Optimizing Test Architecture for TSV based 3D Stacked ICs using Hard SOCs

Surajit Kumar Roy[1], Chandan Giri[2], Arnab Chakraborty[3], Subhro Mukherjee[4], Debesh K. Das[5] and Hafizur Rahaman[6]

Dept. Information Technology
Bengal Engineering and Science University, Shibpur
Howrah, West Bengal, India

Dept. of Computer Sc. & Engg
Jadavpur Univerisy
Kolkata – 700032, India

{[1]suraroy, [2]chandangiri [3]grinit10, [4]subhro.mukherjee007} @gmail.com, [5]debeshkdas@gmail.com [6]rahaman_h@yahoo.co.in

Abstract— **In this paper we have addressed the test infrastructure design for TSV based 3D stacked IC (3D SIC). Each of the die consisting of one or more hard SOCs. Main objective of this work is to design the test architecture for the 3D SIC so that overall test time can be optimized. To prove the efficiency of our proposed algorithm we have considered a 3D stacked IC (SIC) using 5 standard SOCs. Obtained test results show that our proposed solution can achieve up to 59 % reduction in test time compared to the baseline method of sequentially testing all the dies in the stack. We have also shown that increasing the number of test access mechanism (TAM) and through silicon vias (TSVs) help in the reduction of test time but the increase in the number of TAM is unnecessary after a certain limit. In this work we have assumed that the different dies in different layers may consist of two SOCs as opposed to previous work, where each die consists of single SOC.**

I. INTRODUCTION

Semiconductor industry is pushing all the time for advanced performance, higher bandwidth and also aim at minimizing power usage and cost. To achieve such efficiencies, the concept of stacking multiple chips above each other and the use of mixed technology is slowly shifting us to the new design paradigm of semiconductor called 3-Dimensional System ICs (3D SICs). 3-Dimentional SICs have became the most common way to integrate disparate technologies that are being used in image sensors, microelectromechanical systems (MEMS) and different other heterogeneous elements, using the concept of mixed technology implemented on different layers individually before integration. Although several critical problems are yet to be resolved in 3D ICs, but still it is said to occupy a large market share in the near future [1]. For our usage we have designed a 3D SIC using standard SOC benchmarks with vertical interconnects called Through Silicon Vias (TSVs). Different dies are placed in different layers, all stacked one above another using wafer bonding or die tacking. These dies are connected by TSVs. Effective design and placement of each die in respective layers can result in optimization of TSVs, which can help to minimize the total chip manufacturing cost as well. Therefore mixed technology and multiple dies can result in higher on chip data bandwidth, reductions in average interconnect length and the problems associated with long global interconnects [2,3].

Besides the chip architecture and the concept of 3D SIC, testing of these chips require a special kind of test infrastructure to enable us to transport test data to the SOCs, present in each die. This test infrastructure is commonly referred as test access mechanism (TAM). The concept of TAM has been alleviated as 3D TAM to allow transferring of test data from input/output pins to different dies. Efficient utilization of TAM is necessary to minimize test time and also reduce the number of TSVs as well. Though several works have already been done for integrated wrapper and TAM optimization algorithms for 2D SOCs, but these algorithms are not applicable for 3D SICs. Hence TAM optimization algorithms are now an active area of research for 3D SIC.

In this paper, we propose an algorithm for TAM optimization of 3D SICs consisting of hard dies, that is the dies whose test architecture are already fixed (we have no scope to change them). After TAM optimization, we compute the least time required to test the complete stack of dies. While, it is easier to put different dies in different layers, the proposed solution will enable us to put lesser complex SOCs in higher layers to achieve the purpose of effective balance of TAM and TSV in each layer. The SICs has been created using 5 SOCs from the ITC'02 benchmarks [18]. In [13] authors have dealt with the same problem. But the basic difference between our proposed algorithm with the previous work is that during optimization of test time we have assumed each die in the SIC may consists of two SOCs rather than single one.

We have organized the rest of the paper in the following way. Section II provides an outline of the prior work. Section III explains the problem using a simple example and explains it in detail. Section IV describes our algorithm and its functionality using a standard 3D SIC. Section V shows the different experimental results for the 3D-SIC. Finally we conclude our paper in Section VI.

