Power-Aware SoC Test Planning for Effective Utilization of Port-Scalable Testers

ANUJA SEHGAL, SUDARSHAN BAHUKUDUMBI, and
KRISHNENDU CHAKRABARTY
Duke University

Many system-on-chip (SoC) integrated circuits contain embedded cores with different scan frequencies. To better meet the test requirements for such heterogeneous SoCs, leading tester companies have recently introduced port-scalable testers, which can simultaneously drive groups of channels at different data rates. However, the number of tester channels available for scan testing is limited; therefore, a higher shift frequency can increase the test time for a core if the resulting test access architecture reduces the bit-width used to access it. We present a scalable test planning technique that exploits port scalability of testers to reduce SoC test time. We compare the proposed heuristic optimization method to two baseline methods based on prior works that use a single scan data rate for all embedded cores. We also propose a power-aware test planning technique to effectively utilize port-scalable testers under constraints of test power consumption. Experimental results are presented for power-aware test scheduling to illustrate the impact of power constraints on overall test time.

Categories and Subject Descriptors: B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms: Algorithms, Design, Reliability

Additional Key Words and Phrases: SoC test, port-scalable testers, test access architecture, integer linear programming

ACM Reference Format:

Sehgal, A., Bahukudumbi, S., and Chakrabarty, K. 2008. Power-aware SoC test planning for effective utilization of port-scalable testers. ACM Trans. Des. Autom. Electron. Syst. 13, 3, Article 53 (July 2008), 19 pages, DOI = 10.1145/1367045.1367062 http://doi.acm.org/10.1145/1367045.1367062

A. Sehgal is with the NVIDIA Corporation and worked on this article as a graduate student at Duke University, Durham, NC 27708.
This research was supported in part by the National Science Foundation under grant CCR-0204077.
A shorter version of this article was published in *Proceedings of the IEEE International Conference on Computer-Aided Design*, 2005, pp. 88–93.
Authors' addresses: A. Sehgal, NVIDIA Corporation, 2701 San Tomas Expressway, Santa Clara, CA 95050; email: asehgal@nvidia.com; S. Bahukudumbi, K. Chakrabarty, Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708; email: {spb, krish}@ee.duke.edu.
Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.
© 2008 ACM 1084-4309/2008/07-ART53 $5.00 DOI 10.1145/1367045.1367062 http://doi.acm.org/10.1145/1367045.1367062

1. INTRODUCTION

Recent advances in technology have led to a tremendous increase in the complexity of system-on-chip (SoC) integrated circuits. Today's heterogeneous SoCs consist of embedded cores that not only operate in multiple clock domains [Goel et al. 2004; Lin and Thompson 2003; Schmid and Knablein 1999], but also (due to differences in performance levels, design styles, and scan insertion methods) differ in their maximum scan clock frequencies [Vranken et al. 2003]. The difference in scan clock frequencies between embedded cores can also arise due to the integration of various cores derived from different, older-generation SoCs into a single, current-generation SoC [Chickermane et al. 2001]. The test time for such SoCs can be reduced by testing the embedded cores at data-transfer rates that match their maximum scan frequencies.

To test embedded cores at different scan frequencies, the test data needs to be simultaneously transported at multiple data rates on the tester channels. In order to meet this requirement, automatic test equipment (ATE) vendors have introduced a new class of testers that can simultaneously drive tester channels at different data rates [Dorsch et al. 2002]. Examples of such ATEs include the Agilent 93000 series [Agilent Technologies 2002; Agilent 93000 2008] and the Tiger system from Teradyne [Teradyne Technologies 2008], which are based on port scalability and test processor-per-pin architecture. Port scalability allows every port of a tester to be configured at a desired data rate, where each port typically consists of multiple channels.

Modular testing of embedded cores offers a promising test solution for SoCs [Goel et al. 2004; Zorian et al. 1999]. It also lends itself well to the scenario of SoC testing using multiple scan data rates. It involves the isolation of an embedded core from surrounding logic using a test wrapper, and the design of a test access mechanism (TAM) to deliver test data from the I/O pins of the SoC. In many test access architectures, the SoC-level TAM wires are divided into fixed-width TAM partitions [Goel and Marinissen 2002; Huang et al. 2002; Iyengar et al. 2003b; Larsson and Peng 2002]. In the scenario being considered here, all wires of a TAM partition belong to the same ATE port, which can be configured for a predefined scan data rate. TAM optimization methods published in the literature do not handle the general problem of designing TAM architectures that are driven by port-scalable ATEs.

The problem of designing a TAM architecture to minimize the SoC test time has been shown in the literature to be $\mathcal{NP}$-hard [Iyengar et al. 2002]. Therefore, efficient heuristic techniques have been developed for SoC test planning and TAM optimization [Goel and Marinissen 2002; Gonciari et al. 2003; Huang et al. 2002; Iyengar et al. 2003b; Larsson and Peng 2002; Zhao and Upadhyaya 2003; Yoneda et al. 2007; Yu et al. 2007; Xu and Nicolici 2005]. However, it is assumed in all these methods that at any instant in time, the ATE provides test stimuli to the SoC at a single data rate. As a result, existing optimization techniques cannot readily exploit the availability of simultaneous multiple data-transfer rates from the ATE to the SoC. In this work, we focus on the problem of designing an optimized TAM architecture that can benefit from the availability of port scalability in ATEs. As in Goel and Marinissen [2002], Iyengar et al.

[2003b], and Larsson and Peng [2002], we base our TAM design on a test bus model.

The testing of embedded cores with multiple scan data rates was recently addressed in Xu and Nicolici [2004a, 2004b], Sehgal et al. [2004], and Sehgal and Chakrabarty [2007]. The key idea in Xu and Nicolici [2004a] is to use bandwidth matching and to determine appropriate scan frequencies for the TAM partitions to reduce test time; however, no limits are set on the maximum scan frequencies of the cores. In Sehgal and Chakrabarty [2007], the number of data rates for the available ATE channels is set to two. The preceding assumptions are too restrictive in practice. In this article, we consider a more general scenario in which cores with different scan frequencies can be driven by ATE channels operating in a data-rate range given by the range of scan frequencies for the cores.

We optimize the TAM architecture for a set of cores with different maximum scan data rates. We compare this approach to a baseline case in which all cores are tested at their maximum scan frequencies. We use an iterative descent procedure that minimizes the testing time by jointly optimizing the widths of TAM partitions, TAM frequencies, and assignment of cores to TAM partitions. We also present a solution to this problem based on integer linear programming (ILP). Although ILP yields optimal results, it is computationally infeasible for large problem instances. Nevertheless, the ILP model can be used to evaluate the heuristic for small problem instances. We derive a lower bound on the SoC testing time and list these bounds for several ITC'02 SoC test benchmarks [Marinissen et al. 2002]. We also present experimental results for several ITC'02 benchmark SoCs.

While the testing of multiple cores in parallel in a core-based SoC results in test schedules with low test times, the concurrent testing of these cores results in increased power consumption during test application. The permissible power envelope is often exceeded when power constraints are not considered during test scheduling [Chou et al. 1997]; this can lead to thermal runaway, or cause severe irreparable damage to the SoC. Testing an SoC can lead to extremely high switching activity, more than when the circuit is in its functional mode [Larsson and Peng 2001; Zhao and Upadhyaya 2003]. It is therefore important to consider power constraints while designing test schedules for embedded cores with multiple scan data rates. We therefore formulate a TAM design and test scheduling problem for cores with different scan frequencies, and we extend the problem formulation to include constraints that are placed on the test power.

