Test Architecture Design and Optimization for Three-Dimensional SoCs

Li Jiang[†‡], Lin Huang[†] and Qiang Xu[†‡]
[†]CUhk REliable computing laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
[‡]CAS-CUHK Shenzhen Institute of Advanced Integration Technology
Email: {ljiang,lhuang,qxu}@cse.cuhk.edu.hk

Abstract

Core-based system-on-chips (SoCs) fabricated on three-dimensional (3D) technology are emerging for better integration capabilities. Effective test architecture design and optimization techniques are essential to minimize the manufacturing cost for such giga-scale integrated circuits. In this paper, we propose novel test solutions for 3D SoCs manufactured with die-to-wafer and die-to-die bonding techniques. Both testing time and routing cost associated with the test access mechanisms in 3D SoCs are considered in our simulated annealing-based technique. Experimental results on ITC'02 SoC benchmark circuits are compared to those obtained with two baseline solutions, which show the effectiveness of the proposed technique.

1 Introduction

There are unlimited demands for electronic products to provide more functionalities and higher performance with less power consumption in each new generation. With the ever increasing complexity for state-of-the-art SoC designs, however, interconnects have become the bottleneck for the performance and power dissipation of such giga-scale integrated circuits (ICs). Three-dimensional (3D) technology that provide abundant interconnect resources has been proposed as a promising solution to resolve this problem [2].

The benefits provided by 3D integration are manifold. First of all, by stacking planar circuit layers at micrometer distances and using through-silicon vias (TSVs) to interconnect them (as shown in Fig. 1), wire lengths can be dramatically reduced, thus leading to improved performance and less communication energy [3, 16]. Secondly, 3D is a natural way to integrate disparate technologies such as microelectromechanical systems (MEMS), image sensors, and other heterogenous elements demanded by applications, as they can be fabricated on different layers separately before integration [23]. Thirdly, the reduced parasitics for interconnects in 3D ICs simplify the circuit design for high-performance applications [20]. Therefore, although there are still critical issues such as heat dissipation to be resolved in 3D integration, it is generally regarded that 3D ICs will occupy a large market share in the future [27].

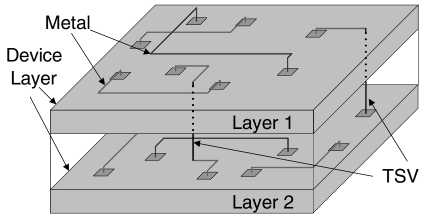

Figure 1. An Example 3D Circuit.

3D ICs can be built in several manners: wafer-to-wafer (W2W) bonding [23], die-to-wafer (D2W) bonding (for 3D ICs built on two semiconductor wafers) [4], or die-to-die (D2D) bonding [14]. The main advantage of W2W bonding is the simplicity of the manufacturing process [14]. However, without "known good die" information, the yield of the 3D chips can be quite low, especially when the die size is large and/or the defect density is high [3]. D2W/D2D bonding, on the other hand, requires a more complex manufacturing process that applies pre-bond wafer-level testing and attaches known good dies only, thus resulting in higher yield compared to W2W bonding [17].

According to [19], tools and methodologies for testing 3D ICs are critical challenges for 3D integration. However, there are only limited prior work in this domain in the literatures. Lewis and Lee [12] proposed a scan-island based design to enable pre-bond test for incomplete circuit at the architecture level. Wu *et al.* [22] studied several scan chain design approaches for 3D ICs and compare their costs. The above scan design methodologies mainly target 3D ICs that put functional blocks in different wafers. For 3D SoCs with entire embedded cores on different layers, modular testing is more attractive as it facilitates the reuse of test patterns. In [21], the authors considered the above problem and presented a test-access mechanism (TAM) optimization technique to minimize the testing time of 3D SoCs, under limits on the number of TSVs utilized by TAMs. The above constraint is important for early 3D technologies with large-sized TSVs. However, recent work has shown that the size of TSVs can be reduced to the μm^2 range [7] and the number of TSVs that can be fabricated on-chip is in the order of millions per cm^2 with technology advancement [14]. Therefore, TSVs should not be a constraint for 3D SoCs test any more. In addition, pre-bond test is not considered in the above work and hence it can only provide cost-effective solutions for 3D SoCs manufactured with W2W bonding technique.