978-0-7695-4570-7/11 $26.00 © 2011 IEEE
DOI 10.1109/ISED.2011.29

IEEE computer society

II. Related Prior Work

Integrated TAM architecture and test schedule optimization for 2D SOCs are well studied in [4, 5, 6]. The optimization methods include ILP[6], rectangular bin packing [4] and some other heuristics like Genetic Algorithm [5]. But none of these methods have addressed the problems related to 3D ICs.

In [7] authors have addressed the testability issues of 3D-SICs by proposing a scan island based approach, which is like the wrapper standards IEEE 1500. Few papers [8-13] in this domain proposed test access architecture optimization for reducing test length and/or associated wire length. Wu et al [8] propose three scan chain length optimization algorithms considering the TSV based interconnects. The authors in [9] also have addressed the optimization of scan chain length along with scan power by reordering the scan chain using Genetic Algorithmic approach. In [13] authors have presented an optimization method to minimize the test time for a 3D-SIC in which each stack level contains exactly one die, either for the final stack test or for any number of multiple test insertions during bonding. Jiang et al. [11, 12] proposed simulated annealing (SA) based algorithms to optimize the pre-bond and post-bond test time of 3D ICs. A heuristic method to reduce weighted test cost with constraints on test pin width in pre-bond and post bond are discussed in [12]. Lo et al [14] propose test architecture for 3D-SICs, considering pre-bond and post-bond, as well as TSV based interconnect testing. This approach assumed that there is no circuitry within the dies in between the wrapper embedded cores which is often not the case. Marinissen et al [15] propose a DfT architecture based in IEEE 1500 standard for both pre-bond and post-bond test for 3D–SICs. In [16] authors also have shown that the DfT architecture can also be designed based on IEEE Std. 1149.1. In [17], Chi et al proposed the DfT architecture that allows placing multiple dies on top of a common base die, resulting in 3D-SICs with multiple “towers”.

III. Problem Definition

In 3D SIC, only the lowest die requires no TSV for testing purpose. But for upper layer dies, we need 3D TAM architecture, where test data can be transported through TAM channels. Also, to enhance the optimal architecture multiple SOCs can be placed in the same die or in stacked dies. Let the maximum available TAM width be denoted by W_{max}. Now it is known that, if all the SOCs are tested parallel, then it is possible that the obtained total test time is minimum and equal to $max\ (n_1, n_2, n_3, \ldots\ n_m)$ where m= number of SOCs and n_i is the test time of the i^{th} SOC. However the total TAM requirement for parallel testing will be equal to: TAM requirement of SOC_1 + TAM requirement of SOC_2 +…….. +TAM requirement of SOC_m , which is not feasible, if it exceeds W_{max}. So, it is not always possible to test all the SOCs in a parallel fashion. Consider an example where 3 SOCs are placed in 3 different layers with the following specifications presented in Table 1.

Table 1: Example SIC and their specifications

SOC	SOC 1	SOC 2	SOC 3
Test time	600	900	1200
TAM	40	70	80

Assume that the maximum available TAM width W_{max} is 140. Now if all the 3 SOCs are tested serially then the total test time will be 2700 clock cycles, and the total TAM width required will be 80. But, if they are all tested in parallel then the total test time will be 1200 clock cycles and the TAM width required will be 190, which exceeds the available TAM width. Considering SOC3 to be placed in lowest layer, the TSV required is 220. Thus a technique must be adopted to minimize test time taking into account the number of maximum available TAM width (W_{max}) and TSVs.

So the problem for optimizing the test infrastructure for SIC consisting of hard dies can be defined as follows:

Given a stack of N dies, where each die consists of at least one SOC, total available number of TAM width W_{max} and the number of available TSVs TSV_{max}, determine an optimal SIC design and test schedule and the position of each SOC in the stack such that the total test time T of the SIC is minimized and the number of TSVs does not exceed TSV_{max} and the number of TAM required does not exceed W_{max}.

