Thermal-Aware Test Scheduling with Cycle-Accurate Power Profiles and Test Partitioning

Thomas Edison YU † Tomokazu YONEDA † Krishnendu CHAKRABARTY ‡ Hideo FUJIWARA †

† Graduate School of Information Science, Nara Institute of Science and Technology 8916-5 Takayama, Ikoma, Nara, 630–0101, Japan

‡ Electrical and Computer Engineering, Duke University, Box 90291, 130 Hudson Hall, Durham, NC 27708

Email: †{tomasu-y, yoneda, fujiwara}@is.naist.jp, ‡ krish@ee.duke.edu

Abstract Higher power densities and the non-linear spatial distribution of heat of VLSI chips put greater emphasis on chip-packaging and temperature control during test. For system-on-chips, power-based scheduling algorithms are used to optimize tests while satisfying power budgets. However, it has been shown that power-constrained test scheduling does not ensure thermal safety due to the non-uniform power distribution across the chip. In this paper, we present a test schedule optimization method for system-on-chips using cycle-accurate power profiles for thermal simulation, test partitioning, and interleaving that ensures thermal safety while still optimizing the test schedule. Our method uses a simplified thermal-cost model and bin-packing algorithm to ensure that the maximum temperatures of SoCs with fixed TAM and core assignments satisfy the temperature constraints with minimum increases in test application time.

Keyword SoC test, thermal constraint, wrapper design, TAM design, test scheduling

1. Introduction

The rapid advancement of VLSI manufacturing processes is being spurred on by the ever growing demand for faster, cooler, more power efficient and reliable chips. Because of this, higher power and heat density have also become a primary concern for the VLSI industry. This is especially true for newer System-on-Chips (SoCs) which integrates various functional cores into one chip while being designed for use in situations which demand low-power, low-heat operation such as in mobile devices. Furthermore, the problem of overheating becomes more significant during testing where there is a larger number of switching activity compared to functional operation due to the need for concurrently testing cores to minimize test time. Overheating can lead to problems such as increased leakage power, random errors and even permanent chip damage. In fact, since for every 15°C rise in temperature, there is approximately a 10-15% delay in timing, timing uncertainties can result in yield loss. Normally, engineers simply use better packaging and cooling methods to solve heating problems but this has become increasingly difficult and expensive. To reduce packaging cost, packages have increasingly been designed for the worst case *typical* application [10] and the cost of cooling during test can become impractical.

The most common design-for-testability technique for SoCs involves the use of a test delivery mechanism, called TAM (Test Access Mechanism) and module isolation circuitry, called *wrappers*. Approaches to optimize wrapper design and test schedules [1, 2, 3] and limiting test power [3, 4, 5] have been proposed. Still, because of the non-uniform spatial power distribution across the chip, limiting the maximum chip-level power dissipation does not ensure a reduction in localized heating (called *hot spots*) which occurs faster than chip-wide heating [7, 10]. Rosinger et al. [7] first proposed using a RC-based thermal model from [10], which takes advantage of the similarities between heat and electrical phenomena, as a basis for test scheduling instead of a chip-level power constraint. In [8], Liu et al. proposed algorithms which try to evenly spread heat over a chip using layout information and a progressive weighting function. In [9], He et al. proposed using test partitioning and interleaving to allow hot cores to cool off while freeing the test resources to test other cores and avoid overheating. For all previous methods, only fixed average power values per core and steady state temperatures were considered. In [13], we first proposed an integrated TAM/wrapper co-optimization and test scheduling method for SoCs with flexible TAM under a thermal constraint.

In this paper, we target SoCs with pre-designed TAM and core assignments, and propose a method to optimize SoC test schedules to satisfy a thermal constraint while keeping the test application time constant. The proposed algorithm combines a schedule rearrangement heuristic based on a simplified thermal-cost model and test partitioning and interleaving, since thermal simulation dominates the overall execution time for the algorithm. We utilize the HotSpot tool [11] for test schedule validation and instead of a fixed power dissipation value per core, we chose to assign a different power value per wrapper configuration. We also used the cycle-accurate power profiles from [5] to generate thermal profiles. To demonstrate the effectiveness of our approach, experiments were done using the ITC'02 benchmark SoC d695[6].