Different from [21], in this work, we consider testing 3D SoCs manufactured with D2W or D2D bonding technology. The test architecture design and optimization problem for 3D SoCs is formulated by considering both post-bond test and pre-bond wafer-level tests. We then propose efficient and effective heuristics to optimize the testing time and the routing cost associated with the test access mechanisms, based on simulated annealing technique. Experimental results on ITC'02 benchmark circuits show that the proposed technique is able to dramatically reduce test cost when compared to two baseline solutions adapted from 2D test architecture optimization methods.

The remainder of this paper is organized as follows. Section 2 reviews related work and motivates this paper. The test architecture design and optimization problem for 3D ICs studied in this work is then formulated in Section 3. In Section 4, we detail our simulated annealing-based solution for the above problem. Experimental results on benchmark circuits are then shown in Section 5. Finally, Section 6 concludes this paper.

2 Preliminaries and Motivation

Modular testing using dedicated bus-based TAMs, such as Test Bus [18] and TestRail [6], is the most popular test strategy for large SoC devices [26]. In modular testing, embedded cores are isolated from their surrounding logic using test wrappers, while TAMs are designed to transport test data between the input/output pins of the SoC and the cores under test (CUTs). The test architecture design and optimization problem for two-dimensional (2D) SoCs, which considers to reduce testing time and the amount of on-chip DfT resources (both logic and routing), has been subject to extensive research in the literature (e.g., [6, 8, 11, 13, 24, 25, 28]). These techniques, however, cannot provide cost-effective solutions for 3D SoCs manufactured with D2W or D2D bonding technique. This is mainly because we need to consider pre-bond wafer-level testing for enhanced yield. We use the following motivating example to illustrate the problem investigated in this work.

Consider a core-based 3D SoC as shown in Fig. 2, wherein six cores are placed on two silicon layers. With a traditional 2D test architecture optimizer that tries to minimize the testing time in post-bond test, we obtain the TAM architecture shown also in this figure. That is, there are totally three TAMs for this example SoC: TAM_1 for core 5, TAM_2 for cores 1, 2, and 3, and TAM_3 for core 4 and core 6. In particular, TAM_2 traverses two layers in this example.

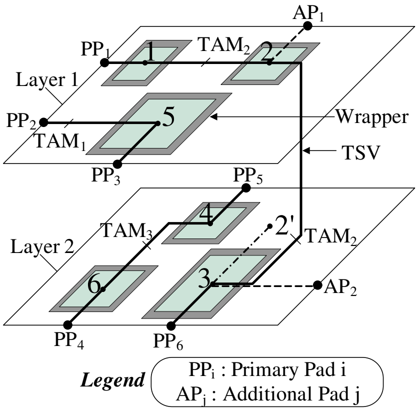

Figure 2. An Example 3D SoC Test Architecture.

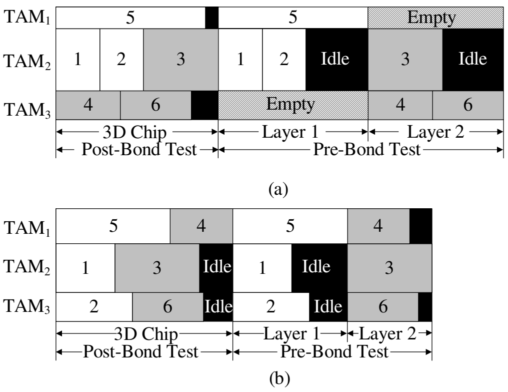

Figure 3. The Impact of Pre-Bond Tests.

When pre-bond tests at the wafer-level are required, however, the test cost model for the 3D SoC changes. For instance, the testing time of the chip is the sum of each layer's pre-bond testing time and the post-bond testing time of the entire chip. That is, for the example shown in Fig. 2, it contains three parts: the pre-bond testing time for layer 1, the pre-bond testing time for layer 2, and the post-bond testing time for the entire chip, represented as three bins in Fig. 3(a), respectively. The cores in different layers are shown in different gray scales, and the TAM can be empty if no cores in that layer are assigned to it. From this figure, it is obvious that the test architecture optimized only for post-bond test in 3D SoCs incurs long idle time on their pre-bond tests (see TAM_2). If we are able to design a 3D-aware test architecture as shown in Fig. 3(b), the total testing time for the 3D SoC can be dramatically reduced, although the testing time of the post-bond test becomes longer. In addition, the routing cost associated with TAMs for 3D SoCs is also different from that of the planar 2D SoCs, as TAMs can use TSVs to go through several layers. Therefore, we should add additional TAM wires (dashed lines) and additional test pads (AP_1, AP_2) for those incomplete TAMs during pre-bond tests (TAM_2), as shown in Fig. 2.

