{LB1} = \frac{E_{total}}{P_{max}} = \frac{\sum_{i=1}^n P_{t_i} \times T_{t_i}}{P_{max}} = \sum_{i=1}^n \frac{P_{t_i}}{P_{max}} T_{t_i}, \tag{1}$$

where TT_{LB1} is a lower bound on SoC test time.

Let $\frac{P_{max}}{P_{t_i}} = F_{t_i}$ be a scale factor by which the clock frequency of a test t_i is varied with respect to the nominal value. This scale factor shall be referred to as *frequency factor* throughout the remainder of this paper. Hence, the total test time for the SoC is lowest when all core tests are scheduled sequentially, one at a time, at clock rate $\frac{P_{max}}{P_{t_i}} f_{nom}$.

The test power of a core test, characterized at nominal supply voltage (V_{nom}), is dependent on voltage. For a given frequency, $energy \propto power \propto V^2$. Therefore, the energy per core test and hence the total test energy also varies with the supply voltage. The lower bound on test time given by Eq. 1 is applicable at the nominal supply voltage.

The energy per core test is given by, $E_{t_i} = P_{t_i} (\frac{V_{min}}{V_{nom}})^2 T_{t_i}$, where V_{min} is the lowest value to which the supply voltage can be reduced without disrupting the circuit’s functionality. Hence, another lower bound on the total test time of the SoC is,

$$TT_{LB2} = \frac{\sum_{i=1}^n P_{t_i} (\frac{V_{min}}{V_{nom}})^2 T_{t_i}}{P_{max}}, \tag{2}$$

Note that a constant V_{min} is assumed for all SoC cores. For simplicity, the general case where V_{min} may vary among the cores is not considered here.

These lower bounds assume that power dissipation of a core can be raised to equal the power budget without any physical limitation on the individual core power limit or the clock rate. In reality, the clock rate of a core is often limited by its structure constraint (e.g., critical path delay) and power constraint (rated maximum power).

The structure and power constraints that limit a core’s maximum frequency are also influenced by the supply voltage. The consumed power (P) varies in direct relation with supply voltage (V_{DD}) and clock frequency (f),

$$P \propto V_{DD}^2 \cdot f \tag{3}$$

This implies that by lowering the operating voltage, the clock rate can be increased without exceeding the power constraint of the core. However, the delay of the circuit also varies with the voltage, as given by the alpha power law [32, 35, 36]:

$$delay \propto \frac{V_{DD}}{(V_{DD} - V_{TH})^\alpha}, \tag{4}$$

where α is the velocity saturation index, whose value lies between 1 (for complete velocity saturation) and 2 (for no velocity saturation) [32, 35, 36]. According to Eq. 4, reducing the voltage causes the delay to increase, which in turn slows down the execution speed, resulting in longer test time. Thus, as we reduce V_{DD} , on one hand, the lowered power consumption allows higher clock rates thereby shrinking the total test time, and on the other hand, the increased circuit delay results in a slower clock rate and longer test time. Therefore, it is required to find an optimal V_{DD} that will allow us to balance the two trade-offs and at the same time achieve a test time reduction.

Let f_p and f_s be the upper limits of frequency corresponding to the power and structure (critical path) constraints of a core, respectively. The relationship given in Eq. 3 can now be written as:

$$P_{core} \propto V_{DD}^2 \cdot f_p$$

where P_{core} and f_p are the power rating and the power constrained frequency limit for a core, respectively. Since the

power rating for a core is a constant, the f_p and V_{DD} relation can be expressed as:

$$f_p \propto 1/V_{DD}^2 \tag{5}$$

The f_s and V_{DD} relationship can be expressed in accordance with the alpha power law (4), as:

$$f_s \propto \frac{(V_{DD} - V_{TH})^\alpha}{V_{DD}} \tag{6}$$

Equations 5 and 6 indicate that as V_{DD} is decreased, f_p increases allowing higher clock rates. At the same time, f_s decreases with decreasing V_{DD} , thus restricting the clock rate. The two constraint are independent of each other, i.e., the power constrained f_p only depends on the rated power of the core with no regard for the critical path of the core and similarly, the critical path of the core dictates the structure constraint limit, ignoring the rated power limit of the core. The maximum allowable clock rate for a core is, therefore, the minimum of the two frequency limits.

The lower bound for the SoC test time, defined in Eq. 2, predicts that the test time continually reduces as V_{DD} is lowered. However, from Fig. 3, we know that beyond an optimal V_{DD} point, the test time increases with decreasing V_{DD} . Therefore, Eq. 2 is revised to include the optimal voltage, V_{sync} , instead of V_{min} .

$$TTLB2 = \frac{\sum_{i=1}^n P_{t_i} \left(\frac{V_{sync}}{V_{nom}}\right)^2 T_{t_i}}{P_{max}}, \tag{7}$$

It can be noted that Eq. 2 would be the same as Eq. 7, when $V_{sync} = V_{min}$.

Let us assume that $f_s = \beta \cdot f_p$ at V_{nom} , i.e., structure constraint (f_s) is β times the power constraint for test frequency, where $\beta \geq 1$. In the worst case, $\beta = 1$ implies that the structure and power constraint for the frequency are same and there can be no further optimization through frequency scaling. The higher the value of β , better is the optimization achieved. As V_{DD} is lowered, both f_s and f_p vary accordingly. At V_{sync} , $f_s(V_{sync}) = f_p(V_{sync})$, i.e., $f_s \cdot \frac{V_{nom}}{V_{sync}} \cdot \left(\frac{V_{sync} - V_{th}}{V_{nom} - V_{th}}\right)^\alpha = f_p \cdot \left(\frac{V_{nom}}{V_{sync}}\right)^2$. Since $f_s/f_p = \beta$,

$$\beta = \frac{V_{nom}}{V_{sync}} \cdot \left(\frac{V_{nom} - V_{th}}{V_{sync} - V_{th}}\right)^\alpha \tag{8}$$

The value of α for a short-channel MOSFETs is 1.3 [35]. For the sake of simplicity, let us assume $\alpha \approx 1$. Now, Eq. 8 can be written as $\beta(V_{sync})^2 - \beta V_{th} V_{sync} - V_{nom}(V_{nom} - V_{th}) = 0$, which is of the form $ax^2 + bx + c = 0$. Solving for V_{sync} ,

$$V_{sync} = \frac{\beta V_{th} \pm \sqrt{(\beta V_{th})^2 + 4\beta V_{nom}(V_{nom} - V_{th})}}{2\beta} \tag{9}$$

4 Optimization of Session-Based Test Schedules

In this section, we will discuss optimization of session-based test schedules through frequency and voltage scaling. As depicted in Fig. 2a, in a session-based test schedule, tests are distributed into sessions. Each session ends when all tests in that session are completed.

4.1 Mixed-Integer Linear Program (MILP) Based Optimization

A mixed-ILP model for optimizing V_{DD} and clock rate per test session is formulated in this section. Let us assume an SoC with n cores. Let there be n core tests, t_1, \dots, t_n , where test t_i has an application time T_{t_i} and power consumption P_{t_i} . These tests are distributed among k sessions, S_1, \dots, S_k , such that each session S_j contains one or more tests to be run concurrently. Sessions, however, are applied sequentially.

A test session may contain any subset to the n tests. Thus, the total number of possible sessions for n tests is $k = \sum_{i=1}^n \binom{n}{i} = 2^n - 1$. The first term, $\binom{n}{1} = n$, corresponds to purely sequential test scheduling where n sessions will apply a single test at a time. The last term, $\binom{n}{n} = 1$, is for all tests included in a single session.