The rest of this paper is organized as follows. The motivation for this work is discussed in Section 2. Section 3 discusses the proposed test scheduling optimization algorithm. Section 4 gives the experimental results while Section 5 concludes this paper.

2. Motivation

For this paper, we focus on optimizing fixed-TAM SoCs by incorporating schedule rearrangement, test partitioning and interleaving with respect to cycle-accurate power and temperature data. As shown in Figure 1, we are assuming a Test Bus Architecture. This architecture assumes that the TAM is partitioned into fixed sub-buses and each IP core is assigned to one of these partitions. This allows for multiple independent test buses on one SoC as can be seen in Figure 1 for the d695 benchmark SoC. While cores cannot use TAM wires belonging to other partitions, cores belonging to the same partition can be tested sequentially in any order, as shown in Figure 2, where the test order of cores c2, c6, and c8 in TAM2 with bus width 8 varies the between the two schedules in Fig. 2(a) and Fig. 2(b). Our method takes advantage of this flexibility in order to derive a schedule with a lower temperature by rearrangement (10°C drop in Fig. 2(b)) and interleaving (additional 5°C drop by partitioning c5 into c5a and c5b, and interleaving them with c3 in Fig. 2(c)).

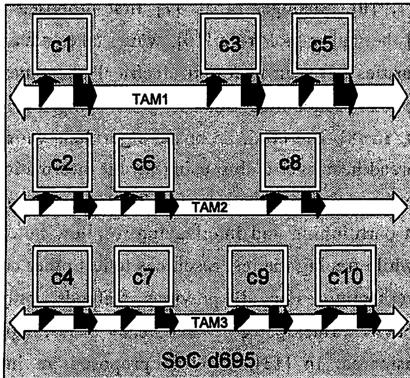
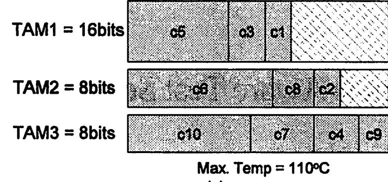


Figure 1. Example of Test Bus Architecture



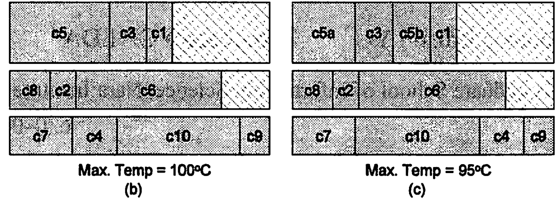


Figure 2. (a)Example Test Bus schedule, (b)after re-arrangement, (c) after test partitioning and

3. Test Schedule Optimization Problem

In this section, we formally present the test schedule optimization problem P_{OP}.

Problem P_{OP}: For an SoC S, given:

$Temp_{max}$: maximum allowed temperature during test

TAM_{cfg}: TAM and core configuration of the SoC which includes:

B: a set of TAMs

For each TAM $b_i \in B$ of S,

- W_i: allotted TAM width
- C_i: a set of cores
- For each core $c_j \in C_i$,
  - P_j: power profile
  - TAT_j: test application time
  - NP_{maxj}: maximum number of test partitions allowed

Determine the following output:

For each core c_j,

- NP_j: set of partitions of the test for c_j
- For each test partition $p_k \in NP_j$
  - $Tstart_k$: test start time
  - $Tend_k$: test end time

such that the temperature does not exceed $Temp_{max}$ while preserving the test application time.

3.1 Thermal Cost Function

The optimization algorithm should be able to decrease the number of thermal simulations needed as this often take a very

