Thermal Aware Test Scheduling for Stacked Multi-Chip-Modules

Vinay N.S.* , Indira Rawat†, Erik Larsson‡, M.S. Gaur§ and Virendra Singh¶
*Cisco Systems, Bangalore, India
Email: ambani.vin.ns@gmail.com
†Electrical Enggineering Dept., Engineering College, Ajmer, India
Email: indirarawat@yahoo.co.in
‡ Dept. of Computer Science, Linkoping University, Linkoping, Sweden
E-mail: erila@ida.liu.se
§ Computer Engineering Dept., Malaviya National Institute of Technology, Jaipur, India
E-mail: gaurms@mnit.ac.in
¶ Computer Design and Test Lab., Indian Institute of Science, Bangalore, India
E-mail: viren@serc.iisc.ernet.in

***Abstract*—In an attempt to increase the area utilization of multi-chip packages, manufacturers have started looking at 3D packaging. The 3D structures have either dies or chips stacked one above the other. These 3D structures have low thermal capabilities, and hence the thermal issues get aggravated during testing. In this paper, thermal aware test scheduling techniques for stacked structures is proposed. Care is taken to schedule the stacked partitions appropriately to prevent local heating and achieve uniform vertical temperature spread. Concurrent testing of dies is considered for reduced test time and existing test architecture is employed.The widely used global peak power model gives a pessimistic estimate of the power profile. A new partition scheme is proposed for the power profile based on power variations and true power. This scheme of power profile partitioning which is a trade-off between a cycle accurate model and global peak model offers both flexibility during scheduling and also reduces total false power . The power profile partitions are scheduled through algorithms based on heuristics. System model of the 3D stack is obtained from the linear RC-models after considering the various factors that affect the heat flow and die temperatures. Simulation results show that the proposed technique achieves uniform temperature spread across the 3D structure.** [1] [2]

I. Introduction

The benefits of incorporating more than one die on a substrate are numerous and compelling. Multiple chip Module (MCM) solutions provide better performance through shorter interconnects as compared to single chip packages connected through the printed circuit board fabric. The latest development in MCM is the 3D orientation of chips called as stacked-MCM. Stacked-MCMs have silicon wafers stacked one above the other and wire-bonded to form a package.The stacked structure has certain drawbacks which have to be addressed. The most important among these are the thermal issues. Due to the stacked nature, inner chips encounter higher temperature resistance and have higher cooling down times. This results in overheating and may damage the chips. Thus care has to be taken during floorplanning, routing and testing stages. Several approaches have been proposed to achieve good thermal spread during the design and placement for planar chips, as well as 3D stacks. The most recent ones [1] tries to achieve a smooth temperature profile and addresses the issue during the placement of modules in a 3D structure during floor-planning. Apart from this, fabrication and packaging issues also have to be addressed. Kim et al.[2] and Economikos et al. [3] discuss the related issues during fabrication and packaging. Testing is one of the most important phases of IC fabrication and deployment, hence it is necessary to address thermal issues during testing as well. Not many attempts have been made in this regard. Liu et al.[4] and He et al. [5] discuss thermal aware testing for planar SoCs. The most recent, Rosinger et al.[8] address the thermal aware scheduling for a core based SOC. The 3D orientation offers more challenges for test scheduling and special considerations have to be done to account for the severe thermal constraints.

In this paper we present thermal aware test scheduling schemes for stacked MCMs along with a special partitioning scheme. The rest of the paper is organized as follows; section 2 discusses the test architecture for stacked MCM. Section 3 describes the thermal model used. Section 4 describes the system modeling aspects. Section 5 discusses the algorithms for thermal aware test scheduling. Sections 6 and 7 have the experimental results and conclusions.

II. Test architecture

The standard boundary scan test-architecture which is the most widely used, has been extended to obtain test architecture for an MCM. The basic idea of this extended test architecture is to have access to all the chips in the stack, either in parallel or sequentially so that all the components of the MCM can be tested. Various methods have been mentioned in

[1] This work was carried out when Vinay N.S. was with CEDT, Indian Institute of Science, Bangalore

[2] This research is partly supported by The Swedish Foundation for International Cooperation in Research and Higher Education (STINT) through Institutional Grant for Younger Researchers for collaboration with Indian Institute of Science (IISc), Bangalore, India.

