Practical test architecture optimization for system-on-a-chip under floorplanning constraints

Sugihara, Makoto
Institute of Systems & Information Technologies/KYUSHU

Murakami, Kazuaki
Department of Computer Science and Communication Engineering, Kyushu University

Matsunaga, Yusuke
Department of Computer Science and Communication Engineering, Kyushu University

Practical Test Architecture Optimization for System-on-a-Chip under Floorplanning Constraints

[†]Makoto Sugihara, [‡]Kazuaki Murakami and [‡]Yusuke Matsunaga

†Institute of Systems & Information Technologies/KYUSHU

2-1-22 Momochihama, Sawara-ku, Fukuoka 814-0001 Japan

‡Department of Computer Science and Communication Engineering, Kyushu University

6-1 Kasuga-Koen, Kasuga 816-8580 Japan

sugihara@isit.or.jp

Abstract

In testing system-on-a-chip (SOC), external pins for test are getting more and more precious hardware resources because the number of external pins is strongly restricted. Cores, which are basic components to build SOCs, are tested via test access mechanisms (TAMs) such as a test bus architecture. When cores are tested via TAMs, test stimuli and test responses for cores have to be transported over these TAMs. There is often the difference between the numbers of input/output ports of cores and the widths of TAMs. This difference causes the serialization of test patterns. It is probable that some parts of TAMs are unused because of the difference. This is a wasteful usage of TAMs. Test scheduling should be done in order to remove such a wasteful usage of TAMs. In this paper, a novel and practical test architecture optimization is proposed such that test time is minimized with floorplanning constraints abided. In this proposal, the computation time for the optimization can be alleviated by floorplanning manipulation. Several experimental results to this optimization are shown to validate this proposal using a commercial LP solver.

1 Introduction

Recent significant advances in semiconductor technology have been increasing the number of transistors available on a chip dramatically. System designers can now build a large system on a single chip as a system-on-a-chip (SOC). They often use multiple pre-designed and pre-verified blocks, hereafter called cores, to reduce the time required for design and verification.

Testing SOCs is getting harder and harder while semiconductor technology improves. Abundant and deep functions per chip make it hard to test SOCs like a former system-on-a-board. The increasing test cost for SOC can be reduced by test time reduction and test time reduction is, therefore, one of major research challenges on SOC. Test access mechanisms (TAMs) and test wrappers were proposed as important components of an SOC test access architecture [13]. TAMs deliver pre-computed test sequences to cores on an SOC, while test wrappers interface between these test data and cores [14].

Test scheduling has been researched to reduce the test time of digital systems [1, 3–5, 7, 9–12]. In [4, 10–12], the combination of BIST and external test is discussed as well as test scheduling. In [4, 5], the relationship between

testing time and TAM widths using ILP was examined, and TAM width optimization under power and routing constraints was studied in [3]. However, the problem of effective test width adaptation in test wrappers was not addressed in [3–5, 10, 11]. A test wrapper design for embedded cores was presented in [8]. However, the issue of reducing the TAM width required for a test wrapper was not addressed. In [7], the problem of effective test width adaptation in test wrappers was addressed. However the problem of floorplanning is not addressed. Floorplanning must be taken into account in order to apply the test architecture optimization to practical SOCs. Moreover, the size of problems solved in [7] is too small to validate the applicability of the approach to practical designs. In this paper, a practical test architecture optimization is proposed which can be done under floorplanning constraints within short computation time. In this proposed optimization, the floorplanner is supposed to determine topological placements and correlations between cores on SOCs.

The remainder of this paper is organized as follows: In Section 2, core clustering for floorplanning is introduced and a mathematical model is shown under which the numbers and widths of TAMs are constant. Some examples of the optimizations which have been solved under this modeling by an LP solver are shown. In Section 3, the restriction of the constant numbers and widths of TAMs is removed and computational complexity to this problem is discussed. In Section 4, several lemmas and theorems are introduced and an efficient algorithm is discussed to seek for a faster computation than the one shown in Section 3. Section 5 concludes this paper with a summary.

2 Core Clustering for Floorplanning

In this section, we discuss a mathematical modeling of test architecture optimization for SOCs with floorplanning constraints. Iyengar et al. presented a research on wrapper/TAM co-optimization in [7] but the mathematical model defined in [7] has two problems. The first problem is that the size of problems which can be solved is too small to be applied to practical SOCs. The next problem is that topological locality of cores are ignored. The mathematical model in [7] assumed that it is discretionary to assign cores to TAMs. Such an assumption is too impractical to apply the optimization to SOCs. Topological locality must be formulated to optimize a test architecture and minimize

test time of practical SOCs.