The rest of this article is organized as follows. In Section 2, we define the test planning problem that exploits the availability of port-scalable ATEs. We develop an integer linear programming model for this problem and derive a lower bound on the SoC testing time. In Section 3, we present a scalable heuristic approach to solve this problem. Experimental results for several ITC'02 benchmark SoCs are also presented. Section 4 describes a power-aware heuristic scheduling method to solve the test planning problem. Experimental results for the ITC'02 SoC test benchmarks are presented to illustrate the impact of

power constraints on the test planning problem. Finally, we present conclusions and directions for future work in Section 5.

2. TAM ARCHITECTURE OPTIMIZATION

In this section, we formulate the TAM optimization problem when port-scalable testers are used. We develop an ILP model to solve this problem and derive a geometric lower bound on the test time.

Problem $P_{port-scalable}$. Given the test-data parameters for N embedded cores in an SoC, the maximum scan frequency $f_i^\star$ for each core i ($1 \leq i \leq N$) and the SoC-level TAM width W determine: (i) the number of TAM partitions B; (ii) for each TAM partition j, the width w_j and the scan frequency f_j, $1 \leq j \leq B$; and (iii) the assignment of cores to TAM partitions. The aforesaid assignment of cores must be such that: (a) the frequency of each TAM partition does not exceed the maximum frequency of any core assigned to this TAM partition; (b) the sum of widths of TAM partitions does not exceed the total TAM width W; and (c) the overall test time of the SoC is minimized.

The test set parameters for each core include the number of primary inputs, primary outputs, bidirectional I/Os, test patterns, scan chains, and scan-chain lengths. The cores are assumed hard, that is, the number and length of scan chains are fixed prior to test planning. These parameters are used to design a wrapper for the cores. The Design_Wrapper algorithm from Iyengar et al. [2002] is used to design a wrapper and determine the testing time for a core for a given TAM width. Note that if the scan frequencies of all cores are equal, $P_{\text{port-scalable}}$ is equivalent to the original $\mathcal{NP}$-hard TAM-design problem described in Iyengar et al. [2002]. Hence $P_{\text{port-scalable}}$ is at least $\mathcal{NP}$-hard.

The testing time of core i on a TAM partition of width w_j is expressed as $T_i(w_j) = \lceil(\max\{si_i, so_i\} \cdot p_i + \min\{si_i, so_i\})/w_j\rceil + p_i$, where si_i, so_i, and p_i are the maximum scan-in time, maximum scan-out time, and the number of test patterns for core i, respectively [Marinissen et al. 1998]. The testing time is expressed in units of clock cycles.

The test time $T_i(w_j, f)$ for core i at frequency f on a TAM partition of width w_j is defined as $T_i(w_j, f) = T_i(w_j)/f$, where the testing time is expressed in units of μs if f is given in MHz. The overall test time of an SoC is the maximum of the test time over all TAM partitions. Let $x_{ij} = 1$, if core i is assigned to TAM j, otherwise $x_{ij} = 0$. The problem $P_{\text{port-scalable}}$ can now be stated as follows.

Minimize $T = \max_j\{\sum_{i=1}^{N} \frac{T_i(w_j)\cdot x_{ij}}{f_j}\}$ **subject to** the following.

(1) $\sum_j^B x_{ij} = 1$, $1 \leq i \leq N$, namely, every core is connected to only one TAM partition;
(2) $\sum_j^B w_j = W$, namely, the sum of widths of TAM partitions does not exceed W;
(3) $f_j = \min_i\{\{f_i^\star \cdot x_{ij}\} \setminus \{0\}\}$, where $\{A \setminus \{0\}\}$ denotes the set difference between A and $\{0\}$; and
(4) $w_j \leq w_{max}$, $1 \leq j \leq B$, where w_{max} is a user-defined limit on the size of a TAM partition.

Each TAM partition j has w_j wires that are connected to w_j tester channels belonging to the same ATE port. This port is configured to operate at frequency f_j. There is an upper limit on the number of channels that can be included in an ATE port, hence the width of each TAM partition cannot exceed an upper limit of w_{max}. For a typical port-scalable ATE such as the Agilent 93000, $w_{max} = 64$ [Khoche 2001].

Next we consider a special case of $P_{\text{port-scalable}}$ which we refer to as $P^{\star}_{\text{port-scalable}}$. This special case is introduced to optimally solve specific instances of $P_{\text{port-scalable}}$. A solution to $P^{\star}_{\text{port-scalable}}$ addresses the assignment of cores and frequencies to TAM partitions. We assume here that TAM partitions have already been determined. We also present an ILP model for this problem.

The problem $P^{\star}_{\text{port-scalable}}$ can also be shown to be $\mathcal{NP}$-hard using the techniques presented in Chakrabarty [2001]. However, it can be solved exactly for small problem instances using an ILP model. The solution $P^{\star}_{\text{port-scalable}}$ can be used to optimally determine the assignment of cores and frequencies to TAM partitions. Let f_{ij} denote the frequency at which core i is tested if it is assigned to TAM partition j. Let $\tau_{ij} = 1/f_{ij}$ denote the corresponding time period. Let $\tau_i^{\star} = 1/f_i^{\star}$ be the minimum possible period of the scan clock for core i. A mathematical programming model for $P^{\star}_{\text{port-scalable}}$ can be derived as follows.

Objective: Min. $\mathcal{T} = \max_j\{T_i(w_j) \cdot x_{ij} \cdot \tau_{ij}\}$ **subject to** the following.

1. $\sum_j^B w_j = W$; 2. $x_{ij} \cdot f_{ij} \leq f_i^{\star}$; 3. $f_{1j} = f_{2j} = \cdots = f_{Nj}$; 4. $\sum_j^B x_{ij} = 1$; 5. $w_j \leq w_{max}$.

Note that the preceding objective function is nonlinear due to the product term $x_{ij} \cdot \tau_{ij}$. We linearize it by replacing it with a new integer variable y_{ij} ($y_{ij} \geq 0$) and adding the following three constraints for every such product term: (i) $y_{ij} - T_{max} \leq 0$, where T_{max} is an upper bound on the minimum time-period limit for all cores; (ii) $-\tau_{ij} + y_{ij} \leq 0$; and (iii) $\tau_{ij} - y_{ij} + T_{max} \cdot x_{ij} \leq T_{max}$.

The new variables and constraints yield the following ILP model.

Objective: Minimize $\mathcal{T} = \max_j\{T_i(w_j) \cdot y_{ij}\}$ **subject to** the following.

1. $\sum_j^B w_j = W$;
2. $x_{ij} \cdot \tau_i^{\star} - \tau_{ij} \leq 0$;
3. $f_{1j} = f_{2j} = \cdots = f_{Nj}$, $1 \leq j \leq B$;
4. $y_{ij} - T_{max} \leq 0$;
5. $-\tau_{ij} + y_{ij} \leq 0$; and
6. $\tau_{ij} - y_{ij} + T_{max} \cdot x_{ij} \leq T_{max}$.

We use this ILP model with the TAM-width partitioning approach from Iyengar et al. [2002] to solve $P_{port-scalable}$. The P_{PAW}_Enumerate procedure described in Iyengar et al. [2002] enumerates unique TAM partitions for given values of B and W. In this work, we use the P_{PAW}_Enumerate procedure with the ILP model for $P^{\star}_{\text{port-scalable}}$.

