

# Test Planning for the Effective Utilization of Port-Scalable Testers for Heterogeneous Core-Based SOCs<sup>1</sup>

Anuja Sehgal and Krishnendu Chakrabarty

Department of Electrical & Computer Engineering

Duke University, Durham, NC 27708, USA

{as,krish}@ee.duke.edu

## Abstract

Many SOCs 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 bitwidth 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 work that use a single scan data rate for all the embedded cores.

## 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 [8, 16, 20], but due to differences in performance levels, design styles, and scan insertion methods, they also differ from each other in their maximum scan clock frequencies [23]. The difference in scan clock frequencies between embedded cores can also arise due to the integration of cores, which have been derived from different older-generation SOCs, into a single current-generation SOC [6]. 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 [7]. Examples of such ATEs include the Agilent 93000 series [1, 3] and the Tiger system from Teradyne [2], 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 [8, 25]. 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 [9, 11, 12, 15]. 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 [13]. Therefore, efficient heuristic techniques have been developed for TAM optimization [9, 10, 11, 12, 15, 24]. 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 [9, 12, 15], we base our TAM design on a Test Bus model.

The testing of embedded cores with multiple scan data rates was recently addressed in [19, 21, 22]. However, no limits are set in prior work on the scan frequencies of the cores, and the number of available ATE channels operating at the two data rates are fixed *a priori*. Such an assumption is too restrictive in practice. In this paper, 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.

In this paper, we optimize the TAM architecture for a set of cores with different maximum scan data rates. We compare this approach 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 the TAM partitions, the TAM frequencies, and the 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 [18]. We also present experimental results for several ITC'02 benchmark SOCs.

The rest of this paper 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

<sup>1</sup>This research was supported in part by the National Science Foundation under grants CCR-9875324 and CCR-0204077.

SOC testing time. In Section 3, we present a scalable heuristic approach to solve this problem. In Section 4, we present experimental results for several ITC'02 benchmark SOCs. Finally, we present conclusion and directions for future work in Section 5.

## 2 TAM Architecture Optimization

**Problem  $P_{\text{port-scalable}}$ :** Given the test data parameters for  $N$  embedded cores in an SOC, the maximum scan frequency  $f_i^*$  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, such that (a) the frequency of each TAM partition does not exceed the maximum frequency of any core assigned to that TAM partition, (b) the sum of the widths of the TAM partitions does not exceed the total TAM width  $W$ , and (c) the overall test time of the SOC is minimized.  $\square$

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 prior to test planning. These parameters are used to design a wrapper for the cores. The *DesignWrapper* algorithm from [13] 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 [13]. 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:

$$T_i(w_j) = \lceil (\max\{s_i, so_i\} \cdot p_i + \min\{s_i, so_i\}) / w_j \rceil + p_i,$$

where  $s_i$ ,  $so_i$ , and  $p_i$  are the maximum scan-in time, maximum scan-out time, and the number of test patterns for Core  $i$  respectively [17]. 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  $\mu\text{s}$  if  $f$  is given in MHz. The overall test time of an SOC is the maximum of the test time over all the 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**

1.  $\sum_j^B x_{ij} = 1$ ,  $1 \leq i \leq N$ , i.e., every core is connected to only one TAM partition;
2.  $\sum_j^B w_j = W$ , i.e., the sum of the widths of the TAM partitions does not exceed  $W$ ;
3.  $f_j = \min_i \{ \{ f_i^* \cdot x_{ij} \} \setminus \{0\} \}$ , where  $\{A \setminus \{0\}\}$  denotes the set difference between  $A$  and  $\{0\}$ ;
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,  $w_{max} = 64$  [4].

Next we consider a special case of  $P_{\text{port-scalable}}$ , which we refer to as  $P_{\text{port-scalable}}^*$ . A solution to this problem 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.

$P_{\text{port-scalable}}^*$  can also be shown to be  $\mathcal{NP}$ -hard using the techniques presented in [5]. However, it can be solved exactly for small problem instances using an ILP model. 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  $\tau_i^* = 1/f_i^*$  be the minimum possible period of the scan clock for Core  $i$ . A mathematical programming model for  $P_{\text{port-scalable}}^*$  can be derived as follows.