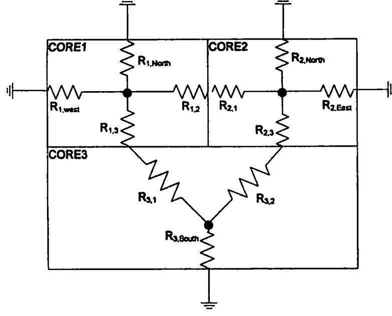


Figure 3. Lateral thermo-resistive model [7]

long time to finish and constitutes a very large part of the overall processing time. [7] proved that there exists a positive correlation between heat and heat dissipation paths represented by lateral thermal resistances, shown in Figure 3. In this work, we have chosen to use lateral thermal resistance as one of the basis for our model and cost function, with the following assumptions. First, since the TAM and wrapper configurations are fixed, minimizing the heat flowing into the core by schedule reconfiguration and interleaving is more practical instead of maximizing heat flow out of the target core as in [7, 13]. Secondly, only active cores contribute heat to other cores so we model the heat flowing into a core as the thermal resistance from the center of adjacent active cores to one of the edges of the hotspot core. Thus, if core 1 and core 2 are tested in parallel, heat flows to core 2 to from core 1 via $R_{1,2}$, and the opposite through $R_{2,1}$. As discussed in [13], we have to consider both the concurrency and precedence of the cores. Furthermore, the time dependence of temperature must also be considered. As in [13], we want to test hot cores with large power densities as short as possible and minimize their effects on other cores. This is done by avoiding concurrency and immediate precedence with cores in the immediate physical periphery of the hot spot core.

Since we are dealing with SoCs with a fixed TAM configuration (i.e. fixed partitioning and width) as well as fixed core distribution among the TAM partitions, the wrapper configuration and power profile for each core are already fixed. The problem of minimizing the hot spot temperature, therefore, becomes a problem of limiting the thermal contributions of the peripheral cores on the hotspot core. For this work, we express the thermal contribution of core c_j on core c_i as the thermal cost function is Equation 1.

In Eq. 1, we assume that the heat flowing from a core c_j to core c_i is proportional to the lateral resistance R_{ji} from the source to the destination core as well as the source's power dissipation, *Pavg*. Moreover, the more heat dissipation paths a source core has, the lesser the heat flowing through each lateral resistance and thus we divide the cost by $R_{TOT,j}$.

$$Tcont_j(c_i) = \frac{R_{ji}}{R_{TOT,j}} \times Pavg_j \times \frac{Trel_{ji}}{TAT_i} \quad (1)$$

$where: R_{ji}$: Lateral thermal resistance from core c_j to c_i, $(R_{ii} = 0)$

$R_{TOT,j}$: Total lateral resitance of core c_j

$Pavg_j$: Average power dissipatio n of c_j

$$Trel_{ji} := \begin{cases} TAT_j - (Tstart_i - Tend_j), \text{if } Tend_j < Tstart_i \\ TAT_j, \text{if } Tstart_i \leq Tend_j \leq Tend_i \\ TAT_j - (Tend_j - Tend_i), \text{if } Tstart_j < Tend_i < Tend_j \end{cases}$$

TAT_i : Test applicatio n time of c_i

$Tstart_i$: Test start time of c_i

$Tend_i$: Test end time of c_i

$Trel_{ji}$ expresses the weight we give on how the relative test times between two cores c_i and c_j affect their thermal contributions to each other and models the fact that the greater the time they have to affect each other, the greater the heat contribution of the cores to each other, but it is set to zero when the value becomes negative. From [13], we know that the average power dissipation gives a closer thermal profile curve to the actual thermal profile derived from cycle-accurate values compared to peak power values. Thus, instead of considering cycle accurate power, we chose to use average power values to greatly simplify cost calculations. The total thermal contribution of other cores to c_i for a certain schedule is computed as follows:

$$Tcont_{TOT}(c_i) = \sum_{j=1}^{N} Tcont_j(c_i) \quad (2)$$

$where: N$: Total number of cores of the SoC

The main idea is to reconfigure the test schedule such that the overall thermal contribution to the hot spot core is minimized to the point that the constraint is satisfied.

3.2 Test Scheduling Algorithm

Rectangular 2-D bin packing has been extensively used to solve the test scheduling problem for embedded cores. Each wrapper configuration of a core is represented by a rectangle whose height and width represents TAM width and test application time, respectively. The rectangles are packed into a bin with unbounded width, representing overall test time, and bounded height representing external TAM width. The aim is to find the optimal way of packing the rectangles such that the overall test application time (e.g. bin width) is minimized. For this paper, previous bin-packing algorithms cannot be directly applied since we cannot simply add the various temperatures of the cores to obtain the overall temperature of the SoC. Furthermore, since it has been shown that the restricted 3-D bin packing problem is *NP-Hard* in [4], this paper proposes a heuristic algorithm to solve the problem. Note that each core is represented by one rectangle since the TAM assignments and wrapper configurations are fixed.

