Supratim Subhra Das, Ria Bose / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 4, Jul-Aug 2013, pp. 347-356

A Genetic Algorithm Based Approach For Optimization Of Test Time And TAM Length For 3D SoC Considering Pre-Bond Test Under The Constraint On The Number Of TSVs

Supratim Subhra Das[1], Ria Bose[2]
[1,2] Department of Electronics & Communication Engg,, Mallabhum Instiute Of Technology,Bishnupur,Bankura-722122,W.B

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. Test-access mechanisms (TAMs) and test wrappers (e.g., the IEEE Standard 1500 wrapper) facilitate the modular testing of embedded cores in a core-based system-on-chip (SoC). Such a modular testing approach can also be used for emerging three-dimensional integrated circuits based on through-silicon vias (TSVs).This paper presents a Genetic algorithm(GA) based solution to Co-optimize test scheduling and TAM length for 3D SoC. A locally optimal best-fit heuristic based bin packing algorithm has been used to determine placement of cores minimizing the overall routing cost. Experimental result on ITC'02 benchmark SoCs shows that the proposed method provides few better test time results compared to earlier work.

Keywords - Genetic Algorithm, Rectangle Packing, System-on-Chip, Test Access Mechanism, Through Silicon Via

I. INTRODUCTION

Modern semiconductor process technologies enable the manufacturing of a complete system on one single die, the so-called system chip. Such system chips typically are very large ICs, consisting of millions of transistors, and contain a variety of hardware modules. In order to design these large and complex system chips in a timely manner and leverage from external design expertise, increasingly reusable cores are utilized. *Cores* are pre-designed and pre-verified design modules. Examples of cores are CPUs, DSPs, media coprocessors, communication modules, memories, and mixed-signal modules. System-on-chip (SoC) designs comprised of a number of embedded cores are widespread in today's integrated circuits. Embedded core-based design is likely to be equally popular for three-dimensional integrated circuits (3D ICs), the manufacture of which has become feasible in recent years. 3D integration offers a number of advantages over traditional two-dimensional (2D) technologies, such as the reduction in the average interconnect length, higher performance, lower interconnect power consumption, and smaller IC footprint. Therefore different test challenges of 3D SoC have become very important topic for today's research field. Optimization of Test Access mechanism lengh is also important because it has a direct impact on reducing routing cost of 3-dimensional System-On-Chip.

II. ISSUES NEED TO BE HANDLED FOR TESTING OF 3D SOC

To test a SoC, we need to test all cores of the chip individually or combinedly. For that purpose we primarily need a test wrapper design for each individual core that can successfully be accessed from outside the chip through a TAM. Wrapper provides an interface between TAM and the core. These wrappers are basically selected on basis of the terminal types of the core. Test wrapper --- operates in 3 modes.

- Functional
- Core internal test mode
- Core external test mode

TAM serves as ' test high way' in the sense that it bridges the physical distance between source and core i.e. TAM transports data between SoC pin and core wrapper. We also need a test source where real time stimulus generation takes places and a test sink where real time response evaluation is carried out. Source and Sink can either be implemented off-chip (Automatic Test Equipment or ATE) or on chip (Built in self Test or BIST).

Some pins in SoC are assigned for TAM --- those pins are termed as TAM wires. If total TAM width for a particular Core exceeds the number of TAM wires then TAM width is partitioned optimally among these wires in such a manner so that test time can be minimized. Therefore TAM width partitioning is another issue to be optimized. In 3D SoC, many cores in same or different level may be connected through a single TAM wire. Interconnection length between cores in same or different layer, in other words, length of a single

TAM wire or TAM length should also be minimized in order to get an optimized Fitness values. In brief, the issues that need to be handled for testing of 3D SoC are highlighted here under

- SoC test access and test scheduling
- Test access optimization
- Test Wrapper Optimization [1]
- Test wrapper And TAM Co-optimization -- The authors in [2], [3], [4] have proposed different efficient approaches for Test wrapper And TAM Co-optimization