Floorplanning makes cores topologically local on SOCs and so it is quite natural that floorplanning should determine several sets of TAMs and all cores should be clustered to the sets of TAMs with the topologically locality abided. Let us define a *group* in order to formulate this topological locality. A *group* is given a corresponding set of TAMs and consists of cores which can share the given TAMs of the group. Every core can be assigned to the TAMs of the groups as far as the core belongs to the group.

Now let us discuss core clustering to groups under floorplanning constraints. It is quite probable that a core can be assigned to several groups if they are topologically close. Here let us assume that any core may be assigned into an only group among possible ones. A Venn diagram for the core clustering is shown in Figure 1. In this example, Core $i \in G_1$ is reachable to TAM set B_1. Core

Figure 1. Venn Diagram for Core Clustering.

$i \in G_1 \cap G_2$ is reachable to both TAM sets B_1 and B_2. And Core $i \in G_1 \cap G_2 \cap G_3$ is reachable to any of TAM sets B_1, B_2 and B_3. Such a core must be assigned to optimal group in order to minimize test time of the SOC. Assignment of such cores to groups and TAMs must be determined in test architecture optimization process.

The TAM sets should be carefully made in compliance with floorplanning. Once TAM sets are determined in all groups, all cores which are accessible to each of TAM set B_j are classified into Group G_j. Thus Group G_j consists of cores which are reachable to TAM set B_j of Group G_j. Now 0-1 integer variable x_{ijk} is introduced as follows to formulate the test time of the SOC.
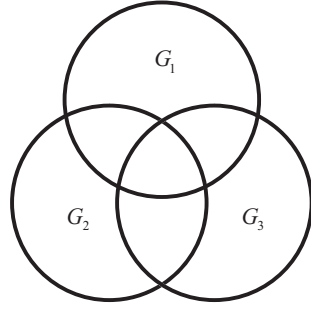
$$x_{ijk} = \begin{cases} 1 & \text{if Core } i \text{ is assigned to the } k\text{th TAM of Group } G_j, \\ 0 & \text{otherwise.} \end{cases}$$

The test time T_{jk} on the kth TAM of Group G_j is

$$T_{jk} = \sum_i T_{W_i}(w_{jk}) \cdot x_{ijk}, \tag{1}$$

where w_{jk} is the width of the kth TAM of Group j and $T_{W_i}(w)$ is testing time for Core i when the width of TAM is w. T_{W_i} can be obtained by solving bin packing problems [2, 7]. For example, T_{W_i} for Core 5 of d695 is shown in Figure 2. The x-axis corresponds to w and the y-axis does to T_{W_i}. $T_{W_i}(w)$ is a monotonically decreasing function between 1 and a upper bound of w. Note that Theorem 1 introduced in [7] is incorrect. The incorrectness of the theorem is detailed in Appendix A. The test time T_j for Group G_j is the maximum one among test times on all TAMs, and formulated as follows.

$$T_j = \max_k T_{jk}$$

The test time for the SOC is the maximum one among test times on all groups, and formulated as follows.

$$T = \max_j T_j$$

Figure 2. Test times for Core 5.

A core must be assigned to one only core, and so 0-1 integer variables must satisfy the following constraints.

$$\sum_{j,k} x_{ijk} = 1, 1 \le i \le N_C,$$

where N_C is the number of cores on the SOC. A mathematical programming model for this problem is shown as follows.
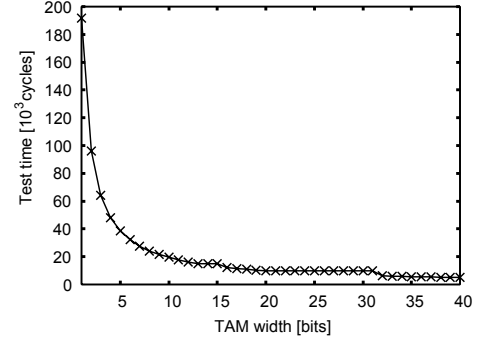
Minimize T, subject to

1. $T \ge \sum_i T_{W_i}(w_{jk}) \cdot x_{ijk}, 1 \le j \le M, 1 \le k \le |B_j|$.
2. $\sum_{j,k} x_{ijk} = 1, 1 \le i \le N_C$.

The number of variables and constraints is a measure of computational complexity. Let the number of TAMs for all groups be N_B. The number of variables for x_{ijk} is $N_C N_B = O(N_C N_B) = O(N_C{}^2)$. Note that the upper bound of N_B is N_C because it is meaningless that there are more TAMs than cores. Floorplanner tool is discretionary to determine this number of groups and TAMs as far as topological locality is abided. The number of constraints is $N_B + N_C \le 2N_C = O(N_C)$.