978-1-4244-9556-6/10/$26.00 ©2010 IEEE

[6]. The different methods offer tradeoff between Test Access Mechanism complexity and test flexibility. The test mechanism for a stacked MCM should provide the following vital features.

- Provision for testing all the chips on the stack.
- Built in self test for certain chips.
- Concurrent testing of chips on the stack.

Test architectures such as parallel-MCM Test Controller(MTC) and reconfigurable MTC[6] provide the essential test features stated above and the choice has no effect on the proposed algorithms. We have taken MTC as the reference test architecture.

III. Thermal characterization

Thermal modeling of the stacked structure is essential for the proposed test scheduling algorithms. The most widely used thermal model is the linear RC-model which not only offers reduced computational complexity which is essential for all scheduling algorithms but also models the system with reasonable accuracy. The thermal model used here can be refereed to in [7]. It models the interface with resistances and the heat sources(chips being tested) as capacitors as shown in Fig:1 . The side wall thermal resistance is high and hence is neglected. The thermal model forms the basis on which the scheduling algorithm related parameters are generated.

IV. System model

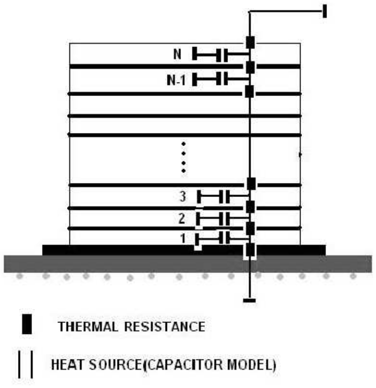

Fig. 1. Thermal model of the stack structure

This section deals with defining and deriving the parameters which directly affect the scheduling process. These parameters give a quantitative estimate of the die temperature, heat flow, temperature gradient, cooling times and power dissipation in the stack structure. These variables are later used by the algorithm to generate the cost function which in turn decides the test schedule.

The naming of chips is done from the bottom of the stack. The structure shown in Fig: 1 has N chips stacked. The bottommost is numbered 1 and the topmost is N. The resistance values can be obtained from the thermal model. Even though all the chips in the stack differ in functionality and hence in area the interface resistances are almost same. The system model parameters are listed below. A detailed explanation is given in the section that follows.

- Chip-number (i): Chip number indicates the location of the chip in the stack. The bottom most chip is numbered as 1 and the topmost as N.
- $d_{i,j}$: This is the distance between two chips in the stack and is defined as the difference between the chip numbers $d_{i,j} = i - j$.
- Chip-exposure (DE_i): Indicates the thermal exposure of Chip i.
- Thermal influence ($T_{i,j}$): Thermal-influence value indicates the effect of the thermal state of chip i on j.
- Power profile (P_i): It is the rectangular box profile model of the single chip i.
- Modified power profile (P_i^m): The power profile which accounts for Chip-exposure.
- Partitioned power profile ($P_{i,j}^m$): The partitioned modified power profile.
- Adjacency factor ($A_{i,j}$): This indicates how close chip j is to a chip i thermally.

A. Die Exposure

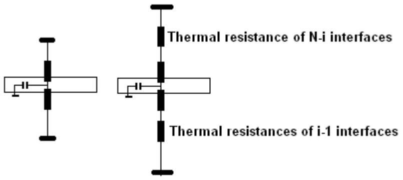
Thermal resistance of N-i interfaces
Thermal resistances of i-1 interfaces

Fig. 2. Die Exposure

As seen from Fig: 2, the thermal impedance seen by a single chip with a planar orientation is lesser than that of a chip in a stack. Therefore while considering the power profile of these chips in the stack scaling has to be done to account for this. If not, the power constraint considered for the structure will be optimistic. Hence we scale the power profile with the chip exposure constant. This scaling factor (chip-exposure) is defined to be the ratio of the thermal-impedance in the two cases. The expression for chip-exposure is derived below.