We next derive a lower bound on the SoC testing time, using a geometric argument. The testing times for a core in the SoC can be represented using a set of rectangles. A set $\mathbf{R}_i$ of rectangles for core i $(1 \leq i \leq N)$ is determined such that the height and width of each rectangle correspond to a TAM width and the corresponding test-application time for the core, respectively. The TAM optimization problem can now be formulated in terms of rectangle packing as follows: Select one rectangle from each set $\mathbf{R}_i$, $1 \leq i \leq N$, and pack the selected rectangles into a bin of fixed height, such that no two rectangles overlap and such that the width to which the bin is filled is minimized. Even though this problem statement addresses a flexible-width TAM architecture as in Iyengar et al. [2003b], it can be used to derive a lower bound.

The area of a bin, with the width representing total testing time T and the height representing total TAM width W, is given by $T \times W$. Each core yields a set of rectangles of different areas. Let $\mathcal{R}_i^{min} \subseteq \mathbf{R}_i$ be the area of the minimum-area rectangle for core i. Let the area of a rectangle representing core i being tested at TAM width w be given by $R_i(w, f) = T_i(w, f) \times w$, where $f \leq f_i^\star$. It follows that $\mathcal{R}_i{}^{min} = \min_i\{R_i(w, f)\}, 1 \leq w \leq W$, and $0 < f \leq f_i^\star$. We next show that the minimum-area rectangle for each core is a rectangle of height 1 and width of $T_i(1, f_i^\star)$, namely, $\mathcal{R}_i{}^{min} = R_i(1, f_i^\star)$.

A lower bound on the testing time $T_i(w, f)$ for core i on a TAM partition of width w and frequency f can be expressed as

$$T_i(w) \geq (\lceil(\max(si_i, so_i) \cdot p_i + \min(si_i, so_i))/w\rceil + p_i) \times 1/f.$$

The numerator of the first term inside the parenthesis on the righthand side of the previous inequality represents the total test-data volume to be applied to the core; it is independent of the number of TAM wires used to apply the test data or the scan frequency of the TAM wires. Hence for any core i, $R_i(w, f) = (\lceil v/w\rceil + p_i) \times w/f$ and $R_i(1, f_i^\star) = (v + p_i)/f_i^\star$, where v is the total test-data volume for the core. Comparing the expressions for $R_i(w, f)$ and $R_i(1, f_i^\star)$ for $w > 1$ and $f < f_i^\star$, we see that $(\lceil v/w\rceil) \times (1/f) \times w > v/f_i^\star$ and $p_i \times w/f > p_i/f_i^\star$. Thus $R_i(w, f) > R_i(1, f_i^\star)$, $\forall w > 1$, and $f < f_i^\star$. We also note that the minimum-area rectangles for the cores might not fill the bin of area $T \times W$ perfectly, owing to variation in the sizes of rectangles. As a result, there may be some unfilled space in the bin. Let us denote the total area of the unfilled space in the bin by Δ, where $\Delta \geq 0$. Now, we know that the total area of the bin cannot be less than the sum of minimum-area rectangles of all cores in the SoC and the sum of all the unfilled space in the bin. Thus

$$\begin{aligned} T \times W &\geq R_1(1, f_1^\star) + R_2(1, f_2^\star) + \cdots + R_N(1, f_N^\star) + \Delta \\ &\geq R_1(1, f_1^\star) + R_2(1, f_2^\star) + \cdots + R_N(1, f_N^\star), \end{aligned}$$

which implies that

$$T \geq \sum_{i=1}^{N} R_i(1, f_i^\star)/W. \tag{1}$$

Let the lower bound obtained from Eq. (1) be denoted by LB_1. We obtain another lower bound LB_2 from Chakrabarty [2001] as follows: $LB_2 = \max_i T_i(w, f_i^\star)$,

```
Procedure. Test_Optimizer
1. for B = 1 to B_max do
2.     TAM_Initialize();
3.     Assign_Core();
4.     descent := true; best_time := testing_time;
5.      while {descent} do
6.       testing_time_old:= testing_time;
7.       Split_TAMs();
8.       Core_shuffle();
9.       Redistribute_TAM();
10.      Merge_TAMs();
11.      if (testing_time < testing_time_old)
12.        then descent:=false;
13.            testing_time:=testing_time_old;
14.        else descent:=true; od
15.      if best_time > testing_time
16.      then do best_time:= testing_time;
17.              best_B:= B; od, od, od
```

Fig. 1. Defect estimation: placement of a core with respect to blocks.

$1 \leq i \leq N$. Specifically, LB_1 is more accurate for smaller TAM widths. However, for larger values of W, the LB_2 is tighter. Hence, the overall lower bound LB_T is determined as $\max\{LB_1, LB_2\}$.

3. OPTIMIZATION PROCEDURES

In this section, we explain the heuristic algorithm used to solve $P_{\text{port-scalable}}$. The algorithm starts with an initial solution and then improves it in an iterative manner, using four iterative descent procedures (IDPs). An IDP reduces (descends) the initial cost, which refers here to the SoC-test time, by reducing the cost with each iteration. It continues to iterate until the cost increases, in which case it exits and outputs the solution from the previous iteration as the final solution. Typically, every IDP has an abort condition to prevent an infinite number of iterations.

The main steps of the algorithm, as shown in Figure 1, are briefly outlined as follows.

1. In procedure *TAM_Initialize*, an initial partition of the total TAM width and the frequencies of TAM partitions are determined based on the scan frequencies for the cores.
2. In procedure *Assign_Core*, a modified best-fit decreasing (BFD) algorithm is used to make the initial core assignments to TAM partitions.
3. Four iterative descent procedures are nested together in a loop; these are executed in an iterative manner, as long as the testing time decreases with each iteration. The four IDPs jointly optimize the TAM-partition widths, TAM-partition frequencies, and the assignment of cores to TAM partitions.

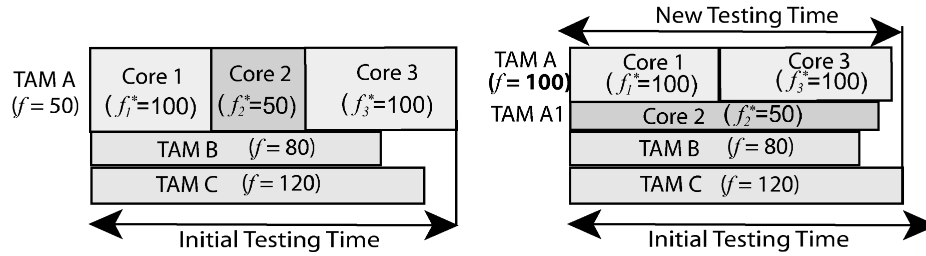

Fig. 2. Illustration of procedure *Split_TAMs*.

3.1 Initial Solution

The *TAM_Initialize* procedure creates B partitions, where B is varied from 1 to B_{max}. The first $B - 1$ partitions have width $\lceil W/B \rceil$, and partition B has width $W - (B - 1) \times \lceil W/B \rceil$. These partitions are assigned TAM frequencies as follows: (i) a list of cores, sorted by their maximum scan frequencies in ascending order, is created; (ii) for every TAM partition j, the frequency of the core with index $((j - 1) \times \lfloor N/B \rfloor + 1)$ in the sorted list is selected; for example, TAM partition 1 is assigned the frequency of that core with index 1. These two steps ensure that the TAM-partition frequencies are evenly distributed between the frequency of the core with lowest maximum scan frequency and that with the highest. It also results in n TAM partitions having the same scan frequency if more than $n \times \lfloor N/B \rfloor + 1$ cores have the same maximum scan frequency.