A hypothetical floorplanning for d695, which is an ITC'02 benchmark SOC, is shown in Table 1. This table shows the number of input and output ports, the number of scan chains, the maximum scan chain length in each module, the number of patterns, and possible assignments to group. The line "Possible assignment to groups" in this table shows which group each core can be assigned to in compliance with floorplanning. These possible assignments mean that $G_1 = \{1, 2, 3, 4, 9\}$, $G_2 = \{2, 3, 4, 5, 6, 7, 8\}$, $G_3 = \{1, 6, 7, 8, 9\}$. Moreover, $G_1 \cap G_2 = \{2, 3, 4\}$, $G_2 \cap G_3 = \{6, 7, 8\}$, $G_3 \cap G_1 = \{1, 9\}$ and

Table 1. Possible core assignments on a hypothetical floorplanning for d695.

Core	1	2	3	4	5	6	7	8	9	10
# Inputs	32	207	34	36	38	62	77	35	35	28
# Outputs	32	108	1	39	304	152	150	49	320	106
# Scan chains	0	0	1	4	32	16	16	4	32	32
Maximum scan chain length	-	-	32	54	45	41	34	46	54	55
# Patterns	12	73	75	105	110	234	95	97	12	68
Possible assignment to groups	{1}	{1}	{2}	{1,2}	{1,2}	{2,3}	{2,3}	{2,4}	{3,4}	{3,4}

$G_1 \cap G_2 \cap G_3 = \phi$. Core assignments should be analyzed and sought out by floorplanner tool. This paper does not cover how a floorplanner tool works for test architecture optimization of SOCs. However, the implementation of the floorplanner tool would be simple. This is a future work for this research. In Table 1, Core 4 can be assigned to either Group 1 or 2. Such cores which can be assigned to one among multiple groups must be assigned to the optimal group so that test time for the SOC is minimized. Table 2 shows optimal core assignments for several TAM width configurations. In these optimization, a commercial LP solver was used [6] and computation times were too short to measure. Column "Total TAM width" in Table 2 is the summation of all widths of TAMs, in other words, the total available TAM width to the SOC. Column "Test time without FP constraints" is the minimal test time when the floorplanning constraint is not taken into account. These test times are same as in [7]. Column "Test time with FP constraints" is the minimal test time sought out by solving the problem shown in this section. Column "TAM width configuration" shows how many TAMs each group has and how many bits the width of each TAM is. For example, "{1}, {1, 2}, {2}" means that there are three groups and there are a 1-bit TAM for Group 1, a 1-bit TAM and a 2-bit TAM for Group 2 and a 2-bit TAM for Group 3. In Column "Assignment to Group", the numbers mean which group each core is optimally assigned to. In Column "Assignment to TAM", the numbers similarly mean which TAM each core is optimally assigned to. For example, in the first configuration for 6 bits, a 1-bit TAM is assigned to Group 1, a 1-bit TAM and a 2-bit TAM to Group 2, and a 2-bit TAM to Group 3. Cores 1, 6, 8 and 9 are optimally assigned to Group 3, Cores 2, 3, 4, 5, and 7 to Group 2 and Core 10 to Group 1. Cores 2, 3, 4 and 5 are optimally assigned to the 2-bit TAM of Group 2 and Core 7 to the 1-bit TAM of Group 2. The minimal test times with and without floorplanning constraints are 120188 and 110641 respectively.

The test architecture optimization with floorplanning constraints resulted in a successful test design at the expense of about 8.63% test time rise for a 6-bit configuration.

3 Optimal Partitioning of TAM Width

In Section 2, a test architecture optimization was discussed for given cores, groups, possible core assignments groups, and a TAM width configuration. TAM width configuration was constant in the previous section but the number of TAMs and their TAM widths may be variable so that shorter test time is achieved. For example, there are three TAM width configurations for 12 bits of total TAM width as shown in Table 2. The number of TAMs and their widths may be variables like these examples so that test time is reduced. In this section, a generalized mathematical modeling

Table 2. Optimal test architecture under FP constraints.