It is assumed that the thermal resistances are equal at the chip-interface and is equal to R. The capacitor acts as open for steady flow of heat and hence does not come into effect during steady state.

Looking at Fig: 2, we have, for chip i in the stack, the thermal impedance Z_i ,

$$Z_i = i * R||(N + 1 - i) * R(||meaninginparallel)$$

Simplifying, we get,

$$Z_i = i * (N + 1 - i) * R/(N + 1)$$

Now for a bare chip the thermal impedance Z_b is

$$Z_b = R/2$$

Thus, we have the ratio:

$$DE_i = \frac{Z_i}{Z_b} \tag{1}$$

$$DE_i = \frac{2*i*(N+1-i)}{(N+1)} \tag{2}$$

This is independent of the thermal resistance and hence the assumption of equal interface resistance does not introduce significant errors.

B. Thermal Influence

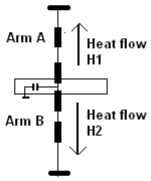

Fig. 3. Heat flow in stack

When a chip is being tested, i.e. the chip under test(CUT) the heat dissipated during testing flows through either of the two arms as shown in Fig: 3. The ratio of heat through the arms is proportional to the ratio of the thermal impedance of the two arms. The thermal influence of the CUT on the other chips in the stack is quantified by the heat flow in the arm. The expression for thermal influence is derived below.

$$T_{i,j} = \frac{R_j}{R_j^{'}} \tag{3}$$

When $d_{i,j} > 0$,

$$T_{i,j} = \frac{(N+1-i)}{i} \tag{4}$$

When $d_{i,j} < 0$,

$$T_{i,j} = \frac{i}{(N+1-i)} \tag{5}$$

where R_j and $R_j^{'}$ are the thermal resistances of the arm containing die j and the arm not containing die j respectively.

C. Adjacency factor

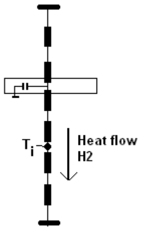

Fig. 4. Effect of heat flow on Temperature of chip

While the thermal influence due to the CUT is same for all the chips on the same arm, the actual effect of this heat flow on the temperature of the wafer is different. This is analogous to a branch with resistances connected in series with a constant current flowing through it, even though the current through the resistances are the same , the actual voltage is different. The adjacency factor models this effect. The temperature of the chip can be derived with the help of thermal-influence and interface resistances and is explained below. From Fig: 4 we can say that adjacency factor is a function of chip-number and interface resistance. Also the temperature of chip j when chip i is being tested is equal to the product of the heat flowing in the arm and ratio of the resistance to chip j from i to the total resistance in the arm. Now the heat flowing in this arm is quantitatively given by the thermal influence as explained before.

Looking at Fig: 4 the ratio of the resistances is given by,

$$A_{i,j} = f(i,j,R)$$

When $d_{i,j} > 0$,

$$A_{i,j} = \frac{j*R}{i*R} = \frac{j}{i} \tag{6}$$

When $d_{i,j} < 0$,

$$A_{i,j} = \frac{(N+1-j)*R}{(N+1-i)*R} = \frac{(N+1-j)}{(N+1-i)} \tag{7}$$

The box representation for the power profile of a chip is not efficient. This is because the difference between the maximum and minimum power in the profile is huge. Global peaks give a pessimistic estimate for the box representation as explained in Fig: 5. In this paper we look at an innovative scheme which involves the partitioning of the power profile around local peaks, thus reducing false power. Also partitioning the power profiles gives flexibility as to when the chip is actually tested. More partitions may result in higher complexity; hence we put a limit on the maximum number of partitions and use the above method. The partitioning scheme puts cap on the ratio

$$\frac{Falsepower}{Truepower}$$

. That is whenever this ratio exceeds a pre-determined value we partition the power profile thereby obtaining partitions which have lower false power.

The above partitioning results in better approximation than the simple box model. The increase in efficiency as compared to the box model for different benchmarks with random excitation is given in table 2.

