Session based Core Test Scheduling for Minimizing the Testing Time of 3D SOC

Surajit Roy, Payel Ghosh, Hafizur Rahaman and Chandan Giri

Abstract – **Three dimensional (3D) VLSI integration based on through-Silicon-Via (TSV) is an emerging technology. It provides heterogeneous integration, higher performance, bandwidth, and lower power consumption. However, 3D-IC suffers from several challenges. The objective of this paper is to design the test access mechanism (TAM) architecture and test scheduling of different modules of an system-on-chip (SOC) such that the overall test time of that SOC gets reduced. In this paper we have used a session based heuristic approach to solve this problem. Experimental results have been tested on different ITC'02 benchmark SOCs that shows promising results for different TAM width allocation.**

Keywords – **3D IC testing, test access mechanism, test scheduling.**

I. Introduction

Moores's law suggests that, for a given size of integrated circuit (IC), the number of transistors on IC gets doubled approximately in every one and half years. In case of 2D IC, if the number of transistors increases, then the size will automatically increase. The gate delay and power consumption also gets increased. So for better performance the size of the IC must be kept as small as possible. Hence 3D integration for VLSI is the effective solution instead of 2D integration.

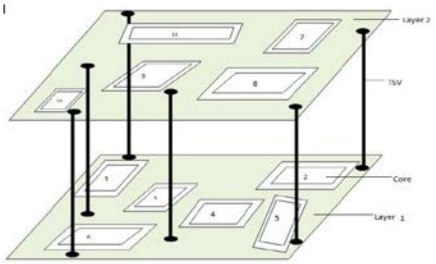


Figure 1. 3D SOC structure

An example of 3D SOC is shown in Figure 1. The entire chip is divided into number of blocks (called cores) which are placed on separate layers and these layers are stacked on top of each other. Layers are connected by *through silicon via (TSV)* and have global interconnection through wires, resistances and capacitances. In 3D architecture, as the design blocks are stacked on top of each other, size of the chip can be kept small. 3D SOCs are also advantageous [2][8] for its scalability, cost, heterogeneous integration, shorter interconnection, power consumption, bandwidth etc. But in spite of all those benefits 3D-IC suffers from great challenge in testing. Testing the cores of an SOC test access mechanism (TAM) is required to transport test patterns and test responses between SOCs test pins and core I/Os. Proper allocation of TAMs to cores reduce this overall testing time of the SOC, which is assumed to be an NP-Hard problem.

In this work, we have proposed a session based heuristic approach for test architecture design and test time optimization problem of 3D SOCs. Experimental results on ITC'02 benchmark SOCs shows the effectiveness of the proposed method compared to already propose 3D test architecture and scheduling optimization methods.

The rest of this paper is organised as follows. In section 2, we have mentioned the related prior works. In section 3, problem formulation is described. Section 4 discusses the proposed method in details. Section 5 shows the experimental results and compared them with other existing approaches and finally section 6, concludes this paper.

II. Prior Work

The first work on TAM design for 3D-SoC [6] presented an integer linear programming (ILP) model. The method divides the total TAM wires into several test buses with fixed width and to assign modular cores to test buses so as to minimize the overall test time under the constraint of TSV count utilized by TAMs. In [1] authors have proposed an efficient thermal aware heuristics to resolve the testing problem for large and complex 3D-SoC devices. A flexible TAM architecture with thermal consideration is designed in [5] that maximize test concurrency by rearranging different sessions. In [3] authors have proposed a TAM optimization technique that does not impose any limits on the number of TSVs used for the TAM, but considers pre-bond testing considerations and wire length limits. A drawback of this approach is that since it does not limit the number of TSVs for the TAM, it ignores constraints related to the keep-out area that is associated with a TSV. In [6] authors have also been reported on the testing of TSVs and an optimization method for 3-D stacked ICs with die level test architecture. Jiang et al. [7] have proposed simulated annealing (SA) based algorithms to optimize modular SOC test architecture

Manuscript received September 9, 2013.

Surajit Roy is with Dept. of Information Technology, Bengal Engineering & Science University, e-mail: suraroy@gmail.com

Payel Ghosh is with Dept. of Information Technology, Bengal Engineering & Science University, e-mail: ghoshpayel89@gmail.com

Hafizur Rahaman is with Dept. of Information Technology, Bengal Engineering & Science University, e-mail: rahaman_h@yahoo.co.in

Chandan Giri is with Dept. of Information Technology, Bengal Engineering & Science University, e-mail: chandangiri@gmail.com

considering both pre-bond tests and post-bond test. In this approach, the same TAMs that traverse multiple layers in post-bond testing are fully reused for pre-bond tests.

