2013 22nd Asian Test Symposium

Formulating Optimal Test Scheduling Problem with Dynamic Voltage and Frequency Scaling

Spencer K. Millican and Kewal K. Saluja
Department of Electrical and Computer Engineering, University of Wisconsin-Madison, WI
1415 Engineering Dr., Madison, WI 53706
smillican@wisc.edu, saluja@ece.wisc.edu

***Abstract*—Various techniques for modern high performance designs, such as clock gating and dynamic voltage frequency scaling (DVFS), have been adapted to address power issues. This is a consequence of technology scaling and it is important and desirable to address reliability needs as well as economic issues. From a testing point of view, introduction of power constraints during testing is needed for the desired product quality and to avoid yield loss. Unlike designers who have often benefited from the design for test hardware introduced for testing, test engineers have rarely taken advantage of the extra hardware introduced to meet design needs. In this paper, we make use of the DVFS technology and its associated hardware to improve test economics. We formulate the power constrained testing problem as an optimization problem that makes use of DVFS technology. We show that we can obtain superior test schedules for both session-based and sessionless testing methods relative to existing and traditional methods of obtaining test schedules.**

I. Introduction

System-on-Chip (SoC) designs have become commonplace in modern integrated circuits (ICs). A SoC consists of existing designs of IP (intellectual property) cores, hard or soft, combining them into a single package, thus creating a large design in a relatively short period of time. However, combining several modules into a single package makes the difficult test problem even more challenging. Modules often have their own proprietary test sets, so the objective of SoC testing often is to apply a given test or tests for each module in the shortest possible time while all modules in the SoC are tested. In this paper, by test we mean a set of vectors needed to test a core. In some cases more then one test may be required to test a core in the sense a core may need to be tested by different sets of vectors at different temperatures, etc.

Test scheduling originated as a hardware-constrained problem [1]–[3]. Since individual modules of an SoC often share common hardware to save chip area or for other reasons, simultaneous testing of modules that have such commonalities is often infeasible. Early goals in SoC test scheduling were to schedule all tests of an SoC in the smallest possible period of time while not violating the given hardware constraints [1], [2]. Although this is a relatively simple problem compared to testing designs with modern temperature and power constraints, this is still an NP-hard problem [1].

However, as feature sizes of devices under test have become smaller, the number of transistors per unit area have grown substantially. A consequence of this has been higher power dissipation. As a result, power during SoC test has become yet another problem to address. A very high power dissipation can cause damage to the device under test, or it can create voltage sag induced false failure(s) [4]. Although this is an issue during normal device operation, it is even more of a concern during test since power dissipation during test can be substantially higher than during normal operation [5]. This is due not only to the testing of the devices in the shortest possible period of time, but also due to the fact the tests often generate higher circuit activity and Design for Test hardware also consumes power during testing [6]. As a result, power has become a new constraint during testing. With power-constrained testing, not only must the hardware constraints be met, but a given maximum power value set either based on the power supply used or by the properties of the devices must not be violated. Like hardware-constrained testing, this is an NP-hard problem, but the practical complexity grows significantly compared to hardware-constrained test scheduling. Like hardware-constrained testing, several methods have been proposed to solve power-constrained testing, such as graph formulations [7], ILP formulations [4], and various other algorithms [5], [8]–[10].

A new technology which can provide potentially better schedules in exchange for more complex scheduling is dynamic voltage and frequency scaling (DVFS). DVFS has been used in recent designs to allow for processors to save power and energy during less demanding intervals of operations by decreasing their operating frequency and voltage supply [11]. This decrease in voltage and frequency effectively slows down the operation of a given core or module while at the same time lowering its power consumption. DVFS technology can be used to scale the power and test application time (TAT) of individual tests to find better and more compact test schedules. Test scheduling using DVFS has been studied in the recent past [12]–[15], but the methods proposed in literature have either ignored the power during test aspect, or the solutions proposed have been less than optimal by relying on session-based formulations and other unnecessary constraints. In this paper we address the power issue explicitly and the main contributions of this research are as follows:

- We give an optimal formulation for DVFS SoC testing under power and hardware constraints, for both session-based and sessionless environments.
- We show that DVFS scheduling can provide better and more compact test schedules for power constrained testing compared to schedules without utilizing DVFS technology.
- We show that sessionless formulations generate superior test schedules compared to session-based formulations without excessive computation overhead.

The rest of the paper is organized as follows: Section II gives a brief history of power-constrained and DVFS testing. Section III gives a session-based formulation for DVFS, while Section IV gives a sessionless formulation. Section V gives the experimental setup used to evaluate the effectiveness of the two formulations, while Section VI provides the results for various benchmark SoCs, and the paper concludes with Section VII.