III. CASE STUDY OF PREBOND AND POSTBOND TEST

In [5], Li Jiang, Lin Huang and Qiang Xu formulated test architecture design and optimization problem for 3D SoCs by considering both post-bond test and pre-bond wafer-level tests.They propose efficient and effective heuristics to optimize the testing time and the routing cost associated with the test access mechanisms,based on Genetic Algorithm.Here is an example that briefly describes their work

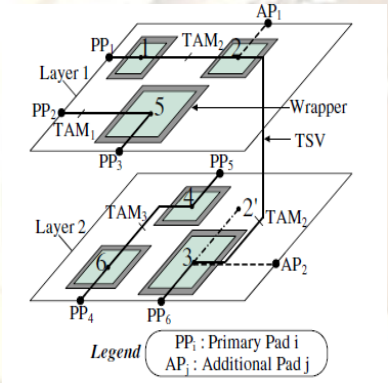

Figure 1. An Example of 3D SoC Test Architecture

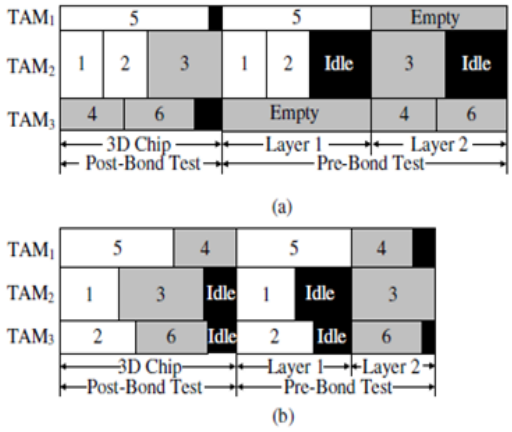

Figure 2. The impact of Pre-bond Test

There are totally three TAMs for this example SoC: TAM1 for core 5, TAM2 for cores 1, 2 and 3, and TAM3 for core 4 and core 6. In particular, TAM2 traverses two layers in this example.When pre-bond tests at the wafer-level are required, 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.1,it contains three parts: the pre-bond testing time for layer1, the pre-bond testing time for layer 2, and the post-bond testing time for the entire chip, represented as three bins in Fig.2(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 prebond tests (see TAM2). 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.In our work, we have considered pre-bond test-pin-count constraint during optimization, as same as that followed in [6].

IV. PROBLEM FORMULATION

We are given with a 3D SoC having H number of cores, L number of layers, T is the total TAM width assigned to it. An optimized fixed core wrapper design is given for each core in all the layers of SoC. We need to optimize

1) Test time

We need to perform TAM width partitioning, core assignment to each TAM width and layer assignment for each core.

2) TAM length

We first need to place the cores (connected to a single TAM wire) in one layer (to which TAM wire is connected) using some placement algorithm. Their placement will be based on their areas (TAM width * Testing Time) and they can be placed using Best Fit placement heuristic approach or using rectangular bin pack approach. After placement, using some efficient tools we need to find out the position or co-ordinate of each core connected to a single TAM wire. If we consider that each core has rectangular shape then position of this core actually corresponds to mid point of that rectangular area i.e. the point of intersection of the line passing through middle of width and the line passing through middle of the length. Then we need to sum up the interconnect lengths of the cores and thus we can get the TAM length. Similar formulation should be done for each TAM wire assignments. The maximum of all the TAM lengths obtained is the TAM length of the SoC.

V. TEST COST MODEL

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

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

where, C_{Test_Time} is the total testing time for both prebond tests and post-bond test, while C_{Wire_Length} is the total TAM wire length. $\boldsymbol{\alpha}$ is a weighting factor designated by users. For the example test architecture and the associated test schedule shown in Fig. 2(a), C_{Test_Time} is the sum of three terms: ($T1$+$T2$+$T3$) for post-bond entire chip, $T5$ for pre-bond layer 1 and ($T4$ +$T6$) for pre-bond layer 2, where Ti is the testing time of core i.

