2015 IEEE 24th Asian Test Symposium

Test Infrastructure Development and Test Scheduling of 3D-Stacked ICs Under Resource and Power Constraints

Rajit Karmakar, Aditya Agarwal and Santanu Chattopadhyay
Dept. of Electronics & Electrical Comm. Engineering
Indian Institute of Technology Kharagpur, India, Kharagpur, 721302
Email: {rajit, adityaagarwal, santanu}@ece.iitkgp.ernet.in

***Abstract*—This paper presents a test infrastructure development and test scheduling strategy for 3D-SICs under resource (test pins and TSVs) and power constraints. Depending upon the various scheduling restrictions, two test scheduling strategies have been proposed with an objective to minimize the overall test time (TT) of the stack. A step-by-step approach deals with the individual dies separately and develops power-restricted test schedules for each die and finally decides test concurrency between the dies satisfying the resources and power limits of the stack. Particle Swarm Optimization (PSO) based meta search technique has been used to select the resource allocation and power distribution to individual dies and also their internal test schedules. Incorporation of PSO in two stages of optimization produces a notable reduction in the overall test time of the SIC. Another integrated approach uses PSO to generate power-constrained test schedule of the entire SIC in a single optimization step. Integrated approach produces better results than the step-by-step approach because of its higher flexibility with lesser restrictions. User may select any of the scheduling strategies depending upon the scheduling criteria.**

***Keywords*—3D-SIC, TSV, Power-constraint test scheduling, 3D-bin packing, Particle Swarm Optimization.**

I. Introduction

In the modern era of miniaturization of feature size and dimension of semiconductor devices, long interconnects between the cores are becoming the main bottleneck of the conventional SoC designs. Interconnects, which cannot be scaled down with transistors, hamper circuit performance with its high delay and power consumption. Recently, 3D-IC has emerged to be a potential solution to this problem. Instead of designing 2D-IC with long global interconnects, interconnect lengths can be reduced significantly by designing circuit components into several layers and bonding them together. Through-silicon vias (TSVs), the vertical metal interconnects, are widely used to bind the dies of different layers of 3D-Stacked ICs (3D-SICs). 3D-integration helps to achieve high bandwidth, low latency circuit with higher packaging density and low footprint.

A major hindrance to widespread adoption of 3D IC technology is the lack of understanding of the 3D testing problems. The test engineers not only have to be concerned about the defects of the individual dies, but also have to sort out the defects in the TSVs as well as in the binding procedure of the dies into the stack. To address all these issues, 3D-SICs are tested in different phases like pre-bond, mid-bond and post-bond testing [1]. This total testing procedure is time consuming. On the other hand, shrinking product development cycle demands reduction in the Test Time (TT) of the system, hence requires a quality development of test architecture and optimized test schedule of the cores in different dies. In 3D-SIC, only the bottom die can be accessed using some limited number of external test pins. The test patterns for all the cores of the non-bottom dies have to be sent through these external test pins only, making the testing problem more complex. TSVs are used for inter-die communication to send the functional signals, power/ground, clock, as well as test access to the logic blocks of different layers of the device. The total number of TSVs must also be within a certain limit to reduce its area overhead as well as TSV defects and TSV-induced defects in devices [2]. The test scheduling and test architecture optimization problem for 3D-SICs not only includes the test infrastructure development of individual dies, it also has to take care of designing and routing of 3D Test Access Mechanism (3D-TAM) to transport the test data to the cores of the dies located at different layers of the stack.

TT may reduce significantly if a number of cores are tested in parallel in the overall test schedule. However, this is constrained by the limited number of on-chip test resources (in terms of the external test pins and TSVs). Overlapped testing of cores may also be prohibited by the system level power limit. The test mode power is often much higher than the functional mode of operation. Managing the power consumption of VLSI systems is now considered to be one of the most important challenges and must be taken into consideration when developing test infrastructure of SoC. The problem gets aggravated with the 3D integration. The test engineer has to check power validation at the time of selecting test parallelism between the cores of a die as well as test concurrency among the dies. It makes the schedule generation difficult as it is necessary to ensure power limit validation at every time instant of the schedule.

Some initial works in 3D-SIC testing have proposed test architecture design [3], while some other works considered test optimization and test scheduling [4], [5] of the entire stack. However, these works do not consider the power constraint during test optimization. Few other works [6], [7] take power limitation into account while developing pre-bond test schedule of individual dies. In this paper, we have proposed a test infrastructure development and test schedule generation strategy for 3D-SIC under resources (test pins and TSVs) and power constraints. A 3D-bin packing heuristic, guided by Particle Swarm Optimization (PSO) [8] based search procedure is designed to generate power-aware test schedules for individual dies as well as the complete stack. Two approaches have been proposed to deal with the power issues in the test schedule generation procedures. In the first approach, power-aware test schedules of the individual dies are generated separately. The total power budget is distributed to different dies to set the die level power limits. Finally, a 3D-TAM is designed for the entire stack and the test concurrency between the dies are decided subject to the resource availability and power validation of the entire stack. Resource and power distribution to the dies are

978-1-4673-9739-1/15 $31.00 © 2015 IEEE
DOI 10.1109/ATS.2015.20

carried out using PSO to get better result in test parallelism between the dies. Use of PSO in both die level and stack level optimizations helps to achieve good reduction in test time of the entire stack. In the second approach, a single power budget is considered for the entire stack and the test resources are distributed among all the cores located at different levels of the stack. A PSO guided test schedule is generated for all the cores satisfying the TSV requirements between the dies to facilitate such schedule and also the power validation.

