Robust Test Architecture Optimization of 3D Stacked ICs based On Uncertainty

Sonika Nayan
Department Of Information Technology
Indian Institute Of Engineering Science
And Technology(IIEST), shibpur,
Howra
sonikanayan@gmail.com

Sonali Nayan
Department Of Information Technology
Indian Institute Of Engineering Science
And Technology(IIEST), shibpur,
Howrah
sonalinayan25@gmail.com

Anjali Chachara
Department Of Computer
K.J.Somaiya College Of Engineering
anjalichachra24@gmail.com

Abstract: 3D integration offers many benefits, such as high band width, low power and small footprint. But test complexity, test cost are main problem for 3D SIC. Recent work on the optimization of 3D test architectures to decrease test cost suffer and have the shortcoming that they do not consider uncertainties in input set. They consider only single input-parameter set. But the fact is that assumed values for input set such as test power, and pattern count of logic cores, which are used for optimizing the test architecture for a die, may vary from the actual values which are fixed at the design stage. Previous optimization methods taken into account the worst-case guess for all input parameters to make sure the workability, which can result in low and hence costly solutions. Objective of this project work is to find robust test architecture optimization of the of 3D stacked ICs to reduce the overall test time of SOC considering the maximum available TAM width in case of uncertainty in TAM configuration. Experimental results are presented for several ITC02 benchmark SOCs which show promising results for different TAM width allocation on.

Index Terms—*TAM (Test access mechanism), Optimization, SOC, Test Architecture*

*****

I. INTRODUCTION

1.1 Core based (3-D SOC)

Three-dimensional (3D) incorporation innovation offers the guarantee of being another method for expanding framework execution. This guarantee is because of various trademark elements of 3D joining, including (a)decreased absolute wiring length and along these lines diminished interconnect delay times (b) drastically expanded number of interconnects amongst chips and (c) the capacity to permit divergent materials, process advances, and capacities to be integrated[10].

Three-dimensional stacking ICs using through-silicon via (TSVs) has empowered another approach to incorporate more gadgets in a solitary package [6]. TSVs are metal by means of that give electrical associations starting with one pass on then onto the next through the silicon substrate [1]. TSV innovation interconnects stacked gadgets on wafer or kicks the bucket level for superior correspondence, in particular hyper between associations.

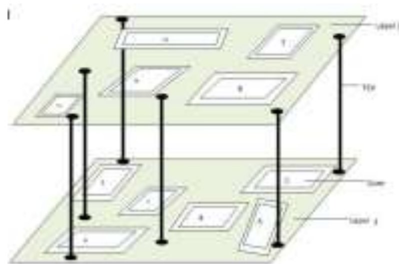

Figure: 1 3D SOC structure

An example of 3D SOC is shown in Fig 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. Testing of the cores of an SOC requires test access mechanism (TAM) for transporting test patterns and test responses between SOCs test pins and core I/O. Proper allocation of TAMs to cores of an SOC for testing reduce the overall testing time of the SOC. 3D ICs is a chip in which two or more layers of active electronic component are integrated both vertically and horizontally into a single circuit[12][9]. Test wrapper is a layer of DFT logic that connects a TAM to a core for the purpose of test. TAM delivers pre-computed test access sequence to the cores on the SOC while test wrapper translates these test sequence into patterns that can be applied directly to the cores.

1.2 3-D stacked ICs offer many benefits over 2D ICs:

- High band width - Where large number of vertical vias allows construction of wide bandwidth and low power consumption.
- Heterogeneous integration - Allows integration of complex system comprising a large number of different modules such as embedded cores in a single die or package. 3D stacking permits heterogeneity, since the dice arranged one above the other need not be similar. This allows a designer to optimize components according to their function.

- Faster interconnection and reduction in average interconnection length - Since the core are close by, the interconnect length is significantly smaller. This results in low latency and higher performance.
- Decrease size and weight - 3D chips save space. Core that used to be centimeters apart on a chip can now be placed millimeters apart, and that too vertically can be seen in figure1.2.
- High performance - 3D interconnects permit data to be moved both horizontally and vertically. This boost performance by 30 to 40 percent.
- Scalability - More functionality can be fits into a small space. It makes devices tiny but powerful.
- Cost - Partitioning a large chip into multiple smaller dies with 3D stacking can improve the yield and reduce the fabrication cost.