II. Past Work

A. Session Based & Sessionless Test Scheduling

After the introduction of hardware-constrained test scheduling [1], the first modification to be studied was the inefficiencies of session-based testing while scheduling under hardware constraints [16]. Initial test-scheduling formulations were for session-based testing due to their simplicity [1]. Scheduling tests in a session-based manner can be viewed as a bin selection problem, where each test must be assigned to a bin, but no two tests can be assigned to the same bin if

1081-7735/13 $31.00 © 2013 IEEE
DOI 10.1109/ATS.2013.39

they are incompatible (due to sharing of resources, such as common hardware). After each test has been assigned to a bin, the TAT of each bin is the TAT of the longest test in the bin, and the TAT of the schedule is the sum of the TAT of each bin.

However, a deficiency with session-based formulations is that they lack the ability to exploit potential overlaps. When tests are assigned to sessions, it is impossible to overlap tests with any other test outside of the session, even if those tests are compatible. This occurs because a session has one or more tests which are incompatible with the desired overlapping test(s) in an other session. If such tests can be overlapped, then the TAT can be reduced (see Figure 1).

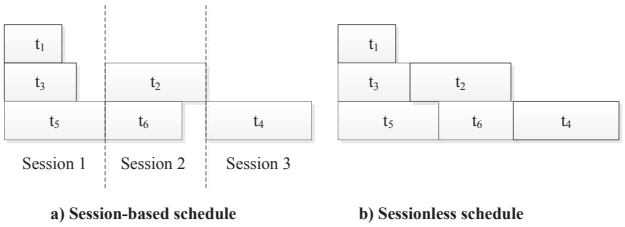

Fig. 1. Presuming that test 2 and 5 are compatible, test time can be reduced by eliminating session requirements.

Formulations for sessionless test scheduling have attempted to remedy the overlapping restraints of session-based formulations as follows. For any sessionless scheduling method, tests must be able to be scheduled arbitrarily in time. This requirement inherently increases the complexity of sessionless scheduling methods. Solutions for sessionless scheduling often make use of heuristics, and a common heuristic makes use of list-based algorithms [17]. However, there have been instances of using deterministic ILP-style formulations for hardware-constrained test scheduling [18].

Although sessionless formulations are successful in finding shorter test schedules, they do so at the expense of longer computation times or heuristic based results providing no guarantee about the quality of the solution. Since sessionless formulations require tests to be scheduled arbitrarily in time, more variables and constraints need to be enforced to implement such a requirement. Some scheduling methods, such as list-based algorithms, are not deterministic and therefore the quality of a schedule depends on the order of the tests in the list [17]. If a good list is given, a good test schedule will be found, but if a bad list is given, the schedule found may be worse than a good session-based schedule.

B. Power & DFVS Test Scheduling

One of the first studies into power-constrained test scheduling was done by Chou et al. [7]. In this study, the complexities of power-constrained test scheduling were first observed as an extension to hardware-constrained test scheduling. The scheduling problem was viewed as a hardware-constrained test scheduling problem with a test or a set of tests being designated as incompatible with another test or tests, except a new power constraint was also enforced. This alone implied that power-constrained test scheduling is more complex than hardware-constrained test scheduling. The approach used to solve the test scheduling problem was a graph-based approach, in which all possible "cliques" and their "subcliques" of a graph are found, with each clique or subclique corresponding to a test session. This approach relied on an earlier implementation with hardware-constrained test scheduling [1], where each test in a clique must be hardware compatible. However, if power constraints are enforced then the additional requirement is that the sum of power from each test in a clique or subclique must also be lower than the given power constraint. This graph-based approach has effectively been implemented in ILP formulation in [14], [15].

As power during test has become larger due to smaller IC feature sizes, DVFS has been explored as a new technique to achieve better test schedules without violating power constraints. The original purpose of DVFS was to decrease the operating voltage and frequency of a module or core when the work load on it decreased so as to save power without hindering performance [11]. However, DVFS can also be used to slow down or speed up individual tests during scheduling to allow for better test compaction under power constraints. If the operating voltage or frequency is lowered for a test, then the test will take longer to complete, but at the same time the test will consume less power. This can allow for two tests that normally could not run in parallel due to their combined power being too high to be made compatible by lower operating voltage or frequency for one or more of the tests. Likewise, a test that must be run individually can increase its operating voltage and frequency to decrease the time needed to apply the test at the expense of increased power consumption.

Initial work on DVFS test scheduling focused on scheduling all possible voltage or frequency combinations of a given core, thereby throughly testing a device [12]. Such studies take the form of a hardware-constrained test scheduling problem, since the goal of such studies is to test all possible DVFS combinations for each IC. However, such test models presume tests need to be applied at multiple VDD values, which is not the case for non-voltage dependent faults. Therefore, such studies have not considered that only a single voltage and frequency pair needs be scheduled for a given test, nor have they considered the power dissipated during test as a constraint.