The rest of the paper is organized as follows. Different power constrained test scheduling problems have been described in Section II. Section III and IV present the PSO guided test architecture optimization strategies for different power optimization problems stated in Section II. Results of our experimentation and related discussions have been presented in Section V. Section VI draws the conclusion of the paper.

II. Power Constrained Test Scheduling

In 3D-SICs, the cores are distributed over different layers of the stacks. Hence power-aware test schedule can be generated for the individual dies, as well as the entire stack. Depending upon the requirements of the users, two different power-constrained test scheduling problems can be formulated:

- *Die First Power Optimization*
- *Integrated Power Optimization*

In *Die First Power Optimization*, the number of test pins that can be allocated to each die, once decided, cannot be changed in the middle of the schedule. The test engineers need to develop power-aware test schedules for individual dies, for the decided amount of test resources allocated to the die. *Die First Power Optimization* requires two steps of optimization, i.e. 1) *die level optimization* and 2) *stack level optimization*. Given an allocated TAM to a die, *die level optimization* deals with resource allocation to individual cores and selection of core ordering during testing to minimize the test time of the die without violating the die level power limit. *die level optimization* is similar to the power-aware test scheduling of 2D-SoC. In *stack level optimization*, a 3D test architecture is developed and test concurrency between the dies are decided under the resource and power constraints. The test engineers have to ensure that at any point of the schedule the cumulative power budgets of the dies tested in parallel must be less than the system level power budget of the stack. It may be noted that, TT not only depends on the allocated test pins, but also on the allocated power budgets of the die. Figure 1 shows the variation of test time for different TAM widths and power budgets of ITC'02 benchmark $d695$ [9]. It clearly shows that, for a given allocated test pins to a die, pareto-optimal points can be observed in its test time for different power budgets. An optimal selection of power limits for all the dies plays an important role to obtain the best results in test parallelism. In *stack level optimization*, a test engineer has to select both the values of allocated test pins and power limits for each die, which makes the test scheduling problem more complex.

The importance of the selection of suitable power limits and test resources in test time reduction can be explained with the following example. Let us assume a 3D-SIC with

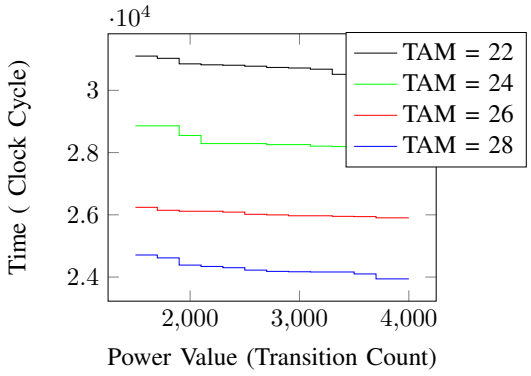


Figure 1: Variation of test time for different TAM widths and power budgets for SoC $d695$

three dies, Table I reports the pareto-optimal points in test times (TT) of each die corresponding to different TAM widths (TAM) and power budgets (P). Maximum allowable TAM width to test the SIC is 20 test pins with maximum system level power limit of 400 units. For the sake of simplicity and better understanding of the importance of optimal power limit selections of the dies, in this particular example, we have not considered the TSV constraint. However, in actual formulation, we have considered the number of TSVs to decide test parallelism between dies. Figure 2 shows two possible cases of distribution of test pins and power limits to different dies. In the first case, 15, 20 and 15 test pins are allocated to the dies D_1, D_2 and D_3 respectively and their corresponding power limit allocations are 250, 400 and 150 units. D_1 and D_3 are tested in parallel and after that D_2 is tested. The total test time of the stack is 275 units. It may be observed from Table I that, test time remains unchanged for D_1, in the power limit range between 175 units to 450 units, when it is tested with 15 test pins. An efficient power allocation to D_1 would be 175 units, instead of 250 units, considered in the example. These extra power limits might be used for other dies to get better test time for them. Same observations can be found for other dies as well. On the other hand, in case 2, power limits and test resources are much efficiently distributed among different dies, which help to reduce the test time of the stack.

