Optimization of Dual-Speed TAM Architectures for Efficient Modular Testing of SOCs

Anuja Sehgal, *Member, IEEE*, and Krishnendu Chakrabarty, *Senior Member, IEEE*

Abstract—The increasing complexity of system-on-chip (SOC) integrated circuits has spurred the development of versatile automatic test equipment (ATE) that can simultaneously drive different channels at different data rates. Examples of such ATEs include the Agilent 93000 series tester based on port scalability and the test processor-per-pin architecture and the Tiger system from Teradyne. The number of tester channels with high data rates may be constrained in practice, however, due to ATE resource limitations, the power rating of the SOC, and scan frequency limits for the embedded cores. Therefore, we formulate the following optimization problem: Given two available data rates for the tester channels, an SOC-level test access mechanism (TAM) width W , an upper limit V ($V < W$) on the number of channels that can transport test data at the higher data rate, determine an SOC TAM architecture that minimizes the testing time. We present an efficient heuristic algorithm for TAM optimization that exploits port scalability of ATEs to reduce SOC testing time and test cost. We present experimental results for the ITC '02 SOC test benchmarks and investigate the impact of dual-speed TAM architectures on power consumption during testing for one of these benchmarks.

Index Terms—Full-chip testing, SOC testing, test scheduling, test access mechanism, dual-speed TAM, TAM optimization.

1 INTRODUCTION

RECENT advances in CMOS technology have led to a significant increase in the complexity of system-on-chip (SOC) integrated circuits. In order to address the test requirements of complex SOCs, automatic test equipment (ATE) vendors have recently announced a new class of testers that can simultaneously drive different channels at different data rates. Examples of such ATEs include the Agilent 93000 series tester based on port scalability and the test processor-per-pin architecture [1] and the Tiger system from Teradyne [3] in which the data rate can be increased through software for selected pin groups to match SOC test requirements. However, the number of tester channels with high data rates may be constrained in practice due to ATE resource limitations, the power rating of the SOC, and scan frequency limits for the embedded cores. Optimization techniques are therefore needed to ensure that the high data-rate tester channels are efficiently used during SOC testing.

Modular testing of the embedded cores in an SOC can simplify the complex problems of test access and application [5]. For modular testing, an embedded core is isolated from the surrounding logic using a test wrapper and a test access mechanism (TAM) is designed to deliver test data from the I/O pins of the SOC and propagate test responses from core terminals to chip pins. This approach facilitates the reuse of precomputed tests for individual cores and

partitions the SOC for test; thus, the test methodology follows the modular design process.

Modular testing also lends itself well to the use of port scalable testers for dual-speed testing of SOCs. In port scalable testers, every port of the tester can be configured at a desired data rate, where each port typically consists of multiple channels [2]. Each channel of the tester connects directly to an on-chip TAM wire; the total number of channels used is equal to the total TAM width of the SOC. The group of TAM wires connected to the same port operate at the same scan data rate. In our scenario of dual-speed testing, the lower-speed TAM wires are connected to channels of the ports that are configured at the lower scan data rate, while the higher-speed TAM wires are connected to channels of ports that are configured at the higher data rate. More than one port may be configured at one of the two data rates depending on the number of available high-speed and low-speed TAM wires.

A number of methods for TAM design [5], [4], [6], TAM optimization, and test scheduling [7], [11] for modular test have been presented in the literature. TAMs have been designed based on direct access to cores multiplexed onto the existing SOC pins [12], reuse of the on-chip system bus [4], utilization of transparent paths through and/or around neighboring modules [13], [14], and utilization of bridge elements such as multiplexers and tristate buffers, and isolation rings around cores [15]. Recently, the most popular techniques appear to be dedicated, scalable TAMs such as Test Bus [6] and TestRail [5]. We focus our attention on the Test Bus model in this paper. Power-constrained test scheduling has also been studied in the literature [8], [9], [10], [11], [29]. These methods focus on the optimization of SOC power together with the optimization of overall SOC test time. In [9] and [10], the power profiles of the embedded cores are considered for the optimization of SOC power and SOC test time.

- A. Sehgal is with the Consumer Products Group, Advanced Micro Devices, One AMD Place, Sunnyvale, CA 94085. E-mail: as@ee.duke.edu.
- K. Chakrabarty is with the Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708. E-mail: krish@ee.duke.edu.

Manuscript received 19 July 2005; revised 11 Dec. 2005; accepted 13 Feb. 2006; published online 22 Nov. 2006.

For information on obtaining reprints of this article, please send e-mail to: tc@computer.org, and reference IEEECS Log Number TC-0242-0705.

The problem of designing a TAM architecture and determining a test schedule to minimize the SOC testing time has been shown in the literature to be $\mathcal{NP}$ -hard. Therefore, a number of efficient heuristic techniques have been developed for TAM optimization [7], [16], [17], [18], [19]. However, in all these methods, it is assumed 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 speeds 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.

We extend the heuristic approach based on rectangle packing that was presented in [17]. The use of rectangles to model core tests was described in [16], [17]. 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 a Core i ($1 \leq i \leq N$, where N is the number of cores in the SOC) can be constructed such that the height of each rectangle corresponds to a different TAM width and the width of the rectangle represents the core test application time for this value of TAM width. The TAM optimization problem can now be formulated in terms of rectangle packing as follows: Select one rectangle $R_{ij} \in \mathbf{R}_i$ 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 the width to which the bin is filled is minimized. The problem formulation and heuristic solution in [17] are based on a single-speed TAM architecture; therefore, they are not directly applicable to the problem of optimizing dual-speed TAM architectures being studied in this paper.

The availability of dual-speed ATEs was recently exploited in [20], where a technique was presented to match ATE channels with high data rates to core scan chain frequencies using virtual TAMs. A virtual TAM is an on-chip test data transport mechanism that does not directly correspond to a particular ATE channel. Virtual TAMs operate at scan-chain frequencies; however, they interface with the higher-frequency ATE channels using bandwidth matching. Moreover, since the virtual TAM width is not limited by the ATE pin-count, a larger number of TAM wires can be used on the SOC, thereby leading to lower testing times. A drawback of virtual TAMs, however, is the need for additional TAM wires on the SOC as well as frequency division hardware for bandwidth matching. Related recent work also requires additional hardware for frequency matching [21]. In our approach, we reduce the hardware overhead by using a smaller number of on-chip TAM wires.

In this paper, we focus on the effective and efficient design of dual-speed TAM architectures for modular testing of SOCs. The major contributions of this paper are the following:

- We provide a method to effectively use ATE channels with high data rates to directly drive SOC TAM wires without requiring frequency division hardware.
- We describe a heuristic method, based on rectangle packing with two bins, to arrive at a test solution

that exploits the port scalability in current-generation ATEs.

- We show that the use of more than two scan data rates does not always result in a reduction of overall test time.
- We provide a geometric lower bound on the test time for dual-speed TAM architectures.
- We study the impact of dual speed testing on the peak and average test power of the SOC.
- We study the impact of power constraints on the overall test time of the SOC.

The rest of this paper is organized as follows: In Section 2, we define the dual-speed TAM optimization problem and formulate it as a generalized version of rectangle packing. In Section 4, we present an efficient algorithm to optimize a dual-speed TAM architecture and to derive a test schedule that minimizes the testing time. In Section 5, we present the experimental results for three ITC '02 benchmark SOCs. In Section 6, we examine the impact on test power for a benchmark SOC for which power models are available. We also investigate the impact of power constraints on the test schedule. Finally, we present conclusions in Section 7.

2 DUAL-SPEED TAM OPTIMIZATION

In this section, we define the dual-speed TAM optimization problem and formulate it in terms of rectangle packing.