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

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

Note that the above objective function is non-linear 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)  $-\tau_{ij} + y_{ij} \leq 0$ ; (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**

1.  $\sum_j^B w_j = W$ ;
2.  $x_{ij} \cdot \tau_i^* - \tau_{ij} \leq 0$ ;
3.  $f_{1j} = f_{2j} = \dots = f_{Nj}$ ,  $1 \leq j \leq B$ ;
4.  $y_{ij} - T_{max} \leq 0$ ;
5.  $-\tau_{ij} + y_{ij} \leq 0$ ;
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 [13] to solve  $P_{\text{port-scalable}}$ . The  $P_{\text{PAW\_Enumerate}}$  procedure described in [13] enumerates unique TAM partitions for given values of  $B$  and  $W$ . In this work, we use the  $P_{\text{PAW\_Enumerate}}$  procedure with the ILP model for  $P_{\text{port-scalable}}^*$ .

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 the width to which the bin is filled is minimized. Even though this problem statement addresses a flexible-width TAM architecture as in [12], 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  $f \leq f_i^*$ . It follows that  $\mathcal{R}_i^{min} = \min_i \{ R_i(w, f) \}$ ,  $1 \leq w \leq W$ , and  $0 < f \leq f_i^*$ . We next show that the minimum-

area rectangle for each core is a rectangle of height 1 and width of  $T_i(1, f_i^*)$ , i.e.  $\mathcal{R}_i^{min} = R_i(1, f_i^*)$ .

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

$$T_i(w) \geq ((\max(s_{i1}, s_{i2}) \cdot p_i + \min(s_{i1}, s_{i2}))/w + p_i) \times 1/f.$$

The numerator of the first term inside the parenthesis on the right-hand side of the above 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  $R_i(1, f_i^*) = (v + p_i)/f_i^*$ , where  $v$  is the total test data volume for the core. Comparing the expressions for  $R_i(w, f)$  and  $R_i(1, f_i^*)$  for  $w > 1$  and  $f < f_i^*$ , we see that  $(\lceil v/w \rceil + p_i) \times w > v/f_i^*$  and  $p_i \times w/f > p_i/f_i^*$ . Thus,  $R_i(w, f) > R_i(1, f_i^*)$ ,  $\forall w > 1$ , and  $f < f_i^*$ . We also note that the minimum-area rectangles for the cores might not fill the bin of area  $T \times W$  perfectly owing to the variations in the sizes of the 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  $\Delta$ , where  $\Delta \geq 0$ . 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 and the sum of all the unfilled space in the bin. Thus,

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

which implies that

$$T \geq \sum_{i=1}^N R_i(1, f_i^*)/W \quad (1)$$

Let the lower bound obtained from Equation (1) be denoted by  $LB_1$ . We obtain another lower bound  $LB_2$  from [5] as follows:  $LB_2 = \max_i T_i(w, f_i^*)$ ,  $1 \leq i \leq N$ .  $LB_1$  is more accurate for smaller TAM widths. However, for larger values of  $W$ ,  $LB_2$  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  $P_{port-scalable}$ . The algorithm starts with an initial solution and then improves it in an iterative manner using four iterative descent procedures (IDP). 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 the 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 the TAM partitions.
3. Four iterative descent procedures are nested together in a loop; these are executed in an iterative manner, as long as the