The test time of session S_j , $j = 1 \dots k$, is given by $T_{S_j} = \max\{T_{t_i}\}$, $\forall t_i \in S_j$, and the power dissipated during session S_j is given by $P_{S_j} = \sum(P_{t_i})$, $\forall t_i \in S_j$.

The general test scheduling problem consists of selecting a sufficient number of sessions such that each core test is included in a session, the total test time of all selected sessions is minimized, and the maximum power during any selected session does not exceed a given peak power limit. This is expressed as an optimization problem:

Objective:

$$\text{Minimize } \sum_{j=1}^k T_{S_j} \cdot x_j$$

where $x_j = \begin{cases} 1, & \text{if } S_j \text{ is scheduled} \\ 0, & \text{otherwise} \end{cases}$

Subject to:

1) Power constraint: $P_{S_j} \cdot x_j \leq P_{max}$, $j = 1 \dots k$, where P_{max} is the power budget for SoC.

2) All n cores are tested: each test, t_m , $m = 1 \dots n$, is executed at least once, i.e.,

$$\sum_{j, t_m \in S_j} x_j \geq 1, m = 1 \dots n$$

We include test clock frequency and supply voltage in the MILP model to allow scaling of both time and power as we

optimize the test time. It must be noted that the maximum clock frequency of a session is decided by the maximum clock frequency of the slowest core in that session, i.e., $f(S_j) \leq \min\{f_{max}(t_i) | \forall t_i \in S_j\}$, where $f(S_j)$ is the clock rate of session S_j and $f_{max}(t_i)$ is the maximum clock frequency of test t_i . Since all cores of the SoC are testable at some nominal clock frequency, f_{nom} , it is reasonable to assume that f_{nom} is the clock rate of the slowest core in the SoC (say f_0). Then, frequency factor of a session is $F_j = \frac{f(S_j)}{f_0}$. Note that the test session containing the slowest core of the SoC will possess a unity frequency factor. The maximum frequency factor is given by:

$$\max\{F_j\} = \min\left[\frac{\min\{f_{max}(t_i) | \forall t_i \in S_j\}}{f_0}, \frac{P_{max}}{P_{S_j}}\right] \quad (10)$$

As stated earlier, the clock frequency of each core is constrained by structure and power limits, and the highest clock rate for a core is actually the minimum of these constraints. Therefore, the clock frequency of a session, which is the same as the slowest core in that session, is given by $f(S_j) \leq \min\{f_p(t_i), f_s(t_i) | \forall t_i \in S_j\}$, where $f_p(t_i)$ and $f_s(t_i)$ are the power and structure constraint limits of core test t_i . Since the frequency factor of a session, $F_j = \frac{f(S_j)}{f_0}$, its maximum value is given by,

$$\max\{F_j\} = \min\left[\frac{\min\{f_p(t_i), f_s(t_i) | \forall t_i \in S_j\}}{f_0}, \frac{P_{max}}{P_{S_j}}\right] \quad (11)$$

The voltage range is divided into multiple steps of voltages and for each step, the test power and frequency limits (structure and power constraint) of each test session is pre-computed. Let P_{ij} be the test power of j^{th} session at i^{th} voltage. Similarly, let $F_{s_{ij}}$ and $F_{p_{ij}}$ be the frequency limit imposed by the structure and power constraints, respectively, for j^{th} session at i^{th} voltage. For each session, the V_{DD} is chosen by a binary variable whereas the clock rate of the session is a real-valued variable. T_j and F_j are the test time and frequency factor of j th test session. We introduce x_{ij} , a [0,1] binary variable that when set to 1 selects i th V_{DD} step for j th test session. The test schedule optimization can then be described as follows:

Objective:

$$\text{Minimize } \sum_{i,j} (T_j / F_j) \cdot x_{ij}$$

where $x_{ij} = \begin{cases} 1, & \text{if } j^{th} \text{ session is scheduled at } i^{th} \text{ voltage} \\ 0, & \text{otherwise} \end{cases}$

Subject to:

- 1) $F_j \cdot \sum_i (P_{ij} \cdot x_{ij}) \leq P_{max}$, i.e., power constraint is satisfied whenever $x_{ij} = 1$ or i th voltage step is chosen for j th session, $j = 1 \dots k$.
- 2) $\sum_i x_{ij} = 1$, i.e., exactly one voltage is chosen for j th session, $j = 1 \dots k$.

- 3) $F_j \cdot x_{ij} \leq F_{s_{ij}}$ and $F_j \cdot x_{ij} \leq F_{p_{ij}}, \forall \{i, j\}$
- 4) Core test $t_m, m = 1 \dots n$, is included in at least one selected session:

$$\sum_{\forall j, t_m \in S_j} \sum_i x_{ij} \geq 1$$

The first constraint of this ILP ensures that the power consumption of any session in the test schedule does not exceed the power budget. The second constraint specifies that each session has exactly one voltage value. The clock frequency is bounded by the structure constraint ($F_{s_{ij}}$) and the power constraint ($F_{p_{ij}}$) in the ILP's third constraint. The fourth constraint ensures test completeness requiring that each core test is included in a selected test session.

F_j is a real variable and x_{ij} is a binary variable. Since the MILP formulation contains objective and constraints that involve products of these variables, the formulation is non-linear. We linearize the formulation by substituting $1/F_j = u_j$ and $u_j \cdot x_{ij} = q_j, \forall \{i, j\}$. The new variables u_j and q_j linearize the MILP formulation, however, their substitution necessitates additional constraints. Thus, for j th test session ($j = 1 \dots k$): 1) $q_j \geq u_j - M(1 - x_{ij}), \forall i$, where M is a large constant, $M \gg u_j$, and 2) $q_j \geq 0$. The new ILP formulation is as follows:

Objective:

$$\text{Minimize } \sum_j T_j \cdot q_j$$

Subject to:

- 1) $\sum_i (P_{ij} \cdot x_{ij}) \leq P_{max} \cdot u_j, j = 1 \dots k$ (Power constraints)
- 2) $\sum_i x_{ij} = 1, j = 1 \dots k$ (Voltage assigned to each session)
- 3) $x_{ij} \leq F_{s_{ij}} \cdot u_j$ and $x_{ij} \leq F_{p_{ij}} \cdot u_j$
- 4) Core test $t_m, m = 1 \dots n$, is included in at least one selected session:

$$\sum_{\forall j, t_m \in S_j} \sum_i x_{ij} \geq 1$$

- 5) $q_j \geq u_j - M(1 - x_{ij}), \forall i, j$
- 6) $q_j \geq 0$

Note that the voltage step size determines the precision of the solution. However, reducing the step size to

enhance the precision would increase the number of variables in the formulation and hence render the ILP solution less intractable.

4.2 Heuristic Based Optimization

Integer Linear Programs are NP-hard in general and are computationally expensive for large SoCs. The CPU time required to obtain an optimal solution increases exponentially as the number of cores and the complexity of the SoC increases. The proposed MILP method also shares the same issues with scalability in terms of the problem size. Hence, we present a simulated annealing (SA) based optimization technique that is much faster than ILP for larger SoCs and also capable of producing results similar to that of the ILP method. Heuristic algorithms, often employing greedy approaches, perform much better in terms of CPU time as compared to exact methods such as ILP. While a heuristic method does not guarantee an optimal solution, a good algorithm can produce near-optimal values consistently. Simulated annealing is a directed search algorithm inspired from the annealing process in metallurgy, first proposed by Kirkpatrick et al. in [20] and has been used in the past for SoC test scheduling [24, 56]. The algorithm accepts a non-improving solution with a finite probability so as to avoid getting stuck at a local optimum.