Problem $P_{\text{dual-speed}}$. Given the test data parameters for the embedded cores, total SOC-level TAM width W , a total of V available high-speed ATE channels ($V < W$), and the ratio f of the high-speed data transfer rate to the low-speed data transfer rate, determine 1) the wrapper design, TAM width, and test data rate for each core, and the SOC test schedule such that a) the total number of TAM wires utilized at any moment does not exceed W , b) the number of TAM wires driven at the high data rate does not exceed V , and c) the SOC testing time 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 to be hard cores, i.e., the number and length of scan chains are fixed.

For a given TAM width and wrapper design for a core, we assume that its testing time at the high data rate is f times less than its testing time at the low data rate. In other words, if it takes $T_{ih}(w_i)$ seconds to test Core i at the high data rate with a TAM width w_i , the time taken to test it at the low data rate is $T_{il}(w_i)$ seconds, where $T_{il}(w_i) = f \times T_{ih}(w_i)$. A core vendor can mandate an upper limit on the scan test frequency for a core and, if this upper limit is lower than the higher data rate, the core can only be tested at the lower data rate. Otherwise, a core can be tested using either the high-speed data channels or the low-speed data channels. We assume that a core is not connected to the ATE by both high-speed and low-speed data channels during scan testing, i.e., it is not possible to assign both high-speed and low-speed TAM wires to a core. While a higher data rate for a given TAM width always leads to

Fig. 1. An example illustrating the impact of the use multiple-speed TAM architecture on test time.

reduced testing time for a core, a higher data rate for a subset of TAM wires can lead to a smaller TAM width that is available for a core. The reduced TAM width for the core can lead to an increase in its testing time, despite the faster test data rate. As a result, an optimization procedure as described in this paper is needed to either select appropriate values of f and V or determine an efficient TAM architecture for given values of f and V .

Recall that the height of a rectangle for a core represents the TAM width assigned to that core and the width of the rectangle denotes the testing time of the core for the corresponding value of the TAM width. The *Design Wrapper* algorithm from [22] is used to design a wrapper and determine the testing time for a core for several possible TAM widths. These precalculated testing times are subsequently used in the TAM optimization procedure. Based on an extension to the heuristic approach of [17], we formulate the dual-speed TAM optimization problem as follows: Given a collection of two sets of rectangles for each core, one representing the testing times for the high data rate and the other representing the testing times for the low data rate, and two bins of fixed heights $W - V$ and V , respectively, denoting the two data transfer rates, select a rectangle for each core and pack it in the appropriate bin such that no two rectangles overlap and the maximum of the widths of the two bins is minimized.

Let m be the number of TAM widths of interest for Core i . Let $Rl_i = \{Rl_{i1}, \dots, Rl_{im}\}$ be the set of rectangles for Core i for the low data rate and $Rh_i = \{Rh_{i1}, \dots, Rh_{im}\}$ be the set of rectangles for the high data rate. Let $R_i = Rl_i \cup Rh_i$. Consider two bins of unbounded width stacked on top each other; the height V of the top bin represents the V high-speed TAM wires, and the height $W - V$ of the lower bin represents the low-speed TAM wires. The optimization problem can now be formally stated as follows:

Problem P_{GRP2} . Given the collection of set of rectangles $R_1, R_2, \dots, R_N$ for an SOC with N cores, select one rectangle R_i^* from each set R_i , pack R_i^* in the bin for high-speed (low-speed) TAM wires if $R_i^* \in Rh_i$ ($R_i^* \in Rl_i$) such that no two rectangles overlap and the maximum of the widths of the two bins is minimized.

P_{GRP2} reduces to the problem P_{GRP} described in [17] if $Rh_i = \emptyset$, $1 \leq i \leq N$, and $V = 0$. Since P_{GRP} was shown to be $\mathcal{NP}$ -hard in [17], we conclude that P_{GRP2} is also $\mathcal{NP}$ -hard.

In [22], the staircase nature of testing time variation with TAM width for cores is exploited to reduce the TAM width assigned to cores to the minimal value required to achieve a specific testing time. The TAM width values for which the testing time decreases are called the Pareto-optimal points of the core and only rectangles corresponding to the Pareto-optimal TAM width values are considered. In the TAM optimization problem addressed here, each core has a set of Pareto-optimal points for low-speed test application and the same number of Pareto-optimal points for high-speed test application.

The problem of dual-speed optimization can be extended to more than two speeds. However, multiple speeds for a given TAM width increases the number of bins, which in turn results in the availability of smaller TAM width per bin. This often results in some cores having reduced test widths at higher data rates. Recall that, in most cores, the relationship of the core's test time to the core's test width is a staircase function. Thus, even if a higher shift frequency is used for a core, its test time can increase if the TAM architecture leads to a reduction in the bitwidth used to access it.

We present an example in Fig. 1 to illustrate that having more than two data rates is not always advantageous. Consider an SOC with cores A, B, C, and D; Fig. 1a lists the test times (in clock cycles) for these cores for different

Data structure *Schedule*

1. $width_l(i)$ /* preferred TAM width for Core i on the low-freq lines*/
2. $width_h(i)$ /* preferred TAM width for Core i on the high-freq lines*/
3. $width(i)$ /* TAM width assigned to Core i */
5. $begin(i)$ /* start time of Core i */
6. $end(i)$ /* end time of Core i */
7. $freq(i)$ /* boolean indicates Core i is scheduled on high-freq or low-freq TAM*/
8. $scheduled(i)$ /* boolean indicates Core i has been scheduled */
9. $complete(i)$ /* boolean indicates test for Core i has completed */

Fig. 2. Data structure for the test schedule.

values of the bitwidth w used to access them. We consider three different TAM architectures. For the sake of this example, we assume that the cores are appropriately wrapped for each value of w . As shown in Fig. 1b, the test time is 965.25 μ s when all tester channels run at the common scan frequency of 20 MHz. In Fig. 1c, five of the 10 channels are configured to operate at twice ($f = 2$) the scan data rate of the low-speed channels of 20 MHz. The test time for this case reduces to 537.2 μ s. In Fig. 1d, when two out of the five high-speed channels are run at three times ($f = 3$) the low-speed channels, the test time increases to 540 μ s. This is because the increase in scan frequency is accompanied by a decrease in the bit width available for testing the cores.

3 LOWER BOUND ON TEST TIME

In order to evaluate our heuristic, we determine a lower bound on the testing time for a dual-speed architecture. In our lower bound and in the test scheduling approach, we do not allow the overlap of the scan-out operation for the last test pattern of a core with the scan-in operation for the first pattern of the next core on the same TAM wire. While this is feasible for a fixed-width TestRail architecture, as in [7], it is difficult to implement for a flexible-width Test Bus architecture.

For a single-speed TAM architecture, 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. The area of a rectangle representing core i being tested at TAM width w is given by $R_i(w) = T_i(w) \times w$, where $T_i(w)$ is the testing time of Core i on TAM width w . Let the area of the minimum-area rectangle for Core i be R_i^{min} , where $R_i^{min} = \min_i \{R_i(w)\}$, $1 \leq w \leq W$. A lower bound on the testing time of a core on a TAM width w can be expressed as:

$$T_i(w) = \lceil (\max(ts_i, tr_i) \cdot p_i + \min(ts_i, tr_i)) / w \rceil + p_i,$$

where ts_i is the number of test bits to be scanned into core i , tr_i is the number of test bits to be scanned out of core i , and p_i is the number of test patterns to be applied to Core i .

