14th Asia and South Pacific Design Automation Conference (ASP-DAC 2009), pp.793-798, Jan. 2009.

Test Infrastructure Design for Core-Based System-on-Chip Under Cycle-Accurate Thermal Constraints

Thomas Edison Yu†, Tomokazu Yoneda†, Krishnendu Chakrabarty‡ and Hideo Fujiwara†

† Graduate School of Information Science, Nara Institute of Science and Technology, Kansai Science City, 630–0192, Japan

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

† E–mail:*{tomasu-y, yoneda, fujiwara}@is.naist.jp*, Tel.:+81-743-72-5222, Fax:+81-743-72-5229

‡ E–mail:*{krish}@ee.duke.edu*, Tel: +1 (919) 660-5244, Fax : +1 (919) 660-5293

Abstract

We present a thermal-aware test-access mechanism (TAM) design and test scheduling method for system-on-chip (SOC) integrated circuits. The proposed method uses cycle-accurate power profiles for thermal simulation; it also relies on test-set partitioning, test interleaving, and bandwidth matching. We use a computationally tractable thermal-cost model to ensure that temperature constraints are satisfied and the test application time is minimized. Simulation results for the ITC'02 SOC Test Benchmarks show that, compared to prior thermal-aware test-scheduling techniques, the proposed method leads to shorter test times under tight temperature constraints.

Keywords

SoC test, TAM design, test scheduling, thermal-aware test, wrapper design.

1. INTRODUCTION

Rapid advances in recent years in semiconductor manufacturing processes and design tools have led to a relentless increase in chip complexity. Greater on-chip functionality has also heightened the demand for faster processors and higher integration levels. As a result, high power consumption and heat densities are major concerns for the semiconductor industry. This problem is greatly exacerbated for system-on-chip (SoC) integrated circuits, which integrate several (and often heterogeneous) functional cores on one chip. While overheating is a serious problem for SoCs in normal functional mode, there is even greater power consumption (and therefore heat dissipation) in test mode. It is well-known that switching activity during test can be several times higher than in functional mode due to concurrent-testing [4]. Moreover, to reduce test time, and therefore, test cost, test scheduling is used to increase concurrency for SOC testing. As a result, there is significantly higher switching activity during test application for core-based SOCs.

Overheating can lead to several problems such as increased leakage power and thermal runaway, soft errors, and even permanent chip damage. Furthermore, for every 20°C rise in temperature, there is approximately a 5-6% delay in timing, which can result in yield loss [12]. One solution to this problem is to use more expensive packaging and cooling methods; however, this solution leads to higher cost in an increasingly cost-sensitive market for SoCs. To reduce packaging cost without limiting performance, packages have increasingly been designed for the worst-case *typical* application [9]. Since the thermal-management system of a chip is designed around this package, and since such details are not immediately accessible during test development, these solutions can make at-speed tests impractical, increase overall test time, or lead to higher cooling cost during test.

Until recently, lowering of the test power has been advocated as an effective method for avoiding overheating during test application. Since a widely-studied design-for-testability technique for SoCs involves the use of a test delivery infrastructure, consisting of a *test-access mechanism* (TAM) and module isolation circuitry (called a *wrapper*), several methods have been proposed for wrapper design, TAM optimization, and test scheduling under test-power constraints [1-4]. However, due to the non-uniform spatial power distribution across the chip, setting a limit on the maximum chip-level power consumption does not ensure a reduction in localized heating (referred to as *hot spots*). It has been widely reported that hot spots are more of a concern than chip-wide heating [6, 9], since they lead to stress-related reliability problems. Moreover, it has been shown in [9] that there is a need for a temperature-based model for thermal management. Due to the effects of thermal capacitance, the correlation between actual temperature and chip power consumption is quite low in practice. Thus, a thermal-aware TAM/wrapper co-optimization and test scheduling method for SoCs was presented in [10].

In this paper, we present a technique for TAM optimization and test scheduling for core-based SoCs under thermal constraints. We assume a fixed-width TAM architecture, as in [1], and we consider test-set partitioning [8] and bandwidth matching [11] to derive more effective solutions. The main contributions of this work are as follows:

1) We study the impact of test-set partitioning and bandwidth matching on thermal-aware TAM optimization and test scheduling. Cycle-accurate power profiles are used for each wrapper configuration of an embedded core.

2) A computationally tractable thermal-cost model is used as the basis for an optimization algorithm for SoC TAM design and test scheduling. We minimize the SoC test time under thermal constraints.

3) Detailed simulation results are presented for two ITC'02 Benchmark SoCs. The results show that (i) the test application time obtained using the proposed method is in most cases less than that using [10]; (ii) the proposed method provides solutions even under tight temperature constraints, including situations where [10] fails to find a solution.

The rest of this paper is organized as follows. Additional motivation for this work, an overview of related prior work, and some key aspects of the proposed method are presented in Section 2. Section 3 describes the proposed TAM optimization and test scheduling method. Section 4 presents simulation results, including a detailed comparison with prior work. Finally, Section 5 concludes the paper.