Next, procedure *Assign_core* assigns each core to one of the B TAM partitions such that each core is tested at a frequency lower than or equal to its maximum scan frequency. The steps of this procedure are as follows. While not all cores have been assigned to a TAM partition: (i) find a TAM partition $TPART_{\min}$ with the lowest testing time among all TAM partitions; and (ii) from all cores with maximum scan frequencies greater than the frequency of $TPART_{\min}$, find the core with the maximum test time on TAM partition $TPART_{\min}$. There can be instances in which TAM partition $TPART_{\min}$ has a frequency higher than the maximum scan frequency of all cores not yet assigned to TAM partitions. In such cases, the next TAM partition with minimum testing time is determined. The TAM architecture obtained from *TAM_Initialize* and *Assign_core* is now used as an initial solution for the IDPs.

3.2 Iterative Descent Procedures

The *Split_TAMs* IDP optimizes the TAM frequencies, TAM widths, and core assignments, based on the initial solution. All TAM partitions have one or more *frequency-bottleneck cores*, which are those cores having the minimum scan frequency among all cores assigned to this TAM partition. The *Split_TAMs* procedure displaces frequency-bottleneck cores from those TAM partitions with maximum testing time, and places them on a separate TAM partition operating at the bottleneck frequency. Figure 2 illustrates the *Split_TAMs* procedure. The main steps of this procedure are as follows.

(1) Identify a TAM partition $TPART_{max}$ that has the maximum testing time. A TAM partition with maximum testing time is also referred to as the bottleneck TAM partition.
(2) Remove frequency-bottleneck cores from $TPART_{max}$. Exit the procedure if all cores on the TAM partition have the same maximum scan frequencies.
(3) Upgrade the scan frequency of $TPART_{max}$ to the frequency of the new frequency-bottleneck cores, and update the testing time of $TPART_{max}$.
(4) Remove a total of δ TAM wires from $TPART_{max}$ in increments of one, until the testing time of $TPART_{max}$ does not exceed the initial testing time of $TPART_{max}$ from step 1. The "slack" TAM wires are removed in this step.
(5) Use all δ TAM wires to test the cores removed from $TPART_{max}$ in step 2. Update the testing time of this newly formed TAM partition, and recompute the maximum testing time of the SoC.
(6) If the testing time has not exceeded the original testing time of $TPART_{max}$ from step 1, return to step 1.

The *Core_Shuffle* IDP jointly optimizes the core assignments and scan frequencies of the TAM partitions by shuffling the core assignments to TAM partitions. (The widths of TAM partitions remain unchanged.) The main steps of the procedure are as follows.

(1) Identify TAM partitions $TPART_{max}$ and $TPART_{min}$ that have the maximum and minimum testing time, respectively.
(2) Identify all cores assigned to $TPART_{max}$ that have a maximum scan frequency greater than the scan frequency of $TPART_{min}$. These cores are "compatible" with TAM partition $TPART_{min}$.
(3) If there are no compatible cores, replace $TPART_{min}$ with a TAM partition with the next-lowest testing time. If $TPART_{min}$ and $TPART_{max}$ point to the same TAM partition, exit the procedure; otherwise repeat step 2.
(4) From the set of compatible cores, select a core that has the maximum testing time on $TPART_{max}$ among those cores that can be assigned to $TPART_{min}$. This choice should not cause the testing time of $TPART_{min}$ to exceed the initial testing time of $TPART_{max}$ from step 1.
(5) Displace the selected core from $TPART_{max}$ and the set of compatible cores. Update TAM-partition testing times.
(6) Repeat steps 4 and 5, until no compatible core can be assigned to $TPART_{min}$, without causing the testing time of $TPART_{min}$ to exceed the initial testing time of the SoC from step 1.
(7) Update the frequency of $TPART_{max}$ to the scan frequency of that core with the minimum scan frequency. If, in step 5, frequency-bottleneck cores are displaced, the scan frequency of $TPART_{max}$ increases. Update testing times for $TPART_{max}$ and $TPART_{min}$.
(8) If the new $TPART_{max}$ and $TPART_{min}$ are different, repeat the *Core_Shuffle* procedure.

The *Redistribute_TAM* IDP optimizes the TAM widths of TAM partitions. It removes slack TAM wires from nonbottleneck TAM partitions, and assigns them to the bottleneck TAM partitions. The main steps of the IDP are as follows.

(1) Identify the bottleneck TAM partition $TPART_{\max}$.
(2) Find a nonbottleneck TAM partition $TPART_{\min}$ which has the minimum testing time.
(3) Remove slack TAM wires from $TPART_{\min}$ and merge them with the bottleneck TAM partition.
(4) Update testing time; if the testing time has reduced over the initial testing time, return to step 1, otherwise exit the procedure.

Thus, the *Redistribute_TAM* IDP continues to remove slack TAM wires from the bottleneck TAM partition, while the testing time of the nonbottleneck TAM partitions does not exceed the initial testing time of the SoC.

The *Merge_TAMs* IDP merges two TAM partitions to reduce the test time of those cores belonging to them by offering a greater bit-width. However, it causes the merged TAM partition to operate at the minimum of the scan frequencies of the two merged TAM partitions. The main steps of this procedure are as follows.

(1) Identify the TAM partition $TPART_{\max}$ with maximum testing time. All the remaining TAM partitions are candidates for merging with $TPART_{\max}$.
(2) For every TAM partition other than $TPART_{\max}$, compute the testing time of the SoC if it were to be merged with $TPART_{\max}$.
(3) Select that TAM partition that results in the highest reduction in testing time of the SoC in step 2. If the merging of every TAM partition in step 2 results in an increase in the initial SoC test time, exit the procedure.
(4) Repeat the previous three steps.

The worst-case time complexity of the overall heuristic procedure is $O(B_{max}^2 N + N \log_2 N)$. Since TAMs are considered in a pairwise manner, the number of TAM-partition choices for core replacement at every step is by B_{max}^2. The parameter N represents the number of candidate cores that can be shuffled between TAM partitions in the worst case. The sort operation performed by the heuristic procedure contributes the term $N \log_2 N$ in the O-notation.

3.3 Experimental Results

We now present experimental results for five ITC'02 benchmark circuits. We compare the results of the heuristic approach to the ILP-based approach, the baseline case, and the derived lower bounds on test time. In the first baseline case, the cores are tested at their maximum scan frequency. In this case, the number of TAM partitions is equal to the number of unique maximum scan frequencies for the embedded cores. The results for the baseline case are obtained using the TAM optimization technique from Iyengar et al. [2003a]. For the second baseline scenario, we consider an SoC with a fixed number of TAM partitions; the assignment of cores to TAM partitions is obtained using methods presented in Iyengar et al. [2003a]. The frequency of a TAM partition

is set to the lowest scan frequency of those cores assigned to this TAM partition. In this article, we compare our methods with SoCs designed with two (*TAM2*) and three (*TAM*3) TAM partitions. We first present results for the ILP-based approach for two benchmark SoCs with a small number, of cores, namely, d695 and a586710.