III. Problem formulation

The objective in this work is to generate an efficient way of designing a test schedule with optimized testing time considering the maximum available TAM width and maximum available TSVs. So the problem can be formulated as follows:

Given a three dimensional SOC with i) n number of cores, ii) total TAM width limit (TAM_{max}), iii) total TSV limit (TSV_{max}), iv) number of input/outputs, number of scan chains and their lengths, iii) core placement details iv) layer details and v) testing time of each core for different TAM widths determine the scheduling order of cores in different sessions with the TAM_{max} such that the overall testing time is optimized with the constraints of TSV_{max}.

IV. Proposed Algorithm with example

Figure 2 describes the proposed algorithm. Our proposed algorithm works in following manner. First sort all the cores of the SOC in ascending order of their test time for given TAM width, store them in *cid* and calculate the total test time. Then the scheduling mechanism is performed.

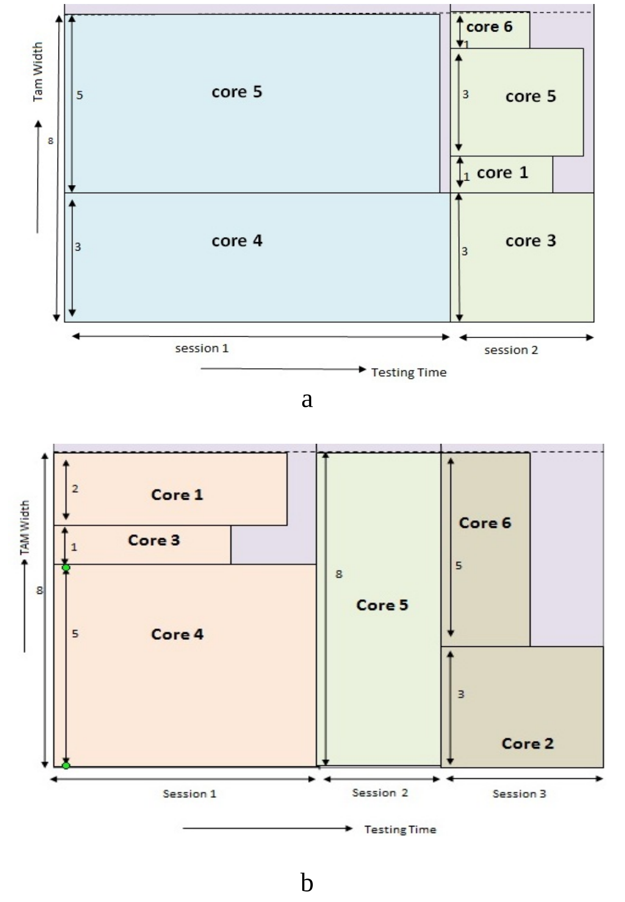
```
Algorithm: Scheduling

1.  cid=list the order of cores of the SOC under test with the
    increasing order of testing time for given maximum
    possible TAM width $TAM_{max}$.
2.  Calculate the overall test time as "t_time" by adding the
    test time (obtained in step 1) of all cores
3.  While the 1st pare-to optimal point of the core with
    maximum testing time ($C_L$) is not reached
    a)  While all cores in cid are not scheduled
        i.   Select the core from the non scheduled core
             list which have the maximum testing time
             for the maximum possible TAM for that
             core
        ii.  Schedule other possible unscheduled cores
             which have lesser or equal test time than
             the core chosen in step a.i for remaining
             available TAM width for that session.
    b)  Record the order of scheduled core for different
        sessions.
    c)  Calculate the total test time as "time_new" by
        adding the maximum test time of each session
    d)  If the time_new < t_time,
            then t_time=time_new
    e)  Update the TAM width of the core $C_L$ by
        decreasing its value by 1 and repeat from step3.
4.  End
```

Figure 2. Proposed Heuristic Algorithm

First select the core (C_L) with maximum testing time and assign to it the maximum possible TAM width. If all the TAM width are allocated to this core then the session is completed and we create a new session for the remaining cores.