Here in our work, we will concentrate on the GA based approach to optimize Test time and TAM length. The computation of wire length (or TAM length), however, is non-trivial. We will 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 will calculate Wire Length as follows. Wire Length for a TAM that involves several layers contains two parts: the intra-layer wire length and the interlayer 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 have to use an algorithm to compute its wire length. As for the inter-layer wire lengths, they are calculated as the Manhattan distance between the corner cores in different layers, e.g., for TAM2 in Fig.1, 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.

VI. CO-ORDINATE SYSTEM

We will use only the first quadrant of an orthogonal coordinate system. The fact that the SOC is in the first quadrant means that all coordinates will be non-negative numbers, which simplifies our calculations. The unit of the coordinates is not specified, but should be consistent for all co-ordinates belonging to the same SoC.

VII. SoC LAYOUT POSITION

The SoC layout is assumed to be a rectangle. Total scan chain length is taken as area of a module or core. 1/3rd of area is taken as width of the core and 2/3rd of area is taken as height of the core. Sum of widths of all the cores of 3D SoC is taken as total width of the floor (in which cores are to be placed) and sum of heights of all the cores of SoC is taken as total height of the floor.

VIII. CORE LAYOUT POSITION

Each Core is assumed to be a rectangle.If co-ordinate or position of a core is determined by (x,y), then x = half of length of its width and y = half of length of its height.

IX. CORE PLACEMENT PROCEDURE

The authors in [3] proposed a approach for wrapper/TAM co-optimization based on generalized rectangle packing also referred to as two-dimensional packing. Based on this approach we have proposed an heuristic based placement algorithm which helps to place the connected cores in same layer so that number of TSVs required to connect them can be optimized.Our algorithm is given below

- Set initial value of l is 1.
- While (l not equal to L)
- Set initial TAM bus number (b) = 1.
- While (b not equal to B)
- Until all the cores connected to TAM bus number b in layer number l is selected.
- Select one core from array of cores assigned to TAM bus number b in layer number l (placing the first core at the upper left most corner of the floor).
- For each such core
- Check whether the space where it is placed is empty or not. If space is not empty, then search for near most adjacent empty space and move the core to that place for placement.
- Check whether aspect ratio of core is properly fitted to the space or not .If not, then swap its width and height and check again. If still not, then find near most larger empty space and move the core to that place for placement.
- Check whether any overlapping is occurred or not.
- Find the co-ordinate or position of cores connected to TAM bus number b in layer number l.
- Calculate TAM length for all W_b ($1 \le b \le B$) for all l ($1 \le l \le L$).

Finally addition of TAM lengths for all W_b ($1 \le b \le B$) is performed to find the overall TAM length or wire length of 3D SoC. The following tree structure is an example which shows the connection between different cores of 3D SOC.

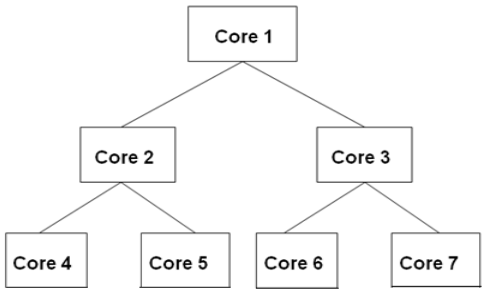

Figure 3: Tree Structure

Here is the implementation of our proposed placement procedure

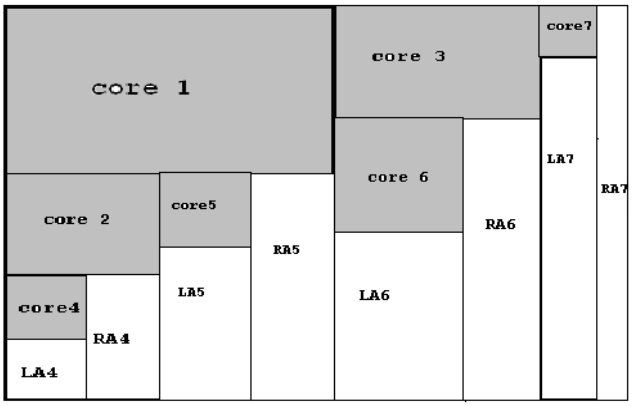

Figure 4: Placement of cores

X. PROPOSED ALGORITHM

Genetic Algorithms (GA) are Stochastic optimization search algorithms based on the mechanics of natural selection and natural genetics. The genetic formulation of this problem should involve the careful and efficient choice of the following.