Total TAM width	Test time without FP constraints	Test time with FP constraints	TAM width configuration	Assignment to group	Assignment to TAM	Δt
3	220288	251480	{1}, {1}, {1}	(1,1,2,1,2,3,3,2,1,1)	(1,1,1,1,1,1,1,1,1,1)	14.16%
6	110641	120188	{1}, {1,2}, {2}	(3,2,2,2,2,3,2,3,3,1)	(1,2,2,2,2,1,1,1,1,1)	8.63%
6	110641	191874	{1}, {1,1}, {3}	(3,1,2,2,2,3,3,3,1,3)	(1,1,1,1,2,1,1,1,1,1)	73.42%
9	73972	95992	{1}, {2,2}, {2,2}	(1,2,2,2,2,3,2,2,1,3)	(1,1,1,1,2,2,1,1,1,1)	29.77%
9	73972	84207	{1}, {3,3}, {2}	(3,1,2,1,2,2,2,2,3,3)	(1,1,2,1,2,1,1,2,1,1)	13.84%
12	55993	60128	{2}, {4,4}, {2}	(3,2,2,2,2,2,3,3,3,1)	(1,2,2,2,1,2,1,1,1,1)	7.38%
12	55993	64070	{2}, {2,3,3}, {2}	(3,2,1,2,2,2,3,2,3,1)	(1,1,1,1,3,2,1,1,1,1)	14.43%
12	55993	62248	{2}, {3,4}, {3}	(1,2,2,1,2,2,3,2,1,3)	(1,2,2,1,2,1,1,2,1,1)	11.17%

in which the restrictions of the constant number and width of TAMs are removed, is shown.

When TAM k of Group j has l of width, the test time for TAM k of Group j is formulated the same way as Equation (1) as follows.

$$T_{jk} = \sum_i T_{W_i}(W_{jk}) \cdot x_{ijk} = \sum_{i,l} T_{W_i}(l) \cdot \delta_{jkl} \cdot x_{ijk}, \quad (2)$$

where

$$\delta_{jkl} = \begin{cases} 1 & \text{the width of } k\text{th TAM of Group } j \text{ is } l, \\ 0 & \text{otherwise.} \end{cases}$$

Any core must be assigned to one only TAM and the following constraint can be introduced.

$$\sum_{j,k} x_{ijk} = 1.$$

Any TAM must have one only width and the following constraint can be introduced.

$$\sum_l \delta_{jkl} = 1.$$

In Equation (2), the term $\delta_{jkl} \cdot x_{ijk}$ is non-linear and that is replaced to linearize Equation (2) by variable y_{ijkl}. Variable y_{ijkl} has the following constraints.

$$\begin{aligned} \delta_{jkl} + x_{ijk} - 2y_{ijkl} &\geq 0, \\ \delta_{jkl} + x_{ijk} - y_{ijkl} &\leq 1. \end{aligned}$$

Let the total bits for all TAMs be W. The following constraint is introduced.

$$\sum_{j,k,l} l \cdot \delta_{jkl} = W.$$

The test time for the SOC is equal to the test time which is maximum among ones on all TAM and shown as follows.

$$T = \max_{j,k} T_{jk} = \max_{j,k} \sum_{i,l} T_{W_i}(l) \cdot y_{ijkl}.$$

A mathematical programming model for this problem is shown as follows.

Minimize T, subject to

1. $T \geq \sum_{i,l} T_{W_i}(l) \cdot y_{ijkl}, \forall j, k.$
2. $\sum_{j,k} x_{ijk} = 1, \forall i.$
3. $\sum_l \delta_{jkl} = 1, \forall j, k.$
4. $\delta_{jkl} + x_{ijkl} - 2y_{ijkl} \geq 0, \forall i, j, k, l.$
5. $\delta_{jkl} + x_{ijkl} - y_{ijkl} \leq 1, \forall i, j, k, l.$

6. $\sum_{j,k,l} l \cdot \delta_{jkl} = W$.

When the number of TAMs is shown by N_B, there are $N_C N_B$ variables for x_{ijk}, $N_B W$ for δ_{jkl}, and $N_C N_B W$ for y_{ijkl}. The total number of variables is $N_B \cdot (N_C + W + N_C W) = O(N_C N_B W) = O(N_C^2 W)$. Note that the upper bound of N_B is N_C because it is meaningless that there are more TAMs than cores. The number of constraints is $N_B + N_C + N_B + N_C N_B W + N_C N_B W + 1 = O(N_C N_B W) = O(N_C^2 W)$. The numbers of cores, TAMs, and the total width of TAMs affect the computation time for this problem. This means that the computation time grows quite rapidly as N_C, N_B, and W grow. It is hard to obtain the optimal solution for large W, though W of SOCs is generally quite large. Only small problems can be solved within practical time.

4 Test Architecture Optimization under FP Constraints

In Section 3, the test architecture optimization in which TAM configurations were treated as variables was discussed. As discussed in the previous section, the optimization is a time-consuming process itself. An algorithm which can optimize a test architecture within shorter and more practical computation time is necessary to apply this optimization to practical SOCs. In this section, a test architecture optimization achieved by enumeration of optimizations shown in Section 2 is discussed.