If all the TAM is not used for that core then select the cores in the order as they are in *cid* such that they satisfy the remaining available TAM width as well as the testing time lesser or equal to C_L for that session. Rest of the cores is scheduled in same fashion in different sessions. After scheduling of all cores of the given SOC the overall testing time is calculated by adding up the maximum test time of all sessions. Repeat the entire scheduling procedure until the first pareto optimal point [9] of C_L is reached. The iteration which gives the optimum scheduling order of the cores, record the details of that iteration as the final scheduling order of the cores with respective TAM width and the corresponding overall test time is considered as the resultant overall test time of that SOC.



a

b

Figure 3. Intermediate Core test architecture with max TAM width 8 (a), Final Core test architecture with max TAM width 8 (b)

Assume an SOC having six cores, namely core1, core2, core3, core 4, core5 and core6 of a SOC are there for testing and the given maximum available TAM width is 8. Assume the test time of the cores are 2507, 1624, 3305, 8342, 5829, 64 respectively for maximum available TAM width. At first all these cores are sorted according to their minimum testing time for given TAM width in ascending order and get the order of cores as 6,2,1,3,5,4. After that the actual scheduling process (Step 3) is started. At first, core 4 is chosen for scheduling in 1st session as it has the largest testing time. Then core 3 and 1 are selected respectively for scheduling in that session as they have lesser or equal testing time than core 4 with remaining available TAM width. Hence cores 4, 3 and 1 can be tested parallel and this will be a test session with maximum available TAM width. The rest of the cores are scheduled in same fashion in 2nd and 3rd session as shown in Figure 3b. When all six cores are scheduled then the overall test time is calculated by adding up the maximum testing time of all the sessions. For further reduction of test time again the sessions are formed. This is done as follows: currently assigned TAM width of the largest core (as in previous session) is decreased by 1 and repeats the same scheduling Step 3. This process is continued until the TAM width of the largest core is reached to the 1st pare-to optimal point of that core. Another scheduling architecture is shown in Figure 3a. Since it does not produce optimum results therefore the final configuration is the previous one (shown in Figure 3b).

V. Experimental Results

The proposed algorithm is implemented using C programming language and GCC compiler in Linux platform is used here for simulation. The program is run in HP workstation with Core i7 processor and 4 GB RAM. Simulations are done on ITC'02 benchmarks SOC like p22810, 993791, p34392. Results are presented in Table I and Table II. ΔT represents the improvement in test time (in %) with respect to our proposed method with other existing methods. ΔT is calculated as follows:

*ΔT= ((Test time corresponding to Existing method - Test time corresponding to our method) / Test time corresponding to existing method) * 100%.*

The values of all testing times are in "Clock Cycles" unit.

Test time results for SOCs p22810, p34392 and p93791 are compared with [4] and [6] as no constraints like thermal, power etc. are presented. Table I shows the results of p22810 with existing methods and Table II presents the experimental results for the SOCs p93791 and p34392 with relative improvements in test time compared to the [4] and [6] respectively.

Table I
Experimental Results and Comparative Study of SOC p22810

TAM WIDTH	p22810, TSV_{max} = 80				
	Our Method	Method [4]	ΔT_1 (%)	Method [6]	ΔT_2 (%)
16	479211	959753	50.07	275537	-73.92
24	357743	7387800	95.16	249274	-43.51
32	243303	653060	62.74	205831	-18.20
40	204687	611087	66.50	181687	-12.66
48	186511	587328	68.24	169392	-10.11
56	156391	563882	72.26	153282	-2.03
64	134820	553497	75.64	142210	5.20

In p22810 our proposed method provides 68.03% improvement in test time with compared to [4] for 64 TAM widths. In p93791 our proposed method provides 71.56% improvement in test time with respect to [4] for 64 TAM widths and In p34392 our proposed method shows 70.07% improvement in test time with respect to [4] for 56 TAM width.

But compared to the other method in [6] though our method does not perform well but it can be noted that for higher order TAM widths our proposed method performs better. Also it can be noted that with respect to the running time of the algorithm our proposed heuristic can do the simulation in a very negligible time compared to [6] as shown in Table III.

VI. Conclusions

In this paper, we have implemented an efficient modular based approach for minimizing the overall test time of a 3D SOC considering the total TAM width and TSV as constraints. The comparative study shows that the obtained results are better than the existing algorithms in most of the cases in terms of testing time. As the TAM distribution and assignment of them to cores for testing is performed in a dynamic way therefore the algorithm works in a much more flexible manner.