Now, we know that the total area of the bin cannot be less than the sum of the minimum-area rectangles of all the cores in the SOC. Thus, for any bin, $T \times W \geq R_1^{min} + R_2^{min} + \dots + R_i^{min}$, which implies that $T \geq \sum_{i=1}^N R_i^{min} / W$. In dual-speed TAM optimization using

rectangle packing, two bins are stacked on top of each other. A core can only be assigned to one of the two bins. Let $x_{i1} = 1$ ($x_{i2} = 1$) if Core i is assigned to the upper (lower) bin. Clearly, $x_{i1} + x_{i2} = 1$, $1 \leq i \leq N$. In order to determine a lower bound on the SOC testing time, we define the following integer linear programming (ILP) problem:

Objective: Minimize C subject to

1. $C \geq T_h$;
2. $C \geq T_l$;
3. $T_h \geq (\sum_{i=1}^N R_{ih}^{min} \cdot x_{i1}) / V$;
4. $T_l \geq (\sum_{i=1}^N R_{il}^{min} \cdot x_{i2}) / (W - V)$;
5. $x_{i1} + x_{i2} = 1$, $1 \leq i \leq N$.

The above ILP model can be solved easily to determine the lower bounds. It takes less than a second of CPU time for the benchmark SOCs. Compared to the lower bound based on the notion of a “bottleneck core” derived in [23], the above lower bound is more accurate for smaller TAM widths. However, for larger values of W , [23] provides a tighter bound in many cases. Hence, we take the maximum of the lower bound obtained from the ILP model and lower bound from [23]. The lower bounds for various TAM widths for several benchmark SOCs are presented in Section 7.

4 OPTIMIZATION PROCEDURE

In this section, we explain the heuristic procedure used to solve the $P_{\text{dual-speed}}$ problem, which was modeled as P_{GRP2} in Section 2. We extend the $TAM_optimizer$ procedure from [17] to solve the rectangle packing problem concurrently over two bins. In the $TAM_optimizer$ procedure, tests are scheduled depending on certain preferred TAM widths. When a core completes its test, the TAM wires being used by it are freed and are available for assignment to other cores. The goal is to assign a preferred TAM width to each core as long as there are enough TAM lines available. We describe the details of our heuristic method TAM_Opt_{dual} in the following paragraphs.

Data structure. The TAM width and the testing time of each core are stored in a data structure, which contains information about the start time, end time, preferred TAM widths, and TAM width frequency assigned. The data structure is presented in Fig. 2. This data structure is updated as the SOC test schedule is developed.

```

1 Compute collection  $R_l$  and  $R_h$  of rectangles using Design.wrapper;
2 Initialize( $C, d, p$ );
3 Set  $w_{avail} = W$ ;  $v_{avail} = V$ ;  $current\_time = 0$ ;
4 While  $C.size \neq 0$ 
5   If  $w_{avail} > 0$  OR  $v_{avail} > 0$ 
6     If Core  $i \in C$  can be found, such that  $width_l(i) \leq w_{avail}$  AND  $T_{il}(width_l(i))$  is maximum OR
7     If Core  $j \in C$  can be found, such that  $width_h(j) \leq v_{avail}$  AND  $T_{jh}(width_h(j))$  is maximum{
8       If  $C.size \neq 1$ 
9         Calculate  $T_{il}(w)$ , where  $w$  = highest Pareto-optimal width for  $i$ , such that  $w \leq w_{avail}$ ;
10        Calculate  $T_{jh}(v)$ , where  $v$  = highest Pareto-optimal width for  $j$ ,
11        such that  $v \leq v_{avail}$ ;
12        If  $T_{ih}(v) \leq T_{il}(w)$ , set  $freq(i) = 1$ , set  $width(i) = w$ ;
13        Else set  $freq(i) = 0$ , set  $width(i) = v$ ;
14      Else if  $T_{jh}(width_h(j)) < T_{il}(width_l(i))$ 
15        set  $freq(j) = 1$ ;  $core = j$ ;  $width(j) = width_h(j)$ ;
16      Else set  $freq(i) = 1$ ;  $core = i$ ;  $width(i) = width_l(i)$ ;
17      Update( $core$ );
18    Else /* find a core that can use the resulting idle TAM wires */
19      Find  $next\_time = end(i)$ , such that  $end(i) > this\_time$  AND  $end(i)$  is minimum;
20      If Core  $i$  can be found, such that  $scheduled(i) = 0$  AND
21       $T_i(w_{avail}) + this\_time \leq next\_time$  AND
22       $T_i(w_{avail}) + this\_time$  is maximum{
23        Set  $width_l(i) = w$ , where  $w$  is the highest Pareto-optimal width for  $i$ , such that  $w \leq w_{avail}$ ;
24      If Core  $j$  can be found, such that  $scheduled(j) = 0$  AND
25       $T_j(v_{avail}) + this\_time \leq next\_time$  AND
26       $T_j(v_{avail}) + this\_time$  is maximum{
27        Set  $width_h(j) = v$ , where  $v$  is the highest Pareto-optimal
28        width for  $j$ , such that  $v \leq v_{avail}$ ;
29      Repeat line 8–10;
30    Else /*if no core found in lines 13 and 15, assign the TAM wires to
31    already schedules cores that will benefit most in testing time reduction */{
32      If Core  $i$  can be found, such that  $begin(i) = this\_time$  AND
33       $T_i(width_l(i)) - T_i(width_l(i) + w_{avail})$  is maximum;{
34        Set  $width(i) = w$ , where  $w$  is the highest Pareto-optimal width,
35        such that  $w \leq width_l(i) + w_{avail}$ ;
36      Else /* declare TAM wires  $w_{avail}$  idle until  $next\_time$  */
37        Set  $w\_idle = w_{avail}$ ;  $w_{avail} = 0$ ;
38      If Core  $j$  can be found, such that  $begin(j) = this\_time$  AND
39       $T_j(width_h(j)) - T_j(width_h(j) + v_{avail})$  is maximum;{
40        Set  $width(j) = v$ , where  $v$  is the highest Pareto-optimal width,
41        such that  $v \leq width_h(j) + v_{avail}$ ;
42      Else /* declare TAM wires  $v_{avail}$  idle until  $next\_time$  */
43        Set  $v\_idle = v_{avail}$ ;  $v_{avail} = 0$ ;
44      Repeat lines 8–10 }
45    Else{
46      Calculate  $next\_time = end(i)$ , such that  $end(i) > this\_time$  AND  $end(i)$  is minimum;
47      Set  $this\_time = next\_time$ ;
48      For every Core  $i$ , such that  $end(i) = this\_time$ {
49        if  $freq(i) = 0$ , Increment  $w_{avail} = w_{avail} + width(i)$ ;
50      Else Increment  $v_{avail} = v_{avail} + width(i)$ ;
51      Set  $complete(i) = 1$ ;}}
52  Return Schedule;

```

Fig. 3. Pseudocode for procedure $TAM_Opt_{dual}(C, W, V, D, p)$.

Preferred TAM widths. We first compute a collection of Pareto-optimal rectangles for both the bins in Line 1 of TAM_Opt_{dual} (Fig. 3). Each core has a “preferred low-frequency TAM width” and a “preferred high-frequency TAM width.” The preferred TAM widths are computed as a small percentage of the maximum allowable TAM widths W_{max} and V_{max} , respectively, in Procedure *Initialize*. The pseudocode for Procedure *Initialize* is shown in Fig. 4. In our algorithm, we choose $W_{max} = 64$ and $V_{max} = 64$. An input parameter p determines the preferred TAM width for each core. It is the TAM width at which a testing time of the core reaches within p percent of its testing time at the maximum allowable TAM width W_{max} (Line 2 and Line 7 of *Initialize*). This input parameter is varied from 1 to 10 and the value that results in the best solution is chosen. To account for the requirements of bottleneck cores, an input difference parameter d is chosen. If the testing time can be improved by adding a few TAM wires ($\leq d$) to the core, then the preferred TAM width is increased (Line 5 and Line 10 of *Initialize*). In our procedure, we use the same

value of p to determine the preferred TAM width for both of the bins.