---

#### 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
```

---

Figure 1. Pseudocode for the proposed approach.

testing time decreases with each iteration. The four IDPs jointly optimize the TAM partition widths, the TAM partition frequencies, and the assignment of cores to TAM partitions.

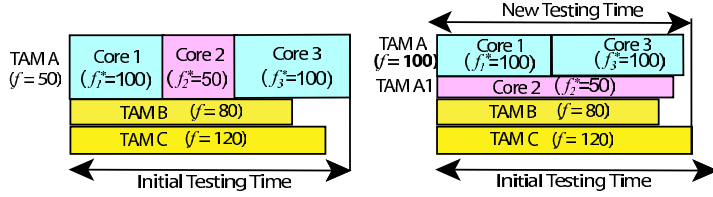
#### 3.1 Initial solution

The *TAM\_Initialize* procedure creates  $B$  partition, 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 the 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; e.g., TAM partition 1 is assigned the frequency of the core with index 1. These two steps ensure that the TAM partition frequencies are evenly distributed between the frequency of the core with the lowest maximum scan frequency and the core with the highest maximum scan frequency. 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*



**Figure 2.** Illustration of procedure *Split\_TAMs*.

cores, which are the cores that have the minimum scan frequency among all cores assigned to that TAM partition. The *Split\_TAMs* procedure displaces frequency-bottleneck cores from the 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  $\delta$  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  $\delta$  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 the scan frequencies of the TAM partitions by shuffling the core assignments to TAM partitions. (The widths of the 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 the 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 the 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 the TAM partitions. It removes the slack TAM wires from non-bottleneck 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 non-bottleneck 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 non-bottleneck 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 the cores belonging to them by offering a greater bitwidth. 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 merged with  $TPART_{max}$ .
3. Select the 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 above three steps.

The overall worst-case complexity of the heuristic procedure is  $O(B_{max}^2 N + N \log_2 N)$ .

## 4 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 baseline case, the cores are

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}	{17,16}	{20,20}	{27,21}	{31,25}	{33,31}
$\{f_j\}$	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}	{80,40}
$T$	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}
$\Delta T_{ILP}(\%)$	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
$\{f_i^*\}$	{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}
$T$	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}
$\Delta T_{ILP}(\%)$	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
$\{f_i^*\}$	{40, 80, 40, 40, 80, 80, 40}						

$$\Delta T_{ILP}: \frac{T - T_{ILP}}{T_{ILP}} \times 100; \Delta T_b: \frac{T - T_b}{T_b} \times 100;$$

**Table 1.** Testing times for d695 (in  $\mu s$ ) and a586710 (in  $ms$ ).

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 [14]. 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 1. The set of maximum scan frequencies for the cores is shown in the last row of the tables (element  $i$  if the set corresponds to the maximum scan frequency of Core  $i$ ). The lower bounds are obtained using (3) and by assuming that every core is tested at its maximum scan frequency. The ILP-based approach (indicated by  $T_{ILP}$ ) outperforms the heuristic ( $T$ ) and the baseline case ( $T_b$ ) for many cases. However, the run time 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 run time 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, i.e., 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.

Next we present results for SOCs p34392, p22810, and p93791 in Table 2. 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. The reduction in testing time using the proposed approach over the baseline case is as high as 50%. Since there are five (nine) distinct frequencies, the number of TAM partitions is limited to five (nine) in the baseline case. However, from these experimental results, we observe that even when the baseline case has the same number of TAM partitions as the proposed approach, the proposed approach results in lower SOC test times. This implies that the frequencies of the TAM partitions and the assignment of the cores to the TAM partitions have a major impact on the overall test time of the SOC. 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 the 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 the benchmarks for TAM width  $W$  to the test time 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.

p34392 (five scan frequencies: 40 MHz to 200 MHz)				
$W$	$LB_T$	$T$	$T_b$	$\Delta T$
16	12724	14683	19209	-23.56
24	8757	12514	13089	-4.33
32	6774	10806	10976	-1.57
40	5584	7707	9438	-18.34
48	4790	7707	9438	-18.34