Table I: An Illustrative Example Data Set

		TAM = 10			TAM = 15			TAM = 20		
Die 1	**P**	>50	>200	>450	>50	>175	>450	>50	>125	>375
	TT	400	300	100	300	200	70	200	150	50
Die 2	**P**	>100	>300	>500	>100	>250	>650	>100	>175	>450
	TT	300	200	100	225	150	70	150	125	75
Die 3	**P**	>75	>225	>400	>75	>200	>400	>75	>175	>350
	TT	200	150	100	150	100	75	100	80	50

The power-constrained test scheduling problem for 3D-SIC can also be solved in an integrated fashion using a single step of optimization. The *Integrated Power Optimization* considers a single power budget for the entire stack and develops a test architecture and generates a test schedule for all the cores of the stack without handling different layers separately. A 3D TAM is designed for the entire stack and the test resources are dynamically partitioned and allocated to the cores of different layers to generate a power-constrained test schedule without violating the criteria of TSV requirements to validate the schedule. *Integrated Power Optimization* makes the best utilization of the test resources by reallocating the idle resources for further test applications. It gives lot of

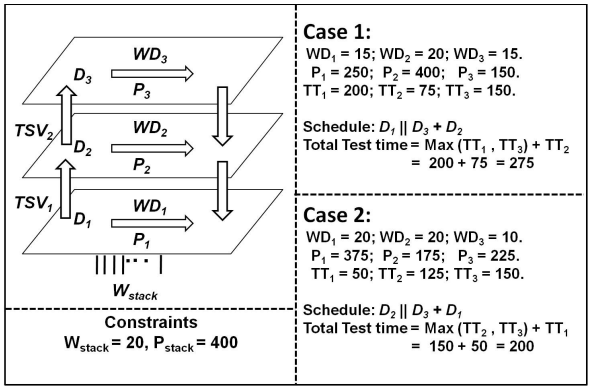


Figure 2: An example solution of efficient stack level TAM and power distributions

flexibility in test schedule and makes the schedule generation procedure simple. However, it cannot handle the situation in which individual dies have limits on maximum power consumption. The $Die\ First\ Power\ Optimization$ can take care of such situation better.

III. Die First Power Optimization

The power-aware test architecture and test schedule problem of 3D-SIC under $Die\ First\ Power\ Optimization$ approach can be described as follows.

A. *Problem Formulation*

Suppose a stack consisting of M dies D_i $(1 \leq i \leq M)$ is to be tested with a maximum available test pins W_{stack}. Assume that the maximum number of allowable TSVs to route the 3D-TAM is TSV_{stack} and the stack level maximum power limit is P_{stack}. Each die D_i consists of N_i number of cores $C_{1_i}, C_{2_i} \ldots C_{N_i}$. Determine an optimal 3D-TAM design of the stack and also select the power budget of each die optimally. After allocating W_{die} test pins and P_{die} power limit to a die, a power constrained test schedule needs to be developed for each die, satisfying the allocated resource and power constraints of the die. Finally, a test schedule is to be generated for the entire stack by selecting test parallelism between the dies, such that the total test time TT for the stack is minimized without violating the resource constraints and the power consumed by the stack at each instant of time during test is less than P_{stack}.

B. *Scheduling Strategy*

Both the problems of (i) test pins and power limit distribution among the dies of the different layers and (ii) determining an optimized test schedule for cores in a die, are NP-hard [4]. We have approached these problems in two steps. First, we have carried out $die\ level\ optimization$. The results of $die\ level\ optimization$ are used in later part of $stack\ level\ optimization$, where we have carried out the 3D-TAM design and power limit distribution to the dies. Particle Swarm Optimization (PSO) guided heuristics have been used in both levels of optimization.

1) ***Die level optimization****:* Lot of research works have been carried out in the past to address the power-aware test scheduling problem of 2D-SoC. Any of the available scheduling strategies can be used to solve the $die\ level\ optimization$ problem. In our previous work [10], we have proposed a PSO guided 3D-bin packing heuristic to solve the power-restricted test scheduling of 2D-SoC. PSO is a population based evolutionary technique [8], which has the advantage of better exploration of large search spaces and faster convergence towards near optimal solution. Our 3D-bin packing heuristic, guided by PSO based search procedure, has enabled us to evolve towards better test schedules than many of the contemporary SoC testing approaches, while working with ITC'02 benchmarks [9]. In the present formulation, we have used the same PSO guided 3D-bin packing heuristic for $die\ level\ optimization$. However, to reduce the scheduling complexity, we have considered single fixed peak power model of each core, instead of considering a relatively complex window-based peak power model proposed in [10].

PSO starts with an initial population of particles. Each particle corresponds to a probable test schedule, which we feed to the 3D-bin packing heuristic. Each particle has its fitness value in terms of test time TT, which is obtained from the bin packing heuristic. Particles evolve towards better solution, over the generations guided by three factors – its own intelligence ($pbest$), global intelligence ($gbest$), and the inertia factor.