IV. PROPOSED ALGORITHM

In our work we have assumed that all the SOCs are hard and each die of the SIC may consist of more than one SOC for optimizing the test time. But the algorithm proposed in [13] works on the principle of trying to test complex SOCs parallel and then test the next set of SOCs serially using Integer Linear Programming (ILP) formulation. This method assumed that each die is consisting of only single SOC. According to the algorithm proposed in [13], the set of TAMs that are allocated for them remain blocked, even if some other SOCs can be tested using the same TAMs. Therefore a technique is implemented to minimize the blocking time of the TAMs. This can help in two aspects:

1. Optimal and efficient use of available TAM width
2. Efficient utilization of TSV

The principal idea behind our proposed algorithm is to enable the parallel testing approach of SOCs in hard dies. The efficient utilization of available TAMs and TSVs are taken as the key focus in this paper. In previous approach [13], as there were many SOCs with different TAMs, when testing them in parallel fashion, the SOC with minimum test time is finished at the earliest. So, the TAM of the allocated SOC that have just ended is left unused. Our work focuses on this weakness of the previous test approach and attempts some recovery. The idea is based upon the use of unused TAMs. Hence a multi-layered approach is considered here with one or more SOC in each layer.

Consider n SOCs $(S_1, S_2, S_3,.,and\ S_n)$ with their test times sorted in increasing order :$T_1, T_2, T_3,.....an,T_n$. Their TAM requirements are: $W_1, W_2, W_3,......and\ W_n$ respectively. If the SOCs are tested serially, then the final test time will be equal to the sum of individual test times, which is the worst-case result. So the SOCs with higher test time are tested in parallel up to maximum limit, till the TAM requirement do not exceed the maximum number of available TAM width W_{max} and the number of TSVs used should be below TSV_{max}. According to our algorithm, SOCs $S_n, S_{(n-1)},...,S_j$ are taken into testing until total TAM width is nearest to minimum available TAM width and total TSVs used is nearest to TSV_{max}. To place the SOCs in different layers, the SOC with highest TAM requirement is kept at the ground level and the dies are stacked upon each other in decreasing order of their TAM requirements. But in case of $W_n=W_{(n-1)}=\ldots=W_j$, S_j is kept at the ground level, because if we keep the die with lowest test time at the ground level then multiple SOCs can be placed in the same level, leading to lower TSV requirement. Also, if we keep the SOCs with the decreasing order of their TAM requirement on top of each other then it is also possible to optimize the number of TSVs. Otherwise, if we keep the SOCs with higher TAM width on top of the SOC with lower TAM width then all the required TSVs for the SOCs requiring higher TAM will need to go through lower levels that can increase the TSV requirement.

Now, if all the SOCs are covered, then the test time is $T_{n,}$ which is equal to the maximum test time of the SOCs under testing. Again, if some SOCs are still left for testing, then the following steps must be implemented, until all the SOCs are tested. The algorithm is prescribed in the following steps:

1. Sort the SOCs with respect to test time
2. Now check whether any of the remaining SOCs fit in the currently available TAM width and TSV.
3. If any SOC is found in step (2) , then do the following:
    (a) Find out the SOC with the highest test time, which fits in the available TAM width and TSV.
    (b) Put the selected SOC obtained in step (3(a)) in the same layer as the SOC that has recently ended.
    (c) Now if all the SOCs are covered then the test time is $max(T_n,..T_{j+1},(T_k+T_j))$, where S_k gets chosen in step 3(a) and T_k is its test time.
4. If no SOC is found in step (2), go to step (1).
5. If any SOC left, continue with step (2), otherwise terminate test.

So, from the algorithm, we can say that:

1. If all SOCs are covered in the first hop then test time is T_n. Here T_n is the test time of S_n which is the SOC with highest test time.

2. In case of the step(2) if SOC S_j ends at the earliest and S_k is tested after S_j, then test time=$max(T_n,\ldots,T_{(j+1)},(T_j+T_k))$

3. In case of of (4) if S_k starts after $S_{(j+1)}$ has ended, then, test time = $max(T_n,\ldots,T_{(j+2)},(T_{(j+1)}+T_k),(T_j+T_k))$.

A. Time Complexity

Let us consider that there are n SOCs to be tested. The sorting in step-1 takes O(n log n) and O(n^2) time in best and worst case respectively. The complexity of checking whether an SOC will fit in the available TAM and TSV is O(log n). In the worst case, only a single SOC is tested in each hop. Then the worst case time complexity is:

$(1+2+3+\ldots.+n)*O(log(n))=n(n+1)/2*O(log(n))$
$=O(n^2)*O(log(n))=O(n^2)$.