The space of the enumeration itself is too vast to completely seek out. In this section, several lemmas and theorems are introduced to prune enumeration. Tree pruning makes the test architecture optimization possible within practical computation time.

Now let us introduce several notations to make a discussion clear. Let test time of Core c on a TAM whose width is w be $T_W(w, c)$. These times are obtained by solving bin packing problems as described in Section 2 [2, 7]. In this section, it is assumed that these times are comprehensively given. Optimal test architecture free from floorplan constraints can be determined by the total TAM bits w and a set of cores C. Let the test time free from floorplan constraints be $T_{NFP}(w, C)$. This test time can be sought out the same way as [7]. Optimal test architecture with floorplan constraints can be determined by the total TAM bits, a group configuration and a TAM bits distribution. Note that a TAM bits distribution is necessary to solve a problem shown in Section 2. Enumeration of such problems, consequently, necessitates a TAM bits distribution. Let a group configuration and a TAM bits distribution be $\mathbb{G} = \{G_1, G_2, \cdots, G_n\}$ and $\boldsymbol{b} = (b_1, b_2, \cdots, b_n)$ respectively. Note that $\sum b_i = w$. Let the minimal test test time with floorplan constraints be $T_{FP}(w, \mathbb{G}, \boldsymbol{b})$.

Lemma 1
The test time for a core monotonously decreases as the width of the TAM for the core increases.
$w_1 < w_2 \Rightarrow T_W(w_1, c) \geq T_W(w, c)$.

Lemma 2
The test time for the SOC without floorplan constraints monotonously decreases as the TAM bits increase.
$w_1 < w_2 \Rightarrow T_{NFP}(w_1, C) \geq T_{NFP}(w_2, C)$.

Lemma 3
The test time for the SOC with floorplan constraints monotonously decreases as the TAM bits increase.
$w_1 < w_2 \Rightarrow \min_{\boldsymbol{b_1}} T_{FP}(w_1, \mathbb{G}, \boldsymbol{b_1}) \geq \min_{\boldsymbol{b_2}} T_{FP}(w_2, \mathbb{G}, \boldsymbol{b_2})$.

Lemma 4
The test time for a group increases as a core is added to the group.
$G_1 \subseteq G_2 \Rightarrow \min_{\boldsymbol{b_1}} T_{FP}(w, \{G_1\}, \boldsymbol{b_1}) \leq \min_{\boldsymbol{b_2}} T_{FP}(w, \{G_2\}, \boldsymbol{b_2})$.

Theorem 1
If there exists a feasible solution which can achieve test time T_{tmp} under floorplan constraints and only if $T_{NFP}(w, C)|_{C=G_M - \sum_{i \neq M} G_i} \geq T_{tmp}$, the optimal TAM bits for Group M are w or more.

Theorem 1 means that enumeration can be pruned depending on test time of temporal solution. The more the enumeration proceeds, the more tree pruning is done because test time of temporal solution becomes shorter and shorter. Theorem 1 prunes the lower TAM bits on each group.

Lemma 5
If $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) > T_{NFP}(b_M, G_M)$ *then* $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) = \min_{\boldsymbol{b'}} T_{FP}(w - w', \mathbb{G}', \boldsymbol{b'})$, *where* $\mathbb{G}' = \{G_1 - G_M, \cdots, G_{M-1} - G_M, G_{M+1} - G_M, \cdots, G_n - G_M\}$.

Theorem 2
If $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) > T_{NFP}(b_M, G_M)$ *then the optimal TAM bits for Group M are b_M or less.*

Theorem 3
If $\min_{b_{M+1}, \cdots, b_n} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) > T_{NFP}(b_M, G_M)$ *then*
$$\min_{b'_{M+1}, \cdots, b'_n} T_{FP}(w, \mathbb{G}, \boldsymbol{b'})|_{b'_i=b_i(\forall i, i<M), b'_M=b_M+1} \geq \min_{b_{M+1}, \cdots, b_n} T_{FP}(w, \mathbb{G}, \boldsymbol{b}).$$

Theorem 3 prunes the upper TAM bits for each group while Theorem 1 prunes the lower TAM bits. Theorem 3 alleviates the computation time of the problems given large TAM bits w.

Theorem 4
If $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) > T_{NFP}(b_M, G_M)$ *and only if* $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b}) < \min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b})$ *then the test architecture which achieves* $\min_{b_i, i \neq M} T_{FP}(w, \mathbb{G}, \boldsymbol{b})$ *is optimal.*

Theorem 4 is a stricter version of Theorem 2. Theorem 4 means that tree pruning of enumeration can finish when test architecture is optimized in ascending order of TAM bits for a group. This theorem can reduce the exploration space of enumeration and the computation time can be greatly reduced consequently.