Rectangular 2D-bin packing has evolved to be a popular method to solve the test scheduling problem for embedded cores. Each core C_i $(1 \leq i \leq N)$ is represented by a set of wrapper configurations R_i. The test resource requirement of core C_i with j^{th} wrapper configuration can be represented by a rectangle whose height and width represent allocated TAM width (w_{ij}) and the corresponding test time $(T(w_{ij}))$ respectively. Power constrained scheduling takes this problem to a 3D-bin packing problem, where power represents the third dimension. To get a schedule for the full die, the rectangles are selected in descending order of their area and packed into a bin of fixed height (W_{die}) so that test time (width of the bin) is minimized without violating TAM requirements and power budgets at every point of the schedule.

2) ***Stack level optimization****:* The main objective of $stack\ level\ optimization$ is optimal allocation of test resources and power distribution to the dies and selection of test concurrency between the dies satisfying resources and power constraints.

As we have shown in Section II, pareto-optimal points in test time of a die can be observed for different allocated test pins and also for different power budgets for a particular number of test pins. To find such pareto-optimal points, we run the $die\ level\ optimization$ part for different test pins and power budgets and create a $pareto\text{-}library$ for each die. Figure 3 shows an example $pareto\text{-}library$ for a die. A second level PSO is used to select a value from $pareto\text{-}library$ of each die and feed to a $3D\_Test\_Schedule$ heuristic, which eventually finds the test time TT of the stack. Figure 4 shows an example of stack level particle for five dies. A particle has two parts, the first part selects the test pins (W_{die_i}) of each die while the second part selects the corresponding power limit (P_{die_i}). For example, in Figure 3, 2^{nd} pareto-optimal TAM and corresponding 3^{rd} pareto-optimal power limit get selected by a particle for the example die. PSO evolves over the generations towards better solution in TT. The $3D\_Test\_Schedule$ (Algorithm 1) heuristic selects the dies in descending order of their test time and decides test concurrency between the dies at the lowest possible scheduling

point. A $Power\_Checker$ checks power validation at every point of the schedule.

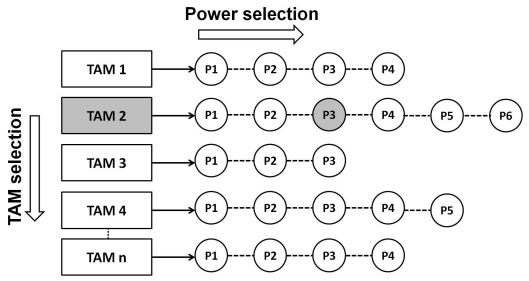


Figure 3: An example of $pareto\text{-}library$

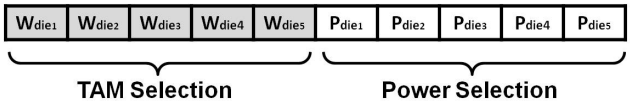


Figure 4: An example particle for $stack\ level\ optimization$

TSVs play a major role in designing test architecture of 3D-SICs. TSVs between each layer has to be decided appropriately to design a 3D-TAM to route the test data to the cores of the non-bottom dies. Test concurrency between the cores largely depends on the TSV limitation along with the test pin and power constraints. The number of TSVs required between the layer l and $l-1$ is determined by two factors-

1) The maximum number of test pins required by a layer at or above l.
2) The sum of test pins for parallelly tested dies at or above layer l at any particular point in the schedule.

TSVs between layers l and $l-1$ must be equal to the maximum of these two quantities. TSVs cannot be allocated dynamically. TSVs between all the layers must be determined before a SIC starts its scheduling. At the time of schedule generations using $3D\_Test\_Schedule$ heuristic, a $TSV\_Allocator$ is used to allocate TSVs between all the layers so that it can facilitate the test concurrency between the dies without violating TSV criteria at any point of the obtained test schedule.

Input : List of dies $D_i(1 \leq i \leq M)$ to be scheduled with assigned test pins W_{Die_i}, power P_{Die_i} and corresponding test time T_{Die_i}; W_{stack}: maximum test pins of the stack; TSV_{stack}: maximum TSV limit; P_{stack}: maximum power limit of the stack;
begin
  current scheduling point ⟵ 0;
  Select a die D_i in descending order of T_{Die_i};
  if D_i satisfies test pins requirement at the current scheduling point **then**
    $TSV\_Allocator$ checks TSV criteria;
    $Power\_Checker$ checks power validation;
    if All satisfied **then**
      Schedule D_i;
      Check the schedulability of the next unscheduled die in descending order of T_{Die_i} at current scheduling point;
  else
    Check next die of the sorted list;
  if *All the dies are yet to be scheduled* **then**
    Move to the next scheduling point;
  Return TT as the maximum test end time among all the dies;

Algorithm 1: *3D_Test_Schedule*

IV. Integrated Power Optimization

The power-aware test architecture and test schedule problem of 3D-SIC under $Integrated\ Power\ Optimization$ approach can be described as follows.