An overview of our SA based optimization algorithm is shown in Fig. 4. Functions used in this algorithm are defined below:

Simulated Annealing Heuristic

T = temperature

K = cooling parameter

T_f = final temperature

X_B = best solution obtained so far

X_C = current solution

generate initial solution, X_0 (test schedule and corresponding test time)

$X_B = X_0, X_C = X_0$

while $T \geq T_f$ **do**

 perform SA move operation (swapping of tests) on X_C
 scale clock rate and voltage to optimize the new test schedule
 compute test time of the optimized test schedule, X_{new}
 $Diff = X_{new} - X_C$

if $Diff \leq 0$ or $\exp(-\frac{Diff}{KT}) \geq \text{random}(0, 1)$ **then**

$X_B = X_{new}, X_C = X_{new}$

else

 discard X_{new} and retain X_C

end if

$T = K \times T$

end while

Fig. 4 Overview of heuristic simulated annealing (SA) algorithm

4.2.1 Initial Solution

The initial solution is developed by inserting a randomly selected test into a session until the session's power consumption (P_{ses}) is close to the peak power budget (P_{max}). This step is repeated until all the tests are grouped into sessions such that no two sessions contain the same test.

The test schedule, thus generated, serves as the starting point for the simulated annealing heuristic. Frequency and voltage scaling (described in *cost calculation*) are also applied to optimize the test time obtained through Fig. 5.

4.2.2 SA Move Operator

The move operator in our simulated annealing algorithm is a swapping of tests between two sessions. Among the many test sessions of the test schedule, two sessions s_1 and s_2 are selected at random, such that $s_1 \neq s_2$. A randomly chosen test in each session is swapped with each other, thus forming a new test schedule. The cost of the resultant solution, which is the test time of the new test schedule, is computed. The new test schedule is accepted if the new solution is better than the best solution obtained so far or if their difference (d) is such that $\exp(-\frac{d}{KT}) \geq \text{random}(0, 1)$, where K is the cooling parameter and T is the annealing temperature (described in *annealing schedule*), else the swap is discarded.

4.2.3 Cost Calculation

The cost in this optimization problem refers to the test time of the test schedule. The overall test time for the session-based test schedule is the sum of the test time of the longest

Init Sol

list1 = list of core tests to be scheduled {initially contains all core tests}

list2 = list of core tests currently executed {initially empty}

$t_{sch} = 0$ {overall test time of the test schedule}

while list1 is **not** empty **do**

 list2 = empty list

while $P_{ses} < P_{max}$ **do**

 insert random test i into list2

 delete test i from list1

$P_{ses} = \sum P_i, \forall i \in \text{list2}$

end while

if $P_{ses} > P_{max}$ **then**

 remove recently added test from list2

end if

$t_{sch} = t_{sch} + \max(t_i, \forall i \in \text{list2})$

end while

Fig. 5 Generating initial solution for simulated annealing (SA) algorithm

test in the session. The test clock frequency and the supply voltage are scaled to optimize the test time. The scaling factor for the clock, referred to as the frequency factor (F), is updated after addition of every test during the initial solution phase and after every swap, in the SA move operation phase. The frequency factor is limited by both the peak power budget and the clock rate constraints of individual core of the SoC, as given in Eq. 10.

Voltage scaling is done for each session as given below:

- reduce V_{DD} by one step.
- Calculate the power and structure constraint limits of the tests using Eqs. 5 and 6, respectively.
- Update the frequency factor using Eq. 11.
- Repeat the steps if the resulting session test time is lower than before, else quit the voltage scaling procedure.

4.2.4 Annealing Schedule

Annealing schedule refers to the temperature (T), the cooling parameter (K) and their effects on the optimization procedure. True to the metallurgical annealing process, the initial value of the temperature is high. Each iteration of the heuristic, which produces a new solution, corresponds to a value of the temperature. After each iteration, the temperature is reduced according to $T_{new} = K \times T$, where $K \leq 1$. Hence, the number of iterations is dependent on both, the temperature and the cooling parameter. The stopping criteria for the procedure would be the temperature dropping below a certain specified limit.

4.3 Results

4.3.1 Experimental Setup

The exact and the heuristic optimization methods were experimented on several ITC’02 benchmarks [15] and the ASIC Z SoCs. The AISC Z was introduced by Zorian [55] and consists of RAM, ROM and other blocks. These blocks, along with their test time and power are shown in Fig. 6. The peak power budget for the ASIC Z is given as 900 mW.

The test time and test power data for the ITC’02 benchmarks have been provided by Millican and Saluja [27, 28]. The peak power budget for these SoCs were assigned based on the test power information of individual cores in the SoCs. To account for power and structure constrained limits on frequencies of individual cores, maximum clock rates are allocated to each core. The values for the power constrained limit (f_p) are computed based on the test length and test power of the blocks/cores. The core with the highest test power and longest time is regarded as the slowest and the rest of the cores in the SoC are normalized with respect to

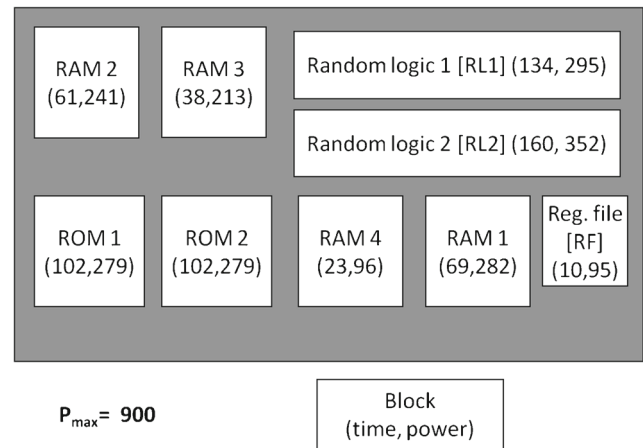


Fig. 6 Components of ASIC Z [55] and their test time (in arbitrary units) and test power (in mW)

the slowest core. Hence, a test with low test power value can be clocked at a faster rate. For assigning the structure constraint limit (f_s), the fact that the test power can be as high as four times the functional power is taken into account, i.e., $P_{func} \leq P_{test} \leq 4P_{func}$. $P_{test} \propto f_p$ and since the structure constraint limit decides the functional clock rate, $P_{func} \propto f_s$. Hence, the structure constraint limit (f_s) for each core is set to $\beta \times f_p$, where β is a uniform random number in the range [1, 4]. For illustration, the complete data of ASIC Z, including the frequency limits, is given in Table 1. This test data set for ASIC Z is specified at a nominal supply voltage for which we assume that the test power and power budget are as provided by Zorian [55]. The frequency constraints for each block were derived by the steps described previously. Similarly, the test data for the remaining benchmarks is given in [40].

Further, the range of operating voltage, in this work, is assumed as 1.0 volt (nominal) to 0.6 volt (minimum). The other parameters for the alpha-power law [32, 35, 36] are

Table 1 Test Data set for ASIC Z [55] at nominal supply voltage (1.0 volt)

Block	Test time in arbitrary units	Test power in mW	Frequency constraints	
			f_p^+	f_s^{++}
RAM1	69	282	1.75	6.65
RAM2	61	241	2	7.55
RAM3	38	213	3	5.6
RAM4	23	96	5	8.8
ROM1	102	279	1.5	4.6
ROM2	102	279	1.5	3.83
RL1*	134	295	1.2	2.74
RL2*	160	352	1	2
RF**	95	10	8	17.6

⁺power constraint ⁺⁺structure constraint *Random Logic **Register File

$V_{th} = 0.5$ volt and $\alpha = 1.0$. These values are in tune with the 45-nm technology [54]. Keeping this in mind, let us revisit the lower bound on SoC test time. As mentioned earlier, the structure constraint on an SoC core clock rate limits the scaling of V_{DD} . Assuming the least restrictive condition on the structure constraint, we have $f_s = 4f_p$. Substituting $V_{nom} = 1.0$ volt, $V_{th} = 0.5$ volt and $\beta = 4$ in Eq. 9 yields $V_{sync} \approx 0.69$ volt. This value of V_{sync} can be used in Eq. 7 to derive the lower bound on the test time of the SoC benchmarks considered in this work.