Assigning preferred TAM widths to cores. In the case when both the high-speed and low-speed TAM wires are available (Line 5 of TAM_Opt_{dual}), a core that has the

Procedure *Initialize*(C, d, p)

```

1 For each Core  $i \in C$ 
2   Calculate  $T_{il} = T_i(W_{max}) + \frac{p}{100} \times (T_i(1) - T_i(W_{max}))$ ;
3   Set  $width_l(i) = w$ , such that  $T_i(w) - T_{il}$  is minimum;
4   Calculate highest Pareto-optimal width  $w_p$ ;
5   If  $w_p - width_l(i) \leq d$  then  $width_l(i) = w_p$ ;
6 For each Core  $i \in C$ 
7   Calculate  $T_{ih} = T_i(V_{max}) + \frac{p}{100} \times (T_i(1) - T_i(V_{max}))$ ;
8   Set  $width_h(i) = w$ , such that  $T_i(w) - T_{ih}$  is minimum;
9   Calculate highest Pareto-optimal width  $w_p$ ;
10  If  $w_p - width_h(i) \leq d$  then  $width_h(i) = w_p$ ;

```

Fig. 4. Pseudocode for procedure *Initialize*.

Procedure Update(*i*)

```

1 Let i be the core to be updated in the test schedule;
2 Set begin(i) = current_time;
3 if freq(i) = 0.4 Set end(i) = current_time + Til(width(i));
5 ELSE end(i) = current_time + Tih(width(i));
6 Set scheduled(i) = 1;
7 If i ∈ C then C = C − {i};

```

Fig. 5. Pseudocode for procedure *Update*.

highest testing time and whose preferred TAM width is less than the available TAM width is found for both the high-speed and low-speed bins (Lines 7 to 10 of *TAM_Opt_{dual}*). Of the two assignments, the assignment that yields a smaller end time is chosen. If only one type of TAM wires (low-speed or high-speed) is available, the core with the largest testing time and whose preferred TAM width is less than or equal to the available TAM width is chosen. On assigning a core to one of the two bins, the data structure for the core is updated using Procedure *Update*. The pseudocode for Procedure *Update* is shown in Fig. 5. Fig. 6 illustrates some of the variables used in our procedures. Two pointers, namely, the parameter *this_time* and *next_time*, are jointly maintained for the two bins. The parameter *this_time* is used to keep track of the earliest time at which TAM lines are available in either of the two bins and *next_time* is the minimum width to which either of the two bins are filled. These two pointers are updated as the tests for the cores are scheduled.

Minimizing idle time. The minimization of idle time is done in the same manner as in the *TAM_optimizer* procedure. If the available TAM widths are less than the preferred TAM width of all the cores, there might be idle spaces in the schedule since the next assignment can be made only when more TAM lines have been freed. These idle spaces may appear in both the high-speed and the low-speed bin. These spaces are minimized, in Lines 13 to 17 of *TAM_Opt_{dual}*, by assigning the freed TAM lines to an unscheduled core that will finish its test before more TAM lines are freed and will use up most of the idle time. However, the idle spaces in the two bins have to be minimized independently since a core cannot be assigned both high-speed and low-speed TAM wires.

Fig. 6. Test scheduling using high-speed and low-speed bins.

TABLE 1
Testing Time Results for p22810 ($f = 2$)

p22810					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
T_n ($W = 16$)	11315.97	14652.25	17236.85	25329.25	22631.95
LB	10536.20	12041.60	14048.65	16858.50	21072.95
$\Delta T_n(\%)$	-50.00	-35.25	-23.83	11.91	0
T_n ($W = 24$)	7694.45	9987.25	10838.75	14807.95	15389.00
LB	7023.90	8027.50	9365.70	11238.80	14048.55
$\Delta T_n(\%)$	-50.00	-35.10	-29.56	-3.77	0
T_n ($W = 32$)	6153.75	6966.00	8641.20	10437.85	12307.50
LB	5267.75	6020.55	7024.15	8429.05	10536.20
$\Delta T_n(\%)$	-50.00	-43.40	-29.78	-15.19	0
T_n ($W = 40$)	4932.37	5890.00	6966.00	7914.70	9864.65
LB	4213.95	4816.20	5619.15	6743.10	8428.75
$\Delta T_n(\%)$	-50.00	-40.29	-29.38	-19.76	0
T_n ($W = 48$)	4181.35	5159.30	6069.95	6691.10	8362.80
LB	3511.55	4013.40	4682.60	5619.20	7023.90
$\Delta T_n(\%)$	-50.00	-38.30	-27.41	-19.98	0
T_n ($W = 56$)	3635.40	4393.70	5150.55	6424.55	7270.85
LB	3009.8	3009.90	4013.50	4816.35	6020.25
$\Delta T_n(\%)$	-50.00	-39.57	-29.16	-11.63	0
T_n ($W = 64$)	3423.45	3985.75	4915.35	5531.70	6847.05
LB	2633.60	3511.80	3511.80	4214.30	5267.75
$\Delta T_n(\%)$	-50.00	-41.78	-28.21	-19.21	0

$\Delta T_n : \frac{T_n - T_0}{T_0}$; LB: Lower bound;

Redistribution of lines to fill idle time. In Lines 18 to 26 of *TAM_Opt_{dual}*, if there does not exist a core that can fill up the idle time before the freeing up of more TAM lines, then the idle TAM lines can be redistributed among the cores that began their testing at the start of the idle time. The idle TAM lines in the low-speed bin are redistributed to the cores in that bin only and, similarly, idle lines in the high-speed bin can be redistributed among cores in the high-speed bin only.

In summary, the proposed heuristic procedure extends the *TAM_optimizer* from [17] by allowing more decisions to be made due to the availability of two bins. The time complexity of the procedure is found via experiments to be similar to that of *TAM_optimizer*. For all the benchmarks that we considered, the CPU time was less than a minute.

5 EXPERIMENTAL RESULTS

In this section, we present experimental results on test scheduling and dual-speed TAM optimization for the three largest SOCs (in terms of the number of cores) from the ITC '02 SOC Test Benchmarks [24]. We first study the testing time reduction obtained with different sized high-speed bins. The testing time is calculated in μs , the low-speed TAM lines are assumed to be driven at 20 MHz, and the high-speed TAM lines are assumed to be driven at $20f$ MHz for different values of f . In Tables 1, 2, and 3, we present the testing time and lower bounds for various values of TAM widths W and a range of values for V , the number of TAM wires driven by the high-speed ATE channels. We also assume in this set of experiments that the high-speed data rate is twice that of the low-speed data rate, i.e., $f = 2$. The percentage change in testing time ΔT_n (percent) is calculated as follows: $\frac{T_n - T_0}{T_0} \times 100$, where T_n represents the testing time when n percent ($n = V/W \times 100$) of the TAM width is made up of high-speed channels, and T_0 represents the base case

TABLE 2
Testing Time Results for p34392 ($f = 2$)

p34392					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
$T_n (W = 16)$	25595.45	33798.55	42318.60	51660.45	51191.00
LB	23322.90	15550.75	23322.90	31095.10	46639.40
$\Delta T_n(\%)$	-50.00	-33.97	-17.33	0.91	0
$T_n (W = 24)$	18021.35	22469.20	25830.20	31902.15	37971.35
LB	15550.75	13614.45	15550.75	20732.20	31095.10
$\Delta T_n(\%)$	-52.53	-40.82	-31.97	-15.98	0
$T_n (W = 32)$	13614.45	18985.65	18404.10	21813.80	27228.95
LB	13614.45	13614.45	13614.45	15550.75	27228.95
$\Delta T_n(\%)$	-50.00	-30.27	-32.40	-19.88	0
$T_n (W = 40)$	13614.45	13614.45	14703.2	17008.75	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
$\Delta T_n(\%)$	-50.00	-50.00	-46.00	-37.53	0
$T_n (W = 48)$	13614.45	13614.45	14703.20	14889.85	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
$\Delta T_n(\%)$	-50.00	-50.00	-50.00	-46.00	0
$T_n (W = 56)$	13614.45	13614.45	13614.45	14703.20	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
$\Delta T_n(\%)$	-50.00	-50.00	-50.00	-46.00	0
$T_n (W = 64)$	13614.45	13614.45	13614.45	14703.20	27228.95
LB	13614.45	13614.45	13614.45	13614.45	27228.95
$\Delta T_n(\%)$	-50.00	-50.00	-50.00	-46.00	0

$\Delta T_n : \frac{T_n - T_0}{T_0}$; LB: Lower bound;

when no high-speed channels are used. We vary W from 16 to 64 in steps of 8, and we also consider five different values of n .