A. *Problem Formulation*

Suppose a stack consisting of M dies D_i $(1 \leq i \leq M)$ is to be tested with a maximum available test pins W_{stack}. Assume that the maximum number of allowable TSVs to route the 3D-TAM is TSV_{stack}. Each die D_i consists of N_i number of cores $C_{1_i}, C_{2_i} \ldots C_{N_i}$. Determine an optimal TAM design and corresponding test schedule for the stack, such that the total test time TT for the stack is minimized without violating resource and TSV constraints and the integrated power consumptions of all the cores (irrespective of their die levels) tested in parallel at any time instant must not violate a stack level maximum power limit of P_{stack}.

B. *Scheduling Strategy*

Integrated Power Optimization allows dynamic TAM allocation to the cores in the scheduling procedure, which encourages us to develop a PSO guided 3D-bin packing approach similar to the $die\ level\ optimization$ of $Die\ First\ Power\ Optimization$ with extra TSV constraint to facilitate 3D-TAM design. However, the particle structure and the PSO formulation is slightly different in $Integrated\ Power\ Optimization$.

a) ***PSO Formulation****:* Each core C_{j_i} $(1 \leq j \leq N_i)$ is represented by a set of wrapper configurations R_{j_i}. A particle selects a test rectangle for each core C_{j_i} from the set of its wrapper R_{j_i}. Let $B = \lceil \log_2^R \rceil$, where R is the maximum number of rectangles a core can have. A particle consists of $\sum_{i=1}^{M} N_i \times B$ number of bits. First B bits identify the test rectangle selected for the first core, second B bits for the second core, and so on. Similarly first N_i cores represent the cores of the 1^{st} die, second N_i cores for the 2^{nd} die, and so on. Figure 5 shows a sample particle with M = 3 and B = 4. Fitness of a particle is equal to the total test time (TT) of the 3D-SIC after scheduling the test rectangles using the $Integrated\_Test\_Schedule$ procedure. For the initial generation, particles are generated randomly. In the successive generations, new particles are created using a $replace$ operator, which attempts to align a particle with its $pbest$ and the $gbest$ particles, with some probability. For the sake of this alignment, the $replace$ operator is applied at bit position of a particle. For bit position i of a particle, the bit is replaced by the corresponding bit of $pbest$ particle with probability α. After the operator has been applied for $pbest$, the same is done with respect to $gbest$ with probability of replacement, β. In our experimentation, we have kept both α and β values at 0.1.

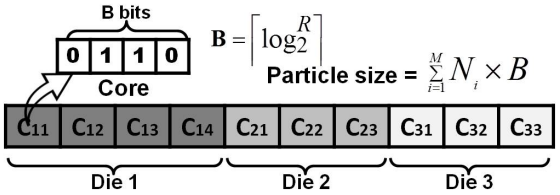


Figure 5: An example particle for $Integrated\ Power\ Optimization$

*b) **Scheduling Algorithm**:* The algorithm takes as input the rectangle set corresponding to a particle, the maximum test pins W_{stack}. SP, ATP and PT keep track of the scheduling points, corresponding resource availability at those points and power values at SPs respectively. As the still unscheduled cores get scheduled, the list SP, ATP and PT get updated. The rectangles are sorted on their area values (test pins (w) $\times$ test time (T)) in a descending order. The scheduling-point list SP is scanned from the minimum to the maximum value. For the scheduling-point sp_k, the algorithm scans the unscheduled rectangle list to check for the largest rectangle that can be scheduled at sp_k. A $Power\_Checker$ checks power validity at each SP and a $TSV\_Allocator$ (as mentioned in $stack\ level\ optimization$), checks TSV requirement between different layers of the stack to facilitate concurrent testing of core C_{j_i} with other cores of different dies tested in parallel. If C_{j_i} does not satisfy resource, TSV and power constraints, we move to the next core until all the cores get scheduled. If none are feasible, the algorithm advances to the next scheduling-point. When rectangles corresponding to all cores have been scheduled, the maximum end time of testing of all cores gives the total test application time for the SoC. The $Integrated\_Test\_Schedule$ algorithm to produce the schedule is presented next.

Input : List of rectangles to be scheduled; W_{stack}, the maximum test pins;
Var : SP**:** A list of scheduling points; ATP**:** List of available test pins at each scheduling point $sp \in SP$; PT**:** List of total power values at each scheduling point $sp \in SP$;
begin
  Sort list of rectangles on decreasing area;
  Mark all rectangles as unscheduled;
  while there exists unscheduled rectangles **do**
    Check if any rectangle picked up in sorted order can be scheduled at next scheduling point sp_k with available TAM resource atp_k;
    $TSV\_Allocator$ checks TSV criteria;
    $Power\_Checker$ checks power validation;
    if All satisfied **then**
      Update SP, ATP, PT and Rectangle List;
      Mark corresponding rectangle scheduled;
    else
      Continue with next $sp_k \in SP$;
  Return the maximum test end time of all rectangles;

Algorithm 2: ***Integrated_Test_Schedule***

V. Experimental Results

In this section, we present the results of our experimentation for both $Die\ First\ Power\ Optimization$ and $Integrated\ Power\ Optimization$. For the sake of comparison, we have considered the same 3D-SIC benchmarks presented in [4]. Figure 6 presents the SICs, which are formed using ITC'02 benchmarks as dies.