In the absence of scan-frequency information for cores in the ITC'02 benchmarks, we use a random number generator to obtain the maximum scan frequencies for the cores. The random number generator is used to select frequencies from a set of predefined scan frequencies for each core. For the two smaller SoCs, d695 and a586710, we use two scan frequencies of 40 MHz and 80 MHz. For the three larger SoCs, p22810, p34392, p93791, the random generator chooses frequencies from a larger set of scan frequencies. The ILP-based approach is run only for $B = 2$ and $B = 3$ because the problem size grows exponentially in B.

The results for d695 and a586710 are shown in Table I. The set of maximum scan frequencies for the cores is shown in the last row of the table (element i if the set corresponds to the maximum scan frequency of core i). We denote the test time obtained using the ILP-based approach as T_{ILP}. The test times for the first baseline scenario are denoted as T_b. Further, $T_{b_{TAM2}}$ and $T_{b_{TAM3}}$ represent the test times obtained using the second baseline method for an SoC with two and three TAM partitions, respectively. The lower bounds are obtained using (3) and by assuming that every core is tested at its maximum scan frequency. The ILP-based approach outperforms the heuristic (T) and the baseline scenarios in most cases. However, the runtime for d695 with $B = 3$ ranges from 37 minutes to 192 minutes for W ranging from 16 to 32. It does not reach a solution for TAM widths greater than 40 for $B = 3$. For a586710, the runtime is less than 10 minutes for $B = 3$ for all values of W. The proposed heuristic and the baseline case require less than 10 seconds for all values of $W \leq 64$ for all the ITC'02 benchmarks. The heuristic approach provides an optimal solution, that is, same test time as the ILP-based method, for several cases ($W = 32$ and $W = 64$ for d695, and $W = 56$ for a586710). For other cases, its testing time is no more than 20% higher than the optimum test time. For d695 and a586710, the test time with three TAM partitions for the second baseline is less than that for the proposed method for some values of W. We attribute this to the fact that d695 and a586710 are simple designs. As expected, the proposed method outperforms the baseline methods for more realistic benchmarks (see Table II). For $W = 32$, the second baseline with $B = 3$ leads to lower test time than the ILP method ($B = 2$). This shows that the minimum test time with $B = 2$ is higher than the test time obtained using the baseline method for $B = 3$.

Next we present results for SoCs p34392, p22810, and p93791 in Table II. The results are presented for two frequency ranges of 40 MHz to 200 MHz, and 10 MHz to 50 MHz, respectively. We performed experiments for sets of five scan frequencies and nine scan frequencies in the two frequency ranges. Using the proposed approach, the reduction in testing time over the first baseline case is as high as 52.54% (in one case) for the three large "p" SoCs from Philips. Since there are five (nine) distinct frequencies, the number of TAM partitions is limited to five (nine) in the first baseline case. However, from these experimental results,

Table I. Testing Times for d695 (in μs) and a586710 (in ms)

d695							
W	16	24	32	40	48	56	64
T_{ILP}	811	619	543	448	426	420	420
$\{w_j\}$	{5,11}	{8,16}	{8,24}	{8,32}	{11,37}	{12,44}	{12,52}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
T_b	1022	775	619	545	543	540	448
$\{w_j\}$	{8,8}	{12,12}	{16,16}	{20,20}	{27,21}	{31,25}	{33,31}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
$T_{b_{TAM2}}$	930	733	552	544	514	514	448
$\{w_j\}$	{8,8}	{8,16}	{16,16}	{20,20}	{20,28}	{31,25}	{33,31}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
$T_{b_{TAM3}}$	889	603	513	510	495	493	421
$\{w_j\}$	{4,6,6}	{6,9,9}	{10,11,11}	{12,15,13}	{14,17,17}	{16,20,20}	{17,24,23}
$\{f_j\}$	{80,40,40}	{80,40,40}	{80,80,40}	{80,40,40}	{40,40,80}	{80,40,80}	{80,40,40}
T	929	742	543	535	439	439	420
$\{w_j\}$	{2,5,9}	{4,7,13}	{10,32}	{7,34}	{3,12,33}	{14,42}	{21,43}
$\{f_j\}$	{80,80,40}	{80,80,40}	{80,40}	{40,80}	{80,80,40}	{80,40}	{80,40}
$\Delta T_{\text{ILP}}(\%)$	14.55	19.87	0	19.42	3.05	4.52	0
$\Delta T_b(\%)$	−9.10	−4.26	−12.92	−1.83	−19.15	−18.70	−6.92
$\Delta T_2(\%)$	−0.10	−1.22	−1.63	−1.65	−14.59	−14.59	−6.25
$\Delta T_3(\%)$	4.44	23.05	5.52	4.67	−11.31	−10.95	≈0
$\{f_i^*\}$	{40, 80, 40, 40, 40, 80, 40, 80, 80, 80}						

a586710							
W	16	24	32	40	48	56	64
T_{ILP}	5275	3768	2713	2260	1959	1959	1658
$\{w_j\}$	{2,2,12}	{3,4,17}	{4,4,24}	{5,5,30}	{5,8,35}	{6,9,41}	{7,11,46}
$\{f_j\}$	{40,40,40}	{40,40,40}	{40,40,40}	{40,40,40}	{40,40,40}	{40,40,40}	{40,40,40}
T_b	9626	6518	5093	4178	3466	3107	2754
$\{w_j\}$	{8,8}	{12,12}	{16,16}	{20,20}	{24,24}	{28,28}	{32,32}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
$T_{b_{TAM2}}$	5446	3768	2863	2659	2417	2101	1813
$\{w_j\}$	{4,12}	{7,17}	{9,23}	{19,21}	{21,27}	{29,27}	{30,34}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
$T_{b_{TAM3}}$	5727	3751	2839	2633	2490	2091	1922
$\{w_j\}$	{1,4,11}	{1,5,18}	{1,7,24}	{1,10,29}	{1,14,33}	{3,16,37}	{3,20,41}
$\{f_j\}$	{40,40,40}	{80,40,40}	{80,80,40}	{40,80,40}	{40,40,80}	{40,40,80}	{40,40,80}
T	5804	3973	3250	2544	2341	1959	1798
$\{w_j\}$	{2,14}	{3,21}	{3,29}	{4,36}	{1,3,10,34}	{8,12,36}	{15,49}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{40,40,40,40}	{40,40,40}	{40,40,40}
$\Delta T_{\text{ILP}}(\%)$	10.03	5.44	19.79	12.57	19.50	0	8.45
$\Delta T_b(\%)$	−39.70	−39.04	−36.19	−39.11	−32.46	−36.95	−34.71
$\Delta T_2(\%)$	6.57	5.44	13.51	−4.32	−3.14	−6.75	≈0
$\Delta T_3(\%)$	1.34	5.91	14.47	−3.38	−5.98	−6.31	−6.45
$\{f_i^*\}$	{40, 80, 40, 40, 80, 80, 40}						

$\Delta T_{\text{ILP}} : \frac{T - T_{\text{ILP}}}{T_{\text{ILP}}} \times 100$; $\Delta T_b : \frac{T - T_b}{T_b} \times 100$; $\Delta T_2 : \frac{T - T_{b_{TAM2}}}{T_{b_{TAM2}}} \times 100$; $\Delta T_3 : \frac{T - T_{b_{TAM3}}}{T_{b_{TAM3}}} \times 100$.

we observe that even when the first baseline case has the same number of TAM partitions as the proposed approach, the latter results in lower SoC test times. This implies that the frequencies of TAM partitions and the assignment of cores to TAM partitions have a major impact on the overall test time of the SoC.