A proper encoding of the solutions to form chromosomes.

1. To decide upon a crossover operator.
2. To identify a proper mutation operator
3. A cost function measuring the fitness of the chromosome in a population.

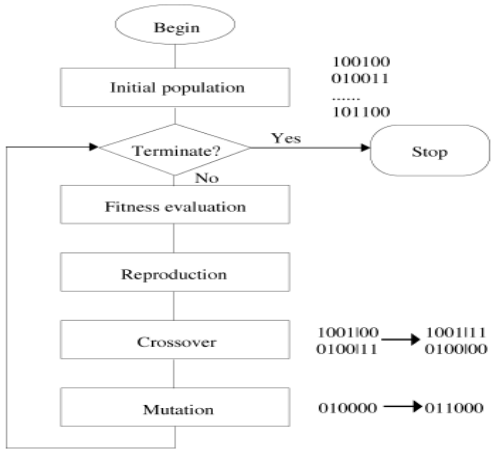

Figure 5: Genetic Algorithm Flow

XI. SOLUTION REPRESENTATION

A chromosome for this problem conceptually consists of three parts. First part named as *partition part* is the number of TAM partitions. Since the total number of partitions N can be encoded with a binary string of (logN/log2) bits , this partition is an array of size (logN/log2).

The 2nd part named as *distribution part* gives the width of each partition of the TAMs.This part is an array of real numbers between 0 and 1, where jth entry of this array multiplied by the TAM width (V) represents the width of the bus j. the array size is equal to the decimal value of the first part. However to keep the chromosome length fixed, 8 entries are used here and only few of them are actually used (depending upon the decimal value in partition part).

The 3rd part named as *assignment part* has 2 subparts. The first subpart gives the assignment of cores to the randomly selected TAM widths available in the 2nd part. Array size of this part is the number of cores available and array index denotes the core numbers.The 2nd subpart gives the assignment of cores to the randomly selected layer numbers available. Array size of this part is the number of cores available and array index denotes the core numbers.

0	1	1	0.3750000	0.0625000	0.562500	x	x	x	x	x	2	2	12	12	2	3	4	4	4	2

Figure 6: An Example of our chromosome structure

In this example, the number of bits in partition part is 3 and the number of partitions is 3 (011). So, in the distribution part of TAM first three entries will be considered and the respective TAM widths are (0.375 * V), (0.0625 * V) and (0.5625 * V), where V is the Total TAM Width available in this context. Now in first sub part of assignment part TAM Width is assigned to core 1, TAM Width is assigned to core 2 and so on. Similarly in 2nd subpart of assignment part layer number is assigned to core 1, layer number is assigned to core 2 and so on.

XII. GENETIC OPERATORS

Two genetic operators ---- crossover and mutation ---- are generally used to evolve new generation. A brief description about these two operators are given below

1) **Crossover**

GA formulation is biased towards selecting the chromosomes with better testing time value or fitness value to participate in crossover. For this purpose, the whole population is sorted according to their fitness values. A certain percentage of population with better fitness value is defined to be the 'Best Class'. To select a

chromosome participating in crossover a random number between 0 and (total number of populations – Best Class populations) is generated. Then the chromosome corresponding to this randomly generated population number is selected. After selecting two such chromosomes to participate in crossover, a single point crossover is applied on each of the *partition part*, *distribution part* and *assignment part of the chromosome.*

2) Mutation

The mutation operator brings more effective variations into the chromosomes introducing newer search options. To select a chromosome participating in mutation, a random number between 0 and (total number of populations – Best Class populations) is generated. Then the chromosome corresponding to this randomly generated population number is selected. Then we select a random point in each of the four fields of the chromosome and change its value. For the first field we complement a randomly selected bit among the (logN/log2) no of bits where N denotes the number of partitions. For the 2nd field or the *distribution part* we replace with randomly generated values in the range 0 and 1. For this part, normalization is carried out to ensure the unity sum requirement.