Furthermore, each core rectangle can only be packed to the pre-assigned TAM.

Initialization: Bin Sorting and Initial Scheduling

The initialization steps first make sure that each core wrapper configuration satisfies the thermal constraint $Temp_{max}$. Each core c_i has a minimum cost $cost_min_i$ and maximum cost $cost_max_i$ (both initially set to infinity), a priority value Pr_i (initially set to *NULL*), and a temporary cost to determine potential hot spot cores, $cost_tmp_i$, computed for each core using Equation 3, where $Area_i$ is the surface area of c_i. It is supposed in Eq. 3 that the core with the highest power density and/or longest test time has the potential to be the hottest core during the test. The core rectangles are then sorted in descending order from the core with the highest *cost_tmp*.

$$cost_tmp_i = \frac{Pavg_i}{Area_i} \times (TAT_i) \quad (3)$$

The bin is first divided into $|B|$ sub-bins representing each TAM partition, b_i ($1 \leqq i \leqq |B|$). The rectangles are packed into their respective pre-assigned sub-bins (e.g. TAM partitions) according to their *cost_tmp*. Thermal simulation is then performed, after all rectangles have been packed, on the finished schedule to determine the hottest core, c_{Hmain}, using the HotSpot simulator developed in [12]. The algorithm ends if the hottest core does not exceed $Temp_{max}$. If it does exceed the constraint, its priority value Pr_{Hmain} is set to 1, where a lower priority value denotes a hotspot core with a greater priority. Priorities determine which cores take precedence when minimizing the thermal cost during the following steps. The overall thermal contribution $Tcont_{TOT}(c_{Hmain})$, to the original hotspot core is computed and set as $cost_max_{Hmain}$.

Step 1: Re-packing Rectangles with Thermal Cost Minimization for Main Hotspot Core

Before re-packing, the algorithm again re-sorts the rectangles according to their *cost_tmp*. The algorithm first packs the hotspot core c_{Hmain} with the highest priority (Pr_{Hmain} = 1) into its assigned sub-bin.

It then looks for the core rectangle, c_p, with the highest *cost_tmp* value but the smallest thermal contribution, $Tcont_p(c_{Hmain})$ to c_{Hmain}. Next, the rectangle representing c_p is packed into its pre-assigned TAM partition. This search and packing routine is repeated until all the core rectangles have been packed. After packing the all the rectangles, another thermal simulation is performed and the hottest core, c_{Hnew} is determined. The algorithm finishes if the temperature of c_{Hnew} satisfies $Temp_{max}$. If not, c_{Hnew} is checked to see if it is the same hotspot core, c_{Hmain}, before re-packing (e.g. Pr_{Hnew}=1). If it is the same core, the algorithm jumps to Step 3, otherwise, the overall thermal contribution, $Tcont_{TOT}(c_{Hmain})$, to the original hotspot core is computed and set as $cost_min_{Hmain}$. The priority value, Pri_{Hnew}, of c_{Hnew} is set to $Pri_{Hmain} + 1$ and the algorithm proceeds to Step 2.

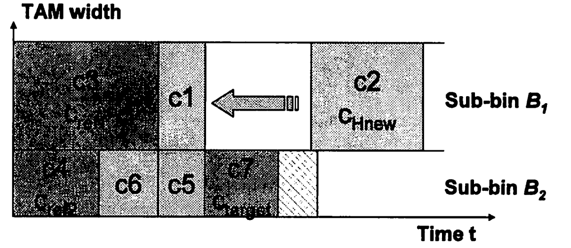


Figure 4. Packing cores with respect to reference and target cores (Step 2)

Step 2: Re-packing with Multiple Hotspot Cores

In this step, we designate the set of scheduled cores with a fixed *cost_min* as reference cores, C_{ref}, and the target hotspot core to be cooled-down as c_{target}, which is initially set to c_{Hnew}. All the sub-bins are again emptied and the cores are sorted into two lists: all rectangles with $Pr \neq NULL$ are put into L_{pri}, the rest into L_{rst}. L_{pri} is sorted in ascending priority value while L_{rst} is sorted according to decreasing *cost_tmp*. L_{rst} is then concatenated to the end of L_{pri} to form a master list, L_{master}.