Table II. Testing Time (in μs) for p34392, p22810, and p93791

p34392 (five scan frequencies: 40 MHz to 200 MHz)								
W	LB_T	T	T_b	$T_{b_{TAM2}}$	$T_{b_{TAM3}}$	ΔT	ΔT_2	ΔT_3
16	12724	14683	19209	25637	24572	−23.56	−42.72	−40.24
24	8757	12514	13089	22140	17096	−4.33	−43.47	−26.80
32	6774	10806	10976	17992	15185	−1.57	−39.93	−28.88
40	5584	7707	9438	13789	12642	−18.34	−44.10	−39.03
48	4790	7707	9438	13148	12381	−18.34	−41.37	−37.74
56	4538	7707	9425	12904	11688	−18.22	−40.27	−34.06
64	4538	7617	7619	10306	9165	−0.026	−26.09	−16.89
$\{f_i^*\}$ (MHz)	{80, 80, 40, 120, 200, 40, 160, 40, 20, 40, 200, 160, 200, 40, 40, 80, 120, 160, 120}							

p22810 (five scan frequencies: 10 MHz to 50 MHz)								
W	LB_T	T	T_b	$T_{b_{TAM2}}$	$T_{b_{TAM3}}$	ΔT	ΔT_2	ΔT_3
16	13554	22567	29464	24652	22524	−23.40	−8.45	−0.01
24	9267	14894	22372	18713	16418	−33.42	−20.40	−9.28
32	7124	12613	15153	16450	13135	−16.76	−23.25	−3.97
40	5838	12607	14125	15239	12898	−10.74	−17.27	−2.22
48	5148	8307	13367	14236	11644	−37.85	−41.64	−28.65
56	5148	7435	9341	11713	9135	−20.40	−36.52	−18.60
64	5148	7435	9294	11247	8843	−20.00	−33.89	−15.92
$\{f_i^*\}$ values (MHz)	{20, 40, 50, 10, 40, 20, 50, 30, 40, 30, 20, 10, 40, 20, 30, 10, 50, 20, 20, 40, 50, 10, 40, 20, 50, 30, 40}							

p93791 (five scan frequencies: 10 MHz to 50 MHz)								
W	LB_T	T	T_b	$T_{b_{TAM2}}$	$T_{b_{TAM3}}$	ΔT	ΔT_2	ΔT_3
16	86175	110790	183186	175031	172928	−39.52	−36.70	−35.93
24	57727	66186	139457	118673	119884	−52.54	−44.22	−44.79
32	43503	54795	94878	90816	88773	−42.25	−39.66	−38.27
40	34969	48016	72748	65634	63417	−33.99	−26.84	−24.28
48	29279	36395	61192	57942	54531	−40.52	−37.18	−33.25
56	25215	30686	56578	51135	47686	−45.76	−39.99	−35.64
64	22168	34872	47644	44398	42178	−26.80	−21.45	−17.32
$\{f_i^*\}$ values	{10, 30, 20, 50, 30, 20, 40, 10, 50, 40, 20, 50, 30, 40, 30, 20, 10, 10, 40, 20, 30, 30, 40, 20, 50, 50, 30, 20, 10, 10, 40, 50}							

$\Delta T : \frac{T-T_b}{T_b} \times 100$; $\Delta T_2 : \frac{T-T_{b_{TAM2}}}{T_{b_{TAM2}}} \times 100$; $\Delta T_3 : \frac{T-T_{b_{TAM3}}}{T_{b_{TAM3}}} \times 100$.

Similar improvements in test time are observed for the second baseline scenario. It is also observed that for the larger SoCs, there are always several TAM partitions running at shift frequencies higher than the minimum shift frequency.

If multiple scan clock signals are transported to the chip from the tester, part of the total TAM width W will need to be allocated for the scan clock signals. Thus, for the case of five frequencies, we also compared the test time for benchmarks for TAM width W to that obtained using a baseline method that tests every core at the minimum scan frequency, but with a TAM width of $W + 4$ bits. We found that even with fewer TAM wires, the proposed approach results in significantly lower SoC test times. For p93791, the improvements ranged from 21% to 39%, and similar improvements in test times were observed for p22810 and p34392.

4. POWER-AWARE TEST PLANNING

The testing of multiple cores in parallel in a core-based SoC results in test schedules with reduced test times. However, the concurrent testing of these cores often results in increased test power. The permissible power envelope is often exceeded when power constraints are not considered during test scheduling; this can cause severe irreparable damage to the SoC. It is well known that scan testing and concurrent testing of embedded cores lead to high switching activity which can be several times higher than that for functional operation [Larsson and Peng 2001]. Since test power is directly proportional to the frequency of test application, the use of port-scalable testers to reduce test time is expected to lead to higher test power.

Test scheduling under power constraints was first presented in Chou et al. [1997]. Power-constrained test scheduling for an SoC was addressed in Zhao and Upadhyaya [2003], Larsson and Peng [2001], and Iyengar and Chakrabarty [2002], and more recently in Su and Wu [2004], Larsson and Peng [2006], and Samii et al. [2006]. Power constraints were also considered in Huang et al. [2002], but only for a flexible-width TAM architecture; the approach presented in Huang et al. [2002] cannot be applied to fixed-width and multifrequency TAMs.

We present a new test planning approach that takes into account the different scan frequencies of individual cores to exploit the port scalability of testers. While the use of port-scalable testers helps to reduce the overall test time of the SoC, operating the different cores of these heterogeneous SoCs at multiple data rates can cause overheating of the device, due to violation of the permissible power envelope P_{max}. The parameter P_{max} refers to the maximum power consumption of the device during test that allows for proper circuit operation. If a core i is assigned to a TAM partition j, then the power consumption of the core during test can be represented as P_{ij}, where $P_{ij} = P_i^* \cdot \lceil \frac{f_{ij}}{f_i^*} \rceil$, and P_i^* is the power consumed by core i when operating at a frequency f_i^*. The dynamic power consumption of a device is directly proportional to the frequency of operation, and the dynamic power is the dominant power component during test [Samii et al. 2006]. We now present a heuristic to solve $P_{PA_port-scalable}$, which is a power-aware version of the problem $P_{port-scalable}$ presented in Section 3.

4.1 Power-Aware $P_{port-scalable}$: $P_{PA_port-scalable}$

In this section, we present a heuristic algorithm to solve $P_{PA_port-scalable}$. The heuristic algorithm determines the cores assigned to TAM partitions and the frequency of TAM partitions. It also determines the order in which the cores are tested on the TAM partitions. The sequence of procedures adopted in this section to solve the problem is built upon procedures developed in Section 3. As in Section 3, the following are the main steps of the algorithm.

—The first step in solving $P_{PA_port-scalable}$ involves determining the initial widths of TAM partitions and the frequencies of TAM partitions. This information is determined based on the maximum scan frequencies for the cores.

—The *PA_Assign_Core* procedure is used to make initial core assignments to the TAM partitions. The power constraints are considered while making these initial core assignments to ensure that the test power never exceeds P_{max}.