The above definitions model the system parameters which affect individual chip temperatures. With these parameters, a cost function can be obtained and the scheduling can be performed by using the cost function. The cost function indicates the effect that one chip has on the other in the stack. Looking at the system parameters it is easy to see that the product of Thermal Influence and the Adjacency Factor gives the temperature of a chip in the stack. Thus when chip i is being tested the effect on chip j is given by $T_{i,j} * A_{i,j}$. Thus if we calculate a cost function table with N rows and N columns(for a stack of N),then we will know the influence

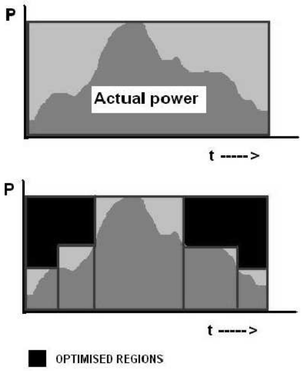

Fig. 5. Partitioning scheme

of each chip on the other. The scheduling algorithm makes use of the above table to determine the sequence of test.

The proposed algorithm is independent of the actual values in the cost function table. If a more accurate cost function is derived based on a more accurate thermal models in the future, the same generic algorithm can still be used for scheduling.

V. Test scheduling

We propose thermal aware test scheduling algorithms to sequence the test process of the above system. In these algorithms we try to schedule the chips within peak power and thermal constraints. The problem is stated formally below.

Problem formulation: Given a stack consisting of N chips with power profiles $P_1, P_2, ..., P_N$ and a well defined thermal model with all the interface resistances R_i, with P_{max} being the peak power constraint, find an efficient thermal aware scheme involving modeling and test-scheduling of chips such that the power constraint is not exceeded.

In this regard we look at two algorithms which use different heuristics and try to achieve the above goals.

The proposed partitioning scheme further sub-divides the power profile P_i into, what we call, power profile partitions. P_i thus consists of partitions $PP_1, PP_2, ..., PP_m$. The number of partitions is restricted to 5.

Thus the aim is to select the power profile partitions and schedule them without exceeding the power constraints. Since conflicting chips implies that the corresponding power profile partitions also conflict, we use the term conflict interchangeably. Also we assume that the chip is a single entity, even though it may consist of independent cores. This assumption is valid as the motivation is to reduce the variation of the temperature in the vertical direction rather than in the horizontal direction.

A. A Simplistic Approach to Thermal Aware Scheduling

In this scheduling scheme we take a strict thermal aware only approach where priority is given to only the thermal constraint. Here we assume that concurrent scheduling of conflicting chips cannot be done. The meaning of the term conflict is explained below Consider the top-most and the bottom most chips. They see little impedance on one of the arms, the outer, and hence most of the heat flows through the outer arm and they are not affected by any activity in the rest of the structure i.e. these chips do not conflict with any of the other chips in the stack. The chips at the center see almost equal impedances in the two arms and hence thermal activity anywhere affects the chip i.e., they conflict with all the chips in the stack. An other observation is that as we move away from the center the chips seem to become lesser and lesser susceptible to any activity on the stack. The plot of impedance of the chips as a function of N is shown in the Fig:7. With the help of the plot we define, around the center, a set of chips which conflict with all the chips in the stack S_i and also a complementary set of chips which only conflict with the chips which lie on the outer arm of the stack SC_i. For a stack of 5, referring to Fig:7 we have $S_i = 2, 3, 4$ and $SC_i = 1, 5$. Thus from the above concept we can prepare a conflict graph and use it for scheduling. A threshold must be chosen to segregate chips into S_i and SC_i . We do this by plotting the resistance in the two arms for all the chips and then define a variance around the point of intersection and then consider all the chips lying within this variance to have equal resistance, as shown in Fig: 7. The chips lying within the circle, conflict with all the chips in the stack. For any chip outside the circle of equal resistance the conflict is only with the chips lying in the path of lower resistance.

The scheduling begins by selecting one of the chip power profile partitions. The graphs for different chips are then run through to find out if the chip is under conflict. The set of partitions with no conflict are selected and scheduled after verifying for the power constraint condition. In this approach the most important task would be to obtain the conflict graph. The conflict graph for a 5 stack structure is shown in Fig: 6.