A pseudo-code of test architecture optimization algorithm is shown in Figure 3. This algorithm mainly consists of three parts. In the first part of the algorithm,

Test Architecture Optimization Algorithm

Procedure OptimizeTestArchitecture
Input w_{SOC}: The total bits of all TAMs.
Input $T_W(w, i)$: Test time for Core i when the width of TAM is w.
Input $\mathbb{G} = \{G_1, \cdots, G_n\}$: A set of n groups.
Output b: Optimal TAM bits distribution.
Output Optimal TAM architecture.
begin
    // *Seek out minimal test times* $T_{NFP}(w, C)|_{C=G_i - \sum_{j \neq i} G_j}$
    // *and maximal test times* $T_{NFP}(w, G)$.
    for $\forall G \in \mathbb{G}$ **do** // *for all groups*
        forall possible TAM bits **do** // *the bits notated by w*
            // *Seek out minimal test time for G and w*
            Compute $T_{NFP}(w, C)|_{C=G_i - \sum_{j \neq i} G_j}$.
            // *Seek out maximal test time for G and w*
            Compute $T_{NFP}(w, G)$.
        endfor
    endfor
    // *Seek for maximal number of TAMs* $N_{Bmax}(w, G)$.
    for $\forall G \in \mathbb{G}$ **do**
        for $\forall C \subseteq G$ **do** // *for all core combinations*
            forall possible TAM bits **do** // *the bits notated by w*
            Seek out $T_{NFP}(w, C)$ and let the number
            of TAMs for the solution be $N_B(w, C)$.
            $N_{Bmax}(w, G) = \max\{N_{Bmax}(w, G), N_B(w, C)\}$;
            endfor
        endfor
    endfor
    // *Optimizations shown in Section 2 are enumerated.*
    Let $\mathbb{P}$ be all partitions of w to groups.
    for $\forall P \in \mathbb{P}$ **do** // *for all partitions of TAM bits to groups*
        Check out the applicability of Theorem 3.
        If applicable, return the solution and exit.
        Let $\mathbb{B}$ be all possible TAM configurations derived from P.
        Note that $\mathbb{B}$ must not exceed $N_{Bmax}(w, G)$.
        for $\forall b \in \mathbb{B}$ **do** // *for all possible TAM configurations*
            Minimize test time
            and let the test time be $T_{FP}(w, \mathbb{G}, b)$.
            if a better solution is obtained **then**
                Narrow $\mathbb{P}$ and $\mathbb{B}$ by Theorems 1, 2, 3 and 4.
            endif
        endfor
    endfor
end

Figure 3. Test design optimization algorithm.

$T_{NFP}(w, C)|_{C=G_i - \sum_{j \neq i} G_j}$ and $T_{NFP}(w, G)$ are sought out for all groups and possibles TAM bits. These times can be solved with short computation time because the size of problems is small. The times are relevant to Theorems 1-4. The times are used in enumeration, that is, in the third part of this algorithm. In the second part, the maximum number of TAMs on every group is sought out by solving all possible combinations of cores and TAM bits. These problems are small and easy to solve because the number of cores is small. The number of TAMs can be the upper bound on enumeration and alleviate the enumeration process. In the third part, an optimal test architecture is sought out by enumerating small problems. Repeatedly speaking, the test architecture optimization is based on enumeration of problems shown in Section 2. Tree pruning is done by the theorems introduced in this section.

Optimizations were done for d695 of ITC '02 benchmark SOCs. Test designs of d695 are shown in Table 1 in order to evaluate the optimization proposed in this section. Two group configurations for d695 are shown in Table 3.

Table 3. Two hypothetical FPs for d695.

Core #	1	2	3	4	5	6	7	8	9	10
Floorplan 1	{1,2}	{1,3}	{1,3}	{1,3}	{3}	{2,3}	{2,3}	{2,3}	{1,2}	{1,2}
Floorplan 2	{1,2}	{1}	{1}	{1}	{2}	{2}	{2}	{2}	{2}	{1,2}

Floorplan 1 has a more discretionary grouping to assign cores to multiple groups than Floorplan 2. In Floorplan 2, only Cores 1 and 10 can be assigned to multiple groups. Optimal test architectures for Floorplans 1 and 2 are shown in Tables 4 and 5. In these tables, the minimal test times with and without floorplanning constraints are shown for the purpose of comparison. In the tables, "Optimal TAM width configurations", "Optimal assignments to groups" and "Optimal assignments to TAMs under FP constraints" are also shown in these tables. Δt means how much test time rises with floorplan constraints. Computation times to obtain an optimal test architecture under FP constraints are shown in the last column. The optimizations were done for 10, 20, 30, 40, 50 and 60 of TAM bits. According to the experimental results shown in the tables, all the test times derived for Floorplan 1 are shorter than the ones for Floorplan 2, though the difference between the minimal test times is quite slight.