The above motivates us to investigate the test architecture design and optimization problem for D2W/D2D bonding fabricated 3D SoCs, as formulated in the following section.

3 Problem Formulation

3.1 Test Cost Model

The test cost model for 3D SoCs to evaluate different test architectures is shown in the following.

$$C_{total} = C_{Test\_Time} \times \alpha + C_{Wire\_Length} \times (1-\alpha) \qquad (1)$$

where, C_{Test_Time} is the total testing time for both pre-bond tests and post-bond test, while C_{Wire_Length} is the total TAM wire length. α is a weighting factor designated by users. For the example test architecture and the associated test schedule shown in Fig. 3(a), C_{Test_Time} is the sum of three terms: $(T_1 + T_2 + T_3)$ for post-bond entire chip, T_5 for pre-bond layer 1 and $(T_4 + T_6)$ for pre-bond layer 2, where T_i is the testing time of core i.

The computation of C_{Wire_Length}, however, is non-trivial. In this work, we assume a TAM involved in several layers will route through all cores tested with this TAM on one layer before it goes through TSVs to connect cores in other layers. Accordingly, we calculate C_{Wire_Length} as follows.

C_{Wire_Length} for a TAM that involves several layers contains two parts: the intra-layer wire length and the inter-layer one. For the former one, the TAM is broken into several segments, each on a single layer. For each segment (or TAM that is on one layer only), we use the algorithm in [5] to compute its wire length[1]. As for the inter-layer wire lengths, they are calculated as the Manhattan distance between the corner cores in different layers, e.g., for TAM_2 in Fig. 2, the inter-layer wire length is the Manhattan distance between core 3 and the core 2 mirrored on layer 2 (i.e., the dot dash line). The wire length for TSVs is ignored due to their tiny sizes.

It is important to note that our proposed algorithms (detailed in Section 4) can be applied to other cost models as well. For example, if a different TAM routing strategy is used [22], partial pre-bond testing is applied [1] or multi-site testing is considered [10]. Designers can just update the above test cost model accordingly and apply our proposed technique.

3.2 Problem Definition

The problem addressed in this paper can be formulated as follows:

Problem: Given

- a set of cores C, and the test parameters for each core $c \in C$, that is, the number of inputs in_c, the number of outputs out_c, the number of bidirectionals bi_c, the number of test patterns p_c, the number of internal scan chains sc_c, and for each scan chain i, its scan chain length in terms of flip-flop count $l_{c,i}$;
- the physical position of every core c, i.e., which layer it sits on and its X-Y coordinate on that layer;
- the maximum available TAM width W_{TAM};

Determine the number of TAMs, the cores assignment associated with each TAM, and the width of each TAM to minimize the total test cost as shown in Eq. (1). Note, test wrapper design and optimization is a subproblem of the above problem, and we use the algorithm in [9] to address it.

4 Proposed Algorithm

In this section, we introduce the proposed simulated annealing-based algorithm for tackling the problem presented in Section 3.2. The proposed approach is applicable for optimizing both TestRail and TestBus architecture, but we mainly focus on the Test Bus architecture for the sake of simplicity.

4.1 Outline of The Proposed Algorithm

The test architecture design and optimization problem for 2D SoCs has been proven to be a NP-hard problem [9]. To reduce computational time, prior work mainly resorts to deterministic heuristics to address this problem (e.g., [6, 8, 11, 13, 24, 25, 28]). As shown earlier, for 2D SoCs without considering pre-bond tests, generally one single TAM is the bottleneck that determines the SoC testing time. Consequently, greedily optimizing the bottleneck TAM by assigning cores to different TAMs and/or allocating more TAM wires to the bottleneck TAM can lead to close-to-optimal solution [6]. The above deterministic optimization strategies, however, are difficult to apply to optimize 3D SoC test architectures as we need to consider both pre-bond tests and post-bond test, which can have multiple bottleneck TAMs in terms of testing time (e.g., TAM_1 for layer 1 pre-bond test and TAM_2 for post-bond test in Fig. 3(a)). We therefore propose to use simulated annealing (SA) based stochastic search algorithms to tackle the problem described in Section 3.2.