The experiments were preformed on a Dell workstation with a 3.4GHz Intel Pentium processor and 2GB memory. The MILP models were solved using IBM CPLEX Optimization Solver (student edition). The simulated annealing (SA) based heuristic algorithm was developed using the Python scripting language [10].

4.3.2 MILP Results

In order to evaluate the optimization results, three cases are considered. Case 1 is the nominal case where V_{DD} and the test clock are fixed at a nominal value. In Case 2, V_{DD} is still fixed at a nominal value but the clock frequency is optimized on a per test session basis [41]. In Case 3, both V_{DD} and the clock are optimized [42].

Results for ASIC Z are shown in Tables 2 and 3. Previously published optimal test times for the ASIC Z are 392 by Zorian [55], 330 by Chou et al. [4, 5] and 300 by Larsson and Peng [24]. In Table 2, for the nominal case (Case 1) the test schedule and test time of 300 units are

Table 2 Optimized Session-Based Test Schedules for ASIC Z [55] for nominal and custom clock rates (Cases 1 and 2)

Case 1: Nominal clock			Case2: Customized session clocks			
Session	Block	Test time	Session	Block	Freq. factor	Test time
1	RL1 RL2 RAM2	160	1	RAM1 ROM2	1.5	68
2	RAM1 ROM1 ROM2	102	2	RAM2 RAM3	1.98	30.77
3	RAM1 RAM4 RF	38	3	RAM4 RF	4.71	4.88
			4	ROM1 RL1 RL2	0.97	164.624
Total test time =		300	Total test time =		268.274	

The supply voltage is at nominal value for both cases

Table 3 Test times (in arbitrary units) for ASIC Z [55] with customized V_{DD} and clock rates (Case 3)

Session	Block	Freq. factor	V_{DD}	Test time
1	RF	12.5	0.8V	0.8
2	RAM1,RAM2, RAM3,RAM4	2.56	0.65V	26.95
3	ROM1,ROM2, RL1,RL2	1.33	0.75V	120.5
Total Test time =				148.25

similar to those obtained by Larsson and Peng [24]. Customizing the test clock for each session (Case 2) reduced the test time by 10.5 %. The frequency factor in the table indicates the speed-up of the clock done to reduce the test time. For examole, a frequency factor of 1.5 implies that the test clock frequency of that session was increased to 1.5 times from the nominal value.

A lower bound on the overall test time for ASIC Z at nominal V_{DD} , as calculated using Eq. 1, is 220.2 units. This is 26.6 % lower than the test time at nominal clock rate and voltage (Case 1) and only 17.9 % below the test time for Case 2 (customized test clock frequency). One can observe that by customizing the clock rate, the test scheduling result moves closer to the lower bound but is constrained by the maximum clock rate of individual cores.

Table 3 shows that customizing both V_{DD} and test clock (Case 3) lowers the test time by as much as 50 % over that of Case 1. It can also be noted from the table that the sessions in the schedule not only have different clock rates but also have different V_{DD} , which is the optimum for the session.

Similarly, the optimized test times for several benchmarks for various cases are given in Table 4. The percent reductions specified in the last two columns refer to the reduction in test time achieved by Case 3 (session-wise V_{DD} and clock scaling) with respect to the other two optimization cases. For instance, as observed before in case of ASIC Z, the test time for Case 3 is about 50 % lower than that of Case 1 (fixed V_{DD} and clock) and 45 % lower than Case 2 (only frequency scaling). Notably, the customiziation of both voltage and frequency can cut the test time in half. Considering the complexity of the algorithms, it can be noted that the optimization through frequency and voltage scaling consumes most CPU time and also the run time grows rapidly with the SoC size.

4.3.3 Heuristic Algorithm Results

A comparison of optimized test times obtained from the MILP and the SA based heuristic test scheduling algorithm is provided in Table 5. Since the heuristic is dependent on the initial solution, the algorithm is repeated for hundred starting points and the best solution among them is selected.

Table 4 Session-based test times (in arbitrary units) for benchmark SoCs, obtained by the MILP method for the three optimization cases considered

Benchmark	No. of cores	P_{max}	Case 1		Case 2		Case 3		% Reduction over	
			Test	CPU	Test	CPU	Test	CPU	Case 1	Case 2
			time	time	time	time	time	time		
a586710	7	800mW	14271856	0.35 sec	13011130.61	0.92 sec	6799115.12	12.03 sec	52.36	47.74
h953	8	800mw	122636	0.6 sec	121715.34	0.95 sec	79318.76	48.71 sec	35.32	34.84
ASIC Z	9	900mW	300	0.62 sec	268.274	1.63 sec	148.25	501.18 sec	50.58	44.74
d695	10	400mW	15188	0.65 sec	12733.2	2.49 sec	7173	3649.52 sec	52.77	43.67

Case 1: V_{DD} and test clock at nominal values; Case 2: V_{DD} at nominal value, clock scaled per test session; Case 3: V_{DD} and clock scaled per test session

The CPU time of the algorithm is averaged over the hundred simulations. Even though the solution is repeated for 100 initial points to account for the algorithm’s dependence on the initial solution, the actual run time from the initial solution to the end result would still be equivalent to a ‘single run’ and this run time is obtained by dividing the total time by the number of runs (100, in this case). As seen from the table, the difference between the heuristic solution and the exact MILP solution is marginal.

The CPU time for the heuristic does not vary much with respect to the SoC size. To emphasize this point, the heuristic methods was employed to solve the test scheduling problem for larger ITC benchmarks, for which the MILP solver would struggle hard to provide a solution. In order to further evaluate the performance of the heuristic, SoCs with 100, 200 and 500 cores (referred to from now on as R100, R200 and R500, respectively) were created. The test time and test power data for the R100 SoC was generated using a uniform random number generator, in the range (10, 100) and (50, 500), respectively. The R200 and the R500 are multiple copies of the R100 SoC. The peak power budget for these SoCs was set to 900mW. Table 6 summarizes the optimized test times obtained through the heuristic method for these SoCs. These results show that the heuristic method also achieves a test time reduction of up to 60 %. The heuristic algorithm is able to provide an optimized test schedule for the 500 core SoC in just over 6 seconds.

Table 5 Session-based test times (in arbitrary units) for MILP and heuristic test scheduling methods, with customized V_{DD} and clock rates

Benchmark	SA based heuristic method		MILP method	
	Test time	CPU time	Test time	CPU time
a586710	6799118.34	0.12 sec	6799115.12	12.03 sec
h953	79319.12	0.09 sec	79318.76	48.17 sec
ASIC Z	150.26	0.11 sec	148.25	501.18 sec
d695	7177.53	0.17 sec	7173	3649.52 sec

5 Optimization of Non-Preemptive Sessionless Test Schedules

As discussed earlier, in sessionless testing, a test is scheduled as soon as some other test ends, without waiting for a session boundary. As a result, sessionless test scheduling often has test time that is at least equal to, but often better than, that of session-based scheduling. In this section, a test optimization algorithm for non-preemptive sessionless test scheduling is proposed. This is a heuristic approach very similar to that of the session-based test scheduling, in that, it also based on a simulated annealing algorithm. Since only a single clock and a fixed V_{DD} are assumed, tests that are scheduled together have the same clock rate and V_{DD} . As a result, now the frequency factor corresponds to a clock scaling factor for set of tests scheduled concurrently. The lower bound on test time provided by Eq. 2 is valid for sessionless test schedules as well. An MILP approach towards optimization of sessionless test schedules is provided in [29].