In the best case, all the SOCs are tested in a single hop. So, the best case is when all the SOCs are tested in parallel in the first hop and the time complexity is $O(1)*O(log(n))=O(log(n))$.

In the average case, some of the SOCs can be tested in a single hop. For simplicity, we consider that in each hop m(m<n) number of SOCs are tested. Now, let us consider that r hops are required to complete the testing of n SOCs. For simplicity, we consider that n=m*r. In this case the time complexity is: $(m+2m+\ldots+r*m)*O(log(n))= m(r+1)/2* O(log(n))=(m*r+m)/2*O(log(n))=(n+m)/2*O(log(n)) < (n+n)/2*O(log(n)) = O(n\ log(n))$.

B. Illustrative Example

Let us consider a hypothetical SIC consisting of four SOCs(SOC1,SOC2,SOC3 and SOC4). Test times of SOC1, SOC2, SOC3 and SOC4 are 400,150,300 and 200 and their TAM requirements are 20, 10, 20 and 15 respectively. Also, the total available input TAM width is 40. So, according to our algorithm, we can say that SOC3 and SOC1 will be tested simultaneously in parallel, because they have higher test time and they fit well in the available TAM width. Thus, we place them in two separate dies. Now, testing of SOC3 ends earlier than SOC1.

Therefore, the 20 TAM of SOC3 are currently unused. So, we allocate SOC4 for testing as the available TAM width (20) = required TAM width (20) for SOC number 4. Therefore we place SOC4 in the same layer as SOC3. During testing of SOC4, SOC2 has to wait until SOC1 is completed. Once SOC1 is completed, we start testing SOC2. Thus we keep SOC2 in the same layer as SOC1. So in one layer (say, L1) we place SOC3, SOC4 and in another layer (say, L2) SOC1, SOC2. L1 requires maximum of 20 TAM width and L2 also requires maximum of 20 TAM width. Now, as the test time of L1 is less than L2, we place L1 at the ground level and L2 above it. So, TSV requirement is 40 (20 for upward and 20 for downward respectively). The final architecture has been shown in Fig.1.