XIII. EXPERIMENTAL RESULTS

In this section, we present experimental results for four bench-mark SoCs: d695 (an academic benchmark from Duke University), p22810, p34392, and p93791 (industrial SOCs from Philips). These four SoCs are part of the ITC'02 SoC test benchmarking initia-tive [7]. The number (e.g.,93791) in each SoC name is a measure of its test complexity. This naming convention is described in [4].

Example SoCs : SoC d695 consists of ISCAS benchmark circuits [4]. SoC p22810 contains 6 memory cores and 22 scan-testable logic cores. SoC p34392 contains 15 memory cores and 4 scan-testable logic cores. SoC p93791 contains 18 memory cores and 14 scan-testable logic cores. SoC p93791 contains 32 cores [4]. Of these 32 cores, 18 are memory cores and 14 are scan-testable logic cores. A summary of the 32 cores is presented in Table [8].

Our proposed algorithm has been implemented in C language and results we obtained are as follows

TABLE-1: Results of Test Time For SoC d695

SoC	TSV Limit	W	B	TAM	Test Time	Min. Test Time (Ours)	In method[9]	Improvement in (%)
d695	80	16	2	1,15	43125	42525	44027	3.41
			3	9,4,3	42525			
			4	5,8,1,2	43296			
			5	5,3,2,1,5	44624			
		24	2	5,19	28944	28289	32542	13.07
			3	13,9,2	28289			
			4	7,12,1,4	29045			
			5	3,9,6,1,5	31241			
		32	2	6,26	23297	20906	24133	13.37
			3	15,12,5	20906			
			4	12,15,3,2	21570			
			5	13,8,8,1,2	23488			
		40	2	14,26	19938	17294	19718	12.29
			3	4,21,15	17294			
			4	19,16,3,2	17610			
			5	12,19,5,2,2	18405			
		48	2	21,27	18906	14568	18837	22.66
			3	20,20,8	14568			
			4	20,2,5,21	14688			
			5	19,8,13,6,2	15557			
		56	2	19,37	18439	12917	15237	15.23
			3	21,30,5	13428			
			4	9,4,16,27	12917			
			5	17,27,6,4,2	13428			
		64	2	26,38	18223	12290	13816	11.68
			3	25,12,27	13228			
			4	15,16,26,7	12791			
			5	8,16,17,10,13	12290			

TABLE – 2: Results of Test Time For SoC p22810

SoC	TSV Limit	W	B	TAM	Test Time	Min. Test Time (Ours)	In method[9]	Improvement in (%)
p22810	80	16	2	10,6	459053	441273	469487	6.01
			3	8,3,5	441273			
			4	5,5,2,4	45341			
			5	3,5,2,42	475587			
		24	2	8,16	335385	308960	376819	18.01
			3	10,6,8	308960			
			4	9,8,4,3	312001			
			5	11,5,4,2,2	333683			
		32	2	21,11	305082	240031	270592	11.29
			3	9,7,16	253300			
			4	15,7,5,5	240031			
			5	12,6,3,5,6	242407			
		40	2	31,9	277167	200548	240542	16.71
			3	17,10,13	219520			
			4	17,7,7,9	200348			
			5	9,16,6,3,6	204685			