One of the most straightforward methods to address this problem is then to construct a unified solution representation including both core assignment and TAM width allocation, and perform simulated annealing on it. That is, we can represent a solution as a few core sets and a TAM width for each set. This method, in spite of its simplicity, is not quite effective due to the huge solution space and the difficulty to specify neighboring solutions in the SA process.

Fortunately, we notice that, given a fixed core assignment for each TAM, it is easy to determine close-to-optimal TAM widths for each TAM by properly adjusting existing deterministic heuristics. Based on this observation, for a given number of TAMs, we propose to separate our optimization procedure into two nested parts: the *outer SA-based core assignment* (Section 4.2) and the *inner heuristic-based TAM width allocation* (Section 4.3). By doing so, the solution space explored by simulated annealing shrinks to the various core assignment solutions only, without loss of its quality. The above outer and inner procedures are for a given number of TAMs, we therefore need to enumerate this value in our algorithm.

The overall algorithm to tackle 3D SoC test architecture design and optimization problem is shown in Fig. 4. In the beginning, we set the minimum number of TAMs (TAM_Num_{min}) and the maximum number of TAMs (TAM_Num_{max}) to be explored in our algorithm. Typically, $TAM\_Num_{min} = 1$ while TAM_Num_{max} is set to be a small number that is much less than $\min\{|C|, W_{TAM}\}$[2]. Then, for a given number of TAMs, we start from a random initial core assignment, and keep on searching for its neighbor solutions. Once a feasible solution is obtained, TAM width allocation is conducted by inner deterministic algorithm. Core assignment solutions are evaluated using the cost model in Section 3.1. Finally, the algorithm outputs the best solution that is obtained during the stochastic search process.

[1] Note that additional pads (APs) and wires (dash lines in Fig. 2) for test may be necessary when reusing chip-level TAMs for wafer-level testing.

[2] Generally speaking, a large number of TAMs results in excessive testing time and hence is not preferred.

```
Set $TAM\_Num_{\min}$ and $TAM\_Num_{\max}$;
For $Tam\_Num$ = $TAM\_Num_{\min}$ to $TAM\_Num_{\max}$
    // Simulated Annealing
    Get initial core assignment with no empty TAM;
    Perform inner TAM width allocation algorithm
    and compute initial cost;
    $Cost_{best} \leftarrow$ initial cost, $Cost_{old} \leftarrow$ initial cost;
    Set temperature $T$ as a high value;
    While ($T >$ end temperature $T_{end}$)
        // Run a few iterations at same T
        For each iteration
            Random move to reach a new core assignment;
            Perform inner TAM width allocation;
            Compute $Cost_{new}$;
            If $Cost_{new} < Cost_{old}$ or $e^{\Delta Cost/T} > rand()$;
                // Accept Move
                $Cost_{old} \leftarrow Cost_{new}$;
                Update core assignment solution;
                If $Cost_{new} < Cost_{best}$
                    $Cost_{best} \leftarrow Cost_{new}$;
                    Record the best solution;
            Else
                Restore old solution;
        Decrease temperature $T$;
Output the best solution;
```

Figure 4. Main Flow of the Proposed Algorithm.

4.2 SA-Based Core Assignment

Suppose we are performing core assignment for m TAMs. Remind there are totally $|C|$ cores to be tested, the problem comes down to dispatching cores to m sets. As a consequence, a solution can be represented as a series of sets $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m$, where $\mathcal{A}_i$ is the core set assigned to TAM i. For instance, assume there are two TAMs and five cores, a valid solution can be $\{(1,3),(2,4,5)\}$, meaning that cores 1 and 3 are assigned to one TAM, and cores 2, 4, and 5 are assigned to the other one.

Apparently, both $\{(1,3),(2,4,5)\}$ and $\{(2,4,5),(1,3)\}$ are both valid representations, but they essentially correspond to the same solution. To eliminate this redundancy and provide a one-to-one mapping between a representation and its corresponding solution, we always keep the smallest core index assigned to TAM i smaller than that assigned to TAM j, provided i is smaller than j. Let α_i be the minimum core index of TAM set $\mathcal{A}_i$. This rule can be expressed as $\forall i < j : \alpha_i < \alpha_j$. According to it, $\{(2,4,5),(1,3)\}$ will be deemed as an invalid solution in our annealing process. With the help of the above rule, the solution space shrinks to $\frac{1}{m!}$ of that without this rule. Also, we do not allow empty sets, because any solution with n empty sets achieved in the iteration where TAM number is set to be m can be revisited in the iteration where TAM number is $(m-n)$ without empty sets.