As expected, the reduction in the testing time is in many cases proportional to the number of high-speed TAM wires. In addition, the testing time is often close to the lower bound derived in Section 2. Note, however, that, since an increase in the number of high-speed TAM wires leads to a reduction in the number of TAM wires available per core,

the decrease in testing time obtained with the dual-speed TAM architecture is not always proportional to the fraction of TAM lines that transport test data at the higher rate. A dual-speed TAM optimization procedure helps the system integrator determine the values of f and n for which the testing time reduction is especially noteworthy. For example, for p93791, the testing time for Core 5 with $f = 1$ and available TAM width of 23 bits is 11,398.9 μs . With $f = 2$ but available TAM width of 10 bits (due to a smaller bin), the testing time increases to 14,026.9 μs .

We make the unexpected observation that, for smaller values of W and n , the testing time is sometimes higher for the dual-speed TAM architecture. The smaller sizes of the bins constrain the heuristic procedure to select small TAM

TABLE 3
Testing Time Results for p93791 ($f = 2$)

p93791					
	$n = 100\%$	$n = 75\%$	$n = 50\%$	$n = 25\%$	$n = 0\%$
$T_n (W = 16)$	46278.15	60845.35	85952.05	102033.4	92556.75
LB	43734.00	49981.85	58312.50	69975.00	81124.00
$\Delta T_n(\%)$	-50.00	-34.26	-7.13	10.23	0
$T_n (W = 24)$	31135.45	43795.65	60795.50	76184.00	62439.75
LB	29155.60	33321.05	38874.75	46649.80	54082.50
$\Delta T_n(\%)$	-50.13	-29.85	-2.63	22.01	0
$T_n (W = 32)$	24375.40	31135.45	37761.05	59911.05	48750.80
LB	21866.70	24990.55	29156.00	34987.30	40561.75
$\Delta T_n(\%)$	-50.00	-36.13	-22.54	22.89	0
$T_n (W = 40)$	19850.50	23184.00	28588.90	33035.45	39701.00
LB	17493.20	19992.30	23324.65	27989.70	32449.20
$\Delta T_n(\%)$	-50.00	-41.60	-27.98	-16.78	0
$T_n (W = 48)$	15698.10	20691.35	23665.90	28436.15	31396.70
LB	14577.55	16660.15	19437.05	23324.65	27040.85
$\Delta T_n(\%)$	-50.00	-34.09	-24.62	-9.42	0
$T_n (W = 56)$	14210.90	17915.55	20383.35	24223.55	28421.80
LB	12494.90	12495.10	16660.30	19992.45	23177.75
$\Delta T_n(\%)$	-50.00	-36.96	-28.28	-14.77	0
$T_n (W = 64)$	12782.15	15727.15	16806.55	22794.75	25564.30
LB	10932.95	14577.80	14577.80	17493.30	20280.60
$\Delta T_n(\%)$	-50.00	-36.96	-28.28	-14.77	0

$\Delta T_n : \frac{T_n - T_0}{T_0}$; LB: Lower bound;

Fig. 7. Effect of variation of frequency factor f on the testing time of SOC p22810 ($n = 50\%$).

Fig. 8. Effect of variation of frequency factor f on the testing time of SOC p34392 ($n = 50\%$).

widths for the cores; this leads to higher testing time for the SOC. The reduction in the testing time due to the higher data rate is not sufficient to outweigh the increase in testing time due to smaller TAM widths for the cores.

The results for SOC p34392 in Table 2 are especially interesting. This SOC is known to have a bottleneck core, due to which the testing time levels off at $27,228.95 \mu s$ for $W \geq 32$ using a single speed TAM architecture [17]. The dual-speed architecture allows us to overcome this “lower bound.” For $n = 25\%$, the testing time drops below $27,228.95 \mu s$ at $W = 29$ and levels off at $14,703.20 \mu s$ at $W = 56$. For $n = 50\%$, the testing time levels off at $13,614.47 \mu s$, (a 50 percent reduction compared to $27,228.95 \mu s$) at $W = 56$.

Next, we study the effect of varying f while keeping n constant. In Figs. 7, 8, and 9, we vary the frequency factor f from 2 to 5 for p22810, p34392, and p93791, keeping n fixed

Fig. 9. Effect of variation of frequency factor f on the testing time of SOC p93791 ($n = 50\%$).

Fig. 10. Illustration of the reduction in the required TAM width due to the availability of high-speed TAM wires for SOC p22810.

Fig. 11. Illustration of the reduction in the required TAM width due to the availability of high-speed TAM wires for SOC p34292.

Fig. 12. Illustration of the reduction in the required TAM width due to the availability of high-speed TAM wires for SOC p93791.

TABLE 4

Comparison between Virtual TAMs and the Proposed Dual-Speed TAM Architecture: (a) $n = 50\%$ and (b) $n = 25\%$

SOC	W^1	$V = W/2$	W_{virt}^2	T_{virt}^3	$T_{dual-speed}^4$	$\Delta T(\%)$
p22810	16	8	40	9709.65	9549.75	-1.64
	32	16	80	6681.40	5148.25	-22.94
	48	24	120	5477.75	4069.95	-25.70
	64	32	160	5477.75	3055.00	-44.22
p34392	16	8	40	27228.95	29363.30	7.83
	32	16	80	27228.95	14703.20	-46.00
	48	24	120	27228.95	10950.15	-59.78
	64	32	160	27228.95	8289.90	-69.55
p93791	16	8	40	36704.25	45250.10	23.28
	32	16	80	23619.40	23583.15	-0.15
	48	24	120	13170.20	15210.95	15.49
	56	28	140	12858.65	12217.70	-4.98
	64	32	160	11179.90	12217.70	9.28

(a)

SOC	W^1	$V = W/4$	W_{virt}^2	T_{virt}^3	$T_{dual-speed}^4$	$\Delta T(\%)$
p22810	16	4	28	14272.50	17252.10	20.87
	32	8	56	7270.85	9426.70	29.65
	48	12	84	6601.25	5469.00	-17.15
	64	16	112	6069.65	5148.25	-15.18
p34392	16	4	28	32757.20	37887.70	15.66
	32	8	56	27228.95	15879.20	-41.68
	48	12	84	27228.95	14703.20	-46.00
	64	16	112	27228.95	14703.20	-46.00
p93791	16	4	28	52292.00	80235.15	53.43
	32	8	56	25741.25	45769.05	77.80
	40	10	70	25741.25	24183.90	-6.05
	48	12	84	20593.00	22794.75	10.69
	56	14	98	20560.25	19494.55	-5.18
	64	16	112	17428.35	17078.70	-2.00

(b)

¹ W : No. of ATE channels; ² W_{virt} : Virtual TAM width; ³ T_{virt} : Testing time using virtual TAMs;⁴ $T_{dual-speed}$: Testing time for dual-speed architecture; ⁵ $\Delta T = \frac{T_{dual-speed} - T_{virt}}{T_{virt}}$.

at 50 percent. It can be seen that the testing times for the higher speed ratios ($f = 4, 5$) for some TAM widths are close to each other. This is because, while the testing time for the high-speed bin tends to decrease with an increase in the speed ratio, the testing time for the low-speed bin tends to remain the same. In such situations, the lower-speed bin dominates the overall testing time.

