SoC TAM Design to Minimize Test Application Time

Huiting Zhang
Department of Electrical and Computer Engineering
Auburn University
Auburn, Alabama 36849
Email:hzz0023@auburn.edu

Vishwani D. Agrawal
Department of Electrical and Computer Engineering
Auburn University
Auburn, Alabama 36849
Email: agrawvd@auburn.edu

Abstract—We propose a new test access mechanism (TAM) design to reduce the System-on-chip (SoC) test application time (TAT) under various hardware and power constraints. Prior works on TAM design focus on designing TAM architecture based on fixed cores parameters, which assumes unchanged internal scan chains and layout arrangement of cores, and therefore is unable to make effective use of SoC resources for testing purpose. Moreover, previous works on SoC test scheduling fail to incorporate the presented hardware and power constraints at once, which may render the testing result inapplicable. The proposed TAM design is tightly integrated with scan chain redesign and layout arrangement of individual core in SoC, and thus able to reduce the wiring complexity and to make effective use of TAM resources. Various hardware and power constraints are considered in this work, which are often ignored or partly addressed by previous SoC test scheduling work. Dynamic voltage and frequency scaling (DVFS) is adopted in test scheduling to minimize testing time. Both session-less and session-based scheduling are formulated in Mixed-integer linear programming. Experimental result of a set of ITC’02 benchmark shows up to 69% of reduction in test application time with the proposed TAM Design and DVFS scheduling.

Keywords - SoC Test Scheduling, DVFS, TAM Design, MILP, TAT, hardware and power constraints.

I. INTRODUCTION

A system-on-chip (SoC) is an entire system that may consist of intellectual property (IP) cores integrated on a single chip. Owing to the proprietary nature of cores, test development is often difficult. Therefore, pre-defined tests are provided to the user by IP core vendors. SoC testing is based on modular testing, which is shown on Fig. 1. Test Wrappers are used to isolate the core for the specific modular testing purpose, while test access mechanism (TAM) serves as a communication bus to transport test data from source to sink. Test source provides test vectors and it could be either on chip or off chip. A typical built-in chip test source is a linear feedback shift register (LFSR) or a ROM. An alternative approach is to apply test patterns externally from automatic test equipment (ATE). Test sink is either an embedded on-chip signature analyzer or an off-chip ATE that provides output comparison with stored response. In this work, a test is referred to as a set of test vectors that is given by the IP vendor and it is assumed that each core could have one or more tests.

One of the primary goal of any SoC testing is to reduce test application time (TAT). With the increasing growth of SoC size, there has been an enormous growth in test application time, test data volume, and rest resource usage. It is necessary to effectively design and use SoC test resources to reduce the test application time. A framework of SoC TAM design is therefore presented in this paper.

The optimization problem we address here is stated as follows: Given the test parameters of cores, as well as total pre-defined SoC TAM width, determine the optimal assignment of TAM, and core arrangements including number of internal scan chains. By integrating design of core arrangement with TAM design, optimal core arrangement is achieved to minimize the TAM wiring complexity by reducing the need for wiring non-contiguous TAM wires to a single core. With more available internal scan chains inside a core, fewer scan shift cycles are required. The limitation on the number of internal scan chains is the available TAM wires to the core. By considering the TAM design and internal scan chains together, optimal test application time is achieved.

The remainder of this paper is organized as follows. Section II provides related work on SoC testing. Sections III elaborates on TAM design features incorporated in Sections IV and V that mathematically formulate session-based and session-less test scheduling problems. Section VI gives an experimental setup. Section VII gives experimental results for five example cases. Section VIII concludes this work.

II. RELATED WORK

Integrating test access mechanism (TAM) design and test scheduling can significantly improve the TAT. An
earlier proposal [3] on flexible-width TAM architecture makes full use of the resources to achieve the optimal test time. However, the core layout arrangement may not be tailored to the need of TAM assignments and may require connectivity of arbitrary TAM wires to a core increasing wiring complexity. To solve this problem, this paper integrates TAM design with layout arrangement of SoC cores. In this way, core arrangement is determined based on the optimal TAM assignment to cores. Such optimal arrangement of SoC cores would allow TAM wires to be partitioned and wired in a way that the possibility of wiring non-continuous TAM wires to a single core is greatly reduced.