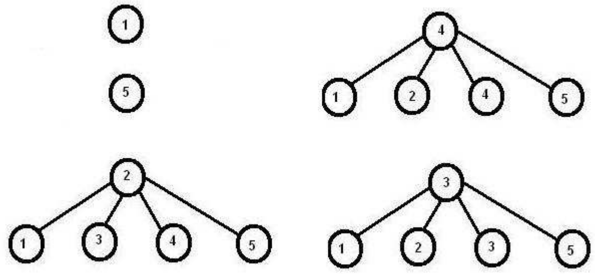

Fig. 6. Conflict graph for a five-stack structure.

For the conflict graph shown in Fig: 6 the schedule order can be 1 and 5, 2, 3 and finally 4.

The flow of the algorithm is as explained below.

```
for all $i$ such that $0 \leq i \leq N$ do
    Prepare a conflict graph
end for
for all $i$ such that $0 \leq i \leq N$ do
    Scale the power profiles according to the chip exposures
end for
for all $i$ such that $0 \leq i \leq N$ do
    Power profile is partitioned using the partitioning algorithm
end for
repeat
    Schedule list =NULL.
    Select one of the unscheduled power profile partitions and add to the schedule list.
    Total power=Peak power of selected partition.
    repeat
        Check for a non-conflicting power profile partition.
        Find the sum of the powers of the partitions selected
        Add the selected partition to the schedule list
    until Power constraint is exceeded
    Remove the last added partition from the schedule list.
    Schedule the partitions in the schedule and remove from the unscheduled partitions list.
until All the partitions are scheduled
```

This method considers the weights of the conflict graph as infinity. The next algorithm makes the weights finite by calculating the cost function. By doing so the complexity of the algorithm is increased but more concurrency and controllability can be brought in.

B. Least Intrusive Algorithm

In this approach we calculate the cost function for all the chips with respect to activity in the other chips and schedule those chips which satisfy the power constraint and also have the least total cost. Here for scheduling we consider both the power constraint and total cost for concurrently scheduled partitions. The maximum total cost provides controllability and can be adjusted depending on the requirements. We schedule a chip partition for testing and then select the partition which has the least cost with respect to the scheduled chip and check if it satisfies the power and the total cost constraints. If it does we schedule it and then look at chips with the same strategy (In the k^{th} sub-iteration i.e. when $(k-1)$ partitions have been selected for scheduling, the partition with the lowest cost considering all the selected partitions will be chosen). If the constrains are not satisfied, we go to the partition with the next least cost and check to see if it satisfies the constraints. Once this iteration is done we remove the tested partitions and then apply the same algorithm for the remaining partitions.

The cost function table for a 4 stack structure is shown in table 1.

The flow of the schedule algorithm is explained below.

TABLE I

TEST-COST OF CHIPS IN THE STACK $T_{i,j} * A_{i,j}$.

Die	1	2	3	4
1	$T_{1,1} * A_{1,1}$	$T_{2,1} * A_{2,1}$	$T_{3,1} * A_{3,1}$	$T_{4,1} * A_{4,1}$
2	$T_{1,2} * A_{1,2}$	$T_{2,2} * A_{2,2}$	$T_{3,2} * A_{3,2}$	$T_{4,2} * A_{4,2}$
3	$T_{1,3} * A_{1,3}$	$T_{2,3} * A_{2,3}$	$T_{3,3} * A_{3,3}$	$T_{4,3} * A_{4,3}$
4	$T_{1,4} * A_{1,4}$	$T_{2,4} * A_{2,4}$	$T_{3,4} * A_{3,4}$	$T_{4,4} * A_{4,4}$

```
for all $i$ such that $0 \leq i \leq N$ do
    Power profile is partitioned using the partitioning algorithm.
end for
for all $i$ such that $0 \leq i \leq N$ do
    Scale the power profiles according to the chip exposures.
end for
for all $i$ such that $0 \leq i \leq N$ do
    The cost function is calculated using the system variables.
end for
repeat
    Schedule list =NULL.
    Select one of the unscheduled power profile partitions and add to the schedule list
    Total power=Peak power of the selected partition.
    Total cost=0
    repeat
        Select a partition which has the lowest cost with respect to the partition(partitions) selected.
        Find the sum of the powers of the partitions selected
        Find the sum of the cost of the partitions selected
        Add the selected partition to the schedule list
    until Both power constraint and total cost are exceeded
    Remove the last added partition from the schedule list.
    Schedule the partitions selected and remove the scheduled partitions from the list.
until All the partitions are scheduled
```