5.1 Heuristic Approach to Optimization

The initial solution and the SA move operator of this method remains the same as that of the heuristic for session-based testing. However, after the swap move, session boundaries in the new test schedule are erased and consecutive test sessions are merged together to form a sessionless test schedule. The cost of the resultant solution is determined; this solution is accepted if the cost is lower than the best solution obtained so far, or if their difference (d) is such that $\exp(-\frac{d}{KT}) \geq random(0, 1)$.

To compute the test time of the sessionless test schedule, firstly, consecutive test sessions in the test schedule resulting from the swap move are merged together by scheduling tests from the next session as soon as a test in the current session completes, as illustrated in Fig. 7. This process of ‘merging’ sessions is repeated until all tests are scheduled.

Table 6 Session-based test times (in arbitrary units) for large SoC benchmarks obtained from the heuristic test scheduling approach

Benchmark	No. of cores	P_{max}	Case 1		Case 2		Case 3		% Reduction over	
			Test	CPU	Test	CPU	Test	CPU	Case 1	Case 2
			time	time	time	time	time	time		
g1023	14	400mW	21245	0.06 sec	19888.7	0.06 sec	12193.05	0.16 sec	42.6	38.7
p34392	19	400mW	952199	0.055 sec	758199.76	0.06 sec	369692.1	0.19 sec	61.17	51.24
t512505	31	400mW	5589002	0.062 sec	5414047.16	0.063 sec	3038172.5	0.19 sec	45.64	43.88
p93791	32	400mW	178568	0.06 sec	160618.71	0.058 sec	90391.8	0.18 sec	49.38	43.72
R100	100	900mW	1347	0.09 sec	1213.56	0.26 sec	730.40	1.36 sec	45.77	39.81
R200	200	900mW	2837	0.13 sec	2502.3	0.5 sec	1536.35	2.6 sec	45.84	38.6
R500	500	900mW	7706	0.28 sec	6653.01	1.21 sec	4212.27	6.28 sec	45.34	36.68

Case 1: V_{DD} and test clock at nominal values; Case 2: V_{DD} at nominal value, clock scaled per test session; Case 3: V_{DD} and clock scaled per test session

The function *Merge* is described in Fig. 8. The test session that will be merged with its predecessor is passed as an argument to the *Merge* function. The tests in this test session are added to the sessionless test schedule as and when the tests in the previous test session complete. The tests currently scheduled are run to completion and new tests are added from the next session as the tests that are currently scheduled, end.

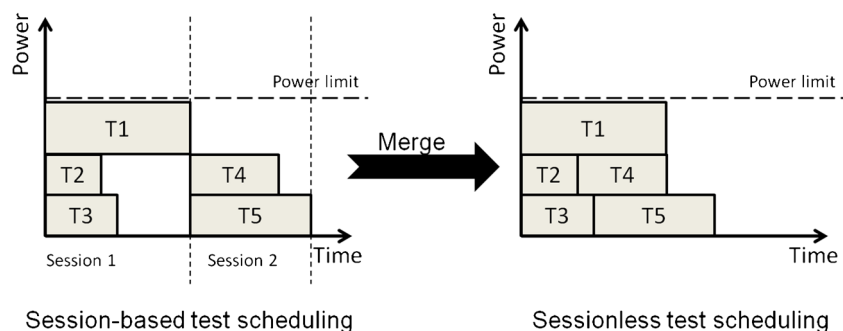
The test clock frequency and the supply voltage are scaled for every set of concurrently scheduled tests to optimize the test time. However, the clock rate and the voltage for concurrently scheduled tests remain constant until the completion of a test; the frequency and voltage scaling is performed after the completion of every test, unlike the session-based test optimization method where the frequency and voltage are scaled after every test session. The annealing schedule remains the same as that for session-based test optimization algorithm.

5.2 Optimization Results

The experimental setup including the benchmarks for sessionless test optimization remains the same as that of

the session-based test optimization. For comparison with voltage and frequency scaled schedules, an algorithm to generate reference sessionless schedules with voltage and frequency fixed at nominal values is provided. As discussed in Section 2, the test scheduling process is modeled as a bin packing problem [13]. An individual core test is treated as a block with test power as height and test time as width. A best-fit decreasing (BFD) heuristic then solves the bin packing problem. The tests are sorted in decreasing order of their power consumption and stacked together in such a way that at any given time in the test schedule the total power does not exceed a specified P_{max} . The algorithm is provided in [43, 44]. As scaling voltage and frequency alters the test time and power of a core test, test schedules with scaled clock and supply cannot be modeled for bin packing. Hence, a simulated annealing (SA) based heuristic algorithm is adopted. The heuristic has some randomization and can be dependent on the initial solution. Therefore, the algorithm is repeated for hundred starting points and the best solution among them is selected. The CPU time of the algorithm is averaged over the hundred solutions. In each iteration, the starting point is a test scheduled with test sessions which are then merged to yield a sessionless test schedule. As the

Fig. 7 Merging test sessions to convert a session based test schedule into a sessionless test schedule



```

Merge(session)
slsch = sessionless test schedule {initially empty}
if slsch is empty then
    add all tests in the session to slsch
else
    while session not empty do
        if test in slsch completes then
            select a test from session and add to slsch
             $P = \sum P_i, \forall i \in slsch$ 
            if  $P > P_{max}$  then
                remove the added test from slsch
            end if
        end if
        perform frequency and voltage scaling
    end while
end if
    
```

Fig. 8 The Merge function erases the session boundaries in a session based test schedule and combines the tests together to form a sessionless test schedule

temperature reduces the algorithm moves from one feasible solution to another, with every new solution being better than the previous one. Occasionally, with a finite probability, a ‘worse-than-previous-best’ solution is accepted to avoid saturation at some local optima. This probability is much lower at lower temperatures.

As in the case of session-based test scheduling, the heuristic method of optimization is applied to several SoCs including R100, R200 and R500 in order to further evaluate the performance of the algorithm. Table 7 summarizes the optimized test times. Here we notice that by scaling the voltage and frequency dynamically the test time could be shortened by 45 to 60 %.

Table 7 Test times (in arbitrary units) for non-preemptive sessionless test scheduling with voltage and frequency scaling

Benchmarks	Test time	Test time	%
	without scaling ¹	with scaling	
a586710	14090716	5797578.6	58.85
h953	122636	60805.62	50.42
ASIC Z	262	137.85	47.38
d695	13301	5210.05	60.83
g1023	18084	8898.82	50.79
p34392	701684	279570.6	60.15
t512505	5344747	2940986.25	44.97
p93791	139008	68638.25	50.62
R100	1208	625.83	48.2
R200	2366	1337.4	43.47
R500	5807	3497.6	39.8

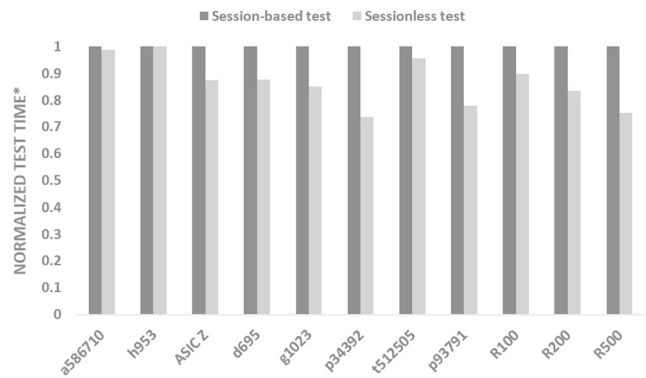
¹At fixed voltage and frequency, obtained from Best-Fit Decreasing algorithm