That work [3] presents a staircase function between TAM width and test time, in which test time only decreases with the increase of TAM width in discrete steps. The reason stated at [3] is that as TAM width increases, the internal scan chains of the core are redistributed among large number of wrapper scan chains. The longest scan chains could only be reduced when the increase of TAM with is sufficient to remove the internal scan chains from the longest wrapper scan chains. With the flexibility on redesigning internal scan chains, as proposed in this paper, the relationship between test time and TAM would be a linearly decreasing function. For instance, the test time of 10 internal scan chains would be twice as that of 20 scan chains.

Another unresolved problem of previous work is the inadequate consideration of various hardware and power constraints. In [1], authors include TAM design in linear programming, however, fail to take some constraints into account. These are constraints on pre-defined shared hardware and shared voltage islands. Thus, some test schedules may become impractical. To solve this problem, new constraints are included in our test scheduling process. Moreover, the authors in [1] focus on the reduction of TAT with increase of SoC testing resources, like a TAM pin or through silicon via (TSV). In general, an even marginal increase of these resources may lead to high cost. For instance, the ATE capacity may not be high enough to support the increase of the TAM width. Besides, that work [1] was done with stacked ICs, in which the number of TSVs would affect the TAM width allocation. Therefore, the effect of TAM resource allocation of that work may not be directly applicable to a single die SoC, which is what the present work focuses on.

III. TAM DESIGN

The optimization of SoC test scheduling is recognized as a hardware-constraint problem [7] [8]. It is infeasible to schedule concurrent testing of modules that have resource conflicts. Therefore, one of the goals of SoC testing is to minimize the overall test time while satisfying the hardware constraints. In this work, we model session-less test scheduling under power constraint, pre-defined constraints on shared hardware (e.g., BIST or memory), and shared voltage islands among cores, which are ignored or only partly addressed previously. Previously proposed session-based scheduling has most these constraints except those from voltage islands. These constraints would be explained with the new scheduling formulation in session-less test scheduling in Section IV. Power constraint is modeled as constant peak power budget, which is chosen to be the maximal peak power of all tests among all test patterns and cycles.

In this paper, dynamic voltage and frequency scaling (DVFS) is adopted to reduce TAT during test scheduling. DVFS could benefit test scheduling by lowering the voltage of cores being tested and therefore accommodate more concurrent tests in a schedule without exceeding the power budget, or it could increase the voltage of a test session to reduce the test application time (TAT) if the tests structurally constrained by the critical path.

Test scheduling is a process of defining an application schedule for tests of various cores on SoC under hardware and power constraints. Existing test scheduling strategies can be classified broadly as session-based test scheduling and session-less test scheduling. [5]

The proposed work is divided into two parts:

1. The first part is test access mechanism (TAM) design. This TAM design is incorporated in the SoC design procedure and therefore allows modifications of internal scan chains and layout arrangement of cores to benefit test application time. Some unique characteristics differentiate this TAM design from previous work and the details will be discussed later in this paper.

2. The second part is test scheduling with DVFS under various hardware and power constraints. The main contribution of this test scheduling is to model SoC testing under all hardware and power constraints at once, which include hardware constraints on voltage islands and power budget during SoC testing. These are either ignored or only partly addressed in the previous work. Details on the constraint handling will be discussed in the following sections.

IV. SESSION-LESS TEST SCHEDULING

Variables of the session-less test scheduling in this paper are imported from the work of Millican and Saluja [1], and Bild et al. [14]. They proposed a formulation for scheduling tests with constraints based on overlapping tests. However, these formulations do not incorporate complete hardware and power constraints into their test scheduling formulation. In the present session-less test scheduling, pre-defined hardware constraints, voltage islands, and power budget are incorporated into the session-less formulation, adding improved practicality to the resulting test schedules.