1.3 Challenges of 3-D ICs

3D stacked ICs has the potential to integrate a number of large SOC, which leads to many challenges such as the test complexity, test cost, test time, design issue, thermal management and distribution of power density. Compared to the 2D designs, the 3D-SoC poses great challenges in testing because of (1) extremely limited test resources such as pin count, TSV density, and routing area that are shared among more and more cores in three dimensional placements. Especially the pins are only located at the bottom tier, and the test access to the upper tiers are through the TSVs. (2) Thermal management becomes even more critical due to the increased power density and the non- uniform distribution of power density. In 3D Stacking multiple active dies are placed directly on top of each other which lead to high concentrations of heat which cannot easily be dissipated. (3) As the cores are deeply embedded and spread among multiple tiers, the three dimensional test access architecture faces a more complex and comprehensive configuration environment. Since there are many extra steps involved in manufacturing 3D ICs, the possibility of defects introduced in each step is also high. Testing of independent dice is relatively tough because of the tight integration between active layers. This introduces further possibility of defects going undetected.

1.4 Motivation

Late work on the streamlining of 3D test designs to lessen test cost experience the ill effects of the disadvantage that they overlook potential instabilities in info parameters; they consider just single point information parameter space. Hence, our inspiration in this anticipate work is to streamline the test time of 3D stacked ICs in the event of instability in information parameters which prompts increment in testing time and there by test cost. Various indeterminate information parameters are there like Test force, Available transmission capacity for test information use, TAM correspondence structure, design tally of rationale centers and test time. An exceptionally restricted work has been completed on the aforementioned issues of 3-D SOCs. This spurs our work for finding powerful test engineering improvement of 3D stacked ICs considering TAM width as questionable parameter.

How uncertain input parameter leads to increase in test cost?

As in 3-D stack ICs a die can be used in multiple stacks each with different property. As a result the originally designed test architecture no longer be optimal which leads to increase in test cost and test time.

TAM being reconfigurable width becomes uncertain input parameter. Thus not known a priori in which configuration die will be tested. In a 3D setting, the test architecture can be designed to be flexible in order to enable compatibility between dies that will be stacked in different stacks. Even though the die manufacturer and the stack integrator have to agree on the physical interface of the dies to be integrated in a stack, the die-level test-access mechanism (TAM) may be reconfigurable, for instance, in order to increase the bandwidth and make maximum use of the channels provided by the test equipment.

1.5 Objective

Objective of this work is to find robust test architecture optimization of the 3D stacked ICs to reduce the overall test time of SOCs considering the maximum available TAM width in case of uncertainty in TAM configuration which leads to increase in testing time and there by test cost.

In this work, a heuristic method has been proposed for robust optimization of test architecture design and test time optimization problem of 3-D SOC. Experimental results show the effectiveness of the proposed methods.

II. LITERATURE SURVEY

Various strategies have been investigated before to streamline the 2D SOC test engineering. Late productions have been concentrating on improvement of test engineering of 3D stacked ICs. However these strategies accept referred to esteem for all info parameters and additionally settled stack-level TAM width. Wrapper-based daisy-chain engineering has been proposed for 3D-SICs which is under thought for institutionalization by the IEEE P1838 Working Group [3]. Take a shot at 3D test methodologies have tended to this issue and introduced a few strategies for test engineering streamlining and test booking. These methods are based on exact optimization techniques such as integer linear

programming (ILP) and heuristics, for example, rectangle pressing [2]–[4]. The proposed wrapper depends on IEEE Std 1500 and takes into account serial and parallel designs. The width of the TAMs in parallel arrangements can be diverse in pre-bond and post-bond modes. We can actualize extra parallel setups with variable TAM width and still guarantee similarity with the initially proposed wrapper. The stream for 3D-wrapper insertion has been robotized in [7] and this structure can be effortlessly reached out to bolster additional parallel post-bond tests with various TAM widths. Daisy chain design at stack-level has been considered. The inspiration for a reconfigurable kick the bucket level TAM depends on the reconfigurable center wrapper approaches displayed in [11]. The proposed center wrappers take into account a flexible center level TAM. The output chains are parceled into gatherings such that the aggregate test time is ideal for each setup [16]. This thought has been stretched out to 3D wrappers for TAM at kick the bucket level.