—Finally, we explain the iterative descent procedures that iteratively reduce the test time while at the same time monitoring the test power for the SoC. These iterative descent procedures, as in Section 3, jointly optimize the TAM-partition widths, TAM frequencies, and the assignment of cores to TAM partitions.

4.2 Initial Solution: $P_{PA_port-scalable}$

The first step in solving $P_{PA_port-scalable}$ remains the same as the *TAM_Initialize* procedure used in Section 3. The *TAM_Initialize* procedure creates B TAM partitions and assigns frequencies to these TAM partitions.

The next procedure *PA_Assign_Core* assigns each core to one of the B TAM partitions such that each core is tested at a frequency lower than or equal to its maximum scan frequency. The following are the steps adopted in this procedure.

(1) While not all cores have been assigned to a TAM partition, find a TAM partition $TPART_{min}$ with the lowest testing time.
(2) From all cores with maximum scan frequencies greater than the frequency of $TPART_{min}$, find a core C with maximum test time on $TPART_{min}$.
(3) Check whether the constraint on power consumption for the SoC P_{max} is violated due to the addition of core C with a power consumption of $P_C\,(f_{TPART_{min}})$.
(4) If P_{max} is exceeded for the core assignment, remove C from the list of cores with maximum scan frequencies greater than $TPART_{min}$.
(5) Repeat step 2 to determine a core with maximum test time on $TPART_{min}$.

The aforesaid sequence of procedures is repeated until all cores in the SoC have been assigned to a particular TAM partition. Exceptions are handled in the same way as described in Section 3.1.

4.3 Iterative Descent Procedures: $P_{PA_port-scalable}$

We now present the three IDP components used to solve $P_{PA_port-scalable}$.

In the *Split_TAM* procedure, the TAM frequencies, TAM widths, and core assignments based on the initial solution are optimized. The *Split_TAM* displaces the frequency-bottleneck cores from those TAM partitions with maximum testing time, and places them in a separate TAM partition. The power-aware *Split_TAM* IDP is similar to the *Split_TAM* procedure presented in Section 3.2. In the power-aware procedure, we check for satisfiability of the power constraint P_{max} when a frequency-bottleneck core is assigned to a new TAM partition. We exit the procedure if no power-compatible frequency-bottleneck cores are found on $TPART_{max}$.

The *Core_Shuffle* IDP jointly optimizes the core assignments and the scan frequencies of TAM partitions by shuffling the core assignment to TAM partitions. The following are the main steps used in this procedure.

(1) The first three steps in *Core_Shuffle* remain the same as from Section 3.2.
(2) We now select that core, from the set of compatible cores, which has a maximum testing time on $TPART_{max}$ that can be assigned to $TPART_{min}$. The choice of core should not cause the test time to exceed the initial test time of $TPART_{max}$ (from step 1), and the constraint on maximum test power P_{max} should not be violated.
(3) Update the testing time of the TAM partitions once the core has been displaced from $TPART_{max}$.
(4) Steps 4 and 5 are repeated until no compatible core can be assigned to $TPART_{min}$, without causing the testing time of $TPART_{min}$ to exceed the initial testing time of the SoC, and without violating the power constraints.
(5) The final two steps in *Core_Shuffle* are the same as the final two steps (steps 7 and 8) in *Core_Shuffle* of Section 3.2.

The final IDP that we use to solve $P_{PA_port-scalable}$ is the *Redistribute_TAM* IDP. We do not use the *Merge_TAM* and *Redistribute_TAM* IDP from Section 3.2 to solve $P_{PA_port-scalable}$ because these procedures will alter the power characteristics of the TAM and will introduce power violations in the test schedule.

4.4 Experimental Results on Power-Aware Test Planning

We present experimental results for three ITC'02 SoC test benchmark circuits in Table III. We compare the results for the power-aware heuristic approach with those obtained using the heuristic presented in Section 3 to solve $P_{port-scalable}$, and a baseline case where cores are tested at their maximum scan frequencies. We use power information (and the units for the power values) for the cores from Samii et al. [2006]; power data for only the three SoCs considered in this section was presented in Samii et al. [2006]. In Samii et al. [2006], a cycle-accurate power modeling approach was developed for core-based SoCs. For a known TAM width we utilize the data in Samii et al. [2006] to determine the peak power consumption for the core over all clock cycles. We use this value of peak power consumption in our experiments. The value of f_i^* is 10 MHz in all our experiments.

The experimental results show that the test times P_{TA} for $P_{PA_port-scalable}$ are higher than those obtained for $P_{port-scalable}$ (T), but significantly lower than the baseline case (T_b) for most values of W and P_{max}. On average, the test time obtained using $P_{PA_port-scalable}$ is 11.24% lower than the baseline scenario (over all power constraints). The maximum values of power consumption during test obtained using $P_{port-scalable}$ (P_{PS}) and the baseline scenario (P_b) are listed in Table III. It is clear from the values of power consumption that $P_{port-scalable}$ results in higher power consumption than $P_{PA_port-scalable}$; this would result in violation of the permissible power envelope during test. The baseline scenario results in 14.56% higher power consumption than $P_{port-scalable}$ on average, over all benchmark circuits. It also results in higher power consumption than $P_{PA_port-scalable}$ in all cases.

Table III. Testing Times (in μs) under Power Constraints and the Test Power Consumption for d695, p22810, and p93791

d695 (five scan frequencies: 10 MHz to 50 MHz)						
	T_{PA}				Power consumption	
W	$P_{max} = 1600$	$P_{max} = 1800$	T	T_b	P_{PS}	P_b
16	1096	996	929	1022	2093	2745
24	812	768	742	775	2093	2745
32	642	591	543	619	1962	2093
40	569	561	535	545	1999	2093
48	526	484	439	543	1813	2093
56	526	484	439	540	1864	1962
64	461	432	420	448	2065	1962

p22810 (five scan frequencies: 10 MHz to 50 MHz)							
	T_{PA}					Power consumption	
W	$P_{max} = 6500$	$P_{max} = 8000$	$P_{max} = 10000$	T	T_b	P_{PS}	P_b
16	29644	25598	24358	22567	29464	9403	10143
24	20189	18644	16096	14894	22372	9403	10143
32	15682	13962	13422	12613	15153	10139	10143
40	14494	13519	13219	12607	14125	10125	10268
48	12646	11375	9638	8307	13367	8123	10268
56	9102	8231	8231	7435	9341	8565	10029
64	8861	8114	8114	7435	9294	8863	10441

p93791 (five scan frequencies: 10 MHz to 50 MHz)						
	T_{PA}				Power consumption	
W	$P_{max} = 15000$	$P_{max} = 20000$	T	T_b	P_{PS}	P_b
16	143452	126787	110790	183186	24306	28726
24	83029	76029	66186	139457	25418	25088
32	67598	61246	54795	94878	22729	27816
40	57571	53921	48016	72748	21331	26328
48	49642	39784	36395	61192	23489	28726
56	42512	36127	30686	56578	22579	25364
64	41868	38614	34872	47644	22579	25364

5. CONCLUSION

We have presented a new test planning technique for core-based SoCs that exploits the port-scalability features of current-generation testers. It is based on a scalable heuristic approach that takes into account multiple scan data rates for an SoC. The heuristic approach involves an iterative descent procedure which jointly optimizes the widths of TAM partitions, TAM-partition frequencies, and the assignment of cores to TAM partitions to reduce the SoC test time. We have also presented an ILP-based approach and two baseline cases for this problem. We have evaluated the proposed heuristic by comparing it to the ILP-based approach, baseline cases, and provable lower bounds on the test time. The heuristic approach performs significantly better than the baseline scenarios. Moreover, in some cases it leads to the same test time as the optimal, but computationally-expensive, ILP approach. We also extend the iterative descent heuristic procedure to incorporate constraints on maximum power consumption during test. Experimental results for several ITC'02 benchmark circuits demonstrate the effectiveness of the proposed methods.