The only move defined in our procedure to find a neighbor solution is M1: pick up a core from a random set $\mathcal{A}_i$ which contains more than one cores, and put it into another randomly selected set $\mathcal{A}_j$. The completeness of the above move can be effectively proved by: starting from a valid solution $\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m$, we are able to reach any other solution $\mathcal{B}_1, \mathcal{B}_2, \cdots, \mathcal{B}_m$ after finite times of M1 move[3].

4.3 Heuristic-Based TAM Width Allocation

With given number of TAMs and the core assignment for each TAM, it might be possible to obtain the optimal TAM width allocation using techniques such as linear programming (e.g., [9]). The inner TAM width allocation procedure however needs to be called every time when we have a feasible core assignment solution during the simulated annealing process. Consequently, the running time for this inner procedure needs to be very short so that we can explore a large number of core assignment solution space. Because of this, instead of acquiring exact optimal solution for the inner TAM width allocation process, we use a greedy heuristic as shown in Fig. 5 to obtain a close-to-optimal solution. Similar to [6], this procedure iteratively assign one-bit wire ($b=1$) to a TAM that leads to the lowest total test cost (Line 6). If this one-bit TAM wire cannot result in cost reduction, we will not allocate it in this iteration (Line 11). Instead, we increase the width of the to-be-assigned TAM wire by one until a lower cost is found (Lines 12-16).

```
1   Allocate one bit width to every TAM;
2   Set $b = 1$;$Cost_{\min} \leftarrow \infty$;
3   While no more unassigned TAM width
4       For each TAM
5           Allocate $b$ bit width to this TAM;
6           Compute the cost of entire TAM architecture;
7           If $Cost < Cost_{\min}$
8               $Cost_{\min} = Cost$;
9               Keep this TAM as the only candidate;
10          Restore this $b$ bit width;
11      If $Cost_{min}$ reduces
12          Allocate $b$ bit to the recorded TAM;
13          Set $b = 1$;
14      Else
15          Increase $b$ by one;
16  Output TAM width allocation;
```

Figure 5. Inner TAM Width Allocation Procedure.

5 Experiments

5.1 Experimental Setup

To demonstrate the effectiveness of the proposed solution for testing 3D SoCs, we present experimental results for two ITC'02 benchmark SoCs: p22810 and t512505 (other benchmarks, e.g., p93791, have similar results.). We map these two SoCs onto three silicon layers randomly and try to balance the total area of each layer, where a core's area is estimated based on the number of internal inputs/outputs and scan cells (if any). An academic floorplanner is utilized to get the coordinates for each core, to be used for wire length calculation. As mentioned in Section 4, we focus on the Test Bus architecture in our experiments.

[3] Due to space limitation, the proof is omitted in this paper.

	p22810										t512505									
Width	Total Testing Time			Ratio (%)		Wire Length			Ratio (%)		Total Testing Time			Ratio (%)		Wire Length			Ratio (%)	
(bit)	TR-1	TR-2	SA	Δ_1^T	Δ_2^T	TR-1	TR-2	SA	Δ_1^W	Δ_2^W	TR-1	TR-2	SA	Δ_1^T	Δ_2^T	TR-1	TR-2	SA	Δ_1^W	Δ_2^W
16	1888667	1327398	1062281	43.75	14.04	5386	13694	12095	-120.66	13.21	47658726	30913144	27190110	42.95	7.82	2743	4560	13234	-382.46	-190.22
24	1302783	1046844	780763	40.07	20.42	7729	17041	14614	-104.41	7.29	34450110	30758094	26778656	22.27	11.55	5459	5824	21245	-289.17	-264.78
32	1031366	836821	627148	39.19	20.33	7693	15051	18062	-126.97	-16.01	33867447	18036401	17510811	48.30	1.55	4596	11590	8621	-87.58	25.62
40	802558	723546	534329	33.42	23.58	8582	21328	24926	-124.91	9.50	23417347	18890284	13028909	44.36	25.03	6313	8460	17540	-177.84	-107.33
48	703142	639377	500868	28.77	19.70	9184	17380	32084	-245.90	-82.78	23417347	18890284	12967247	44.63	25.29	7171	9724	11775	-64.20	-21.09
56	593840	562916	435664	26.64	21.43	9709	22240	37302	-333.20	-89.11	23417347	18890284	12967247	44.63	25.29	7797	10988	13605	-74.49	-23.82
64	527732	490439	421677	20.10	13.03	10560	26862	35286	-241.11	-34.10	23417347	18890284	12967247	44.63	25.29	8985	12252	15846	-76.36	-29.33

Δ_1^T / Δ_2^T: Difference ratio on total testing time between SA and TR-1 / TR-2; Δ_1^W / Δ_2^W: Difference ratio on wire length between SA and TR-1 / TR-2.