The goal of any SoC scheduling is primarily to reduce the test time, therefore, we define a decision variable \( t_{final} \), which identifies the finish time of an SoC test schedule, For
each single test, the finish time \( F(t) \) would be the sum of its start time \( S(t) \) and test time \( L(t) \):

\[
\forall t \in T: t_{\text{finish}} \geq F(t) \\
\forall t \in T: F(t) = S(t) + L(t) \\
\forall t \in T: S(t) \geq 0
\]

To enforce this constraint, it is necessary to know the overlapped tests. \( \eta(t_1, t_2) \) is used to define the relationship of start time and finish time of two tests, which would determine the value of \( \text{overlap}(t_1, t_2) \) later. \( \lambda_i \) is defined as the TAT of the longest possible schedule, which corresponds to the TAT when each test operates on lowest voltage and frequency. Thus,

\[
\eta(t_1, t_2) = \begin{cases} 
1, & \text{If the test } t_1 \text{ finishes before } t_2 \text{ starts} \\
0, & \text{Otherwise}
\end{cases}
\]

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

\[
\forall t_1, t_2 \in T: F(t_1) \leq S(t_2) + (1 - \eta(t_1, t_2)) \cdot \lambda_i
\]

\[
\forall t_1, t_2 \in T: S(t_2) < F(t_1) + \eta(t_1, t_2) \cdot \lambda_i
\]

\[
\text{Overlap}(t_1, t_2) = \begin{cases} 
1, & \text{If the test } t_1 \text{ overlaps with } t_2 \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\forall t_1, t_2 \in T: \text{Overlap}(t_1, t_2) = 1 - \eta(t_1, t_2) - \eta(t_2, t_1)
\]

The run time of each test depends on two variables: the voltage/frequency pair, and the TAM width it is assigned. The decision variable that contains these two attributes is noted as \( \text{Assign}(t,n,w) \). In this work, there are 4 choices of voltage/frequency pairs and TAM width, e.g., \( n \in N = \{1, 2, 3, 4\}, w \in C = \{1, 2, 3, 4\} \). However, some SoC benchmarks are large so the TAM width choice is reduced to 3 to save the scheduling time. The performance is not compromised if the input data is carefully and reasonably selected. It is upon users choose to modify the number of choices for voltage/frequency pairs and TAM width. The actual value of voltage/frequency pairs and TAM width correspond to each choice of \( n \) and \( w \) are discussed later. Below, the constant \( L(t), n, w \) denotes the run time of test \( t \) when executed at DVFS index \( n \) and TAM width index \( w \). It is required that for any test \( t \), only one voltage/frequency pair and TAM width choice should be assigned at a time.

\[
\text{Assign}(t, n, w) = \begin{cases} 
1, & \text{If test is assigned is assigned with } n \text{th v/f pairs and } w \text{th width choice} \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\forall t \in T: \sum_{n \in N} \sum_{w \in C} \text{Assign}(t, n, w) = 1
\]

\[
\forall t \in T: L(t) = \sum_{n \in N} \sum_{w \in C} \text{Assign}(t, n, w) \cdot L(t, n, w)
\]

Also, variable \( \text{compat}(t_1,t_2) \) is used to define the compatibility between two tests. It is not uncommon that cores tests share same hardware, e.g., BIST or memory, which makes them incompatible, in this case, we need to enforce a constraint to avoid the incompatible tests to be overlapped with each other. Below is the formulation of compatibility.

\[
\text{Compat}(t_1, t_2) = \begin{cases} 
1, & \text{If the test } t_1 \text{ incompatible with } t_2 \\
0, & \text{compatible}
\end{cases}
\]

\[
\forall t_1, t_2 \in T: \eta(t_1, t_2) + \eta(t_2, t_1) \geq \text{Compat}(t_1, t_2)
\]

In this work, we adopt the constant power model, which assumes that the power of a test will not increase after it starts. Therefore, it is not necessary to keep checking the power and TAM bounds of each core from the start time to end time. A variable \( \pi \) is defined to assist in checking of TAM and power constraint when the test begins. With this \( \pi(t_1,t_2) \) variable, a relation between the start times of tests \( t_1 \) and \( t_2 \) is determined. Next, we have a new variable \( p(t_1,t_2) \), which equals the power of \( t_2 \) when \( t_1 \) starts. If \( t_1 \) and \( t_2 \) are not overlapped then \( p(t_1,t_2) \) must be zero. It is the same result if \( t_1 \) starts earlier than \( t_2 \). However, if \( t_1 \) and \( t_2 \) are overlapped, and \( t_2 \) starts earlier or at the same time as \( t_1 \) starts, then \( p(t_1,t_2) \) equals to the power of \( t_2 \). Below, \( P_{t,n} \) identifies the power of test ‘t’ when it is assigned the nth voltage/frequency pair. It is required that power consumption of any time compatible tests should be within \( P_{\text{bound}} \) at any time instances. TAM width does not affect the power consumption of test and is therefore ignored in power calculation. \( \lambda_p \) corresponds to the maximal power budget in the scheduling, e.g., all tests are overlapped and operate at highest voltage and frequency pairs.