TABLE 3: Results of TAM Length For SoC p22810

SoC	TSV Limit	L	W	B	TAM	TAM Length	Min. TAM Length (Ours)	In [6]	Improvement in (%)
p22810	80	2	16	2	14,2	2182	2182	12648	82.75
				3	10,3,3	2987			
				4	2,6,1,7	3528			
				5	1,9,3,1,2	4523			
			24	2	22,2	2182	2182	14567	85.02
				3	15,5,4	2987			
				4	3,10,1,10	3528			
				5	2,14,4,1,3	4523			
			32	2	29,3	2182	2182	10898	79.98
				3	21,7,4	2987			
				4	4,13,1,14	3528			
				5	3,19,6,1,3	4523			
			40	2	37,3	2182	2182	8240	73.52
				3	27,8,5	2987			
				4	5,17,1,17	3528			
				5	3,25,8,1,3	4523			
			48	2	26,22	2508	2508	19189	86.93
				3	12,24,12	3209			
				4	41,1,3,3	3727			
				5	42,2,1,1,2	4323			
			56	2	38,18	2382	2382	10754	77.85
				3	23,8,25	3039			
				4	28,20,2,6	3608			
				5	48,1,2,2,3	4284			
			64	2	11,53	2278	2278	11320	79.88
				3	29,18,17	3182			
				4	9,52,1,2	3651			
				5	51,8,1,2,2	4055			

It can be observed from TABLE 1,2 and 5 that improvement in test time reduction has been occurred from minimum 0.29% to maximum 22.66% for SoCs d695,p22810 and p34392. This reduction has been possible in this case due to

alignment of cores in several layers for which multiple core access is possible through TSVs which obviously reduces test time.

Later in TABLE 7, it is also shown that for SoC p93791and TSV limit 80, the improvement in test time occurs from minimum 0.11% to maximum 17.52%.

TABLE 4: Results of TAM Length For SoC p93791, TSV=80, L=2

W	B	TAM	TAM Length	Min. TAM Length(Ours)	In [6]	Improvement in (%)
16	2	14,2	3414	3414	18803	81.84
	3	11,1,14	4228			
	4	5,6,1,4	5527			
	5	5,4,1,4,2	6765			
24	2	21,3	3414	3414	23727	85.61
	3	1,2,21	4228			
	4	9,9,1,5	5527			
	5	8,7,1,5,3	6765			
32	2	29,3	3414	3414	20953	83.71
	3	1,2,29	4228			
	4	12,12,1,7	5527			
	5	11,10,1,7,3	6765			
40	2	37,3	3414	3414	24709	86.18
	3	1,3,36	4228			
	4	15,16,1,8	5527			
	5	14,12,2,9,3	6765			
48	2	8,40	4369	4369	22749	80.79
	3	42,2,4	4743			
	4	39,6,1,2	5577			
	5	23,18,3,2,2	5841			
56	2	20,36	4195	4195	33205	87.37
	3	33,2,21	4658			
	4	23,10,4,19	5049			
	5	6,32,6,7,5	6042			
64	2	18,46	4025	4025	26974	85.08
	3	10,9,4,5	4459			
	4	36,16,8,4	5324			
	5	24,13,14,4,9	5860			

TABLE 5: Results of Test Time For SoC p34392

SoC	TSV Limit	W	B	TAM	Test Time	In method[10]	Improvement in (%)
P34392	80	16	2	8,8	1044538	1134919	7.96
			3	6,8,2	979824	999543	1.97
			4	5,6,2,3	990846	1288341	23.09
		24	2	15,9	797963	913550	12.65
			3	10,6,8	689444	762841	9.62
			4	7,8,1,8	675098	798449	15.45
		32	2	20,12	722709	843301	14.29
			3	16,10,6	570047	665445	14.33
			4	3,2,16,11	582295	684524	14.93
		40	2	17,23	711289	752782	5.51
			3	17,11,12	544836	552231	1.34
			4	11,17,2,10	544579	584301	6.79
			3	17,18,13	544579	546152	0.29
			4	11,28,3,6	544579	544579	0
			4	15,19,15,7	544579	544579	0