To deal with the issue of uncertainties in input parameters for optimization, a method called "robust optimization" has been proposed in the past [5]. This approach takes variations in input parameters into account and finds a solution that does not deviate much from the optimum in case the parameters change [14].

Integer linear programming (ILP) is used to formulate the robust test-architecture optimization problem, and the resulting ILP model serves as the basis for a heuristic solution that scales well for large designs [13]. The proposed optimization frame work is evaluated using the ITC'02 SOC benchmarks [8]. And it is found that robust solutions are superior to single-point solutions in terms of average test time when there are uncertainties in the values of input parameters.

A robust solution may be inferior to a solution optimized for certain values of the input parameters in case their estimates were accurate; however, a solution performs better than non-robust solutions by staying closer to the optimum in the presence of variations.

ILP is impractical for solving the complete robust optimization problem. So it is exploit for solving sub-problem in our large framework in order to optimize the test schedule for a given test architecture and a given set of input parameter. We show that robust solutions result in lower average test time compared to non-robust Single-point solutions.

For larger systems, for which the exact solution of the ILP model becomes intractable, "randomized divide-and-conquer" heuristic for robust optimization that uses the exact ILP method to solve sub-problems are used. The proposed heuristic scales linearly with the number of cores and is hence practical for large systems [15]. In addition, since the iterations during the randomized divide-and-conquer procedure are independent from each other, the performance of our heuristic can be increased by running multiple instances (or thread) in parallel. A flexible TAM architecture is designed which maximizes test concurrency by efficiently utilizing high density vertical interconnection while tackling thermal safety issue by incorporating thermal compatibility into scheduling.

This section presents examples which show how TAM becomes an uncertain input parameter. In this case, the width of the TAM becomes an uncertain parameter at the design stage. Since it is not known *a priori in* which configuration the die will be tested. Even though the die manufacturer and the stack integrator have to agree on the physical interface of the dies to be integrated in a stack, the die-level test-access mechanism (TAM) may be reconfigurable, for instance, in order to increase the bandwidth and make maximum use of the channels provided by the test equipment. With this flexible 3D test architecture, the die can be integrated in different stacks that use n, $2n$, or $4n$ test inputs. Hence the die level TAM width in test-architecture optimization for the stack becomes a distributed input parameter at the design stage. If such potential uncertainties in input parameter is ignored and consider only single point in input parameter space then it leads to undesirable increase in test cost.

Maximum possible TAM width is used to find the least possible testing time of all core. We use the rectangle-packing based formulation to model the scheduling problem. The cores are represented as rectangles with the height denoting the number of TAM wires assigned to the core and the width denoting the test time in terms of the number of clock cycles. Without exceeding the total TAM limit, we try to allocate more cores in parallel with it in time dimension. Concept of Multi-Visit TAMs, i.e parallel TAMs which are allowed to visit the same die more than once is used. At the expense of a minimal adaptation of the die wrapper hardware, the Multi-Visit TAMs offer more flexibility in identifying TAMs in existing interposer designs, hence achieve significantly higher success rates and lower test lengths. Multi visit is used to reduce post bond length.

III. Heuristic method for robust Optimization of 3-D SOC in different configuration of TAM to reduce the overall test time

Uncertain parameter that we take into account is the width of the TAM. A reconfigurable parallel TAM can be there to enable extra bandwidth for test data in case the stack level test architecture allows it in order to avoid non robust solution. Conventional optimization methods have to consider worst-case estimates for all input parameters to ensure feasibility, which can result in conservative and hence expensive solutions. We propose the use of robust optimization for test-architecture design and test scheduling. The goal of this approach is to find a solution that remains close to optimal in

the presence of parameter variations**.** A bus may connect several modular cores spanning on multiple tiers. The cores on the same bus are tested in sequence while the cores on different buses are tested in parallel. Buses can merge and fork to provide the best sharing of the TAMs and to accommodate the flexible wrapper interfacing.

3.1 Problem formulation

The goal in this work is to generate robust test architecture optimization for 3D stacked ICs considering the maximum available TAM width in case of uncertainty in TAM configuration. 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}), and iii) testing time of each core for different TAM widths determine the scheduling order of cores in different configuration with the TAM_{max} such that the overall testing time is optimized.

3.2 Proposed Algorithm:

I have proposed a method, regarding the above mentioned problem. Algorithm is a configuration based robust approach.