Some of the recent works focus on scheduling tests such that a test is scheduled with a single voltage and frequency pair under power constraints [14], [15], with the goal being to schedule each test at least once. Initial work focused on only frequency scaling [14], which was considered to be a simpler problem since power scales linearly with operating frequency. After frequency scaling alone was studied, voltage scaling was added [15]. Both cases showed that allowing for voltage or frequency scaling would result into significantly better test schedules with power constraints. However, these studies not only presumed that scheduling was done in sessions, but they also presumed that for any given session the voltage and frequency pair for every test in the session must be the same, which can lead to inferior results.

III. Session Based Testing

The first formulation presented in this paper is for session-based testing of SoCs. This formulation is presented for the purpose of gaining an understanding of DVFS scheduling, as well as to later gain a perspective on the deficiencies of session-based formulations.

A. Test Environment and modeling

Before a formulation can be made, some assumptions have to be made about the environment in which tests are scheduled. As shown in Section II, different models exist describing how DVFS is used during test. Section VII will discuss what effect other models will have on this study and how the proposed formulations can be extended to fit other models.

The first presumption is that the power dissipated by a test is constant throughout the entire test. This assumption is made so that the power values of a test can be directly used in an ILP-style formulation, for changing power values during test drastically increases the complexity of any formulation. This assumption, that the power during test is constant, has been made by the majority of formulations in the past due its simplicity. For tests that do not have

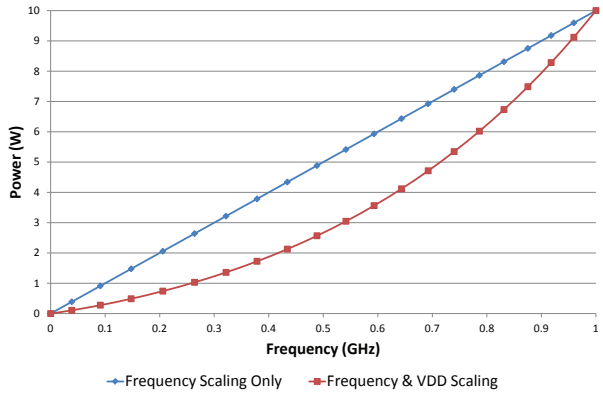

Fig. 2. By scaling V_{DD} with frequency as opposed to frequency alone, better power reduction can be achieved.

constant power dissipation, the power dissipation can be presumed to be the maximum power dissipation of a test. Although it is possible to achieve better test schedules if power is not assumed to be constant by exploiting more opportunities for overlap, it is presumed that this overlap is small relative to any scheduling benefit that can be achieved through other scheduling methods. In the case where 2 or more cores share DVFS hardware, specific constraints can be added to enforce incompatibility between tests being tested at different voltage and frequency values.

It is also assumed that the V_{DD} and frequency of any test can be independently controlled at any time, regardless of what session it is scheduled in or what other tests it overlaps with. Modeling in this manner will make this formulation unique from the formulation proposed in [15]. In [15], it was assumed that every test in a session must have the same V_{DD} and frequency applied to it. By eliminating this assumption, better schedules can be obtained by allowing for better compaction of tests within a session.

In this paper it is assumed that the time overhead required to change the operating V_{DD} and frequency of a core is negligible. If such is not the case, overhead can be modeled by increasing test time corresponding to DVFS switching overhead.

The last aspect of modeling is how operating V_{DD} and frequency is chosen for test scheduling. An initial assumption is that during normal V_{DD} operation (in this study, $1V$), the operating frequency is chosen so as to not violate timing constraints. This means the operational frequency of a test can be increased (therefore decreasing the length of the test) only if the operating voltage is increased. As well, if the frequency of a test is decreased (test length increased), the operating voltage may be decreased since there is no requirement to have a higher voltage drive the lower operating frequency. Given this assumption, every V_{DD} value should have only the highest possible frequency value assigned to it, since assigning any lower frequency value would only waste power and increase test time or both. The same can be said of assigning V_{DD} values to frequency values, but in this study pairs are chosen based on V_{DD} since it is presumed that frequency can be scaled more finely, whereas V_{DD} can only be scaled by set amounts. From this, for a given V_{DD} value, the frequency can be set according to the alpha power law [11], [19], where f_s is the factor by which frequency is scaled:

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

The motivation for scaling V_{DD} with frequency as opposed to scaling frequency alone is that scaling V_{DD} along with frequency allows for less power to be consumed. If scaling frequency is done alone, dynamic power consumption is scaled by the same factor. However, scaling to a lower frequency allows for V_{DD} to be scaled lower as well. By scaling V_{DD} to the lowest possible value for a given frequency, the relation between frequency and power is no longer linear, but is instead cubed, since the relation between dynamic power consumption and DVFS is $P \propto F \cdot V_{DD}^2$. This concept is illustrated in Figure 2, which presumes the assumptions stated below with a normal operation frequency of $1GHz$ and normal power consumption of $10W$.