We first set the core with the lowest priority value as $c_{next} \cup C_{ref}$ and pack it first into its sub-bin.

Step 2.1:

Choose the next rectangle c_{next} in L_{master} and

i. if c_{next} has a fixed $cost_min>0$, then $c_{next} \cup C_{ref}$

ii. else if $Pr_{next} > Pr_{target}$, set c_{next} as new c_{target}

We then compute the current thermal contribution of this new core and the scheduled cores, $Tcont_{current}(c_i)$ for each $c_i \in C_{ref}$. If $Tcont_{current}(c_i) \leqq cost_max_i$ for each $c_i \in C_{ref}$, then compute its thermal contribution to c_{target}, $Tcont_{next}(c_{target})$. Do this for all possible c_{next} in L_{master}, then pack the rectangle with the lowest $Tcont_{next}(c_{target})$. If no rectangle can be found, revert to the schedule at the end of Step 1 and go to Step 3. Otherwise, continue doing Step 2.1 until all rectangles have been packed and go to Step 2.2. In Figure 4, before packing c2, the overall cost of the reference cores c3 and c4 must still satisfy their maximum values while contributing as little as possible to the new target hotspot core, c7.

Step 2.2:

Perform thermal simulation on the finished schedule and check if $Temp_{max}$ is satisfied. If not, determine the hottest core c_{Hnew} and

i. if $c_{Hnew} \in C_{ref}$, and $cost_max_{Hnew}*(1-\beta) \geqq cost_min_{Hnew}$ (β is a heuristic percentage value), then decrease $cost_max_{Hnew}$ by β percent and go to

Step 2.

ii. else if $c_{Hnew} \in C_{ref}$, and $cost_max_{Hnew}*(1-\beta) < cost_min_{Hnew}$, then revert to the schedule at the end of Step 1 and go to Step 3.

iii. else if $c_{Hnew} \equiv c_{target}$, revert to the schedule at the end of Step 1 and go to Step 3.

iv. else determine the core $c_{min} \in C_{ref}$ with the highest priority value, Pr_{min} and set $Pr_{Hnew} = Pr_{min}+1$. Then $Tcont_{TOT}(c_{target}) = cost_max_{target}$, and go to Step 2.

Step 3: Test Partitioning and Interleaving

The algorithm takes note of the time t_{HOT} when the temperature of the hottest core $Temp_{Hnew}=Temp_{max}$, then the test of c_{Hnew} is partitioned at t_{HOT} as long as $NP_{Hnew} \leq NP_{Hnewmax}$. The rectangle representing c_{Hnew} is essentially split into two, creating two new virtual cores, $c_{Hnew,1}$ and $c_{Hnew,2}$ replacing c_{Hnew} and having the following characteristics:

i. $c_{Hnew,1}$ retains the priority and cost limits of c_{Hnew} while $c_{Hnew,2}$ is treated as a completely new core

ii. $c_{Hnew,1}$ and $c_{Hnew,2}$ has infinite thermal contribution to each other which prevents them from being scheduled right next to each other

iii. $c_{Hnew,2}$ can only be scheduled after finishing the test of $c_{Hnew,1}$ and this holds for further partitions of c_{Hnew}

The algorithm updates the core list and returns to Step 2, but this time with the added precedence constraint of the partitions. Furthermore, all cores there were active on or before t_{HOT} will retain the previous schedule configuration as shown in Figure 5. It terminates when the constraint if satisfied, no more tests can be partitioned, or when successive partitioning and interleaving does not yield any drops in temperature.

4. Experimental Results

The experiments were done using one SoC from the ITC'02 SoC Benchmark suite [6], d695. For thermal simulation, cycle-accurate power profiles provided by the authors of [5] were used. Note that the actual power profiles were originally expressed as number of transitions per clock cycle. We converted the values into Watts by simply dividing them by 20 to reflect power dissipation during test. Experiments were done using an HP ProLiant Workstation with 4 Opteron CPU's operating at 2.4GHz with 32GB of memory.