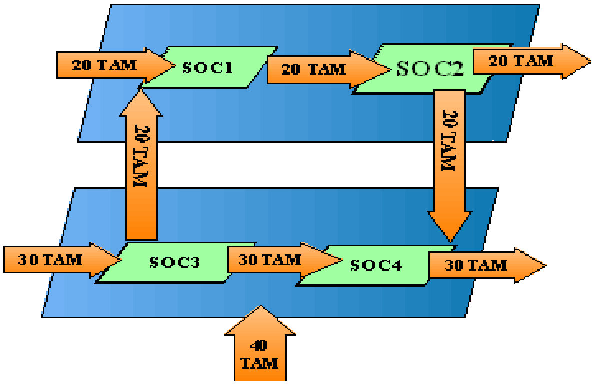

Figure 1: Final placement of 4 SOCs of the example

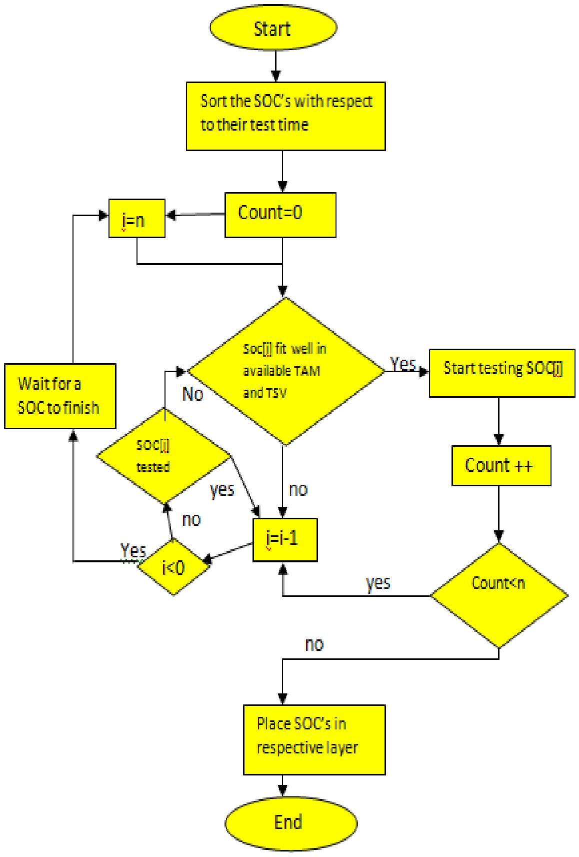

Figure 2: Flowchart for the proposed algorithm

V. EXPERIMENTAL RESULTS

Experimental results are presented based on the ITC'02 benchmarks [18]. Simulation of the results are done on a Dual Core processor having 2 GB RAM. Programs are written in C++ programming language. To illustrate the proposed algorithm, we have considered the same handcrafted a 3D-SIC using 5 SOCs as in [13]. The SOCs used are d695, f2126, p22810, p34292 and p93791. The results are presented in two aspects, where in one case we have varied TAM, keeping TSV constant and in the other case vice versa. Table 2 presents the specifications of each SOC which are hard.

Table 2: Test Lengths and number of Test Pins for dies as required in Hard Dies[13]

SOC	d695	f2126	p22810	p34292	p93791
Test Length	96297	669329	651281	1384949	1947063
Test Pins	15	20	25	25	30

Table 3 and Table 4 represent the simulation results obtained for variation of test time with respect to TSV (keeping TAM width fixed) and TAM (keeping TSV fixed) respectively. The test pins are assigned on the basis of the size of the die, to prevent large dies from taking longer test time. The work proposed in [13] considers each layer consists of one SOC, but in our work our proposed algorithm consider each die of a layer in SIC can contains more than two SOC for optimising overall test time of SIC. This is the basic difference between our work and work proposed in [13].

Table 3 shows the different SIC test time results, where we vary TSV keeping TAM width fixed. Column 1 of Table 2 shows the variation of TSVs keeping maximum TAM width fixed to 80 initially and then decreasing it to 75. Column 2 represents the number of test cycles obtained using our proposed algorithm for testing SIC. Column 3 shows the number of test cycles required when all the SOCs are tested sequentially. Column 4 provides the % improvement in test time results over serial testing approach. It is observed that an an increase in the TSV from 90 to 100 results in a 59% reduction in test time keeping W_{max}=80. For W_{max}=75, and a TSV_{max}=90, we achieve a reduction of 57.12% in test time.

Thus an increase in TAM and TSV results in a greater reduction in test time, but the increase in TAM width results in a greater improvement in test time than a subsequent increase in TSV.

Table 3: SIC test time with the variation of TSV keeping TAM width fixed

W_{max}	**TSV max**	**Required Test cycle**	**Serial Test cycle**	**% Impr. Over serial test**
80	**50**	2598344	4748920	45.29
	60	2598344	4748920	45.29
	70	2598344	4748920	45.29
	80	2598344	4748920	45.29
	90	2598344	4748920	45.29
	100	1947060	4748920	59
75	**50**	2598340	4748920	45.29
	60	2598340	4748920	45.29
	70	2598340	4748920	45.29
	80	2598340	4748920	57.12
	90	2036230	4748920	57.12
	100	2036230	4748920	57.12

Table 4 shows the experimental results for variation of test time with TAM width. The first column represents maximum TAM width used. Column 2 represents the required TSV to test the SOCs optimally with the available TAM. Column 3 shows the TSV requirement in case [13] for the same SIC.

Column 4 is the test time of the SIC obtained on the basis of the available input in column 1 and 2. Column number 5 depicts the test time achieved by [13]. Column number 6 represents the reduction of test time with respect to the serial testing approach. Finally, the column number 7 represents the reduction of test time with respect to the approach proposed in [13]. It is seen that an increase in TAM width (W_{max}) gradually results in a greater reduction in test time. Maximum reduction of 59% is achieved for W_{max} =80 keeping

TSV_{req}=100, which gives 25% improvement in test time over [13]. Hence we can see that the test time is decreasing with the increase in TAM width and the maximum reduction is achieved with TAM width 80. So, our convention places multiple SOCs in a single die optimises the test time better than keeping an SOC in single die. It is also evident from Table 4 that by placing multiple SOCs in a single die can utilize the TSVs efficiently.