In this paper, it is assumed that modern short-channel MOSFET technology is used, with $V_{DD} = 1V$, $V_{TH} = 0.5V$, and $\alpha = 1.3$.

B. Formulation

The formulation for both session-based and sessionless scheduling is given in the form of a mixed-integer linear programming (MILP) formulation. This style of formulation was chosen due to its deterministic nature. Below we give a detailed formulation of the session based scheduling problem.

The first constraint of any scheduling formulation is to define the objective, and the objective in this case is to minimize the total TAT of the schedule. Since the constraint must be made linear, this can be done by forcing the overall TAT to be greater than or equal to the finishing time of each session, $t_f(s)$.

$$\text{minimize } t_{finish} \text{ subject to...}$$

$$\forall s \in S : t_{finish} \geq t_f(s)$$

The start time and finish time of each session, $t_s(s)$ and $t_f(s)$, is dependent on the TAT length of the session, $L(s)$. However, the start time of each session is guaranteed to be the finish time of the previous session, since doing otherwise would only waste time. Note that the maximum possible number of sessions is equal to the number of tests, since the worst-case schedule would have every test assigned to its own session.

$$\forall s \in S : t_f(s) = t_s(s) + L(s)$$

$$\forall s = 2 \ldots |S| : t_s(s) = t_f(s-1)$$

$$t_s(1) = 0;$$

Every test must now be assigned to a session. A new binary variable is created for the purpose of assigning tests (t) to sessions.

$$\Delta(t,s) = \begin{cases} 1 & \text{if the test } t \text{ is assigned to session } s \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t \in T : \sum_{\forall s \in S} \Delta(t,s) = 1$$

If two tests are incompatible due to hardware constraints, then it is impossible for them to be scheduled in the same session.

$$\Gamma(t_1,t_2) = \begin{cases} 1 & \text{if tests } t_1 \text{ and } t_2 \text{ are incompatible} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t_1 \in T, \forall t_2 \in T, \forall s \in S : \Delta(t_1,s) + \Delta(t_2,s) \leq 2 - \Gamma(t_1,t_2)$$

Every test must be assigned to one and only one V_{DD}/frequency pair, assuming there are N V_{DD}/frequency selections available to choose from for each test.

$$\Psi(t,n) = \begin{cases} 1 & \text{if the test } t \text{ is assigned to} \\ & V_{DD}\text{/frequency pair } n \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t \in T : \sum_{\forall n=1 \ldots N} \Psi(t,n) = 1$$

The next variable defined, $\Theta(t,s,n)$, is for the purpose of stating the TAT and power of each session in a linear formulation.

$$\Theta(t,s,n) = \begin{cases} 1 & \text{if the test } t \text{ is assigned to session } s \\ & \text{and assigned to } V_{DD}\text{/frequency pair } n \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t \in T, \forall s \in S, \forall n \in N : \Theta(t,s,n) \geq -1 + \Delta(t,s) + \Psi(t,n)$$

$$\forall t \in T, \forall s \in S, \forall n \in N : \Theta(t,s,n) \leq \Delta(t,s)$$

$$\forall t \in T, \forall s \in S, \forall n \in N : \Theta(t,s,n) \leq \Psi(t,n)$$

The TAT of each session is defined as the longest time amongst all tests within the session. Since the number of sessions is set to the number of tests, some sessions may be unused. If such is the case, the TAT of each session must be at least 0. Here, $L_{c,n}$ is the TAT of a test t if the n^{th} V_{DD}/frequency pair is chosen for it.

$$\forall s \in S : L(s) \geq 0$$

$$\forall s \in S, \forall t \in T, \forall n \in N : L(s) \geq L_{t,n}\Theta(t,s,n)$$

To satisfy a given power constraint, the sum of power from all tests in a session must be lower than the given power bound in every session. Here, $P_{c,n}$ is the power of a test t if the n^{th} VDD/frequency pair is chosen for it.

$$\forall s \in S : \sum_{\forall t \in T} \sum_{\forall n \in N} P_{t,n}\Theta(t,s,n) \leq P_{BOUND}$$

IV. Sessionless Test Scheduling

As was stated in Section II-A, session-based formulations will often fail to find a low test time solution due to their restricted nature. It is easy to see that any possible session-constrained schedule can be converted to a sessionless schedule in the sense that no resource and power constraints are violated, but not every sessionless schedule can be stated as a session-based schedule. Therefore, if there is any test set which has a better sessionless schedule than an optimal session-based formulation, then sessionless formulations are guaranteed to find a schedule no worse (but often better) than a session-based formulation. Such a schedule has already been presented in Figure 1.