VI. RESULTS

We have used ISCAS'85 benchmarks for each chip in the stack. The results have been obtained for power profile partitioning. The power profile is generated with random excitation. Actual Test vector patterns can also be used without any change. We have made use of combinational circuits as it is state independent, but sequential circuits can also be used without affecting the algorithm.

Fig: 9 demonstrates the power profile partition method described and shows the partitions obtained for C6288. An increase in efficiency(true power / total power) of 40.12% is obtained with the random excitation for C6288 benchmark. Table 2 shows the efficiency obtained for each of the benchmarks due to partitioning. The partition results are a trade off between efficiency and number of partitions. We can obtain higher efficiency with more number of partitions but the complexity increases linearly with the number of partitions and hence we put a cap on it.

TABLE II
EFFICIENCY INCREASE DUE TO PARTITION

BENCHMARK	EFFICIENCY
C7552	11.11%
C5315	12.52%
C1908	25%
C2670	14.31%
C6288	40.12%

For experimental purpose we have assumed a stack structure consisting of the above benchmarks. The order in the stack is the same as Table 2.

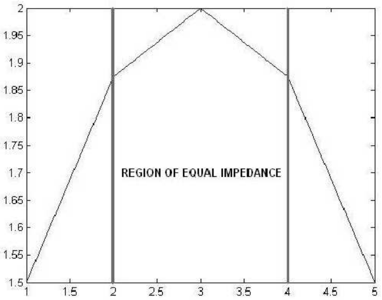

Fig. 7. Region of equal impedance for 5 stacked chips.

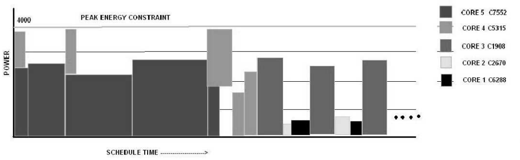

Fig. 8. Partial schedule order for algorithm 1

Fig: 7 shows the region of equal impedance for a stack of 5 chips, here the X-axis is the chip number(defined only for integer values, but has been extrapolated) and the Y-axis is the resistance . Fig: 8 shows the schedule of chips for simplistic scheduling algorithm. The shaded boxes are the power profile partitions and the x-axis is the time. From the schedule we can see that there is hardly any concurrency in the scheduling. Only chips 1 and 5 can be scheduled simultaneously, while the rest have to be done singly.

The results for Least Intrusive Algorithm are given in Fig: 10. The cost values for a stack of 5 are shown in Table 3. The schedule is shown in Fig: 10 and Table 5. The Table 5 gives the exact value of the iteration and power profile value at each iteration. The step value in the table refers to the iteration number. The fields in the next column are in the format(core(peak power,test time)) i.e. 1(x,y) means power partition of the core 1 with peak power of x and test duration of y(y is in terms of iterations).

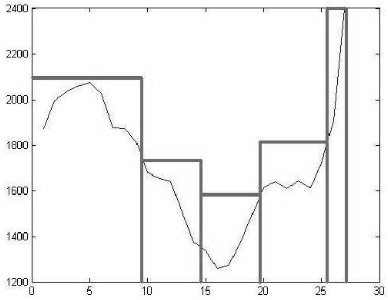

Fig. 9. Power profile partitioning result

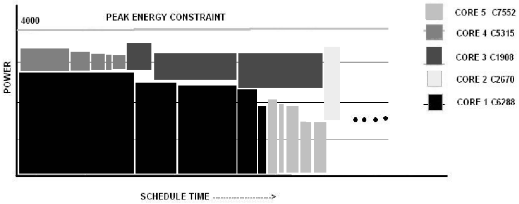

Fig. 10. Schedule using the least intrusive algorithm.