\[
\pi(t_1, t_2) = \begin{cases} 
1, & \text{If test } t_1 \text{ starts earlier or when } t_2 \text{ starts} \\
0, & \text{Otherwise}
\end{cases}
\]

\[
\forall t_1, t_2 \in T: S(t_1) \leq S(t_2) + (1 - \pi(t_1, t_2)) \cdot \lambda_i
\]

\[
\forall t_1, t_2 \in T: S(t_2) > S(t_1) - \pi(t_1, t_2) \cdot \lambda_i
\]

\[
\forall t \in T: \pi(t, t) = 1
\]

\[
\forall t_1 \in T: P(t_1) + \sum_{t \neq t_1} P(t_1,t_2) \leq P_{\text{bound}}
\]

\[
\forall t_1, t_2 \in T: P(t_1,t_2) \leq \text{Overlap}(t_1,t_2) \cdot \lambda_p
\]

\[
\forall t_1, t_2 \in T: P(t_1,t_2) \leq \pi(t_2,t_1) \cdot \lambda_p
\]

\[
\forall t_1, t_2 \in T: P(t_1,t_2) \geq \sum_{n \in N} \sum_{w \in C} \text{Assign}(t_2,n,w) \cdot P_{t_2,n} - (1 - \text{Overlap}(t_1,t_2)) \cdot \lambda_p - (1 - \pi(t_2,t_1)) \cdot \lambda_p
\]

As we discussed earlier, the assignment of TAM width allows flexibility on modifying internal scan chains of cores and the TAM layout arrangement. With these formulations, optimal TAM width assignment is found to benefit the test application time. SoC designers could therefore modify internal scan chains within cores and the TAM layout arrangement accordingly. In the following formulation, a TAM width constraint can be enforced in a similar fashion as power constraint, Below, \( W(t_1,t_2) \) refers to the TAM width
of \( t_2 \) when \( t_1 \) starts, and the constant \( W_{l, w} \) refers to the number of TAM width assigned to \( t_2 \). Below, \( \lambda_w \) is the memory of TAM at any time instant, e.g., sum of all tests using their maximum number of wires:

\[
\forall t_1 \in T: W(t_1) + \sum_{t_2 \in T} W(t_1, t_2) < W^\text{Bound}
\]

\[
\forall t_1, t_2 \in T: W(t_1, t_2) \leq \text{Overlap}(t_1, t_2) \lambda_w
\]

\[
\forall t_1, t_2 \in T: W(t_1, t_2) \leq \pi(t_2, t_1) \lambda_w
\]

\[
\forall t_1, t_2 \in T: W(t_1, t_2) \geq \sum_{w \in C} \sum_{n \in N} \text{Assign}(t_2, n, w) W_{l, w}

- (1 - \text{Overlap}(t_1, t_2)) \lambda_w - (1 - \pi(t_2, t_1)) \lambda_w
\]

In many designs, it is not uncommon that different cores share the same voltage island. However, the voltage island constraint and the above pre-defined hardware constraints are often ignored or only partly addressed in the previous work. Voltage island constraint requires that cores reside in the same voltage island, if scheduled concurrently, and must operate at the same voltage. To enforce this constraint, we have,

\[
\forall t_1, t_2 \in T, \forall n \in N: \sum_{w \in C} \text{Assign}(t_1, n, w) \geq \sum_{w \in C} \text{Assign}(t_2, n, w) - (1 - \text{overlap}(t_1, t_2))
\]

\[
\forall t_1, t_2 \in T, \forall n \in N: \sum_{w \in C} \text{Assign}(t_1, n, w) \geq \sum_{w \in C} \text{Assign}(t_2, n, w) - (1 - \text{overlap}(t_1, t_2))
\]

V. SESSION-BASED TEST SCHEDULING

As stated earlier, session-based scheduling may fail to find the best solution due to its restricted nature. However, it has advantages over session-less scheduling of lower cost and the ease of implementation. Therefore, a session-based algorithm is also being given for SoC test scheduling and results are compared with those of session-less scheduling. This is the first work to integrate the TAM design and test scheduling in the session-based MILP formulation. The basic methodology of session-based formulation is, however, borrowed from [11], [14]. Additional hardware and power constraints are included in this formulation, which are often ignored or partly addressed by previous work. In this presented session-based test scheduling, pre-defined hardware constraint and power budget are included.