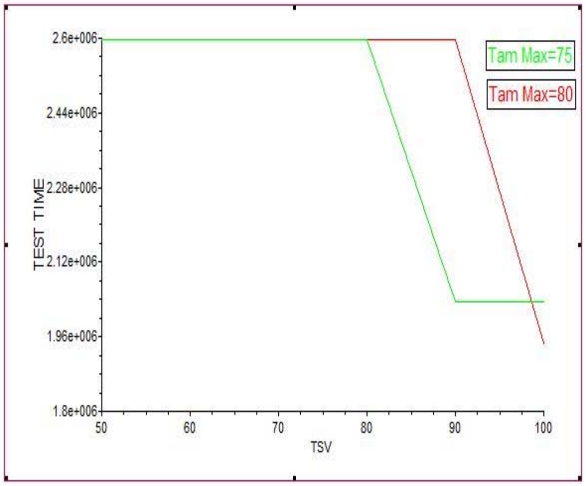

Figure 3: Graph showing variation of test cycles with TSV

Figure 3 demonstrates the variation of Test cycle with TSV. Input TAM requirement has been fixed at 75 and 80 respectively, to present two separate distributions. We can see that a variation in test time occurs drastically, when the TSV is increased from 90 to 100 but the same cannot be obtained for TAM width=75. Therefore a greater increase in TAM width results in a more efficient reduction in test time.

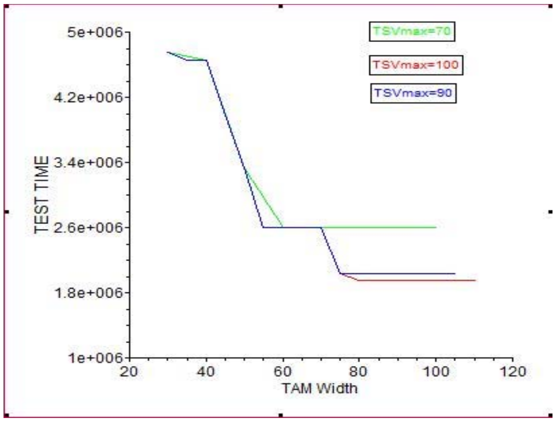

Figure 4: Graph showing variation of testcycle with TAM

Similarly, Figure 4 is used to depict the variation of test time with TAM width keeping TSV constant to 70, 90 and 100 respectively. Again we can see that increasing the TAM results in a greater reduction in test time. But the increase in TSV is also necessary to reach optimality. Thus efficient utilization of TAM and TSV is required to achieve optimal test time.

Table 4: Experimental results for variation of TAM

W *(max)*	TSV req(our)	TSV req for [13]	Test cycle	Test cycle [13]	% Impp. w.r.t serial testing	% Impro. w.r.t [13]
30	0	160	4748920	4748920	0.0	0
35	50	160	4652620	4652620	2.03	0
40	50	160	4652620	4652620	2.03	0
45	40	160	3983290	3983290	16.12	0
50	50	160	3332010	3428310	29.84	2.8
55	50	160	2598340	2712690	45.29	4.22
60	50	160	2598340	2616390	45.29	0.7
65	50	160	2598340	2616390	45.29	0.7
70	50	160	2598340	2616390	45.29	0.7
75	90	160	2036230	2598340	57.12	21.63
80	100	160	1947060	2598340	59	25.07
85	100	160	1947060	2598340	59	25.07
90	100	160	1947060	2598340	59	25.07
95	100	160	1947060	2598340	59	25.07
100	100	160	1947060	2043360	59	4.71
105	100	160	1947060	2043360	59	4.71

VI. CONCLUSION

In this paper we have tried to optimize the testing time of a SIC consists of hard dies with respect to TAM and TSV requirements. For hard dies, the test architecture is fixed and given. Results show that an increase in the number of TAM and TSV, result in a decrease in the test time. Also, an increase in the TAM provides greater reduction in test time. The algorithm helps to efficiently use TAM and TSV which results in lesser cost and complexity. Also the 59% reduction in test time has been achieved with respect to the serial testing approach. The result shows three key improvements compared to the prior works in this field. First is optimal reduction in test time 59% with respect to serial test, second is 25 % reduction compared to the prior work in [13] at the same hardware usage, and third is efficient usage of TAM and TSV which minimizes test cost.