Table II
Experimental Results and Comparative Study of SOC p93791 and p34392

TAM WIDTH	p93791, TSV_{max} = 80					p34392, TSV_{max} = 80				
	Our Method	Method [4]	ΔT_1 (%)	Method [6]	ΔT_2 (%)	Our Method	Method [4]	ΔT_1 (%)	Method [6]	ΔT_2 (%)
16	1824647	3726714	51.04	1779298	-2.55	1024937	2088962	50.94	999543	-2.54
24	1263957	2791223	54.72	1197683	-5.53	692273	1791078	61.35	762841	9.25
32	898061	2390750	62.44	894463	-0.4	625938	1595286	60.76	685445	8.68
40	763782	2153380	64.53	778296	1.86	549700	1538795	64.28	552231	0.46
48	680940	1946263	65.01	713347	4.54	544579	1567728	65.26	544579	0
56	624697	1842030	66.09	635095	1.64	544579	1819571	70.07	540069	-0.84
64	494236	1737586	71.56	572342	13.65	544579	1758060	69.02	534212	-1.94

TABLE III

EXECUTION TIME of SOC p22810, 093791, p34392 COMPARING WITH [6]

TAM Width	**P22810**			**P93791**			**P34392**		
	(ours) (sec)	**Method[6] (sec)**	**Δt (%)**	**(ours) (sec)**	**Method[6] (sec)**	**Δt (%)**	**(ours) (sec)**	**Method[6] (sec)**	**Δt (%)**
16	0.001407	20.24	99.99	0.000845	15.70	99.99	0.001049	12.2	99.99
24	0.001491	30.89	99.99	0.001754	18.70	99.99	0.001097	6.35	99.98
32	0.001549	48.53	99.97	0.001830	16.43	99.99	0.001129	7.26	99.98
40	0.00935	32.59	99.99	0.001951	40.59	99.99	0.001161	12.59	99.99
48	0.001005	53.82	99.99	0.002054	81.82	99.99	0.001212	23.82	99.99
56	0.001304	122.21	99.99	0.002108	180.67	99.99	0.001234	122.21	99.99
64	0.001664	33.71	99.99	0.002192	332.71	99.99	0.001261	33.71	99.99

VII. REFERENCES

[1] Unni Chandran and Dan Zhao,"Thermal Driven Test Access Routing in Hyper-interconnected Three-Dimensional System-on-Chip", pages: 410-418, 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2009.

[2] Chao Zhang, Guangyu Sun, "Fabrication cost analysis for 2D, 2.5D and 3D IC Designs", in Proc. 3D Systems Integration Conference (3DIC), pp. 1-4, 2012

[3] Li Jiang, Qiang Xu, Krishnendu Chakrabarty, and T. M. Mak, “Layout-driven test architecture design and optimization for 3D SOCs under pre-bond test-pin-count constraint”, in Proc. of the 2009 International Conference on Computer-Aided Design, pp. 191-196, 2009.

[4] Chi-Jih Shih, Chih-Yao, Chum-Yi Kuo, James Li, Jiann-Chyi Rau, K. Chakrabarty, “Thermal Aware Test Scheduke and TAM Co-Optimization fir Three-Dimensional IC”, in Proc. Active and Passive Electronic Components, Vol 2012, 10 pages, 2012.

[5] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test Access Mechanism Optimization for Core-based Three-dimensional SOCs", in Proc. of IEEE International Conference on Computer Design, pp. 212-218, 2008.

[6] B. Noia, K. Chakrabarty, S. K. Goel, Erik Jan Marinissen, Jouke Verbree, "Test-Architecture Optimization and Test Scheduling for TSV-Based 3-D Stacked ICs", in proc. of Computer-Aided Design of Integrated Circuits and Systems, pp. 1705-1718, 2011

[7] Li Jiang, Lin Huang, Qiang Xu,“Test Architecture Design and Optimization for Three-Dimensional SoCs", in Proceedings of Design, Automation & Test Europe Conference & Exhibition, pp. 220-225, 2009.

[8] Khan, N.H., Alam, S.M., Hassoun, S., "System-level Comparison of Power Delivery Design for 2D and 3D ICs", in 3D System Integration, pp. 1-7, 2009.

[9] Vikram Iyengar and Krishnendu Chakraborty, “Test Wrapper and Test Access Mechanism Co-optimization for System-On-Chip”, In Journal of Electronic Testing: Theory and Applications (JETTA), vol. 18, pp. 213-230, April 2002.