The present objective is same as that for session-less scheduling, therefore, we define a decision variable \( t_{\text{finish}} \) to identify the finish time of test scheduling:

\[
t_{\text{finish}} \geq F(s)
\]

Unlike the session-less scheduling, session-based scheduling divides tests into sessions. Therefore, tests are considered as tests in sessions instead of individual tests. For each session, the finish time \( F(s) \) would be the sum of its start time \( S(s) \) and test time \( L(s) \). The test time of a session \( L(S) \) equals that of the longest test in that session. The decision variable \( \text{final}(s, t, n, w) \) specifies that for each session, and for each test assigned in this session, which voltage/frequency pair \( (n \in N) \) is chosen, and which TAM width option \( (w \in C) \) is assigned to this test. \( L_t, n, w \) specifies the test time of a test which operates at nth voltage/frequency pairs and assigned with TAM choice index \( w \).

\[
\forall s \in S, F(s) = S(s) + L(s)
\]

\[
\forall s \in S, \forall t \in T, \forall n \in N, \forall w \in C:

L(s) \geq L_{t, n, w} \text{ final}(s, t, n, w)
\]

\[
S(1) = 0
\]

The run time of each test depends on two variables: the voltage/frequency pair, and the number of TAM width. In this work, there are four choices of voltage/frequency pairs and four choices for TAM width assignment, e.g., \( n \in N = \{1, 2, 3, 4\}, w \in C = \{1, 2, 3, 4\} \). However, some SoC benchmarks are large so the TAM width choice is reduced to save the scheduling time. It is upon users choice to modify the number of choices for voltage/frequency pairs and TAM width.

\[
\text{final}(s, t, n, w) = \begin{cases} 1, & \text{If a session is assigned with a test, which operates at nth v/f pair and} \\ 0, & \text{otherwise} \end{cases}
\]

\[
\forall t \in T, \sum_{w \in C} \sum_{n \in N} \text{final}(s, t, n, w) = 1
\]

It is not uncommon that some core tests may share the same hardware, e.g., BIST or memory), which makes them time incompatible. In this case, we need to enforce a constraint to avoid the incompatible tests to be overlapped with each other. Below is the constraint formulation variable \( \text{comp}(t_1, t_2) \) that specifies the compatibility of tests [1]:

\[
\text{Comp}(t_1, t_2) = \begin{cases} 1, & \text{If the test } t_1 \text{ incompatible} \\ 0, & \text{compatible} \end{cases}
\]

\[
\forall t_1, t_2 \in T, \forall s \in S: \text{Assign}(s, t_1)

+ \text{Assign}(s, t_2) \leq 2 - \text{Comp}(t_1, t_2)
\]

In the session-based scheduling, a constant power model is assumed as described for the session-less case of the previous section. \( P_{\text{Bound}} \) and \( W_{\text{Bound}} \) represent the power budget and the pre-defined total SoC TAM:

\[
\forall s \in S: \sum_{w \in C} \sum_{n \in N} P_{w, n} \text{ final}(s, t, n, w) \leq P_{\text{Bound}}
\]

\[
\forall s \in S: \sum_{w \in C} \sum_{n \in N} W_{l, w} \text{ final}(s, t, n, w) \leq W_{\text{Bound}}
\]
VI. EXPERIMENTAL SETUP

The proposed session-less and session-based MILPs were applied to five ITC’02 benchmarks [2]. Vector by vector power consumption, and the test application time (TAT) of individual cores in each SoC at nominal voltage (1V) and clock frequency for ITC’02 benchmark is provided by Sheshadri et al. [5]. In that work, different cores have different ranges of operation based on the longest test of the core. In this paper, DVFS is done by setting the voltage first, and then tuning the frequency by the alpha-power law [10]:

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

Frequencies corresponding to these voltages are the highest possible operating frequencies for corresponding voltages. Voltages are fine grained and the range from 1.1V to 0.6V. In the present work, we assume that threshold voltage \( V_{TH} = 0.5V \) and \( \alpha = 1 \).

VII. RESULTS

Experiments contain five steps. The TAT Tables 2 through 5 are for session-less scheduling with DVFS and the proposed TAM design, using under pre-defined hardware constraint, power budget and voltage island constraint.