ACKNOWLEDGEMENT

This work is sponsored partly by VLSI Design Project from Govt. of West Bengal and by the UGC, Govt. of India under the grant of F.No.40-1/2011(SR), dated 29.07.2011.

REFERENCES

[1]. Yole Development. Market Trends for 3D Stacking. http://www.yole.fr.
[2]. G.Loh, Y.Xie, and B.Black, "Processor Designing 3D-Die Stacking Technologies ",*IEEE Micro* Vol27, No.3, 2007, pp.31-48

[3]. W.R.Davis et al., “Demystifying 3DICs : the Pros and Cons of Going Vertical”, *IEEE Design and Test of Computers*, vol.22, no.6, pp.498-510, 2005.

[4]. Y.Huang et al, “Optimal Core Wrapper width Selection and SOC Test Scheduling based on 3-D Bin Packing Algorithm”, In *IEEE International Conference on Computer Design(ICCD)*, pp.74-82,2009.

[5]. C. Giri and S. Sarkar and S. Chattopadhyaya, “A Genetic Algorithm Based Heuristic Technique for Power Constrained Test Scheduling in Core-based SOCs”, In *Intl. Conf on IEEE VLSI-SOC*, pp.320 – 323, 2007

[6]. V. Iyengar and K. Chakrabarty and E. J. Marinissen, “Test Wrapper and Test Access Mechanism Co-optimization for System-on-a-Chip”, In *Journal of Electronic Testing, Theory and Applications*, Vol. 18, pp. 213-230, 2002.

[7]. D. L. Lewis and H.-Hsin S. Lee, Testing Circuit Partitined 3D IC Desing, In *Proc. ISVLSI*, pp.139-144, May 2009.

[8]. Xiaoxia Wu, Paul Falkenstern, and Yuan Xie, “Scan Chain Design for Three-Dimensional Integrated Circuits (3D ICs). In *Proc. IEEE ICCD*, pp: 208-214, October 2007.

[9]. C. Giri et al, “Scan Chain Design Targeting dual Power and Delay Optimization for 3D Integrated Circuits, in *Proc. IEEE Intl. conf. on Advanced computing, Communication and Technology(ACT*), 2009.

[10]. Xiaoxia Wu et al. Test Access Mechanism Optimization for Core-based Three Dimensional SOCs. In *Proc. IEEE ICCD*, pp. 212-218, 2008.

[11]. L.Jiang,L.Huang,andQ.Xu,“TestArchitectureDesignandOptimizationfor Three-dimensional ”, *In Proc. DATE*, pp.220-225,2009

[12]. L. Jiang, Q.Xu, K.Chakrabarty and T.M.Mak, “Layout driven test-architecture design and optimization for 3D under pre-bond test-pin-count constraint”, *Proc. IEEE International Conference on Computer-Aided Design*, pp.191-196, 2009.

[13]. B. Noia, et al, “Test Architecture Optimization for TSV Based 3D-Stacked ICs.”, Proc. *15th IEEE ETS*, pp. 24-29,2010

[14]. Chih-Yen Lo et al, “SOC Test Architecture and method for 3D-ICs”. In *IEEE Transaction on CAD*, 29(10):1645-1649, Oct. 2010.

[15]. Erik Jan Marinissen et al., “A Structured and Scalable Test Access Architecture for TSV-Based 3D Stacked ICs”, *In Proc.IEEE VTS*, pp:269-274, April 2010.

[16]. Erik Jan Marinissen et al., “3D DfT arcitecture for Pre-bond and Post-bond Testing”, in *Proc. IEEE Intl. 3DIC*, Nov. 2010.

[17]. Chun-Chuan Chi et al, “DfT Architecture for 3D-SICs with Multiple Towers”, In *Proc. ETS,* pages 51-56, 2011.

[18]. E. J. Marinissen et al. “A Set of Benchmarks for Modular Testing of SOCs”. In *IEEE ITC*, pp.519-528,Oct.2002.