Table 1. Experimental Results for $\alpha = 1$.

	$\alpha = 0.6$										$\alpha = 0.4$									
Width	Total Testing Time			Ratio (%)		Wire Length			Ratio (%)		Total Testing Time			Ratio (%)		Wire Length			Ratio (%)	
(bit)	TR-1	TR-2	SA	Δ_1^T	Δ_2^T	TR-1	TR-2	SA	Δ_1^W	Δ_2^W	TR-1	TR-2	SA	Δ_1^T	Δ_2^T	TR-1	TR-2	SA	Δ_1^W	Δ_2^W
16	47658726	30913144	25854529	45.75	10.61	2743	4560	7272	-165.11	-59.47	47658726	30913144	27556426	42.18	7.04	2743	4560	3957	-44.26	13.22
24	34450110	30758094	24742962	28.18	17.46	5459	5824	4714	13.65	19.06	34450110	30758094	28320290	17.79	7.08	5459	5824	4144	24.09	28.85
32	33867447	18036401	14449906	57.33	10.59	4596	11590	9933	-116.12	14.30	33867447	18036401	27355499	19.23	-27.52	4596	11590	3986	13.27	65.61
40	23417347	18890284	13359723	42.95	23.62	6313	8460	7354	-16.49	13.07	23417347	18890284	27360934	-16.84	-36.17	6313	8460	4029	36.18	52.38
48	23417347	18890284	13359680	42.95	23.62	7171	9724	7440	-3.75	23.49	23417347	18890284	27350658	-16.80	-36.13	7171	9724	4175	41.78	57.06
56	23417347	18890284	13361473	42.94	23.61	7797	10988	7470	4.19	32.02	23417347	18890284	27350409	-16.80	-36.13	7797	10988	4102	47.39	62.67
64	23417347	18890284	13372901	42.89	23.56	8985	12252	7374	17.93	39.81	23417347	18890284	28867718	-23.27	-42.61	8985	12252	4043	55.00	67.00

Table 2. Experimental Results for SoC t512505 Considering Both Testing Time and Wire Length.

We compare the proposed algorithm with two baseline solutions, constructed from a traditional 2D optimization algorithm TR-ARCHITECT [6]. In the first one (referred as TR-1), we apply TR-ARCHITECT algorithm to the 3D SoC layer by layer, i.e., no TAM wires is allowed to traverse multiple silicon layers, and we adjust the TAM width among layers iteratively until the testing time of these layers are as balanced as possible. In the second baseline solution (referred as TR-2), we simply apply TR-ARCHITECT algorithm to the whole 3D chip, minimizing the testing time of the post-bond test. In this paper, we do not compare against [21] because [21] is essentially an optimizer for post-bond test with TSV constraints, but we do not consider TSV as constraints in our work. Also, the layout used in [21] is not available to us.

5.2 Experimental Results

Table 1 presents our experimental results when our algorithm considers testing time only (i.e., $\alpha = 1$ in the test cost model). As expected, our algorithm leads to significant improvement in terms of testing time reduction when compared with the other two baseline solution. The benefit can be as high as 48.3% compared with TR-1, and 25.3% compared with TR-2 for SoC t512505. TR-1 leads to longer testing time because TAMs are not allowed to walk through different layers, which significantly constrains the solution exploration space. At the same time, as TAM wire length is not considered in this experiment, typically long TAM wires are obtained using our algorithm, especially when compared to TR-1.

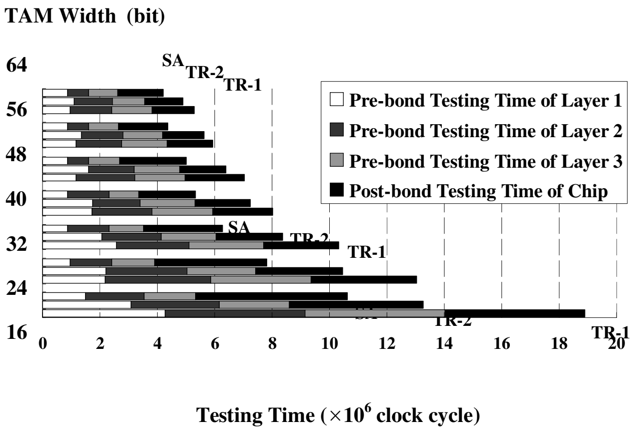

Figure 6. Detailed Testing Time of p22810.