First step compares the TAT reduction between proposed scheduling with DVFS N and TAM N. Note that TAM Y and TAM N refer to techniques with the TAM design of this paper and without the TAM design, respectively. The same naming schemes apply to DVFS N, DVFS Y. Second step compares the TAT variation between the proposed scheduling with DVFS N and TAM Y. Second step compares the TAT variation between proposed scheduling with DVFS N and TAM Y. Third step compares the TAT variation between proposed scheduling with DVFS Y and TAM N. The fourth step compares the TAT reduction by the proposed scheduling over the work of a referenced paper [5]. The final step compares the test scheduling results of the two test scheduling strategies (session-based scheduling and session-less) of this work. The reference work [5] uses DVFS technique to schedule the test only under power constraint. Hardware compatibility, voltage island and TAM bound are not considered in that work. This comparison is used to demonstrate that even when the proposed work in this paper has more constraints, the TAM allocation design together with DVFS could further reduce the test time.

<table>
<thead>
<tr>
<th>Benchmark name</th>
<th>Number of Cores</th>
<th>Power budget (mw)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a586710</td>
<td>7</td>
<td>800</td>
</tr>
<tr>
<td>h953</td>
<td>8</td>
<td>800</td>
</tr>
<tr>
<td>d695</td>
<td>10</td>
<td>400</td>
</tr>
<tr>
<td>g1023</td>
<td>14</td>
<td>400</td>
</tr>
<tr>
<td>p34392</td>
<td>19</td>
<td>400</td>
</tr>
</tbody>
</table>

As we see from Table 2, the proposed TAT achieves reduction in the range 50.9% to 69%, while leaving the TAM bound of each SoC unchanged. TAM allocation is able to achieve optimal test application time by assigning TAM width to cores based on their needs. For instance, for a specific module tested in a SoC, if it requires a very long time to complete its execution, it is highly possible that the SoC scheduling time will be controlled by this single core test time. In this case, linear programming would assign more TAM width to this single test to reduce its test time, as long as the TAM bounds are not violated, and the SoC test time would benefit from this assignment.

<table>
<thead>
<tr>
<th>Benchmark name</th>
<th>Proposed TAT</th>
<th>DVFS N TAM N</th>
</tr>
</thead>
<tbody>
<tr>
<td>a586710</td>
<td>6918324</td>
<td>50.9%</td>
</tr>
<tr>
<td>h953</td>
<td>54654.32</td>
<td>55.4%</td>
</tr>
<tr>
<td>d695</td>
<td>6117.78</td>
<td>69.1%</td>
</tr>
<tr>
<td>g1023</td>
<td>905716</td>
<td>68%</td>
</tr>
<tr>
<td>p34392</td>
<td>298263.6</td>
<td>57%</td>
</tr>
</tbody>
</table>

In Table 3, when only TAM is used to boost the test time, the result is only within 10% variance of the proposed optimal TAM, which signifies the effectiveness of proposed TAM design on the reduction of TAT.

As we see from Table 4, when only DVFS is used, the variation in test application time is huge for some benchmarks, which again signifies the importance of the proposed TAM design. The reason for benchmarks that do not have much variation is that they have less available room for TAM width allocation. However, overall we can conclude that TAM width allocation is more effective than DVFS on the reduction of TAT when we compare Table 3 to Table 4, because the percentage variation of the two techniques favors TAM scheduling.

The results corresponding to referenced work [5] were obtained with no pre-defined hardware constraints, no restrictions of cores that share the same voltage island, and no constraint on TAM bounds. In terms of the technology being used, the reference work incorporates DVFS in
scheduling. As we observe from Table 5, even though these hardware and power constraints are ignored by the reference work, the proposed test scheduling method still gets 8-12% reduction on the test time, which shows the effectiveness of the proposed TAM design in TAT reduction.

Another contribution of this work is that it is perhaps by far the most complete work to model SoC testing under hardware and power constraints, which are addressed individually in the previous work but not addressed together all at once.

Although the present work has advantages as described, future work could incorporate voltage dependent tests into consideration. For instance, some faults are voltage/frequency dependent and are only testable within certain voltage/frequency ranges. Moreover, SoC model presented in this work has no hierarchy. Besides, certain flexibility may not be allowed in a hard core of SoC. Further improvements are left for the future work.

### REFERENCES