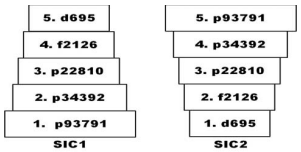
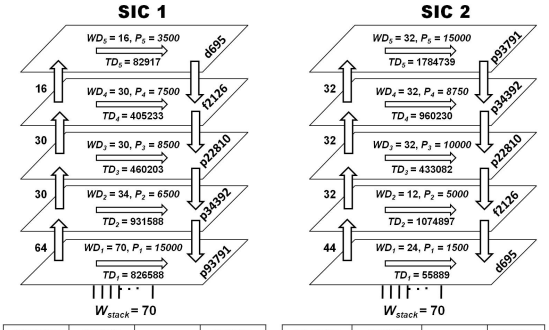


Figure 6: 3D-SIC benchmarks [4]

Table II presents the test pins and TSVs allocation and power distribution among the dies and the test times TT obtained for the entire stack, using $Die\ First\ Power\ Optimization$, for SIC1 and SIC2, for different values of W_{stack} and P_{stack}, considering TSV_{stack} = 140. Each of the columns marked as test pins and power for both SICs has five values representing the test pins and power limits of the five dies in an ascending order of dies from D_1 to D_5. The columns marked as TSVs have four values each, representing the TSVs allocated between the dies D_1 and D_2, D_2 and D_3 and so on till D_5. Figure 7 pictorially shows the resource and power distribution among the dies and also the validation of different constraints at different scheduling points for both SICs for $W_{stack} = 70$, $P_{stack} = 15000$ and $TSV_{stack} = 140$. It may be noted that, for both SICs, our obtained test schedules do not violate resource and power constraints. An obvious observation is that, TT reduces with the relaxation of resource and power constraints, which encourage more test parallelism. However, it may be noted that, for the same resource and power constraints, the test time of SIC2 is generally higher than the SIC1. This is due to the fact that, in SIC2, more complex dies are placed on the higher levels of the stack, which requires more number of TSVs to test them concurrently to minimize TT. Because of this TSV limitation only, we found no improvements in TT for SIC2 even if we increase the test pins after a certain value.



Scheduling Interval	Parallelly Tested Dies	Total Power	Test Pins Used
0 - 826588	D_1	15000	70
826588 - 1286791	D_2, D_3	6500+8500 = 15000	34+30 = 64
1286791 - 1692024	D_2, D_4	6500+7500 = 14000	34+30 = 64
1692024 - 1758176	D_2, D_5	6500+3500 = 10000	34+16 = 50
1758176 - 1774941	D_5	3500	16

Scheduling Interval	Parallelly Tested Dies	Total Power	Test Pins Used
0-1784739	D_5	15000	32
1784739 - 2744969	D_2, D_4	5000+8750 = 14750	32+12 = 44
2744969 - 2859636	D_2, D_3	5000+10000 = 15000	12+32 = 44
2859636 - 2915525	D_1, D_3	1500+10000 = 11500	24+32 = 56
2915525 - 3178051	D_3	10000	32

Figure 7: Pictorial illustration of $Die\ First\ Scheduling$

Table III presents a comparative analysis of our proposed $Die\ First\ Power\ Optimization$ (noted as DF) and $Integrated\ Power\ Optimization$ (noted as IPO) strategies with the test scheduling strategy proposed in [4]. It also shows the variation of TT of SIC1 for different power and test pin limits. It may be noted that, for no power constraint scenario, average improvement of DF over [4] is 34.17%, while IPO produce 35.3% better results than [4] on an average. It proves the quality of our scheduling strategy. However, we could not carry out the comparison under power constraint, because [4] does not consider power restriction in its formulation. $Integrated\ Power\ Optimization$ performs slightly better than the $Die\ First\ Power\ Optimization$ due to its flexibility of better utilization of test resources and dynamic TAM allocation.