The computation times for Floorplan 1 are much longer than the ones for Floorplan 2 because there are more test architectures to be sought in Floorplan 1 than in Floorplan 2. There is a trade-off between discretion of floorplan and computation time. Figure 4 shows optimal test architectures under Floorplans 1 and 2 when total TAM bits

Table 4. Optimal test architecture of d695 under Floorplan 1.

Total TAM bits	Test time without FP constraints	Test time with FP constraints	TAM width configuration	Assignment to group	Assignment to TAMs	Δt	Computation time (HH:MM:SS)
10	67146	68013	{3}, {1}, {6}	(1,1,1,1,3,3,2,3,1,1)	(1,1,1,1,1,1,1,1,1,1)	1.29 %	00:00:03.34
20	33895	34616	{1}, {5}, {6,8}	(1,3,1,1,3,3,3,2,2,2)	(1,1,1,1,2,1,2,1,1,1)	2.13%	00:00:33.63
30	22670	23275	{1}, {7}, {1,4,17}	(1,1,1,3,3,3,3,3,2,2)	(1,1,1,2,3,3,2,1,1,1)	2.67%	00:04:14.32
40	17523	17523	{2}, {3,19}, {16}	(1,3,1,1,3,2,3,2,2,2)	(1,1,1,1,1,2,1,1,1,2)	0%	00:24:12.80
50	13844	13844	{2}, {11}, {2,16,19}	(2,3,3,1,3,3,3,3,2,2)	(1,2,1,1,2,3,3,1,1,1)	0%	02:08:34.58
60	11420	11420	{3}, {11}, {2,4,6,17,17}	(2,3,1,3,3,3,3,3,1,2)	(1,2,1,2,4,5,3,1,1,1)	0%	11:11:02.20

Table 5. Optimal test architecture of d695 under Floorplan 2.

Total TAM bits	Test time without FP constraints	Test time with FP constraints	TAM width configuration	Assignment to group	Assignment to TAM	Δt	Computation time (HH:MM:SS)
10	67146	68914	{1}, {4,5}	(2,1,1,1,2,2,2,2,2,2)	(2,1,1,1,2,1,1,1,2,2)	2.63 %	00:00:00.44
20	33895	35135	{1,4}, {7,8}	(1,1,1,1,2,2,2,2,2,1)	(2,2,1,1,1,2,2,1,2,2)	3.66 %	00:00:05.13
30	22670	23442	{3}, {1,8,18}	(1,1,1,1,2,2,2,2,2,2)	(1,1,1,1,3,3,2,1,3,2)	3.41 %	00:00:30.88
40	17523	17624	{1,2}, {3,4,11,19}	(1,1,1,1,2,2,2,2,2,2)	(1,1,2,2,3,4,2,1,1,4)	0.58 %	00:01:35.82
50	13844	14144	{4}, {2,9,16,19}	(1,1,1,1,2,2,2,2,2,2)	(1,1,1,1,3,4,4,1,3,2)	2.17 %	00:04:42.58
60	11420	11519	{5}, {2,17,17,19}	(2,1,1,1,2,2,2,2,2,2)	(1,1,1,1,3,4,2,1,4,2)	0.87 %	00:13:14.22

are 50. In the figure, the x-axis corresponds to test time and the height of each box means the width of the corresponding TAM.

Floorplan 1
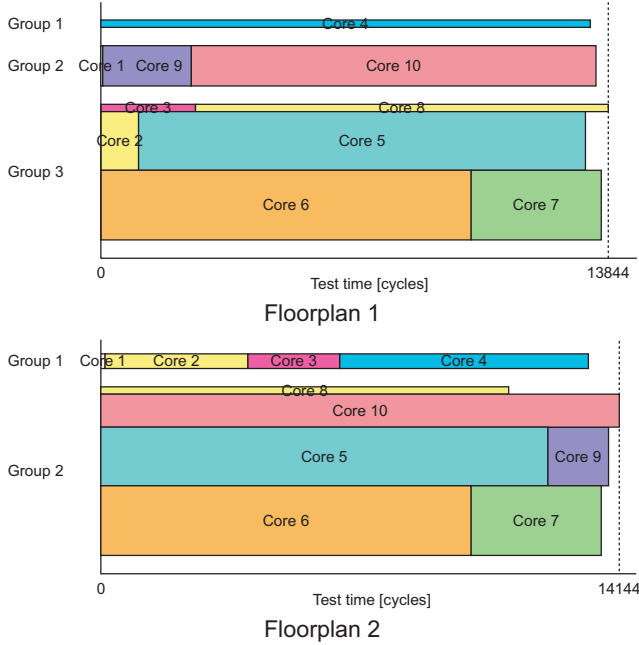
Floorplan 2

Figure 4. Optimal architecture when $w_{\mathrm{SOC}} = 50$.