²Percentage with respect to the reference test time without scaling (column 2)

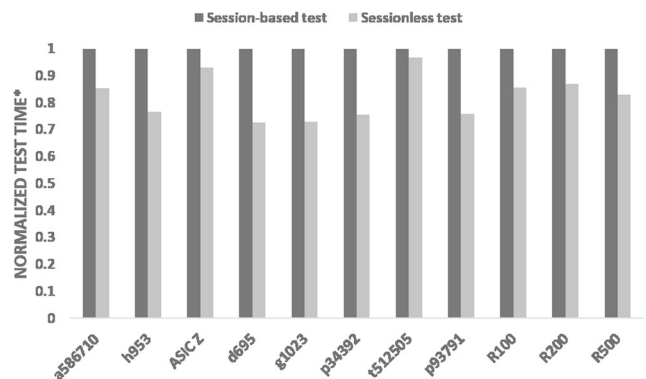
5.2.1 Comparison with Session-Based Testing

The test times for session-based and sessionless test schedules are compared for the case of fixed voltage and test clock frequency (Fig. 9a) and the case of scaled voltage and test clock frequency (Fig. 9b). It can be inferred from these figures that the difference in test time between the two test scheduling strategies is less than 30 % for the SoC benchmarks considered. Session-based testing introduces some idle time gaps in the test schedule by waiting for the longest test in the session to complete and hence leads to longer test times than sessionless testing. In comparison, sessionless testing does not allow idle time gaps since the test scheduling occurs immediately after completion of older tests.

While sessionless test scheduling provides the lowest test time compared to sequential or session-based test scheduling (the same as session-based, in the worst case), it requires a complex control infrastructure. This is because concurrently executing tests require independent control resources, such as scan-enable signal, shift and capture clocks, wrapper instruction register (WIR), etc. [26, 31]. In cases where the



(a) Clock rate and V_{DD} fixed at nominal values.



(b) Clock rate and V_{DD} scaled.

Fig. 9 Comparing test time results for session-based (left bar) and sessionless (right bar) test schedule optimization for fixed (a) and scaled (b) voltage and clock rate. The optimal test time is normalized with respect to that of the session-based test schedule

test time of a sessionless test schedule is only marginally lower than that of a session-based schedule (e.g., t512505 in Fig. 9a or b), the control overhead costs may not be offset by the test cost reduction achieved by the sessionless testing scheme.

6 Discussion

6.1 Lower Bound

Table 8 compares the SoC test times for the session-based and non-preemptive sessionless test schedules to the lower bound given by Eq. 7. The value for V_{opt} is substituted as 0.7V. Note that $V_{opt} = V_{sync}$, as shown in Fig. 3, is the voltage for the minimum test time for a core. The optimal test time of sessionless testing is evidently closer to the lower bound than is the test time of session-based testing. The difference between the lower bound and the optimal test time can be attributed to the fact that while calculating the optimal V_{DD} point, a least restrictive condition was assumed for the structure constraint. This, however, is not the case for all cores of the SoC and therefore, V_{opt} for such cores will be higher than the value 0.69V calculated in Section 4.3.

It can be noted from Table 8 that for benchmarks h953 and t512505 the difference between the lower bound and the optimal test time is much larger than the rest of the benchmarks. This follows from a theorem [40] stating that the lower bound on test time for a core is achieved when the test clock is set by scaling the nominal clock rate by a ratio P_{max}/P_{test} . For some cores in these two benchmarks, this ratio is as large as 2000. The individual clock constraints, however, are not as high as the ratio, P_{max}/P_{test} . As a result,

there is a marked difference between the lower bound and the optimal test times for these two benchmarks.

6.2 Frequency and Voltage Scaling

In this work, we vary the test clock frequency within the bounds of structure constraint (such as critical path) and power constraint (rated power limit) of the cores. For timing tests where frequency is critical during the capture cycle, varying the clock frequency may be limited to the shift cycles used to shift test data in/out. Shifting of test data involves multiple clock cycles to move data through shift register as opposed to the capture cycle which is a single, often at-speed, clock cycle. Hence, shifting data faster by varying clock frequency can lead to a considerable reduction in the overall test time of the SoC. The constraints on the shift clock rate would be the critical delay of the scan path (structure constraint) and the shift power limit (power constraint) of the scan tested core. Our proposed method of finding the optimal V_{DD} may also be confined to the shift cycles during the SoC testing, such that voltage can be reduced to regulate the shift power such that the shift cycle clock frequency can be increased (as done by PMScan [7]).

The voltage and frequency scaling schemes presented in this work are intended only for the reduction of test time and should not interfere with the fault coverage of the test. It has been shown that while V_{DD} does not affect stuck-open defects, it may affect the behavior of resistive opens [8, 25]. Chang and McCluskey conclude from their experiments that low voltage testing captures defects that can cause early-life and intermittent failures and that these defects are undetected at nominal voltage [2, 3]. However, Engelke et al. showed that testing at very low voltages may contribute

Table 8 Test times (in arbitrary units) for non-preemptive sessionless test scheduling with voltage and frequency scaling

Benchmarks	Lower bound	Sessionless testing		Session-based testing	
	(LB) ¹	Test time	% Difference ²	Test time	% Difference ²
a586710	5464175.96	5797578.6	5.75	6799115.12	19.63
h953	19763.41	60805.62	67.5	79318.76	75.08
ASIC Z	104.83	137.85	23.95	148.25	30.23
d695	4376.98	5210.05	15.99	7173	39.02
g1023	5427.69	8898.82	39.0	12193.05	55.48
p34392	245784.34	279570.6	12.08	369692.1	33.5
t512505	755712.11	2940986.25	74.30	3038172.5	75.12
p93791	57836.72	68638.25	15.74	90391.77	36.01
R100	539.07	625.83	13.86	730.40	26.2
R200	1078.14	1337.4	19.38	1536.34	29.82
R500	2695.35	3497.6	22.93	4212.27	36.01

¹Lower Bound calculated at $V_{opt} = 0.69V$, by Eq. 7

²Percent difference is computed with respect to the lower bound

to coverage loss [8]. These observations do not invalidate the proposed method but only restricts the available voltage range for the voltage and frequency scaling scheme. Thus, in the method of this paper the user may select the range of voltages based on the defect coverage requirement. Most of the previously reported work is on *very low voltage testing* [2, 3, 8, 11, 25].

7 Conclusion

System-on-chip (SoC) devices continue to grow in size and complexity due to continued advancement in IC technology. The test data required to test such large, complex SoCs is voluminous, leading to long test times. Also, since power consumption during test mode is much higher than during functional mode, the SoC test power may cause local hotspots and supply voltage droops. In this work, a power-aware optimization method of SoC test schedules through voltage and frequency scaling is discussed. The test clock frequency can be scaled by a factor (referred to as frequency factor in this work) to speed-up the SoC test. This factor, however, is limited by SoC power budget and also by the maximum clock rates of individual cores of the SoC. Restrictions on the core-level clock rate may be imposed by either a power constraint (maximum power dissipation limit of the core) or a structure constraint (critical path delay). Voltage can be reduced to lower the power dissipation and possibly increase the clock frequency without exceeding the power limit of the core, thereby achieving test time minimization. However, in accordance with the alpha power law [32, 35, 36], any reduction in voltage causes the critical path delay to increase which, in turn, leads to increased test time. Hence, a proper choice of both V_{DD} and test clock rate is required to optimize the SoC test time. The voltage and frequency scaling method of optimization is applicable to both session-based and sessionless SoC test schedules and has been demonstrated using several SoC benchmarks. The main contribution of the paper can be summarized as providing a heuristic approach to session-based and sessionless test schedules which can handle large complex SoCs and provide solutions comparable to the exact methods.