When the total TAM width gets larger, the total testing time decreases for p22810. However, for t512505, after the TAM width is larger than 40, its testing time does not decrease any more, mainly due to a bottleneck core in the system. Also, we can observe that when TAM width increases to be more than 32, the testing time of TR-2 becomes even higher, mainly due to the increase of its pre-bond testing time.

Fig. 6 shows the testing time of the pre-bond test for each layer and the post-bond test for the entire chip for SoC p22810. With TR-1, we can observe balanced testing time among all layers (as expected), while the other two algorithms do not have this feature. Compared to TR-2, the proposed algorithm often has longer testing time for the post-bond test, but achieves a much shorter testing time in the pre-bond tests (e.g., when TAM width is 16), thus resulting in improved total testing time.

In Table 2, we present experimental results for SoC t512505 with two sets of weight parameter: $\alpha = 0.6$ and $\alpha = 0.4$, in which the former one is associated with balanced testing time and TAM wire length cost factors while the latter one emphasizes more on the impact of wire length cost. As can be observed from this table, for the former case, the TAM wire length of the proposed solution is still higher than that of TR-1 in many cases, but already much smaller than the case in Table 1. In several cases (when W_{TAM} is 24, 56 or 64), we can achieve both testing time improvement and wire length reduction. As the total TAM width increases, the total testing time of the proposed algorithm declines first, and then increases (see Column 4). We attribute it to the fact that wire length accounting for more share in the cost function with the increment of TAM width.

For the latter case, where wire length has a much heavier weight, since the wire length increases dramatically with TAM width increment, when the TAM width is large, we can achieve much shorter wire length compared to that with TR-1 or TR-2. For instance, when TAM width is 64, the difference ratio on wire length between the proposed algorithm and TR-1 / TR-2 is as high as 55.00% / 67.00%.

Finally, the computational time of the proposed technique is in the range of minutes for all our experiments, which is acceptable for test architecture design and optimization.

6 Conclusion

Existing test architecture design and optimization techniques for 2D SoCs are not readily applicable for emerging 3D SoCs. In this paper, we propose efficient and effective method to optimize the test cost for 3D SoCs with D2W or D2D bonding technique. Experimental results show that the proposed method can significantly reduce test cost when compared to those baseline solutions that are adapted from 2D test architecture optimization methods.

7 Acknowledgements

This work was supported in part by the General Research Fund 417406, 417807, and 418708 from Hong Kong SAR Research Grants Council (RGC), in part by National Science Foundation of China (NSFC) under grant No. 60876029, in part by a grant N_CUHK417/08 from the NSFC/RGC Joint Research Scheme, and in part by the National High Technology Research and Development Program of China (863 program) under grant no. 2007AA01Z109.

References

[1] S. Bahukudumbi and K. Chakrabarty. Wafer-Level Modular Testing of Core-Based SoCs. *IEEE Transactions on VLSI Systems*, 15(10):1144–1154, 2007.

[2] K. Banerjee, S. J. Souri, P. Kapur, and K. C. Saraswat. 3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration. *Proceedings of the IEEE*, 89(5):602–633, 2001.

[3] W. Davis *et al.*. Demystifying 3D ICs: The Pros and Cons of Going Vertical. *IEEE Design & Test*, pp. 498–510, 2005.

[4] T. Fukushima, Y. Yamada, H. Kikuchi, and M. Koyanagi. New Three-Dimensional Integration Technology Using Chip-to-Wafer Bonding to Achieve Ultimate Super-Chip Integration. *Janpaese Journal of Applied Physics Part 1*, 45(4B):3030, 2006.

[5] S. Goel and E. Marinissen. Layout-driven SOC test architecture design for test time and wire length minimization. In *Proceedings IEEE/ACM Design, Automation and Test in Europe*, pp. 738–743, 2003.

[6] S. K. Goel and E. J. Marinissen. Effective and Efficient Test Architecture Design for SOCs. In *Proceedings IEEE International Test Conference (ITC)*, pp. 529–538, 2002.