Unlike with session-based test scheduling, tests can be scheduled arbitrarily in time as long as no two overlapping tests have a hardware constraint. Because of this, timing constraints need to be redefined. First, test application time is no longer based on sessions, but based on individual tests.

$$\text{minimize } t_{finish} \text{ subject to...}$$

$$\forall t \in T : t_{finish} \geq t_f(t)$$

$$\forall t \in T : t_f(t) = t_s(t) + L(t)$$

$$\forall t \in T : t_s(t) \geq 0$$

The TAT of each test, $L(t)$, is dependent on the DVFS pair chosen for that test. The variable $\Psi(t,n)$ can be kept in use for this purpose.

$$\forall t \in T : L(t) = \sum_{\forall n \in N} L_{t,n}\Psi(t,n)$$

To keep tests from overlapping that have a hardware incompatibility between them, the constant $\Gamma(t_1,t_2)$ can be reused. However, the fashion in which the constant is used must be changed, since tests are no longer assigned to sessions, but instead can overlap arbitrarily. To define how tests can overlap, the method from [18] is borrowed to accomplish this. Here, λ is defined as the longest possible test schedule, which is the sum of run times of all tests at their slowest frequency. Setting λ to such or higher allows for the arbitrary scheduling of the start time of any test from time 0 to $\lambda - L(t)$ while defining overlapping tests.

$$\eta(t_1,t_2) = \begin{cases} 1 & \text{if the test } t_1 \text{ finishes} \\ & \text{before the test } t_2 \text{ begins} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t_1 \in T, \forall t_2 \in T : t_f(t_1) \leq t_s(t_2) + (1 - \eta(t_1,t_2)) \cdot \lambda$$

$$\forall t_1 \in T, \forall t_2 \in T : t_s(t_s) \leq t_f(t_1) + \eta(t_1,t_2) \cdot \lambda$$

Two tests t_1 and t_2 are overlapping if both $\eta(t_1,t_2)$ and $\eta(t_2,t_1) = 0$. Using this, hardware compatibility can be enforced.

$$\forall t_1 \in T, \forall t_2 \in T : \eta(t_1,t_2) + \eta(t_2,t_1) \leq 1$$

$$\forall t_1 \in T, \forall t_2 \in T : \eta(t_1,t_2) + \eta(t_2,t_1) \geq \Gamma(t_1,t_2)$$

To enforce a power constraint in a sessionless formulation, the sum of all power values for any combination of overlapping tests must not exceed a given bound.

$$\Omega(t_1,t_2) = \begin{cases} 1 & \text{if test } t_1 \text{ overlaps with test } t_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t_1 \in T, \forall t_2 \in T : \Omega(t_1,t_2) = 1 - (\eta(t_1,t_2) + \eta(t_2,t_1))$$

The variable $\Theta(t_1,t_2,n)$ is redefined to allow the power constraint to be stated linearly. Its formulation is nearly identical to the $\Theta(t,s,n)$ formulation in Section III-B, except $\Psi(t,n)$ and $\Omega(t_1,t_2)$ are used instead of $\Psi(t,n)$ and $\Delta(t,s)$.

$$\Theta(t_1,t_2,n) = \begin{cases} 1 & \text{if test } t_1 \text{ overlaps with test } t_2 \\ & \text{\& the } n^{th} \text{ VDD/frequency pair for} \\ & t_1 \text{ is chosen} \\ 0 & \text{otherwise} \end{cases}$$

$$\forall t_1 \in T : \sum_{\forall t_2 \in T} \sum_{\forall n \in N} \Theta(t_1,t_2,n) \cdot P_{t_3,n} \leq P_{BOUND}$$

V. Experimental Setup

A. Benchmarks

Benchmarks used in this study are based off the ITC'02 benchmarks [20]. These benchmarks represent SoC and provide test length information. Although they also provide module information for the purpose of generating hardware compatibility, this study generates hardware compatibility using a randomized graph-based approach, since hardware compatibility is not the focus of this study [17]. Since the original benchmarks do not contain significant power information, power traces for individual modules and their tests are generated by simulating ITC'99 circuits [21] and assigning those power traces to modules by matching the size and complexity of the ITC'02 modules. Leakage power is not addressed in this study. However, if leakage power is modeled as constant power dissipation proportional to core area than this has the equivalent effect of increase dynamic power consumption by a constant factor proportional to V_{DD} which in turn can be incorporated in DVFS pairs. Specifics on the benchmarks can be found at [22]. These benchmarks represent 3D-stacked ICs, which in this study has the effect of increasing the number of modules for a given benchmark.