3.2.1 Algorithm Description

The proposed algorithm takes n cores of different SOCs and grouped them together in different configuration of TAM width like n, 2n, 4n bit. The proposed algorithm is discussed below.
The algorithm proposed partitioned the maximum TAM width into two parts having a TAM width for each part. Now all the cores of the given SOC are assigned randomly to the two divided part of the TAM width. The cores which fall in the same partition of TAM width share same TAM wire hence those cores are in series. Thus we take the summation of test time of all those cores. Similarly the test time of all other cores grouped in other partition are added as those cores are also sharing the same TAM wire, hence in series. But these two partitions of TAM width are in parallel with each other. Thus we take the maximum test time from the two as final test time for the particular partition. Similarly the maximum test time for all possible partition of TAM width is recorded. Now minimum test time among all the maximum test time is recorded which is taken as the final test time for given TAM width in particular configuration. Similarly other minimum test time or other configuration is taken by going for different configuration gives the test time for given TAM width in any of the configuration.

Algorithm:

Input: set of cores in each SOC, W_{max}->Maximum available TAM width
Output: Total test time of SOC for different configuration of TAM width.
For any of the TAM configuration:
1. Divide the total TAM width (W_{max} into two parts
2. Assign the cores of the SOC to the two different TAM widths one by one until all cores get assigned.
3. Taking the sum of testing time of all those cores assigned to each TAM width separately.
4. Now record the maximum test time among these two divisions.
5. Store this test time as $T_{max.}$
6. Repeat step 1 to 5 all for all possible division of W_{max} and calculate the T_{max} for each division.
7. Select the minimum testing time among all the T_{max} as T_{min} for each configuration.
8. Repeat step 1 to 7 for all different TAM configuration. Record T_{min} for each configuration.
9. Calculate the total test time as (T_{total}) as summation of all T_{min} corresponding to different TAM configuration.
10. END

Example

Assume an SOC having ten cores, namely c1,c2,c3,c4,c5,c6,c7,c8,c9,c10. Maximum available TAM width is 10 and the test time of these cores with respect to different TAM widths are shown in table 3.1.

Table 1: Testing time of different cores for different TAM Width of Example SOC

TAM	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1	428	15292	5058	26602	191874	185794	65686	22427	26351	120188
2	220	7719	2582	13354	95992	93014	32891	11262	13182	60128
3	155	4067	2507	11129	64070	62248	21959	8721	8802	40137
4	116	3275	2507	6782	48106	46741	16493	5680	6597	30132
5	103	2655	2507	5829	38481	37364	13243	4605	5310	24701
6	90	2070	2507	5829	32161	31241	11027	4605	4440	21182
7	77	1774	2507	5829	27613	28199	9695	4605	3798	17663
8	64	1624	2507	5829	24163	23488	8342	4605	3305	15100
9	64	1405	2507	5829	21518	20906	7383	4605	2964	14144
10	63	1257	2507	5829	19757	19034	6717	4605	2820	14144

The algorithm will find the test time for different configuration of TAM Width. Here for solving the example the value of TAM width i.e. n is taken as 2 and similarly will take the value of 2n and 4n as 4 and 8. According to the algorithm all possible partition of the TAM width will be made for 2 it will be (1, 1). Now the cores of the given example are assigned alternatively to the two partitions. After assignment of all the cores the two portions contain the cores as (c1, c3, c5, c7,) and in another as cores (c2, c4, c6, c8, c10). Cores falling in same partition their testing time will be added. And as these two partitions are in parallel so maximum of the testing time is taken it is found to be (419642clock cycles). Now the algorithm will go for TAM width 4. It will make all possible partitions that are (1,3) (2,2) (3,1). Now similarly the cores will be assigned and algorithm gets different test time

corresponding to partitions like (312893 clock cycles, 210084 clock cycles, 370303 clock cycles). Algorithm fetches the minimum test time among the different test time and it is found to be 210084 clock cycles in partition (2, 2). Therefore this would consider as the overall testing time of given SOC for given configuration. Similarly algorithm will go for 4n i.e. TAM width is 8. All possible partitions for it are (1,7) (1,7) (2,6) (3,5) (4,4) (5,3) (6,2) (7,1). And testing time corresponding to different partitions are found to be (291514, 157964,119214, 120652). The overall testing time for given TAM configuration is found to be (119214 clock cycles) in partition (3, 5). Therefore this would consider as the overall testing time of given SOC for 4n configuration. Thus total testing time corresponding to all three configurations will be 648940) clock cycles

3.3 Experimental Result:

The proposed algorithm is implemented using C programming language with Core i7 processor and 4 GB RAM in a Linux operating system. To demonstrate the effectiveness of the proposed algorithm, SOCs from ITC'02 benchmarks namely d695, p22810, p93791, p34392 and f2126 are used. For different configuration of TAM width like n, 2n, 4n testing time is calculated where value of n is taken up to 16.