[7] S. Gupta, M. Hilbert, S. Hong, and R. Patti. Techniques for Producing 3D ICs with High-Density Interconnect. In *Proceedings of the 21st International VLSI Multilevel Interconnection Conference*, 2004.

[8] Y. Huang et al. Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm. In *Proceedings IEEE International Test Conference (ITC)*, pp. 74–82, 2002.

[9] V. Iyengar, K. Chakrabarty, and E. J. Marinissen. Co-Optimization of Test Wrapper and Test Access Architecture for Embedded Cores. *Journal of Electronic Testing: Theory and Applications*, 18(2):213–230, Apr. 2002.

[10] V. Iyengar, S. K. Goel, K. Chakrabarty, and E. J. Marinissen. Test Resource Optimization for Multi-Site Testing of SOCs Under ATE Memory Depth Constraints. In *Proceedings IEEE International Test Conference (ITC)*, pp. 1159–1168, 2002.

[11] E. Larsson and Z. Peng. A Reconfigurable Power-Conscious Core Wrapper and its Application to SOC Test Scheduling. In *Proceedings IEEE International Test Conference (ITC)*, pp. 1135–1144, 2003.

[12] D. L. Lewis and H.-H. S. Lee. A Scan-Island Based Design Enabling Pre-bond Testability in Die-Stacked Microprocessors. In *Proceedings IEEE International Test Conference (ITC)*, paper 21.2, 2007.

[13] Q. Xu and N. Nicolici. Delay fault testing of core-based systems-on-a-chip. In *Proceedings IEEE/ACM Design, Autmation and Test in Europe (DATE)*, pp. 744–749, 2003.

[14] G. Loh, Y. Xie, and B. Black. Processor Design in 3D Die-Stacking Technologies. *IEEE Micro*, pp. 31–48, 2007.

[15] E. Marinissen, S. Goel, and M. Lousberg. Wrapper design for embedded core test. In *Proceedings IEEE International Test Conference (ITC)*, pp. 911–920, 2000.

[16] A. Rahman and R. Reif. System-level performance evaluation of three-dimensional integratedcircuits. *IEEE Transactions on VLSI Systems*, 8(6):671–678, 2000.

[17] G. Smith, L. Smith, S. Hosali, and S. Arkalgud. Yield considerations in the choice of 3D technology. In *Proceedings International Symposium on Semiconductor Manufacturing*, pp. 1–3, 2007.

[18] P. Varma and S. Bhatia. A Structured Test Re-Use Methodology for Core-Based System Chips. In *Proceedings IEEE International Test Conference (ITC)*, pp. 294–302, 1998.

[19] T. Vucurevich. The Long Road to 3-D Integration: Are We There Yet? *Keynote speech at the 3D Architecture Conference*, 2007.

[20] R. Weerasekera, L. Zheng, D. Pamunuwa, and H. Tenhunen. Extending systems-on-chip to the third dimension: performance, cost and technological tradeoffs. In *Proceedings IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 212–219, 2007.

[21] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie. Test-Access Mechanism Optimization for Core-Based Three-Dimensional SOCs. In *Proceedings International Conference on Computer Design (ICCD)*, to appear, 2008.

[22] X. Wu, P. Falkenstern, and Y. Xie. Scan Chain Design for Three-Dimensional Integrated Circuits (3D ICs). In *Proceedings International Conference on Computer Design (ICCD)*, pp. 208–214, 2007.

[23] Y. Xie, G. Loh, B. Black, and K. Bernstein. Design space exploration for 3D architectures. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 2(2):65–103, 2006.

[24] Q. Xu and N. Nicolici. Multi-frequency test access mechanism design for modular SOC testing. In *Proceedings IEEE Asian Test Symposium (ATS)*, pp. 2–7, 2004.

[25] Q. Xu, Y. Zhang, and K. Chakrabarty. SOC test architecture optimization for signal integrity faults on core-external interconnects. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pp. 676–681, 2007.

[26] Q. Xu and N. Nicolici. Resource-constrained system-on-a-chip test: a survey. *IEE Proceedings: Computers and Digital Techniques*, 152(1):67–81, Jan. 2005.

[27] Yole Development. Market Trends for 3D Stacking. http://www.yole.fr.

[28] F. Yuan and Q. Xu. SoC test architecture design and optimization considering power supply noise effects. In *Proceedings IEEE International Test Conference (ITC)*, paper 26.2, 2008.