For the session-based test scheduling, a power-aware SoC test optimization technique by session-wise optimal selection of V_{DD} and clock rate has been proposed. A mixed-integer linear program (MILP) model is formulated to obtain an optimized solution. Results show more than 50 % test time reduction over conventional reference test schedules where V_{DD} and clock rate are fixed at given nominal values. While the MILP method was able to provide optimal solutions for smaller benchmarks, typical SoCs may contain hundreds of cores. Customizing V_{DD} and clock for such SoCs by use of MILP is not practical since the CPU

time required to obtain an optimal solution increases exponentially as the number of cores and the complexity of the SoC increases. Hence, a simulated annealing based heuristic optimization method that can provide near-optimal solutions with much lower runtime than the MILP method for large SoCs is presented. The effectiveness of the algorithm was demonstrated through experiments on SoCs as large as 500 cores. From the results it can be concluded that the size of the SoC did not have a large impact on the performance of the heuristic approach, unlike that of the MILP method. The optimization technique through frequency and voltage scaling was also applied to sessionless test scheduling. The heuristic method, similar to the one proposed for the session-based scheduling, is employed to provide optimal test times. Results show up to 60 % test time reduction over conventional reference test schedules where V_{DD} and clock are fixed at given nominal values. It can be inferred from the comparison between session-based and sessionless testing schemes that the latter could yield the lowest test time but at an additional cost for the test control architecture. The algorithms presented in this paper are only for non-preemptive test schedules and might be extended to preemptive schedules in the future, though keeping in mind that their application could be even more complex.

Acknowledgments Authors are grateful to the National Science Foundation for Grants CCF-1116213, IIP-0738088 and IIP-1266036. They express gratitude to Dr. Spencer Millican and Prof. Kewal Saluja, University of Wisconsin, Madison, for providing power profiles for ITC'02 benchmarks. They thank Erik Jan Marinissen, IMEC, Belgium, and Prof. Nicola Nicolici, McMaster University, Canada, for their valuable inputs on sessionless and session-based test scheduling, and Prof. Alice Smith, Auburn University, for guidance on optimization heuristics.

References

1. Beigné E, Clermidy F, Miermont S, Vivet P (2008) Dynamic voltage and frequency scaling architecture for units integration within a GALS NoC. In: Proceeding Second ACM/IEEE international symp. networks-on-chip, pp 129–138
2. Chang JTY, McCluskey EJ (1996) Detecting delay flaws by very-low-voltage testing. In: Proceeding International test conf, pp 367–376
3. Chang JTY, McCluskey EJ (1996) Quantitative analysis of very-low-voltage testing. In: Proceeding 14th IEEE VLSI test symp., pp 332–337
4. Chou RM, Saluja KK, Agrawal VD (1994) Power constraint scheduling of tests. In: Proceeding 7th International conference VLSI design, pp 271–274
5. Chou RM, Saluja KK, Agrawal VD (1997) Scheduling tests for VLSI systems under power constraints. IEEE Trans VLSI Syst 5(2):175–185
6. Craig GL, Kime CR, Saluja KK (1988) Test scheduling and control for VLSI built-in self-test. IEEE Trans Comput 37(9):1099–1109
7. Devanathan VR, Ravikumar CP, Mehrotra R, Kamakoti V (2007) PMScan: a power-managed scan for simultaneous reduction of

- dynamic and leakage power during scan test. In: Proceeding International test conf., pp 1–10
8. Engelke P, Polian I, Renovell M, Seshadri B, Becker B (2004) The pros and cons of very-low-voltage testing: an analysis based on resistive bridging faults. In: Proceeding 22nd IEEE VLSI test symp., pp 171–178
 9. Girard P, Nicolici N, Wen X (2010) Power-aware testing and test strategies for low power devices. Springer
 10. Guzdial M (2016) Introduction to computing and programming in python, 4th edn. Prentice-Hall
 11. Hao H, McCluskey EJ (1993) Very-low-voltage testing for weak CMOS logic ICs. In: Proceeding international test conf., pp 275–284
 12. Harmanani HM, Salamy HA (2006) A simulated annealing algorithm for system-on-chip test scheduling with power and precedence constraints. *Int J Comput Intell Appl* 6(4):511–530
 13. Huang Y, Reddy SM, Cheng WT, Reuter P, Mukherjee N, Tsai CC, Samman O, Zaidan Y (2002) Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm. In: Proceeding International test conference, pp 74–82
 14. International Technology Roadmap for Semiconductors 2008 Update Overview. <http://www.itrs.net/Links/2008ITRS/Update/>
 15. ITC 2002 SOC Benchmarking Initiative. <http://www.extra.research.philips.com/itc02socbenchm>
 16. Iyengar V, Chakrabarty K (2001) Precedence-based, preemptive and power constrained test scheduling for system-on-chip. In: Proceeding 19th IEEE VLSI test symp., pp 368–374
 17. Iyengar V, Chakrabarty K, Marinissen EJ (2002) On using rectangle packing for SOC wrapper/TAM co-optimization. In: Proceeding 20th IEEE VLSI test symp., pp 253–258
 18. Iyengar V, Chakrabarty K, Marinissen EJ (2002) Test wrapper and test access mechanism co-optimization for system-on-chip. *J Electron Test Theory Appl* 18:211–228
 19. Kavousianos X, Chakrabarty K, Jain A, Parekhji R (2012) Time-division multiplexing for testing SoCs with DVS and multiple voltage islands. In: Proceeding European test symposium, pp 1–6
 20. Kirkpatrick S Jr, Gelatt D, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
 21. Koranne S (2003) Design of reconfigurable access wrappers for embedded core based SoC test. *IEEE Trans VLSI Syst* 11(5):955–960
 22. Larsson E (2005) Introduction to advanced system-on-chip test design and optimization. Springer
 23. Larsson E, Fujiwara H (2002) Power constrained preemptive TAM scheduling. In: Proceeding Seventh IEEE European test workshop, pp 119–126
 24. Larsson E, Peng Z (2002) An integrated framework for the design and optimization of SOC test solutions. *J Electron Test Theory Appl Spec Issue Plug-and-Play Test Autom System-On-Chip* 18:385–400
 25. Li JCM, Tseng CW, McCluskey EJ (2001) Testing for resistive opens and stuck opens. In: Proceeding International test conf., pp 1049–1058
 26. Marinissen EJ (2013) Private communication
 27. Millican SK, Saluja KK (2013) 3D-IC benchmarks. <http://3dsocbench.ece.wisc.edu/> [Online: accessed 03-Jan-2014]
 28. Millican SK, Saluja KK (2013) Private communication
 29. Millican SK, Saluja KK (2014) Optimal test scheduling formulation under power constraints with dynamic voltage and frequency scaling. *J Electron Test Theory Appl* 30(5):569–580
 30. Muresan V, Wang X, Muresan V, Vladutiu M (2000) A comparison of classical scheduling approaches in power-constrained block-test scheduling. In: Proceeding International test conference, pp 882–891
 31. Nicolici N (2013) Private communication
 32. Nose K, Sakurai T (2000) Optimization of V_{DD} and V_{TH} for low-power and high-speed applications. In: Proceeding Asia and South Pacific design automation conference, pp 469–474
 33. Pouget J, Larsson E, Peng Z, Flottes ML, Rouzeyre B (2003) An efficient approach to SoC wrapper design, TAM configuration and test scheduling. In: Proceeding IEEE European test workshop, pp 51–56
 34. Ravi S (2007) Power-aware test: challenges and solutions. In: Proceeding International test conf., pp 1–10
 35. Sakurai T (2004) Alpha power-law MOS model. *Solid-State Circ Soc Newslett* 9(4):4–5
 36. Sakurai T, Newton AR (1990) Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas. *IEEE J Solid-State Circ* 25(2):584–594
 37. Samii S, Larsson E, Chakrabarty K, Peng Z (2006) Cycle-accurate test power modeling and its application to SoC test scheduling. In: Proceeding International test conf., pp 1–10
 38. Sehgal A, Chakrabarty K (2005) Test planning for the effective utilization of port-scalable testers for heterogeneous core-based SoCs. In: Proceeding IEEE International conference on CAD, pp 88–93
 39. Sehgal A, Bahukudumbi S, Chakrabarty K (2008) Power-aware SoC test planning for effective utilization of port scalable testers. *ACM Trans Des Autom Electron Syst* 13(3):53–71
 40. Sheshadri V (2014) Power-aware system-on-chip test optimization through frequency and voltage scaling. PhD thesis, Auburn University, ECE Department
 41. Sheshadri V, Agrawal VD, Agrawal P (2012) Optimal power-constrained SoC test schedules with customizable clock rates. In: Proceeding 25th IEEE system-on-chip conference, pp 271–276
 42. Sheshadri V, Agrawal VD, Agrawal P (2013) Optimum test schedule for SoC with specified clock frequencies and supply voltages. In: Proceeding 26th International conference on VLSI design, pp 267–272
 43. Sheshadri V, Agrawal VD, Agrawal P (2013) Power-aware SoC Test optimization through dynamic voltage and frequency scaling. In: Proceeding 21st IFIP/IEEE International conf. very large scale integration (VLSI-SoC), pp 102–107
 44. Sheshadri V, Agrawal VD, Agrawal P (2013) Session-less SoC test scheduling with frequency scaling. In: Proceeding 22nd IEEE North atlantic test workshop (NATW)
 45. Truong DN, Cheng WH, Mohsenin T, Yu Z, Jacobson AT, Landge G, Meeuwssen MJ, Watnik C, Tran AT, Xiao Z, Work EW, Webb JW, Mejia PV, Baas BM (2009) A 167-processor computational platform in 65 nm CMOS. *IEEE J Solid-State Circ* 44(4):1130–1144
 46. Venkataramani P (2014) Reducing ATE test time by voltage and frequency scaling. PhD thesis, Auburn University, ECE Department
 47. Venkataramani P, Agrawal VD (2013) Reducing test time of power constrained test by optimal selection of supply voltage. In: Proceeding 26th International conference on VLSI design, pp 273–278
 48. Venkataramani P, Agrawal VD (2013) Test time reduction using asynchronous clocking. In: Proceeding International test conf., Paper 15.3
 49. Venkataramani P, Sindia S, Agrawal VD (2013) A test time theorem and its applications. In: Proceeding 14th IEEE Latin-American test workshop
 50. Venkataramani P, Sindia S, Agrawal VD (2013) Finding best voltage and frequency to shorten power-constrained test time. In: Proceeding 31st IEEE VLSI test symp., pp 19–24
 51. Venkataramani P, Sindia S, Agrawal VD (2014) A test time theorem and its applications. *J Electron Test Theory Appl* 30(2):229–236