Figure 8(a) shows the results of $Integrated\ Power\ Optimization$ (IPO) for different power limits for SIC2. It also shows the result of $Die\ First\ Power\ Optimization$ (DF) for $P_{stack} = \infty$. Again, no improvement in TT is observed with increase of W_{stack} for the same TSV limit. To observe the importance of number of TSVs in test time reduction, we have varied the number of TSVs as well as

Table II: Test Time (TT) Obtained Using $Die\ First\ Power\ Optimization$ For SIC1 And SIC2 For Different Values Of W_{stack} And P_{stack} (TSV_{stack} = 140)

	SIC1				SIC2			
W_{stack}	TT	Test Pins	TSVs	Power Distribution	TT	Test Pins	TSVs	Power Distribution
	P_{stack} = **15000**							
30	**3826299**	30,30,30,30,30	30,30,30,30	15000,9500,12500,7500,3700	**3826299**	30,30,30,30,30	30,30,30,30	3700,7500,12500,9500,15000
40	**2892732**	40,30,40,10,20	40,40,30,20	15000,6750,12750,7500,3100	**3269948**	12,10,28,28,34	38,34,34,34	3100,5000,9500,8500,15000
50	**2340756**	50,36,50,14,14	50,50,14,14	15000,6750,9750,7500,1900	**3178051**	16,12,32,32,32	44,32,32,32	1500,5000,10000,8750,15000
60	**1999945**	60,30,30,30,10	60,30,30,10	15000,6750,8000,7500,1900	**3178051**	10,12,32,32,32	44,32,32,32	1900,5000,10000,8750,15000
70	**1774941**	70,34,30,30,16	64,30,30,16	15000,6500,8500,7500,3500	**3178051**	24,12,32,32,32	44,32,32,32	1500,5000,10000,8750,15000
80	**1747090**	80,46,46,18,12	64,46,18,12	15000,8500,12750,5500,3100	**3178051**	44,10,32,32,32	42,32,32,32	3100,5000,10000,8750,15000
90	**1606237**	90,44,50,18,10	62,50,18,10	15000,8500,9750,5500,3100	**3178051**	30,12,32,32,32	44,32,32,32	1700,5000,10000,8750,15000
100	**1546279**	100,32,34,30,10	66,34,30,10	15000,7000,8000,7500,1900	**3178051**	36,12,34,30,32	42,34,32,32	1500,7500,12750,6750,15000
	P_{stack} = **20000**							
30	**3779616**	30,16,14,14,14	30,14,14,14	19500,9750,9250,7500,1700	**3779616**	14,14,14,16,30	30,30,30,30	1900,7500,10000,9750,19500
40	**2855419**	40,22,18,18,32	40,32,32,32	19500,8500,8250,7000,3700	**2959155**	40,18,40,16,22	40,40,38,22	3500,5000,12750,5000,15000
50	**2321515**	40,34,16,10,10	50,16,10,10	15000,9750,10250,5000,1700	**2925569**	10,16,40,16,22	40,40,38,22	3100,5000,12750,5000,15000
60	**1963451**	60,46,42,14,10	60,56,14,10	19500,9750,12500,7500,1700	**2925569**	20,18,40,16,22	40,40,38,22	2300,5000,12750,5000,15000
70	**1668548**	70,34,30,30,16	64,30,30,16	19000,6750,12500,7500,3500	**2925569**	16,16,40,16,22	40,40,38,22	1500,5000,12750,5000,15000
80	**1482714**	64,56,24,16,10	80,24,16,10	15000,8750,11250,5000,1900	**2925569**	56,10,12,28,30	40,40,30,30	1700,5000,11500,8500,15000
90	**1351597**	74,56,26,16,16	82,26,16,16	15000,9250,8000,5000,2500	**2925569**	56,10,12,28,30	40,40,30,30	1700,5000,11500,8500,15000
100	**1291106**	82,56,28,18,10	84,28,18,10	15000,9250,8500,5000,2900	**2925569**	56,10,12,28,30	40,40,30,30	1700,5000,11500,8500,15000
	P_{stack} = **25000**							
30	**3775474**	30,16,14,14,12	30,14,14,12	24500,9750,9750,7500,1900	**3746065**	12,14,14,16,30	30,30,30,30	2100,7500,9500,9750,24500
40	**2860239**	20,20,20,20,34	34,34,34,34	16500,8500,8500,7000,2900	**2854484**	40,20,20,20,20	40,40,40,20	3500,7000,8250,7750,16500
50	**2298293**	50,28,12,10,24	50,36,24,24	24500,8500,8250,7500,3700	**2830521**	10,18,40,16,22	40,40,38,22	3100,5500,12750,5250,19500
60	**1968859**	44,16,26,16,12	58,42,28,12	19000,5750,10000,5000,3100	**2822347**	12,20,20,20,20	40,40,40,20	1900,7000,8750,7750,16000
70	**1662709**	70,38,16,16,12	70,32,16,12	24500,6500,12250,5000,2100	**2822347**	12,20,20,20,20	40,40,40,20	1900,7000,8750,7750,16000
80	**1461896**	62,56,24,18,10	80,24,18,10	20000,9250,12250,5000,1700	**2822347**	12,20,20,20,20	40,40,40,20	1900,7000,8750,7750,16000
90	**1342647**	72,56,24,18,18	80,24,18,18	20000,9250,12250,5000,1900	**2822347**	12,20,20,20,20	40,40,40,20	1900,7000,8750,7750,16000
100	**1263339**	84,54,28,16,14	82,28,16,14	20000,9750,8500,5000,1700	**2822347**	12,20,20,20,20	40,40,40,20	1900,7000,8750,7750,16000