TABLE I
BENCHMARK INFORMATION

Bench #	# Tests	Max Power (W)	Bench #	# Tests	Max Power (W)
1	9	10.9	9	29	11.6
2	15	11.6	10	25	11.6
3	10	4.2	11	24	5.4
4	8	14.2	12	12	34.5
5	14	5.4	13	39	11.6
6	4	34.5	14	33	34.5
7	4	13.1	15	22	34.5
8	7	13.4	16	47	14.2

TABLE II
BENCHMARK RESULTS

Bench #	TAT (us)		Scheduling Runtime (s)	
	Sessionless	Session-based	Sessionless	Session-based
1	13672.39	13672.39	0.29	13.62
2	250.32	264.79	1.1	30.5
3	75.24	75.24	0.27	15.72
4	645.34	645.34	0.18	1.85
5	102.73	102.73	2.15	779.43
6	1219.79	1219.79	0.03	0.03
7	8048.03	8048.03	0.03	0.1
8	55143.92	60166.34	0.95	2.09
9	238.74	259.00	68.76	22689.2
10	238.74	263.34	47.71	2686.36
11	82.47	82.47	71.45	1031.55
12	1219.79	1234.26	0.93	3.47
13	238.74	260.45	216.81	31305.73
14	1222.68	1240.05	18553.6	2081.72
15	8048.03	8076.97	172.18	5843.37
16	436.98	455.79	2816.44	5387.26

B. Implementation

The first experiment is done to evaluate the effectiveness of session-based scheduling as opposed to sessionless scheduling across a series of benchmarks. Information on the sixteen benchmarks used to evaluate the scheduling methods is provided in Table I, which includes the number of tests (and modules, since ITC'02 benchmarks have one test per module) in the benchmark and the maximum power consumed by any given test in the benchmark. The two results recorded for each scheduling method on each benchmark are the optimal scheduled TAT and the computation time required to obtain the optimal schedule.

The second experiment is done to evaluate the effectiveness of DVFS scheduling as more DVFS pairs become available to schedule. Several benchmarks are taken from the original set of sixteen benchmarks and are run with a varying number of DVFS pairs available for scheduling. This is done to observe the effect DVFS has on TAT.

All MILP formulations are implemented using IBM ILOG CPLEX, and they are run to completion to find an optimal solution for a given formulation. For all experiments, the maximum power bound (P_{BOUND}) is set to 20 Watts, which is purposely set lower than the max power of some benchmarks to show the requirement of DVFS in test scheduling in such benchmarks.

VI. RESULTS

Table II shows the results of session-based and sessionless DVFS scheduling on sixteen benchmarks, presuming a normal operating V_{DD} value of $1V$ and a normal operating frequency of $120MHz$. For these benchmarks, four V_{DD} values were available for each test to choose from (1, 0.9, 0.8, and 0.7 volts). The frequency values corresponding to these V_{DD} values are obtained using the method as described in Section III-A.

The results in Table II clearly show that for any benchmark the schedule obtained by sessionless scheduling is always equal to or better than that achieved by session-based scheduling. The reason for this, as explained in Section II-A, is that any session-based schedule is possible with sessionless scheduling, but certain sessionless schedules cannot be expressed as session-based schedules. However, there are cases when both scheduling methods obtain the same or nearly same total TAT. These instances appear more often for benchmarks with a lower number of tests to schedule. The more tests there are to schedule, the more possible schedules exist, which in turn means there are more schedules that sessionless scheduling can take advantage of that session-based scheduling cannot.

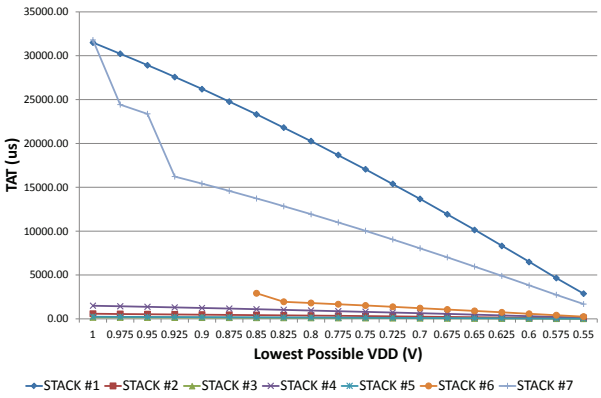

Fig. 3. As more V_{DD} options become available, better schedules can be obtained.