Since the original SoC benchmarks did not include layout information, we handcrafted the layout of the SoC. Further more, a set of 10 different TAM and core configurations, shown in Table 1, were designed for the experiments. In Table

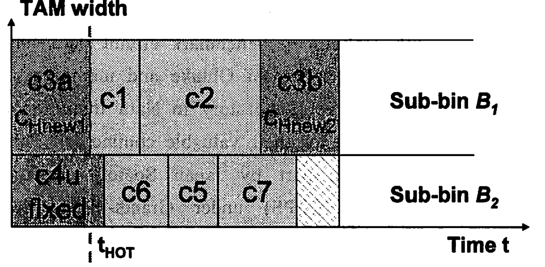


Figure 5. Partitioning and interleaving of cores

1, information on the total TAM width (W_{tot}), the number of partitions, the overall test time (TAT), and the maximum temperatures ($maxT$) from the initialization step of the algorithm are given. The TAM width (W_i) of each TAM partition as well as their member cores ($W_i(cores)$) are also given in no particular order. It is assumed that each core can be partitioned a maximum of 3 times.

The experimental results for each schedule are shown in Table 2. We set the temperature constraint, $Temp_{max}$, at an initial value of 125°C and decreased it by 5°C intervals. Here $maxT$ is the actual maximum temperature of a schedule from thermal simulation. Temperatures due to partitioning and interleaving are grayed-out to distinguish it from results due to simple schedule re-arrangement. It can be seen from the results for schedule 4 that temperature can be decreased by as much as 10°C through re-arrangement alone. Furthermore, greater temperature drops can be achieved when applying partitioning and interleaving, with an average drop of around 10.7°C (around 9% average). Specifically, a maximum temperature drop of 15.6°C or 14% of the highest temperature for schedule 4 was achieved.

5. Conclusion

In this paper, we have presented a thermal-aware test schedule optimization algorithm for system-on-chips with fixed-width TAMs that ensures thermal safety while preserving the test application time. The proposed method allows us to further explore, beyond the limits of peak-power based test scheduling, possible variations of a schedule which can lead to further reductions in temperature using test reconfiguration, partitioning and interleaving. Using cycle-accurate power profiles per wrapper configuration and considering both the spatial and temporal dimensions of heat transfer, overall, allows us to more closely approximate real world thermal phenomena.

Acknowledgements

The authors would like to thank Dr. Erik Larsson of Linköping University, Sweden, for providing us with the

power profiles for the benchmark SoCs, Dr. Paul Rosinger for providing us with preliminary benchmark circuit data, and Prof. Michiko Inoue, Dr. Satoshi Ohtake and members of Computer Design and Test Laboratory in Nara Institute of Science and Technology for their valuable comments.. This work was supported in part by Japan Society for the Promotion of Science (JSPS) under Grants-in-Aid for Scientific Research B(No.15300018) and for Young Scientists(B) (No.18700046).

References

[1] E. J. Marinissen, S. K. Goel and M. Lousberg, "Wrapper Design for Embedded Core Test," *Proc. of IEEE International Test Conference (ITC)*, pp. 911-920, 2000.

[2] S. K. Goel and E. J. Marinissen, "Effective and Efficient Test Architecture Design for SoCs," *Proc. of IEEE International Test Conference (ITC)*, pp. 529-538, 2002.

[3] V. Iyengar, K. Chakrabarty and E. J. Marinissen, "Test Access Mechanism Optimization, Test Scheduling, and Tester Data Volume Reduction for System-on-Chip," *IEEE Trans. On Computers*, vol. 52, no. 12, pp. 1619-1632, December 2003.

[4] Y. Huang et al., "Optimal Core Wrapper Width Selection and SOC Test Scheduling Based on 3-D Bin Packing Algorithm," *Proc. of IEEE International Test Conference (ITC)*, pp. 74-82, 2002.

[5] S. Samii, E. Larsson, K. Chakrabarty and Z. Peng, "Cycle-Accurate Test Power Modeling and its Application to SoC Test Scheduling," *Proc. of IEEE International Test Conference (ITC)*, pp. 1-10, 2006.