56	4538	7707	9425	-18.22
64	4538	7617	7619	-0.026
$\{f_i^*\}$ (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)				
$W$	$LB_T$	$T$	$T_b$	$\Delta T$
16	13554	22567	29464	-23.40
24	9267	14894	22372	-33.42
32	7124	12613	15153	-16.76
40	5838	12607	14125	-10.74
48	5148	8307	13367	-37.85
56	5148	7435	9341	-20.40
64	5148	7435	9294	-20.00
$\{f_i^*\}$ 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)				
$W$	$LB_T$	$T$	$T_b$	$\Delta T$
16	86175	110790	183186	-39.52
24	57727	66186	139457	-52.54
32	43503	54795	94878	-42.25
40	34969	48016	72748	-33.99
48	29279	36395	61192	-40.52
56	25215	30686	56578	-45.76
64	22168	34872	47644	-26.80
$\{f_i^*\}$ values (MHz)	{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}			

$$\Delta T: \frac{T - T_b}{T_b} \times 100.$$

**Table 2.** Testing time (in  $\mu s$ ) for p34392, p22810 and p93791.

## 5 Conclusion

We have presented a new test planning technique for core-based SOC designs 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, the 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 a baseline case for this problem. We have evaluated the proposed heuristic by comparing it to the ILP-based approach, the baseline case, and provable lower bounds on the test time. The heuristic approach performs significantly better than the baseline case, and in some cases it leads to the same test time as the optimal, but computationally-expensive, ILP approach. Experimental results for several ITC'02 benchmark circuits demonstrate the effectiveness of the proposed approach.

## References

- [1] Agilent Technologies. Winning in the SOC market, available online at: <http://cp.literature.agilent.com/litweb/pdf/5988-7344EN.pdf>
- [2] Teradyne Technologies. Tiger: Advanced digital with silicon germanium technology. <http://www.teradyne.com/tiger/digital.html>
- [3] Agilent Corporation. Agilent 93000 multi-port testing user manual, revision 4.0.0, 2002.
- [4] A. Khoche, Agilent Corporation. *Private communication*, 2004.
- [5] K. Chakrabarty. Optimal test access architectures for system-on-a-chip. *ACM Trans. Design Automation of Electronic Systems*, vol. 6, pp. 26–49, January 2001.
- [6] V. Chickermane et al. A building block BIST methodology for SOC designs: A case study. *Proc. ITC*, pp. 111–120, 2001.
- [7] R. Dorsch et al. Adapting an SoC to ATE concurrent test capabilities. *Proc. ITC*, pp. 1169–1175, 2002.
- [8] S. K. Goel et al. Test Infrastructure Design for the Nexperia<sup>TM</sup> Home Platform PNX8550 System Chip. *Proc. DATE Designers' Forum*, pp. 108–113, 2004.
- [9] S. K. Goel and E. J. Marinissen. Effective and efficient test architecture design for SOC designs. *Proc. ITC*, pp. 529–538, 2002.
- [10] P. T. Gonciari et al. Addressing useless test data in core-based system-on-a-chip test. *IEEE Trans. CAD*, vol. 22, pp. 1568–1580, November 2003.
- [11] Y. Huang et al. Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm. *Proc. ITC*, pp. 74–82, 2002.
- [12] V. Iyengar et al. Test access mechanism optimization, test scheduling and tester data volume reduction for system-on-chip. *IEEE Trans. Computers*, vol. 52, pp. 1619–1632, December 2003.
- [13] V. Iyengar et al. Test wrapper and test access mechanism co-optimization for system-on-chip. *JETTA*, vol. 18, pp. 213–230, April 2002.
- [14] V. Iyengar et al. Efficient test access mechanism optimization for system-on-chip. *IEEE Trans. CAD*, vol. 22, pp. 635–643, May 2003.
- [15] E. Larsson and Z. Peng. An integrated framework for the design and optimization of SOC test solutions. *JETTA*, vol. 18, pp. 385–400, August/October 2002.
- [16] X. Lin and R. Thompson. Test generation for designs with multiple clocks. *Proc. DAC*, pp. 662–667, 2003.
- [17] E. J. Marinissen et al. A structured and scalable mechanism for test access to embedded reusable cores. *Proc. ITC*, pp. 284–293, 1998.
- [18] E. J. Marinissen et al. A set of benchmarks for modular testing of SOC designs. *Proc. ITC*, pp. 519–528, 2002.
- [19] Q. Xu and N. Nicolici. Time/Area tradeoffs in testing hierarchical SOC designs with hard mega-cores. *Proc. ITC*, pp. 1196–1202, 2004.
- [20] J. Schmid and J. Knablein. Advanced synchronous scan test methodology for multi clock domain ASICs. *Proc. VTS*, pp. 106–113, 1999.
- [21] A. Sehgal et al. SOC Test Planning Using Virtual Test Access Architectures. *IEEE TVLSI*, vol. 2, pp. 1262–1276, 2004.
- [22] A. Sehgal and K. Chakrabarty. Efficient Modular Testing of SOC designs using Dual-Speed TAM Architectures. *Proc. DATE*, pp. 422–427, 2004.
- [23] H. Vranken et al. ATPG padding and ATE vector repeat per port for reducing test data volume. *Proc. ITC*, pp. 1069–1078, 2003.
- [24] D. Zhao and S. Upadhyaya. Power constrained test scheduling with dynamically varied TAM. *Proc. VTS*, pp. 273–278, 2003.
- [25] Y. Zorian et al. Testing embedded-core-based system chips. *IEEE Computer*, vol. 32, pp. 52–60, June 1999.