Although sessionless scheduling can clearly give better TAT results, the primary motivation behind session-based scheduling is the simplicity of implementation, and thus reduced design effort and hardware complexity. Also, it is intuitively expected that computation effort to find solution to session-based scheduling is lower than that for sessionless schedules. However, the results in Table II show a contrary phenomenon. The computation time for all benchmarks except for one (Bench 14) favors sessionless scheduling. The reason for this anomaly is that both methods require the same number of variables to solve, even though the session based scheduling problem is apparently simpler. In sessionless scheduling, the primary variable in question is the start time of each test ($t_s(c)$), while in session-based scheduling the primary variable is which session a test belongs to (in this study, implemented using $\Delta(c, s)$). From the point of view of a MILP solver, session-based scheduling is actually more complex since the "primary variable" is implemented using several binary variables as opposed to a single integer variable. It may be possible to make the session-based formulation more efficient, but the point remains clear that a sessionless formulation is by no means overly complex to implement or time-consuming.

Figure 3 gives the TAT of sessionless formulations for selected benchmarks (selected for feasible computation time) as the allowed range of V_{DD} values increases. The value at the X-axis represents the lowest possible V_{DD} value allowed, with the all V_{DD} values allowed between that value and $1V$ with a $0.025V$ granularity. For instance, for the $0.9V$ data point, the V_{DD} values available to schedule for each test is 1, 0.975, 0.95, 0.925, and $0.9V$.

One observation based on Figure 3 is that as lower V_{DD} values are allowed for scheduling, the TAT is greatly reduced. It is true that lowering V_{DD} values can allow for the overlapping of tests that was not possible before due to power constraints, but at the same time lowering V_{DD} values (and in turn lowering operating frequency) increases individual test lengths. This leads to the conclusion that the effect of reducing test power outweighs the effect of increasing test time due to greater test overlap potential.

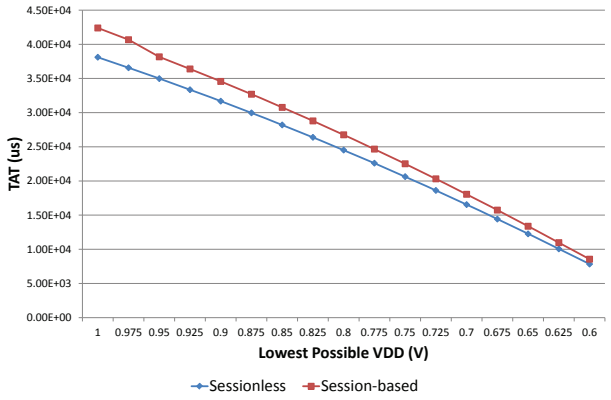

Fig. 4. As more V_{DD} options become available, sessionless scheduling will still give better schedules than session-based schedules.

A second observation from Figure 3 is that DVFS may make test scheduling possible for circuits that were not testable without DVFS. This is shown well with Stack #6, which is not schedulable until $0.85V$ V_{DD} is allowed. Although it is not wise to create a design with test power higher than a given limit, this may be the case with high-power designs, since test power is often higher than normal operating power [5].

Figure 4 gives the TAT of both session-based and sessionless formulations for Benchmark #8 as the allowed range of V_{DD} values increases. An observation based on Figure 4 is for any DVFS values available for scheduling, sessionless schedules always performs better than session-based scheduling. This is a trend observed in all stacks. This is confirmation of the statement made in Section II-A, since adding DVFS generates more scheduling possibilities and sessionless scheduling can take more advantage of these scheduling options than session-based scheduling can.

VII. Conclusions

Scheduling tests in a DVFS environment shows the potential for greatly reducing TAT for multi-core SoCs under power constraints. As voltages are scaled lower, individual test times are sacrificed for reduced power consumption. This trade off allows for greater test time compaction under power constraints, which in turn leads to better test economy.

The benefit of the TAT reduction achieved by the proposed formulations can be further applied to different DVFS test scheduling models. For models that presume tests must be scheduled for every DVFS voltage, each test is duplicated to represent running a test at a particular voltage level and the variable determining operating V_{DD} will determine the operating frequency instead. To model the existence of voltage islands (multiple modules sharing the same voltage/frequency source), a new MILP constraint can be added to force the Ψ values of two tests to be equal if they are scheduled simultaneously and share the same voltage/frequency source. In both cases, the proposed formulations will achieve lower TAT compared to previous formulations, and sessionless scheduling will allow for lower TAT than session-based scheduling.