REFERENCES

AGILENT 93000. 2008. *Agilent 93000 Multi-Port Testing User Manual, revision 4.0.0.*

AGILENT TECHNOLOGIES. 2002. *Winning in the SoC market, available online at:* http://cp.literature.agilent.com/litweb/pdf/5988-7344EN.pdf.

CHAKRABARTY, K. 2001. Optimal test access architectures for system-on-a-chip. *ACM Trans. Des. Autom. Electron. Syst. 6*, 26–49.

CHICKERMANE, P. G. V., GREGOR, S., AND PIERRE, T. S. 2001. A building block BIST methodology for SoC designs: A case study. In *Proceedings of the International Test Conference*, 111–120.

CHOU, R. M., SALUJA, K. K., AND AGRAWAL, V. D. 1997. Scheduling tests for VLSI systems under power constraints. *IEEE Trans. VLSI Syst. 5*, 175–185.

DORSCH, R., RIVERA, R. H., WUNDERLICH, H. J., AND FISCHER, M. 2002. Adapting an SoC to ATE concurrent test capabilities. In *Proceedings of the International Test Conference*, 1169–1175.

GOEL, S. K., CHIU, K., MARINISSEN, E. J., NGUYEN, T., AND OOSTDIJK, S. 2004. Test infrastructure design for the Nexperia™ home platform pnx8550 system chip. In *Proceedings of the DATE Designers' Forum*, 108–113.

GOEL, S. K. AND MARINISSEN, E. J. 2002. Effective and efficient test architecture design for SoCs. In *Proceedings of the International Test Conference*, 529–538.

GONCIARI, P. T., HASHIMI, B. A., , AND NICOLICI, N. 2003. Addressing useless test data in core-based system-on-a-chip test. *IEEE Trans. Comput. Aided Des. 22*, 1568–1580.

HUANG, Y., REDDY, S. M., CHENG, W. T., REUTER, P., MUKHERJEE, N., TSAI, C. C., SAMMAN, O., AND ZAIDAN, Y. 2002. Optimal core wrapper width selection and SoC test scheduling based on 3-D bin packing algorithm. In *Proceedings of the International Test Conference*, 74–82.

IYENGAR, V. AND CHAKRABARTY, K. 2002. System-on-a-Chip test scheduling with precedence relationships, preemption, and power constraint. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst. 21*, 1088–1094.

IYENGAR, V., CHAKRABARTY, K., AND MARINISSEN, E. J. 2003a. Efficient test access mechanism optimization for system-on-chip. *IEEE Trans. Comput. Aided Des. 6*, 635–643.

IYENGAR, V., CHAKRABARTY, K., AND MARINISSEN, E. J. 2003b. Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip. *IEEE Trans. Comput. 52*, 1619–1632.

IYENGAR, V., MARINISSEN, E. J., AND CHAKRABARTY, K. 2002. Test wrapper and test access mechanism co-optimization for system-on-chip. *J. Electron. Test. Theory Appl. 18*, 213–230.

KHOCHE, A. 2001. Agilent corporation. Private communication.

LARSSON, E. AND PENG, Z. 2001. Test scheduling and scan division under power constraint. In *Proceedings of the Asian Test Symposium*, 259–264.

LARSSON, E. AND PENG, Z. 2002. An integrated framework for the design and optimization of SoC test solutions. *J. Electron. Test. Theory Appl. 18*, 385–400.

LARSSON, E. AND PENG, Z. 2006. Power-Aware test planning in the early system-on-chip design exploration process. *IEEE Trans. Comput. 55*, 227–239.

LIN, X. AND THOMPSON, R. 2003. Test generation for designs with multiple clocks. In *Proceedings of the Design Automation Conference*, 662–667.

MARINISSEN, E. J., ARENDSEN, R. G. J., BOS, G., DINGEMANSE, H., LOUSBERG, M., AND WOUTERS, C. 1998. A structured and scalable mechanism for test access to embedded reusable cores. In *Proceedings of the International Test Conference*, 284–293.

MARINISSEN, E. J., IYENGAR, V., AND CHAKRABARTY, K. 2002. A set of benchmarks for modular testing of SoCs. In *Proceedings of the International Test Conference*, 519–528.

SAMII, S., LARSSON, E., CHAKRABARTY, K., AND PENG, Z. 2006. Cycle-Accurate test power modeling and its application to SoC test scheduling. In *Proceedings of the International Test Conference*.

SCHMID, J. AND KNABLEIN, J. 1999. Advanced synchronous scan test methodology for multi clock domain ASICs. In *Proceedings of the VLSI Test Symposium*, 106–113.

SEHGAL, A. AND CHAKRABARTY, K. 2007. Optimization of dual-speed TAM architectures for efficient modular testing of SoCs. *IEEE Trans. Comput. 56*, 120–133.

SEHGAL, A., IYENGAR, V., AND CHAKRABARTY, K. 2004. SoC test planning using virtual test access architectures. *IEEE Trans. VLSI Syst. 2*, 1262–1276.

SU, C. P. AND WU, C. W. 2004. A graph-based approach to power-constrained SoC test scheduling. *J. Electron. Test. Theory Appl. 19*, 45–60.

TERADYNE TECHNOLOGIES. 2008. Tiger: Advanced digital with silicon germanium technology. http://www.teradyne.com/tiger/digital.html.

VRANKEN, H., HAPKE, F., ROGGE, S., CHINDAMO, D., AND VOKERINK, E. 2003. ATPG padding and ATE vector repeat per port for reducing test data volume. In *Proceedings of the International Test Conference*, 1069–1078.

XU, Q. AND NICOLICI, N. 2004a. Multi-Frequency test access mechanism design for modular SoC testing. In *Proceedings of the Asian Test Symposium*, 2–7.

XU, Q. AND NICOLICI, N. 2004b. Time/Area tradeoffs in testing hierarchical SoCs with hard mega-cores. In *Proceedings of the International Test Conference*, 1196–1202.

XU, Q. AND NICOLICI, N. 2005. Resource-Constrained system-on-a-chip test: A survey. *IEE Proce. Comput. Digital Techniq. 152*, 67–81.

YONEDA, T., IMANISHI, M., AND FUJIWARA, H. 2007. An SoC test scheduling algorithm using reconfigurable union wrappers. In *Proceedings of DATE*, 231–236.

YU, T. E., YONEDA, T., ZHAO, D., AND FUJIWARA, H. 2007. Using domain partitioning in wrapper design for IP cores under power constraints. In *Proceedings of the VLSI Test Symposium*, 369–374.

ZHAO, D. AND UPADHYAYA, S. 2003. Power constrained test scheduling with dynamically varied TAM. In *Proceedings of the VLSI Test Symposium*, 273–278.

ZORIAN, Y., MARINISSEN, E. J., AND DEY, S. 1999. Testing embedded-core-based system chips. *IEEE Comput. 32*, 52–60.

Received June 2007; revised December 2007; accepted March 2008