TABLE III
TEST-COST OF CHIPS IN THE STACK WITH RESPECT TO OTHERS.

Die	1	2	3	4	5
1	1.5000	0.9000	0.7000	0.5000	0.3000
2	1.5000	1.5000	1.0000	0.7500	0.5000
3	0.8333	1.1667	1.5000	1.1667	0.8333
4	0.5000	0.7500	1.0000	1.5000	1.5000
5	0.3000	0.5000	0.7000	0.9000	1.5000

TABLE IV
TEST SCHEDULE FOR LEAST INTRUSIVE ALGORITHM

Step	(Time)	Schedule
1	0	1(2722.8,96),4(742.24,40)
2	40	1(2722.8,96),4(696,15+40)
3	55	1(2722.8,96),4(663.76,21+55)
4	76	1(2722.8,96),4(563.24,5+76)
5	82	1(2722.8,96),4(526,11+82)
6	93	1(2722.8,96),3(851,19+93)
7	96	3(851,19+93),1(2620.1,18+96)
8	112	1(2620.1,18+96),3(633.84,43+112)
9	114	3(633.84,43+112),1(2456.8,41+114)
10	155	1(2431.8,155+6),3(606.84,30+155)

The direct comparison of these algorithms cannot be made with any existing algorithms as they are mostly for planar SOC, but the complexity is of the order $O(N)$ for the algorithm 1 and $O(N^2)$ for algorithm 2 which is fairly optimum for test scheduling algorithms.

VII. CONCLUSION

In this paper we have proposed two algorithms for efficient testing of stacked structures with importance given to thermal activity. We demonstrated two methods employing entirely unique methods to achieve thermal aware testing. We have generated a cost function based on the thermal parameters. The results show that the chip temperature is regulated as the chips are scheduled based on the cost function.

The two heuristics look at reducing the local heating, an other approach could be to look at the actual flow of heat in the stack during testing.

References

[1] Haixia Yan Qiang Zhou,XianlongHong,Efficient Thermal Aware Placement Approach Integrated With 3D DCT Placement Algorithm,9th International Symposium on Quality Electronic Design.

[2] Bruce Kim, Member, IEEE, Madhavan Swaminathan, Member, IEEE, Abhijit Chatterjee, Senior Member, IEEE, and David Schimmel, Member, IEEE,A Novel Test Technique for MCM Substrates,IEEE transactions on components, packaging, and manufacturing technology, vol 20, no 1,February 1997.

[3] L. Economikos, T. Morrison, and F. Crnic,Electrical test of multichip substrates,IEEE Trans. Comp., Packag., Manufact. Technol.,vol 17, page 56-61,February,1994

[4] C. Liu, K. Veeraraghavant, and V. Iyengar., Thermal-Aware Test Scheduling and Hot Spot Temperature Minimization for Core-Based Systems, 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT05), page 552-560,2005

[5] Zhiyuan He, Zebo Peng, Petru Eles, Paul Rosinger,BashirM,Al-Hashimi, Thermal-Aware SoC Test Scheduling with Test Set Partitioning and Interleaving, TUGboat, vol 12, no 2, page 291-301, June, 1991.

[6] E.Sasidhar Leon Alkalai A. Chatterjee,Test Strategies for a 3-D Stack Multichip Module Space Flight Computer, 1997 International Conference on Multichip Modules., 1997

[7] Andrs Poppe, Yan Zhang, Gbor Farkas, Hon Wong, John Wilson, Pter Szab, Thermal Characterization of Multi-Die Packages, Electronics Packaging Technology Conference, 2006

[8] Paul Rosinger, Bashir M. Al-Hashimi,Thermal-Safe Test Scheduling for Core-Based System-on-Chip Integrated Circuits. IEEE transactions on computer aided design of integrated circuits and systems, vol 25, no. 11, November 2006