To take maximum advantage of DVFS scaling, the session-based test scheduling barrier must be broken. By breaking this barrier, better test schedules can be found, thereby reducing TAT and leading to better test economy. Although session-based scheduling has been previously implemented to reduce schedule computation time through simpler formulations, this study has shown that the computation time of sessionless formulations is by no means a hindrance to their implementation.

References

[1] C. Kime and K. Saluja, "Test Schduling in Testable VLSI Circuits," in *International Symposium on Fault-Tolerant Computing*. Santa Monica: IEEE, 1982, pp. 406–412.

[2] K. Chakrabarty, "Test scheduling for core-based systems using mixed-integer linear programming," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 10, pp. 1163–1174, 2000.

[3] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in *VLSI Test Symposium*. IEEE Comput. Soc. Press, 1993, pp. 4–9.

[4] M. Nourani and J. Chin, "Power-time tradeoff in test scheduling for SoCs," in *Proceedings 21st International Conference on Computer Design*. IEEE Comput. Soc, 2003, pp. 548–553.

[5] S. Samii, M. Selkala, E. Larsson, and K. Chakrabarty, "Cycle-Accurate Test Power Modeling and Its Application to SoC Test Architecture Design and Scheduling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 5, pp. 973–977, May 2008.

[6] P. Girard, N. Nicolici, and X. Wen, *Power-Aware Testing and Test Strategies for Low Power Devies*. Springer, 2010.

[7] R. 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*, vol. 5, no. 2, pp. 175–185, Jun. 1997.

[8] Z. He, Z. Peng, and P. Eles, "Power Constrained and Defect-Probability Driven SoC Test Scheduling with Test Set Partitioning," in *Design, Automation & Test in Europe*. IEEE, 2006, pp. 1–6.

[9] E. Larsson and H. Fujiwara, "Power Constrained Preemptive TAM Scheduling," in *IEEE European Test Workshop*. IEEE Comput. Soc, 2002, pp. 119–126.

[10] D. Zhao and S. Upadhyaya, "Dynamically partitioned test scheduling with adaptive TAM configuration for power-constrained SoC testing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 6, pp. 956–965, Jun. 2005.

[11] K. Nose and T. Sakurai, "Optimization of VDD and VTH for Low-Power and High-Speed Applications," in *Asia and South Pacific Design Automation Conference*. IEEE, 2000, pp. 469–474.

[12] X. Kavousianos, K. Chakrabarty, A. Jain, and R. Parekhji, "Test Scheduling for Multicore SoCs with Dynamic Voltage Scaling and Multiple Voltage Islands," in *2011 Asian Test Symposium*. IEEE, Nov. 2011, pp. 33–39.

[13] H. Salamy and H. M. Harmanani, "An optimal formulation for test scheduling network-on-chip using multiple clock rates," in *anadian Conference on Electrical and Computer Engineering*. IEEE, May 2011, pp. 215–218.

[14] V. Sheshardi, V. D. Agrawal, and P. Agrawal, "Optimal Power-Constrained SoC Test Schedules With Customizable Clock Rates," in *ACM Symposium on Cloud Computing*, San Jose, CA, Oct. 2012, pp. 271–276.

[15] V. Sheshardi, V. Agrawal, and P. Agrawal, "Optimum Test Schedule for SoC with Specified Clock Frequencies and Supply Voltages," in *VLSI Design*, Pune, Jan. 2013.

[16] G. Craig, C. Kime, and K. K. Saluja, "Test scheduling and control for VLSI built-in self-test," *IEEE Transactions on Computers*, vol. 37, no. 9, pp. 1099–1109, 1988.

[17] C. Yao, K. K. Saluja, and P. Ramanathan, "Power and Thermal Constrained Test Scheduling Under Deep Submicron Technologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 2, pp. 317–322, Feb. 2011.

[18] D. R. Bild, S. Misra, T. Chantemy, P. Kumar, R. P. Dick, X. S. Huy, and A. Choudhary, "Temperature-aware test scheduling for multiprocessor systems-on-chip," in *IEEE/ACM International Conference on Computer-Aided Design*. IEEE, Nov. 2008, pp. 59–66.

[19] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[20] E. Marinissen, V. Iyengar, and K. Chakrabarty, "A set of benchmarks for modular testing of SOCs," in *Proceedings. International Test Conference*. Baltimore, MD: IEEE, Oct. 2002, pp. 519–528.

[21] S. Davidson, "ITC'99 Benchmark Circuits - Preliminary Results," in *International Test Conference 1999. Proceedings (IEEE Cat. No.99CH37034)*. Int. Test. Conference, 1999, pp. 1125–1125.

[22] S. K. Millican and K. K. Saluja. (2013) 3D-IC Benchmarks. [Online]. Available: http://3dsocbench.ece.wisc.edu/