TABLE 5
Peak and Average Power Data for ISCAS-85 Cores
in d695 [28], [27]

Core	Peak power (PSF)	Average power (mW)
c432	660	1.12
c499	602	2.05
c880	823	2.75
c1355	275	5.45
c1908	690	9.22
c2670	354	10.80
c3540	530	14.64
c5315	753	23.10
c6288	641	70.32
c7552	1144	37.52

An advantage of the dual-speed TAM architecture is that, compared to a single-speed TAM architecture, a desired testing time for the SOC can be achieved with a smaller number of TAM wires. Let W (W') be the SOC-level TAM width for the single-speed (dual-speed) TAM architecture

Fig. 13. Testing time results for d695 ($f = 2$).

Fig. 14. Peak power consumption using n percent of high-frequency lines in d695 ($f = 2$).

Fig. 15. Average power consumption using n percent of high-frequency lines in d695 ($f = 2$).

that is required to achieve a desired testing time $\mathcal{T}$. Figs. 10, 11, and 12 show the ratio W^*/W versus $\mathcal{T}$ for p22810, p34392, and p93791, respectively, for two values of n and for $f = 2$. For $n = 25\%$, the ratio W^*/W is sometimes larger than one, which implies that no benefit is obtained with the dual-speed architecture for these cases.

Finally, in Table 4, we compare the testing time obtained with the dual-speed architecture to that obtained with the virtual TAM architecture described in [20] for $n = 50\%$, $n = 25\%$, and $f = 4$. We find that, for a given number of ATE channels, the dual-speed TAM architecture tends to outperform the virtual-TAM architecture for p22810 and p34392. The improvement in the virtual-TAM architecture is more pronounced for larger values of W . Note that, for

TABLE 6
Testing Time in μs , with Power Limits of 2,500 PSF and 3,000 PSF for $n = 0\%$, $n = 25\%$, and $n = 50\%$

$n = 0\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	2146.95	2146.95	0.00	2146.95	0.00
24	1463.55	1463.55	0.00	1463.55	0.00
40	884.50	898.75	1.61	884.50	0.00
48	743.65	789.35	6.15	762.50	2.53
56	621.05	668.80	7.69	647.05	4.19
64	580.20	624.80	7.69	618.80	6.65
$n = 25\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	2786.10	3016.35	8.26	2786.10	0.00
24	1738.30	1738.25	0.00	1738.30	0.00
32	987.85	1207.85	22.27	1110.60	12.43
40	881.20	1070.60	21.49	917.90	4.16
48	623.50	815.45	30.79	876.60	40.59
56	551.65	704.95	27.79	625.60	13.41
64	531.25	704.95	32.70	587.20	10.53
$n = 50\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	1724.50	2117.65	22.80	1928.30	11.82
24	1131.25	1466.30	29.62	1202.75	6.32
32	801.15	1145.20	42.94	894.60	11.66
40	672.55	1031.55	53.38	788.95	17.31
48	551.65	927.80	68.19	682.30	23.68
56	521.70	848.75	62.69	639.10	22.50
64	464.10	767.90	65.46	602.35	29.79

T_{p0} : Testing time with no power limit; T_{p1} : Testing time with power limit of 2500 PSF; T_{p2} : Testing time with power limit of 3000 PSF; $\Delta T_{px} = \frac{(T_{px} - T_{p0}) \times 100}{T_{p0}}$, $x \in \{0, 1\}$

p34392, the testing time is determined by the bottleneck core for the virtual-TAM architecture and it levels off at 27,228.95 μs . However, the use of a higher data rate for a subset of TAM widths allows us to reduce the testing time further without resorting to additional on-chip interconnect for the virtual TAM.

6 IMPACT OF DUAL-SPEED TAMs ON TEST POWER

Excessive power dissipation in an SOC during scan testing can cause overheating, which might lead to irreversible damage to the chip [25]. Therefore, along with efficient TAM design to minimize the testing time of the SOC, it is also important to investigate the effect of the TAM design on power consumption during scan testing. In this section, we study the impact of dual-speed TAM architecture on overall SOC test power. Conversely, we study the impact of power constraints on the overall SOC test time.

The use of high-frequency TAM lines leads to a reduction in the testing time of the SOC; however, it also causes an increase in power consumption during scan testing because power is directly proportional to the scan frequency [26]. In this section, we study the impact of using high-frequency TAM lines on test power for one of the ITC '02 benchmark circuits. Since power estimation models for the ISCAS benchmark circuits have been published in the literature, we utilize one of the SOC benchmarks that consists of ISCAS circuits.

In [27], the peak power consumption for the ISCAS-85 benchmark circuits is estimated based on the maximum switching activity. Since the ISCAS-85 circuits form the cores in SOC d695, we use the power data reported in [27] for estimating the test power for d695. These power estimates have also been used in [16] to evaluate power-constrained TAM optimization. Although the power estimates in [27] are for functional patterns, we use these values for scan test power as in [16] due to the lack of any additional power data for these circuits. In [28], a Monte Carlo approach is used to estimate the average power consumption for the ISCAS-85 benchmark circuits; we use these power estimates to study the average power consumption in d695 when high frequency TAMs lines are used. The power of a core scheduled in the high-frequency bin is calculated as the frequency factor f times its power for the low-frequency bin. The peak power is measured as peak switching frequency (PSF) per node [16], [27] and the average power is measured in mW [28]. Table 5 shows the peak power and average power data for every core in d695. To remove any bias in the selection of the power values, we also carry out simulations using unit power values for the cores in d695. In these experiments, each embedded cores in d695 is assigned a peak power value of one unit.

The instantaneous peak power $P_{peak}(t)$ of an SOC is given by the sum of the peak powers of the cores being tested at time instant t . The SOC testing time is measured in test clock cycles. Let y_{ik} be a binary variable that takes the value 1 if core i is being tested at test cycle $t = k$, $0 \leq k \leq T$, where T is the total number of test cycles taken for the SOC; else $y_{ik} = 0$. Thus, if the peak power of core i is given by P_i and there are N cores in an SOC, the instantaneous peak power of the SOC is given by:

Fig. 16. (a) Test schedule for d695 with no constrain on power limit. (b) Test schedule for d695 with a peak power limit of 2,000 PSF.

$$P_{peak}(t) = \sum_i^N P_i \cdot y_{it}.$$

The overall peak power P_{peak} of the SOC is the maximum of the instantaneous peak power over all time instants during the testing of the SOC. This implies that:

$$P_{peak} = \max_t P_{peak}(t), 0 \leq t \leq T.$$

If it takes T_i clock cycles to test core i and it takes T clock cycles to test the SOC, the average power consumption of the SOC is calculated as:

$$P_{avg} = \frac{\sum_i^N P_i \times T_i}{T}.$$

In Figs. 13, 14, and 15, we study the trade-off between test time reduction and the increase in SOC test power consumption due to the use of high-frequency TAM lines. Fig. 13 shows the testing time results in clock cycles for d695, for TAM widths varying from $W = 16$ to $W = 64$ in steps of 8. We consider three values for the percentage of high-frequency lines, namely, $n = 0\%$, $n = 25\%$, and $n = 50\%$. We also assume that all high-frequency lines operate at a frequency factor of $f = 2$.