52. Zhao D, Huang R, Yoneda T, Fujiwara H (2007) Power-aware multi-frequency heterogeneous SoC test framework design with floor-ceiling packing. In: Proceeding IEEE ISCAS, pp 2942–2945
53. Zhao D, Upadhyaya S (2003) Power constrained test scheduling with dynamically varied TAM. In: Proceeding 21st IEEE VLSI test symp., pp 273–278
54. Zhao W, Cao Y (2006) New generation of predictive technology model for sub-45 nm early design exploration. *IEEE Trans Electron Dev* 53(11):2816–2823
55. Zorian Y (1993) A distributed control scheme for complex VLSI devices. In: Proceeding 11th IEEE VLSI test symp., pp 4–9
56. Zou W, Reddy SM, Pomeranz I, Huang Y (2003) SOC Test scheduling using simulated annealing. In: Proceeding 21st IEEE VLSI test symp., pp 325–330

Vijay Sheshadri is with Intel Corporation, Hillsboro, OR. He received his PhD degree from Auburn University, Auburn, Alabama in 2014 and has published papers in several IEEE conferences. Prior to his doctoral degree, Vijay obtained his MS degree from Vanderbilt University, Nashville, Tennessee in 2010. His research interests include power optimization in SoC testing, design-for-testability, single-event upsets and soft-errors. He is a Member of IEEE. Email: vijay.sheshadri@intel.com.

Vishwani D. Agrawal is a Professor Emeritus in the Department of Electrical and Computer Engineering at Auburn University, Alabama, USA. Prior to retirement in 2016, he was the James J. Danaher Professor in the same department. He has over forty years of industry and university experience, working at Bell Labs, Murray Hill, NJ, USA; Rutgers University, New Brunswick, NJ, USA; TRW, Redondo Beach, CA, USA; IIT, Delhi, India; EG&G, Albuquerque, NM, USA; and ATI, Champaign, IL, USA. His areas of expertise include VLSI testing, low-power design, and microwave antennas. He obtained his BE degree from the Indian Institute of Technology Roorkee, Roorkee, India, in 1964; ME degree from the Indian Institute of Science, Bangalore, India, in 1966; and PhD degree in electrical engineering from

the University of Illinois at Urbana-Champaign, in 1971. He has published over 400 papers, has coauthored five books and holds thirteen United States patents. He is the Editor-in-Chief of the *Journal of Electronic Testing: Theory and Applications*, and a past Editor-in-Chief (1985–87) of the *IEEE Design & Test of Computers* magazine. He was the Keynote Speaker at the 25th International Conference on VLSI Design, Hyderabad, India in 2012, and an invited Plenary Speaker at the 1998 International Test Conference, Washington, D.C., USA. He served on the Board of Governors (1989–90) of the IEEE Computer Society, and in 1994 chaired the Fellow Selection Committee of that Society. He has received nine Best Paper Awards, two Lifetime Achievement Awards, and the Distinguished Alumnus Award of the University of Illinois. Agrawal is a Fellow of the ACM, IEEE and IETE-India. He has served on the Advisory Boards of the ECE Departments of the University of Illinois at Urbana-Champaign, New Jersey Institute of Technology, and the City College of the City University of New York. See his website: <http://www.eng.auburn.edu/~vagrawal>.

Prathima Agrawal is a Professor Emeritus of electrical and computer engineering at Auburn University. Previously, she was the Samuel Ginn Distinguished Professor of Electrical and Computer Engineering and Director of the Wireless Engineering Research and Education Center at Auburn and site Director of NSF's Wireless Center for Advanced Technologies. She spent over 25 years in various capacities in Bell Telephone Laboratories Research at Murray Hill, NJ and Bellcore/Telcordia at Morristown, NJ. She holds 50 US patents. She is a Fellow of the IEEE. She has won several awards and was a member of the Advisory Board of the Electrical Engineering Department at the University of Southern California (USC) and on the advisory board of the Technology Leaders Program at the University of Virginia. She is committed to the engineering education of women and minorities and has mentored several African Americans, Hispanic Americans and women at AT&T and Auburn. She received her PhD in Electrical Engineering from the University of Southern California and the B.E. and M. E. degrees in Electrical Communication Engineering from the Indian Institute of Science, Bangalore. See her website: <http://eng.auburn.edu/users/agrawpr>.