5 Conclusions

In this paper, a test architecture optimization under floorplanning constraints was discussed. There are mainly two contributions in this proposal. The first contribution is that test architecture optimization taken floorplanning constraints into account is proposed. The models of the former researches were idealized too much to optimize test architectures of practical SOCs. There are strict place-and-route constraints in designing practical SOCs and our proposal offers system designers a powerful test architecture optimization in such practical SOCs. The second contribution is to adjust computation time of test architecture optimization. Floorplanner can determine how an SOC should be floorplanned, how many groups there are on the SOC and how many cores each group has. Our experiments showed that the differences between floorplannings made computation times vary from minutes to hours. For example, given 60 of TAM bits, it took more than 2 hours to optimize a test architecture under a floorplan for d695, though less than 5 minutes for the other floorplan. This validates computation time can be adjusted by the number of cores per group which floorplanners can determine.

In future work, floorplanning consideration should be extended to the other TAM architectures. There are several TAM architectures proposed before. It must be also researched how floorplanner determines groups and core assignment to groups.

References

[1] J. Aerts and E. J. Marinissen, "Scan chain design for test time reduction in core-based ICs," *IEEE International Test Conference*, pp. 448–457, October 1998.

[2] G. Ausiello et al., Complexity and approximation – Combinatorial optimization problems and their approximability properties, Springer-Verlag Berlin Heidelberg 1999.

[3] K. Chakrabarty, "Design of system-on-a-chip test access architectures under place-and-route and power constraints," *Design Automation Conference*, pp. 432–437, 2000.

[4] K. Chakrabarty, "Test scheduling for core-based systems using mixed-integer linear programming," *IEEE Transactions on CAD/ICAS*, pp. 1163–1174, October 2000.

[5] K. Chakrabarty, "Optimal test access architectures for system-on-a-chip," *ACM Transactions on Design Automation of Electronic Systems*, vol. 6, pp. 26–49, January 2001.

[6] ILOG, CPLEX 8.1 Reference Manual, December 2002.

[7] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "Test wrapper and test access mechanism co-optimization for system-on-a-chip," *Journal of Electronic Testing: Theory and Applications*, vol. 18, pp. 213–230, April 2002.

[8] E. J. Marinissen, S. K. Goel and M. Lousberg, "Wrapper design for embedded core test," *IEEE International Test Conference*, pp. 911–920, 2000.

[9] M. Nourani and C. Papachristou, "An ILP Formulation to optimize test access mechanism in system-on-chip testing," *IEEE International Test Conference*, pp. 770–777, 2000.

[10] M. Sugihara, H. Date and H. Yasuura, "A novel test methodology for core-based system LSIs and a testing time minimization problem," *IEEE International Test Conference*, pp. 465–472, October 1998.

[11] M. Sugihara, H. Date and H. Yasuura, "Analysis and minimization of test time in a combined BIST and external test approach," *Design, Automation and Test in Europe Conference*, pp. 134–140, March 2000.

[12] M. Sugihara and H. Yasuura, "Optimization of test accesses with a combined BIST and external test scheme," *Asia South Pacific Design Automation Conference/VLSI Design*, pp. 683–688, January 2002.

[13] Y. Zorian, E. J. Marinissen and S. Dey, "Testing embedded core based system chips," *IEEE International Test Conference*, pp. 130–143, 1998.

[14] IEEE P1500 Standard for Embedded Core Test. *http://grouper.ieee.org/groups/1500/*.

A The Upper Bound of TAM Width

A theorem was introduced in [7] as follows.

Theorem 1 in [7]
If a core has n functional inputs, m functional outputs, and sc internal scan chains of lengths $l_1, l_2, \cdots, l_{sc}$, respectively, an upper bound k_{max} on the TAM width required to minimize testing time is given by $\lceil \frac{\max\{n,m\}+\sum_{i=1}^{sc} l_i}{\max_i\{l_i\}} \rceil$.

This theorem is false. For example, let us assume a core has 4 internal scan chains of lengths 5, 3, 3, 3 respectively. To make the example clear, there are no functional inputs and outputs for this core. According to the theorem, $k_{max} = \frac{\max\{0,0\}+(5+3+3+3)}{\max\{5,3,3,3\}} = \lceil 2.8 \rceil = 3$. The true upper bound is, however, 4 because scan chains cannot be split. This contradicts the theorem.