For $W \geq 24$, the testing time decreases with an increase in the number of high-frequency lines. However, for $W = 16$, the testing time for $n = 25\%$ is more than the testing time for $n = 0\%$, because, in this case, the TAM

TABLE 7
Testing Time in μs , with Power Limits of Four Units and Five Units for $n = 0\%$, $n = 25\%$, and $n = 50\%$

$n = 0\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	2146.95	2146.95	0.00	2146.95	0.00
24	1463.55	1635.10	11.72	1463.55	0.00
32	1130.65	1261.50	11.57	1164.65	3.01
40	884.50	973.65	10.08	884.50	0.00
48	743.65	921.00	23.85	815.00	9.59
56	621.05	848.75	36.66	711.50	14.56
64	580.20	772.10	33.07	583.85	0.63
$n = 25\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	2786.10	2786.10	0.00	2786.10	0.00
24	1738.30	1738.30	0.00	1738.30	0.00
32	987.85	1598.35	61.80	987.85	0.00
40	881.20	1401.80	59.08	881.20	0.00
48	623.50	1126.30	80.64	723.70	16.07
56	551.65	1000.85	81.43	661.30	19.88
64	531.25	977.65	84.03	586.75	10.45
$n = 50\%$					
W	T_{p0}	T_{p1}	ΔT_{p1}	T_{p2}	ΔT_{p2}
16	1724.50	2230.85	29.36	1911.10	10.82
24	1131.25	1515.20	33.94	1168.95	3.33
32	801.15	1145.20	42.94	957.75	19.55
40	672.55	996.55	48.17	717.80	6.73
48	551.65	996.55	80.65	622.10	12.77
56	521.70	901.70	72.84	621.10	19.05
64	464.10	901.70	94.29	537.55	15.83

T_{p0} : Testing time with no power limit; T_{p1} : Testing time with power limit of 4 units; T_{p2} : Testing time with power limit of 5 units; $\Delta T_{px} = \frac{(T_{px} - T_{p0}) \times 100}{T_{p0}}$, $x \in \{0, 1\}$

width is very small and, when it is split into two bins, the packing of rectangles into the bins becomes more constrained. From Fig. 14 and Fig. 15, we find that the peak and average power consumption increase with an increase in the number of high-frequency lines for most values of the TAM widths W . There are some notable exceptions however; for several cases, the peak and average power consumption reduces with an increase in n owing to the nature of the test schedule. For example, for $W = 16$ and $n = 25\%$, a lower value for the test time is accompanied by a drop in the average power consumption; a similar trend is seen for $W = 48$, for $n = 25\%$ and $n = 50\%$. This is because the reduction in the availability of the TAM width in the two bins resulted in reduced test parallelism. Since the overall TAM width is divided equally in the two bins, the TAM width in the low-speed bin is reduced and the TAM width in the high-speed bin is increased compared to the case of $n = 25\%$. In this particular case, the reduction in TAM width in the low-speed bin results in a reduction in test parallelism that offsets the increase in power due to higher power consumption in the high-speed bin.

Next, we extend the test scheduling algorithm based on two bins such that the peak power of the SOC is not exceeded at any time during test application. The extension is straightforward, based on a check on power constraints during rectangle insertion in the bin; thus, details are omitted here. We present results for power-constrained test scheduling for d695 in Table 6. The results are presented for

power limits of 2,500 PSF and 3,000 PSF for $n = 25\%$ and $n = 50\%$. Core c7552 has the maximum peak power of 1,144 PSF. In Table 6, we compare the test time (T_{p1} and T_{p2} for power limits of 2,500 PSF and 3,000 PSF, respectively) of the power-constrained test schedule for different values of W to that of unconstrained test schedule (T_{p0}). It is important to note that, in the case of power-constrained test scheduling, we do not attempt to optimize the test schedule for test power. The goal here is minimize the test time under power limits.

From Table 6, we see that the test time for most cases increases compared to the case of unconstrained test scheduling. With a limit on the peak power of the SOC, the test schedules have reduced parallelism and more idle time. Fig. 16 shows the difference in the test schedule for the case of unconstrained test scheduling and constrained test scheduling for a power limit of 2,000 PSF. A small perturbation (the positions of Core 3 and Core 4) is introduced in the test schedule due to power constraints. It is also interesting to note that, for smaller TAM widths ($W < 32$), the test times for the constrained test schedule and unconstrained test schedule are the same. This is because smaller TAM widths inherently impose reduced test parallelism resulting in lower test power in the test schedule.

In Table 6, we also note several nonintuitive results. For unconstrained test scheduling, the test time (T_{p0}) increases for $n = 25\%$ compared to $n = 0\%$ for lower TAM width values of $W = 16$ and $W = 24$. This is due to the reduction

in the availability of total bit width for a given core. It is also interesting to note that, for the constrained test schedules, similar trends are seen even for the higher TAM width values ($W > 32$) between $n = 25\%$ and $n = 50\%$. This is because it is not possible to exploit the availability of the total TAM width in the high-speed bins due to the need for reduced parallelism. Thus, the increased idle time in the high-speed bin and the reduction in the bit width of the lower-speed bin results in an increase in the overall SOC test time.

In Table 7, we present test time results for d695, assuming that each core has a peak power of one unit. We use power limits of four units and five units. We see similar trends in test time variation as seen in Table 6.

From these results, we conclude that, for smaller TAM widths, power constraints do not impact the overall test time of the SOC significantly. For greater TAM widths, it can be advantageous to have a smaller high-speed bin since test parallelism is more limited in the high-speed bin compared to the low-speed bin due to additional power consumption at higher speeds.

7 CONCLUSION

In this paper, we have presented a new technique for dual-speed test scheduling and TAM optimization for SOCs. The proposed approach, which is based on a generalized rectangle packing algorithm, reduces testing time and helps reduce the overall test cost. We have derived a lower bound on the test time using a geometric argument. Experimental results have been presented for several ITC '02 SOC test benchmarks and these results have been compared with the corresponding lower bounds. The testing times obtained using the heuristic rectangle packing method are often close to the lower bounds. These results highlight the improvements that can be obtained using a dual-speed TAM architecture. They also show that the dual-speed architecture should be used judiciously because, in many cases, the advantage of the higher speed TAM is offset by the reduced bitwidth of TAM partitions. Finally, we have examined the impact of using high-frequency TAM lines on the test power. While the test power tends to increase with an increase in the proportion of high-frequency TAM lines, thereby giving rise to interesting trade-off issues, efficient test scheduling can, in some cases, reduce the test power as well as the test time. Conversely, we have also investigated the impact of power constraints on overall SOC test time.

ACKNOWLEDGMENTS

The authors thank Erik Jan Marinissen from Philips Research Labs for helpful pointers to port-scalable testers. This research was supported in part by the US National Science Foundation under grants CCR-9875324 and CCR-0204077. A preliminary version of this paper appeared in the *Proceedings of the IEEE Design, Automation and Test in Europe (DATE) Conference*, pp. 422-427, 2004.