Table 2: Experimental results of SOC d695 for different TAM configuration

d695				
TAM width	Testing time (T_n) (clock cycles)	Testing time (T_{2n}) (clock cycles)	Testing time (T_{4n}) (clock cycles)	Total testing time T_{total} (clock cycles)
n=4	189048	86246	46817	322111
n=6	114295	71616	35046	220957
n=8	88908	48276	26293	163477
n=10	78965	39076	27805	145846
n=12	59362	33458	22607	115427
n=14	48391	31691	20127	100211
n=16	48276	28328	19062	95666

Table 3: Experimental results of SOC p22180 for different TAM configuration

p22180				
TAM width	Testing time (T_n) (clock cycles)	Testing time (T_{2n}) (clock cycles)	Testing time (T_{4n}) (clock cycles)	Total testing time T_{total} (clock cycles)
n=4	1774586	993415	516367	3284368
n=6	1382298	650510	396657	2429465
n=8	1348452	516146	352718	2218316
n=10	728931	412682	317241	1458854
n=12	728931	386732	302979	1418642
n=14	684479	350523	270955	1305957
n=16	548235	344799	285912	1178946

Table 4: Experimental results of SOC p34392 for Different TAM configuration

d34392				
TAM width	Testing time (T_n) (clockcycles)	Testing time (T_{2n}) (clock cycles)	Testing time (T_{4n}) (clockcycles)	Total testing time T_{total} (clock cycles)
n=4	6537099	2199805	1262572	9999476
n=6	2537777	1474307	910390	4922474
n=8	2332562	1206190	765409	4304161
n=10	1681473	1042763	750267	3474503
n=12	1415388	876968	742718	3035074
n=14	1275246	898053	740812	2914111
n=16	1102140	794277	740583	2637000

Table 5: Experimental results of SOC p93791 for Different TAM configuration

P93791				
TAM width	Testing time (T_n) (clockcycles)	Testing time (T_{2n}) (clockcycles)	Testing time (T_{4n}) clockcycles)	Total testing Time T_{total} (clock cycles)
n=4	9088651	3996631	1802235	14887517
n=6	5318115	2525760	1327639	9171514
n=8	3812979	2341483	1028470	7182932
n=10	2841199	1460836	743493	505528
n=12	2816887	1279529	707101	4803517
n=14	2187623	1063003	620376	3871002
n=16	1857596	1026018	513031	3396645

Table 5: Experimental results of SOC f2126 for Different TAM configuration

f2126				
TAM width	Testing time (T_n) (clockcycles)	Testing time (T_{2n}) (clockcycles)	Testing time (T_{4n}) clockcycles)	Total testing Time T_{total} (clock cycles)
n=4	9088651	3996631	1802235	14887517
n=6	5318115	2525760	1327639	9171514
n=8	3812979	2341483	1028470	7182932
n=10	2841199	1460836	743493	505528
n=12	2816887	1279529	707101	4803517
n=14	2187623	1063003	620376	3871002
n=16	1857596	1026018	513031	3396645

4. Experimental Result Discussion:

Table 2 to Table 6 shows the test time of different SOC for different TAM width. Tables describes different test time of SOCs d695, p22810,p34392, p93791 and f2126 in different configuration of TAM for different TAM width $T_{n,}$,$T_{2n,}$$T_{4n}$ represent the testing time for n,2n,4n bit TAM. width. T_n corresponds to testing time of given SOC for n bit TAM width, $T_{2n,}$ corresponds to testing time of given SOC for 2n bit TAM width and T_{4n} corresponds to testing time of given SOC for 4n bit TAM width .For each configuration corresponding to given TAM width the test time is calculated. It is found that the test time corresponding to n bit of TAM is greater than the 2n and 4n bit. Algorithm takes the TAM width as input and calculates the test time for different configuration of TAM width like $T_{n,}$,T_{2n} T_{4n} in clock cycle. TAM width will be taken as input which will give the value of n corresponding to which test time will be calculated. Then test time for 2n bit TAM width and 4n bit TAM width will be calculated. Finally after adding the test time for different configuration T_{total} is calculated. It is found that with increasing the TAM width T_{total} goes on decreasing.Proposed methods find the test time in such a way that whenever the architecture is going for the test for any of the configuration. It is giving optimal result. Different results are tabulated above.