[6] E. J. Marinissen, V. Iyengar and K. Chakrabarty, "A Set of Benchmarks for Modular Testing of SoCs," *Proc. of IEEE International Test Conference (ITC)*, pp. 519-528, 2002.

[7] P. Rosinger, B. Al-Hashimi and K. Chakrabarty, "Rapid Generation of Thermal-safe Test Schedules," *Proc. Design, Automation and Test in Europe (DATE) Conf.*, pp. 840-845, 2005.

[8] C. Liu, K. Veeraraghavan and V. Iyengar, "Thermal-Aware Test Scheduling and Hot Spot Temperature Minimization for Core-Based Systems," *Proc. of the 20th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'05)*, pp. 552-560, 2005.

[9] Z. He et al., "Thermal-Aware SoC Test Scheduling with Test Set Partitioning and Interleaving," *Proc. of the 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT'06)*, pp. 477-485, 2006.

[10] K. Skadron et al., "Temperature-aware Microarchitecture," *Proc. of the 30th Annual Int'l Symposium on Computer Architecture (ISCA'03)*, pp. 2-13, 2003.

[11] M. R. Stan et al., "Hotspot: A dynamic compact thermal model at the processor-architecture level," *Microelectronics Journal: Circuits and Systems 34*, pp. 1153-1165, Dec. 2003.

[12] S. Lee et al., "Constriction/Spreading Resistance Model for Electronics Packaging," *Proc. of ASME/JSME Thermal Engineering Conf.*, pp. 199-206, Mar. 1995.

[13] T. Yu, T. Yoneda, K. Chakrabarty and H. Fujiwara, "Thermal-Safe Test Access Mechanism and Wrapper Co-optimization for System-on-Chip," Proc. of the 16th IEEE Asian Test Symposium (ATS'07), pp. 187-192, Oct. 2007.

Table 1. TAM configuration and core assignments for d695 SoC

schedule#	W_{tot}	*#partitions*	W_1	W_1 *(cores)*	W_2	W_2*(cores)*	W_3	W_3*(cores)*	W_4	W_4*(cores)*	*TAT*	*maxT(°C)*
1	32	2	16	4,5,9,10	16	1,2,3,6,7,8	0	∅	0	∅	27401	101.29
2	32	2	24	4,5,9,10	8	1,2,3,6,7,8	0	∅	0	∅	41002	91.23
3	32	2	8	4,5,9,10	24	1,2,3,6,7,8	0	∅	0	∅	48668	110.06
4	32	3	16	1,3,5	8	2,6,8	8	4,7,9,10	0	∅	32847	110.55
5	32	3	8	1,3,5	16	2,6,8	8	4,7,9,10	0	∅	32847	120.1
6	32	3	8	1,3,5	8	2,6,8	16	4,7,9,10	0	∅	30089	115.09
7	32	4	8	5,7	8	6,9	8	1,2,3,8	8	4,10	32504	121.13
8	32	4	16	5,7	4	6,9	4	1,2,3,8	8	4,10	53337	110.46
9	32	4	16	5,7	8	1,2,6	4	3,8,9	4	4,10	37222	112.54
10	32	4	8	5,7	16	1,2,6	4	3,8,9	4	4,10	37222	124.57

Table 2. Experimental results for various TAM configurations under thermal constraint

Temp$_{max}$	*maxT(°C) of schedule #*									
(°C)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
125	101.29	91.23	110.06	110.55	120.1	115.09	121.13	110.46	112.54	124.57
120	101.29	91.23	110.06	110.55	112.62	115.09	119.98	110.46	112.54	119.99
115	101.29	91.23	110.06	110.55	112.62	114.98	114.97	110.46	112.54	114.97
110	101.29	91.23	110	100.52	109.99	109.99	109.99	107.44	104.95	na
105	101.29	91.23	104.99	100.52	104.96	na	na	104.98	104.95	na
100	99.7	91.23	99.98	99.99	na	na	na	99.99	99.97	na
95	94.99	91.23	na	94.97	na	na	na	na	na	na
90	89.9	89.97	na	na	na	na	na	na	na	na
85	na	84.94	na	na	na	na	na	na	na	na
80	na	na	na	na	na	na	na	na	na	na