TABLE 6: Results of TAM Length For SoC t512505, L=2, TSV=80

W	B	TAM	TAM Length	Min. TAM Length(Ours)	In [5] SA(α=0.6)	Improvement in (%)
16	2	6,10	2147	2147	7272	70.47
	3	9,1,6	2749			
	4	9,2,1,4	3628			
	5	1,3,1,1,10	4524			
24	2	9,15	2147	2147	4714	54.45
	3	14,1,9	2749			
	4	15,3,1,5	3628			
	5	2,5,2,1,14	4524			
32	2	12,20	2147	2147	9933	78.38
	3	20,1,11	2749			
	4	21,4,1,6	3628			
	5	2,6,2,1,21	4524			
40	2	15,25	2147	2147	7354	70.80
	3	25,1,14	2749			
	4	27,5,1,7	3628			
	5	3,8,3,2,24	4524			
48	2	32,16	2426	2426	7440	67.39
	3	23,21,4	2859			
	4	14,22,1,11	3666			
	5	36,5,4,1,2	4219			
56	2	20,26	2322	2322	7470	68.92
	3	44,9,3	2969			
	4	36,14,2,4	3667			
	5	48,1,1,3,3	4476			
64	2	20,44	2372	2372	7374	67.83
	3	21,27,16	3237			
	4	57,1,2,4	3496			
	5	1,58,1,1,3	4255			

TABLE 7: Results of Test Time For SoC p93791

W	B	TAM	Test Time(Ours)	In method[10]	Improvement in (%)
16	2	7,9	1783973	1800413	0.91
	4	2,3,3,8	1810564	2179527	16.93
	5	7,5,1,1,2	1914954	2030868	5.71
24	2	8,16	1203730	1259711	4.44
	4	6,2,10,6	1238525	1337682	7.41
32	2	23,9	893402	894463	0.11
	4	5,7,12,8	932827	944807	1.27
	5	6,9,6,5,6	955698	1008684	5.25
40	2	23,17	728421	778296	6.41
	3	9,24,7	732163	817805	10.47
	4	16,6,1,17	764655	890145	14.09
48	2	25,23	667654	758806	12.01
	3	12,12,24	608515	722042	15.72
	4	6,23,10,9	619629	713347	13.14
	5	16,13,5,6,8	663601	727078	8.84
56	2	29,27	612732	662686	7.54
	3	17,17,22	562904	635095	11.37
	4	11,24,8,13	548030	66447	17.52
	5	21,16,5,7,7	599316	680635	11.95
64	2	20,44	526608	630365	16.46
	3	16,25,23	495148	572342	13.49
	4	17,13,16,18	515751	604553	14.69
	5	17,25,1,13,8	518780	621235	16.49

TABLE 8: Results of TAM Length For SoC p34392, L=2, TSV=80

W	B	TAM	TAM Length	Min. TAM Length(Ours)	In [6]	Improvement in (%)
16	2	14,2	2266	2266	7760	70.79
	3	10,3,3	2964			
	4	8,4,2,2	3338			
	5	9,3,1,1,2	3986			
24	2	22,2	2266	2266	17422	86.99
	3	16,4,4	2964			
	4	13,6,3,2	3338			
	5	15,5,1,1,2	3986			
32	2	30,2	2266	2266	13949	83.75
	3	22,5,5	2964			
	4	17,8,5,2	3338			
	5	22,6,1,1,2	3986			
40	2	38,2	2266	2266	24439	90.73
	3	28,6,6	2964			
	4	22,10,5,3	3338			
	5	28,8,1,1,2	3986			
48	2	20,18	2517	2517	23705	89.38
	3	18,12,18	2838			
	4	1,7,23,17	3391			
	5	27,15,2,2,2	3683			
56	2	38,18	2606	2606	23423	88.87
	3	38,2,16	2875			
	4	30,1,1,24	3236			
	5	12,13,7,12,12	3483			
64	2	38,26	2519	2519	23145	89.12
	3	2,58,4	2596			
	4	49,6,4,5	3127			
	5	51,2,8,1,2	3283			

In TABLE 3,4,6,8 and 9 our results have been compared to the results obtained in [6] where SA (Simulated Annealing) technique was used to achieve routing cost for 3D SoCs.

Authors in [6] used flexible pre-bond test architecture in SA scheme to save routing cost where in our case we have considered rectangle bin packing heuristic based algorithm to reduce TAM length. It is definitely observed that in our technique a reduction in TAM length is possible from minimum 41.33% to maximum 90.73% for SoCs p22810, p93791, t512505 and p34392.