REFERENCES

- [1] Verigy, "Agilent 93000 Flexible Parallel Test Solution," <http://www.verigy.com/content/dav/verigy/Internet/Products/V93000%20SOC%20Series/5989-2598EN.pdf>, 2006.
- [2] A. Khoche, Agilent Corp., private communication, 2004.
- [3] Teradyne Technologies, "Tiger: Advanced Digital with Silicon Germanium Technology," <http://www.teradyne.com/tiger/digital.html>, 2006.
- [4] P. Harrod, "Testing Reusable IP—A Case Study," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 493-498, Sept. 1999.
- [5] E.J. Marinissen et al., "A Structured and Scalable Mechanism for Test Access to Embedded Reusable Cores," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 284-293, 1998.
- [6] P. Varma and S. Bhatia, "A Structured Test Re-Use Methodology for Core-Based System Chips," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 294-302, 1998.
- [7] S.K. Goel and E.J. Marinissen, "Effective and Efficient Test Architecture Design for SOCs," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 529-538, 2002.
- [8] E. Larsson and H. Fujiwara, "Power Constrained Preemptive TAM Scheduling," *Proc. IEEE European Test Workshop (ETW)*, pp. 119-126, May 2002.
- [9] M. Nourani and J. Chin, "Test Scheduling with Power-Time Tradeoff and Hot-Spot Avoidance Using MILP," *Proc. IEEE Computers and Digital Techniques*, pp. 341-355, 2004.
- [10] P. Rosinger, B. Al-Hashimi, and N. Nicolici, "Power Constrained Test Scheduling Using Power Profile Manipulation," *Proc. Int'l Symp. Circuits and Systems (ISCAS)*, vol. V, pp. 251-254, May 2001.
- [11] D. Zhao and S. Upadhyay, "Power Constrained Test Scheduling with Dynamically Varied TAM," *Proc. VLSI Test Symp.*, pp. 273-278, 2003.
- [12] V. Immaneni and S. Raman, "Direct Access Test Scheme—Design of Block and Core Cells for Embedded ASICs," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 488-492, Sept. 1990.
- [13] I. Ghosh, N.K. Jha, and S. Dey, "A Low Overhead Design for Testability and Test Generation Technique for Core-Based Systems," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 50-59, Nov. 1997.
- [14] I. Ghosh, S. Dey, and N.K. Jha, "A Fast and Low Cost Testing Technique for Core-Based System-on-Chip," *Proc. ACM/IEEE Design Automation Conf. (DAC)*, pp. 542-547, June 1998.
- [15] N. Toubia and B. Pouya, "Testing Embedded Cores Using Partial Isolation Rings," *Proc. IEEE VLSI Test Symp. (VTS)*, pp. 10-16, Apr. 1997.
- [16] Y. Huang et al., "Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 74-82, 2002.
- [17] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "On Using Rectangle Packing for SOC Wrapper/TAM Co-Optimization," *Proc. IEEE VLSI Test Symp. (VTS)*, pp. 253-258, 2002.
- [18] E. Larsson and Z. Peng, "An Integrated System-on-Chip Test Framework," *Proc. Design, Automation, and Test in Europe (DATE) Conf.*, pp. 138-144, 2001.
- [19] E. Larsson and Z. Peng, "An Integrated Framework for the Design and Optimization of SOC Test Solutions," *J. Electronic Testing: Theory and Applications*, vol. 18, pp. 385-400, 2002.
- [20] A. Sehgal, V. Iyengar, and K. Chakrabarty, "SOC Test Planning Using Virtual Test Access Architectures," *IEEE Trans. VLSI Systems*, vol. 12, pp. 1196-1202, Dec. 2004.
- [21] Q. Xu and N. Nicolici, "Time/Area Tradeoffs in Testing Hierarchical SOCs with Hard Megacores," *Proc. IEEE Int'l Test Conf.*, pp. 1196-1202, 2004.
- [22] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, "Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores," *J. Electronic Testing: Theory and Applications*, vol. 18, no. 2, pp. 213-230, Apr. 2002.
- [23] K. Chakrabarty, "Optimal Test Access Architectures for System-on-a-Chip," *ACM Trans. Design Automation of Electronic Systems*, vol. 6, pp. 26-49, Jan. 2001.
- [24] E.J. Marinissen, V. Iyengar, and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SOCs," *Proc. IEEE Int'l Test Conf. (ITC)*, pp. 519-528, 2002, <http://www.extra.research.philips.com/itc02socbenchm/>.
- [25] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Test Power: A Big Issue in Large SOC Designs," *Proc. DELTA Workshop*, pp. 447-449, 2002.
- [26] N. Nicolici and B.M. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*. Kluwer Academic, 2003.
- [27] M.S. Hsiao, E.M. Rudnick, and J.H. Patel, "Effects of Delay Models on Peak Power Estimation of VLSI Sequential Circuits," *Proc. Int'l Conf. Computer-Aided Design (ICCAD)*, pp. 45-51, 1997.

- [28] R. Burch, F. Najm, P. Yang, and T. Trick, "A Monte Carlo Approach for Power Estimation," *IEEE Trans. VLSI Systems*, vol. 1, pp. 63-71, Mar. 1993.
- [29] Y. Zorian, "A Distributed BIST Control Scheme for Complex VLSI Devices," *Proc. IEEE VLSI Test Symp. (VTS)*, pp. 6-11, Apr. 1993.
- [30] A. Sehgal and K. Chakrabarty, "Efficient Modular Testing of SOCs Using Dual-Speed TAM Architectures," *Proc. Design, Automation and Test in Europe (DATE)*, pp. 422-427, 2004.

Anuja Sehgal received the BE degree in electronics engineering from the Ramrao Adik Institute of Technology, University of Mumbai, India, in 2001. She received the MS and PhD degrees in electrical and computer engineering from Duke University in 2003 and 2005, respectively. Her research interests lie in test planning and test cost reduction for digital, mixed-signal, and hierarchical system-on-chips. She is a recipient of the best speaker award at the 2004 North Atlantic Test Workshop. She is also a recipient of a best paper award at the 2005 IEEE International Conference on Computer Design. She is currently a senior design engineer at Advanced Micro Devices, Sunnyvale, California. She is a member of the IEEE and the IEEE Computer Society.

Krishnendu Chakrabarty received the BTech degree from the Indian Institute of Technology, Kharagpur, in 1990, and the MSE and PhD degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively, all in computer science and engineering. He is now an associate professor of electrical and computer engineering at Duke University. Dr Chakrabarty is a recipient of the US National Science Foundation Early Faculty (CAREER) award and the US Office of Naval Research Young Investigator award. His current research projects include design and testing of system-on-chip integrated circuits, design automation of microfluidics-based biochips, microfluidics-based chip cooling, and distributed sensor networks. He has authored four books: *Microelectrofluidic Systems: Modeling and Simulation* (CRC Press, 2002), *Test Resource Partitioning for System-on-a-Chip* (Kluwer, 2002), *Scalable Infrastructure for Distributed Sensor Networks* (Springer, 2005), and *Digital Microfluidic Biochips: Synthesis, Testing, and Reconfiguration Techniques* (CRC Press, 2006), and edited the book volumes *SOC (System-on-a-Chip) Testing for Plug and Play Test Automation* (Kluwer 2002) and *Design Automation Methods and Tools for Microfluidics-Based Biochips* (Springer, 2006). He has published more than 230 papers in journals and refereed conference proceedings. He holds a US patent in built-in self-test and he has a pending US patent on sensor networks. Dr. Chakrabarty is a Distinguished Visitor of the IEEE Computer Society for 2005-2007 and a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006-2007. He is a recipient of best paper awards at the 2005 IEEE International Conference on Computer Design and the 2001 IEEE Design, Automation and Test in Europe (DATE) Conference. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany. Dr Chakrabarty is an associate editor of the *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *IEEE Transactions on VLSI Systems*, *IEEE Transactions on Circuits and System I*, and *ACM Journal on Emerging Technologies in Computing Systems* and he is an editor of the *Journal of Electronic Testing: Theory and Applications (JETTA)*. He is a member of the editorial board for *IEEE Design and Test of Computers*, *Sensor Letters* and the *Journal of Embedded Computing* and he serves as a subject area editor for the *International Journal of Distributed Sensor Networks*. In the recent past, he has also served as an associate editor of the *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*. He is a senior member of the IEEE, a member of the IEEE Computer Society, a senior member of the ACM and ACM SIGDA, and a member of Sigma Xi. He serves as vice chair of technical activities on the IEEE's Test Technology Technical Council and is a member of the program committees of several IEEE/ACM conferences and workshops. He served as the program chair for the 2005 IEEE Asian Test Symposium.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.