Table III: Comparison Of Test Time(TT) Between Different Power-Aware Test Scheduling Strategies For SIC1 (TSV_{stack} = 140)

Test Pins	Test Time TT					
	[4]	DF	IPO			
	P_{max} ∞	P_{max} ∞	P_{max} 15000	P_{max} 20000	P_{max} 25000	P_{max} ∞
30	4795930	3724309	3751765	3696978	3680484	3666034
40	3841360	2802280	2827639	2804366	2799428	2768077
50	3090720	2265193	2256484	2250509	2242455	2215398
60	2873290	1875655	1921861	1899615	1853489	1847130
70	2743320	1623521	1673925	1623343	1610358	1593334
80	2439760	1448635	1467448	1457778	1413967	1403787
90	2395760	1267313	1348467	1324112	1278399	1252576
100	2369680	1153169	1208888	1166024	1153080	1136263
Avg. Imp over [4]		**34.17%**	-	-	-	**35.3%**

W_{stack} and reported the TT of SIC2 in Figure 8(b). A notable reduction in TT can be observed with the increase of TSV_{stack} as well as W_{stack} values. Therefore, a proper selection of both of the test resources (test pins and TSVs) with the power limit helps to reduce the test time.

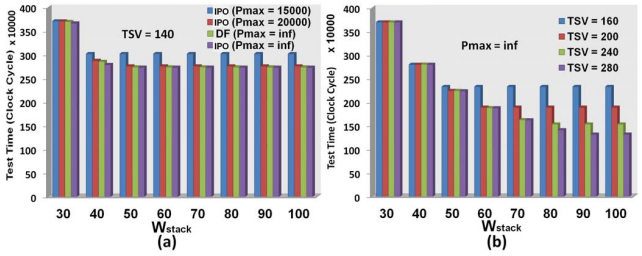


Figure 8: (a) $Integrated\ Power\ Approach$ results for SIC2; (b) Variation of TT with W_{stack} and $TSV stack$ for SIC2

VI. Conclusion

In this paper we have proposed two different solution strategies for resource and power constrained test scheduling of 3D-SIC. PSO guided heuristic has been used to generate the test schedule of the individual dies as well as to select resource and power distribution among the dies for step-by-step approach. A single step PSO based solution strategy has also been proposed to solve the problem in an integrated fashion. Both of our proposed strategies can produce near optimal results without violating resource and power constraints.

References

[1] M. Agrawal and K. Chakrabarty, "Test-cost optimization and test-flow selection for 3d-stacked ics," in *VLSI Test Symposium (VTS), 2013 IEEE 31st*, April 2013, pp. 1–6.

[2] K. Chakrabarty, S. Deutsch, H. Thapliyal, and F. Ye, "Tsv defects and tsv-induced circuit failures: The third dimension in test and design-for-test," in *Rel. Physics Symp. (IRPS), 2012 IEEE Int.*, 2012, pp. 5F–1.

[3] C.-Y. Lo, Y.-T. Hsing, L.-M. Denq, and C.-W. Wu, "Soc test architecture and method for 3-d ics," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 10, pp. 1645–1649, 2010.

[4] B. Noia, K. Chakrabarty, S. K. Goel, E. J. Marinissen, and J. Verbree, "Test-architecture optimization and test scheduling for tsv-based 3-d stacked ics," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 11, pp. 1705–1718, 2011.

[5] R. Karmakar, A. Agarwal, and S. Chattopadhyay, "Testing of 3d-stacked ics with hard-and soft-dies-a particle swarm optimization based approach," in *VLSI Test Symposium (VTS)*. IEEE, 2015, pp. 1–6.

[6] S. Roy, J. Sengupta, C. Giri, and H. Rahaman, "Power constraints test scheduling of 3d stacked ics," in *Design and Test Symp, 2013 Int. 8th*, pp. 1–6.

[7] B. Sen Gupta, U. Ingelsson, and E. Larsson, "Scheduling tests for 3d stacked chips under power constraints," in *Electronic Design, Test and Application (DELTA), 2011 Sixth IEEE Int. Symp. on*, pp. 72–77.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE Int. Conf. on*, pp. 1942–1948 vol.4.

[9] E. J. Marinissen, V. Iyengar, and K. Chakrabarty, "Itc'02 soc test benchmarks web site," 2006.

[10] R. Karmakar and S. Chattopadhyay, "Window-based peak power model and particle swarm optimization guided 3-dimensional bin packing for soc test scheduling," *Integration, the VLSI Journal*, vol. 50, pp. 61–73, 2015.