TABLE 9: Results of TAM Length For SoC t512505, L=2, TSV=80

W	B	TAM	TAM Length	Min. TAM Length(Ours)	In [5] SA(α=0.4)	Improvement in (%)
16	2	6,10	2147	2147	3957	45.75
	3	9,1,6	2749			
	4	9,2,1,4	3628			
	5	1,3,1,1,10	4524			
24	2	9,15	2147	2147	4144	48.19
	3	14,1,9	2749			
	4	15,3,1,5	3628			
	5	2,5,2,1,14	4524			
32	2	12,20	2147	2147	3986	46.14
	3	20,1,11	2749			
	4	21,4,1,6	3628			
	5	2,6,2,1,21	4524			
40	2	15,25	2147	2147	4029	46.71
	3	25,1,14	2749			
	4	27,5,1,7	3628			
	5	3,8,3,2,24	4524			
48	2	32,16	2426	2426	4175	41.89
	3	23,21,4	2859			
	4	14,22,1,11	3666			
	5	36,5,4,1,2	4219			
56	2	20,26	2322	2322	4102	43.39
	3	44,9,3	2968			
	4	36,14,2,4	3667			
	5	48,1,1,3,3	4475			
64	2	20,44	2372	2372	4043	41.33
	3	21,27,16	3236			
	4	57,1,2,4	3496			
	5	1,58,1,1,3	4255			

II. CONCLUSION

We have presented a GA based technique that lead to an efficient procedure for TAM width allocation and test scheduling. We have also presented a new algorithm based on generalized rectangle packing heuristic approach in order to find an optimized solution for placement of cores which is an NP hard problem. This represents

several order magnitude improvement over the methods presented in earlier work.

XIV. ACKNOWLEDGEMENTS

The idea of using rectangle packing based heuristic algorithm and GA came from Dr. Chandan Giri, Asst. Prof. of BESUS, for which we gratefully acknowledge him. We thank him also for his help with providing data for bench-mark SoCs.

REFERENCES

[1] Brandon Noia, Krishnendu Chakrabarty and Yuan Xie, "*Test-Wrapper Optimization for Embedded Cores in TSV-Based Three-Dimensional SOCs"* 978-1-4244-5028-2/09/$25.00 ©2009 IEEE.

[2] Vikram Iyengary, Krishnendu Chakrabarty and Erik Jan Marinissen "*On Using Rectangle Packing for SOC Wrapper/TAM Co-Optimization"* Proceedings of the 20th IEEE VLSI Test Symposium (VTS'02) 1093-0167/02 $17.00 © 2002 IEEE.

[3] V. Iyengar, K. Chakrabarty, and E. J. Marinissen "*Efficient wrap-per/TAM co-optimization for large SOCs".* Proc. Design Automation and Test in Europe (DATE) Conf., 2002, in press.

[4] V. Iyengar, K. Chakrabarty and E. J. Marinissen "*Test wrapper and test access mechanism co-optimization for system-on-chip".* J. Electronic Testing: Theory and Applications, vol. 18, pp. 211–228, March 2002.

[5] L. Jiang, L. Huang and Q. Xu, *''Test Architecture Design and Optimization for Three-Dimensional SOCs'',* Proc. IEEE/ACM Design, Automation and Test in Europe Conf.(DATE 09), IEEE CS Press, 2009, pp. 220-225.

[6] 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",* ICCAD'09, November 2–5, 2009, San Jose, California, USA.

[7] E.J. Marinissen, V. Iyengar and K. Chakrabarty. ITC 2002 SOC bench-marking initiative. http://www.extra.research.philips.com/itc02socbenchm.

[8] P. Varma and S. Bhatia,"*A structured test re-use methodology for core-based system chips.*" Proc. Int. Test Conf., pp. 294–302, 1998.

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

[10] X. Wu et al., "*Test-Access Mechanism Optimization for Core-Based Three-Dimensional SOCs*." Proc. IEEE Int'l

[11] Conf. Computer Design, IEEE CS Press, 2008,pp.212-218.