IV. CONCLUSION

In this work a method has been proposed for robust test architecture optimization of 3D stacked ICs in the presence of input parameter variation for minimizing the overall test time of a 3D SOC considering the maximum available TAM width. The goal of this approach is to find a solution that remains close to optimal in the presence of input parameter variation. Previous optimization solutions have been targeted at specific values of the inputs parameters. However, such “point solution” are not effective in the presence of uncertainties in input parameters, such uncertainty are likely to arise in practical scenarios. The robust optimization problem for five SOCs from the ITC’02 SOC Test Benchmarks have been solved and it has been observed that robust solutions reduce test time more effectively and obtained result for all the SOCS are better and promising. In our experiments only considered uncertain input parameter is the variations in TAM configuration. TAM distribution and assignment of them to cores for testing is performed in a dynamic way therefore the algorithms can perform in much more adjustable manner. Traditional optimization methods have to consider worst-case estimates for all input parameters to ensure Workability, which can result in low and hence costly solutions. It has been observed that neglecting variation in input parameter will result in ineffective single point solutions with high test time. The proposed method goes well with the complexities of the die.

V. FUTURE SCOPE

Uncertainty is a severe problem of 3D stacked ICs. But in this work many uncertain parameters are not considered. The designed framework can easily be extended with additional constraint and more uncertain parameters. In future, we will definitely work on those to complete this project

REFERENCE

[1] S.Koranne.A Novel Reconfigurable Wrapper for Testing of Embedded core-Based SOCs Scheduling. Journal of Electronic Testing, pages 415-434,2002.

[2] Li Jiang et al. Layout-Driven Test-Architecture Design and Optimization For 3D SOCs and pre-bond Test-pin-count Constraint.pages191-196, November 2009.

[3] IEEE 3D-Test P1838 Working Group.

[4] B.Nona et al. Test-Architecture Optimization and Test Scheduling for TSV based stacked IC 30(11):1705-1718, November 2011.

[5] J M. Mulvey, R. J. Vanderbei, and S. A. Zenios Robust optimization of large-Scale Systems Operations Research, 264- 281, 1995.

[6] J.H. Lau and T.G. Yue. Thermal Management of 3D IC Integration with TSV (Through Sili-con via).Electronic Components and Technology (ECTC).pages 635-604,2009.

[7] S.Deutsch et al. Automation of D-DFT Insertion, pages 395- 400, 2011.

[8] E.J. Marinissen, V. Iyengar, and K. Chakraborty.A set of Benchmarks for Modular Testing Of SOCs.In ITC, 602-633, May 2001.

[9] E.J.Marinissen S.K.Goel and M.Lousberg wrapper Design for bedded core Test.pages 911- 920, 2000.

[10] P.J.Van Laarhoven and E.H.L.arts. Simulated Annealing. Theory and Applications, volume 37. Sringer, 1987.

[11] G Gils, J.Wang, A.Seghal, K.J. Balakrishnan and j Wingfield.Test Access Mechanism for Multiple Identical Cores.In ITC, pages 1-10, 2009.

[12] A.Dembla, Y.Zhang, and M.S.Bakir. Fine Pitch TSV Integration in Silicon Micropin-Fin Heat sink for 3D ICs, In proceeding IEEE INTERNATIONAL INTERCONNECT TECHNOLOGY CONFERENCE (IITC), PAGES 1-2012.

[13] K.Chakraborty.Test scheduling for core-Based Systems Using Mixed Integer Linear Programming.TCAD, pages 1163-1174, 2000.

[14] W. Zou et al. SOC Test Scheduling using Simulated Annealing. In Proceeding IEEE VLSI Symposium on Circuits (VLSI), pages 325-330, 2003.

[15] S.Deutsch et al. Automation of 3D-DfT Insertion.In ATS,pages 395-400,2011.

[16] 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 International Conference on Computer-Aided Design, page. 191-196, 2009