2. LIMITATIONS OF RELATED PRIOR WORK

Rosinger et al. [6] first proposed the use of an RC-network based thermal model (based on [9]) for SoC test scheduling. This work draws upon the analogy between heat transfer and electrical current flow, which serves as the basis for test scheduling under thermal constraints. In [7], Liu et al. proposed scheduling algorithms that attempt to evenly spread heat over a chip using layout information and a progressive weighting function. In [8], He et al. proposed the use of test-set partitioning and test interleaving to allow hot cores to cool (while test resources are used to exercise other cores) and thereby avoid overheating. A drawback of all the above methods is that they consider *fixed* average power values per core and steady-state temperatures. Such an assumption is too restrictive in practice due to the temporal and spatial variation of hot spots and chip temperatures [9]. Fur-

978-1-4244-2749-9/09/$25.00 ©2009 IEEE

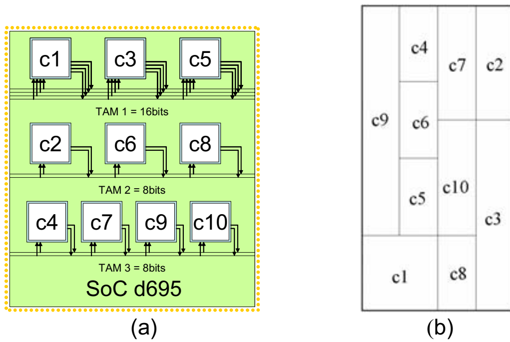

Figure 1. (a) Example Test Bus architecture and (b) layout for d695 SoC

thermore, it was assumed that temperature influences between cores are negligible, which was shown in [10] to be too optimistic and unrealistic.

A thermal-aware TAM/wrapper co-optimization and test scheduling method for SoCs with a flexible-width TAM architecture was presented in [10]. This approach uses cycle-accurate power profiles for accurate thermal simulation. The computation time for test scheduling is reduced by the use of a computationally tractable thermal-cost model which considers the thermal effects between cores, and a heuristic bin-packing algorithm for test scheduling. Simulation results in [10] showed that, while the proposed solution is useful in many situations, especially for wide SOC-level TAMs, it is relatively ineffective under tight thermal constraints and narrow TAM widths.

2.1 Test-Schedule Reshaping, Test-Set Partitioning, Test-Interleaving, and Bandwidth Matching

In this section, we incorporate test-schedule reshaping, test-set partitioning, test interleaving, and bandwidth matching techniques with respect to cycle-accurate power and temperature data. As shown in Figure 1(a), we assume that a Test Bus architecture, as in [1], is used for the target SoC. This architecture assumes that the TAM is partitioned into several fixed width test buses and each core is assigned to one of these partitions, as illustrated in Figure 1 for the d695 benchmark SoC.

To show the effects of test-schedule reshaping and the importance of considering temperature effects between cores, consider the example floor plan for the d695 SOC with the ten cores laid out as shown in Figure 1(b). Given a TAM architecture and core assignment shown in Fig. 1(a), the test schedule in Figure 2(a) yields a maximum temperature of 110°C using the HotSpot temperature simulation tool presented in [9]. During test re-shaping, by reordering the test of cores c2, c6, and c8 on TAM2, and c7, c4, c10, and c9 on TAM3, as shown in Fig. 2(b), we are able to decrease the temperature to 100°C. This is because the new schedule avoids the concurrent testing of c5 with c6 and c10, which are placed next to each other and are the hottest, 2nd-hottest, and 3rd-hottest cores, respectively. Furthermore, partitioning c5 into c5a and c5b and interleaving them with c3 in Fig. 2(c) leads to an additional 5°C drop. Note that in [8], temperature simulations were done for each test per core to determine the partitioning and cooling periods prior to actual scheduling. Simulating the interleaved test of core 5 and 10 (Fig. 3(a)), the thermal profile for core 5 (Fig. 3(b)) shows that ignoring inter-core effects [8] and/or using fixed power profiles is too optimistic and only cycle-accurate thermal simulation will yield realistic results. For this work, partitioning and interleaving are done during scheduling, which ensures more realistic thermal profiles.

Under very tight temperature constraints, we propose using bandwidth-matching circuitry to significantly reduce test temperature. Frequency throttling has been combined with bandwidth matching circuitry and virtual TAM techniques in [11] to reduce dynamic power while minimizing the increase in test application time. Given an ATE frequency f_{ATE} with n TAM wires and a target virtual TAM frequency $freq(b_i)$, by inserting a pair of demultiplexing (DeMUX) and multiplexing (MUX) circuitry between the

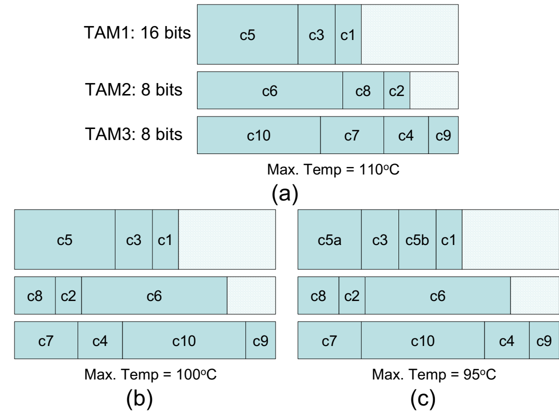

Figure 2. (a) Example test schedule, (b) after reshaping, (c) after test partitioning and interleaving

ATE and the internal TAM b_i and increasing the number of virtual TAM wires to $[f_{ATE} / freq(b_i)]*n$ bits, we can reduce the virtual TAM frequency (and therefore, power consumption) while minimizing the increase in test time. To simplify the clock generation circuitry, we assume that only repeated halving of the virtual test bus frequency (thereby doubling the virtual-bus wire count) is allowed. Since increasing the virtual TAM allotted to a core does not always results in a test time reduction, repeatedly halving the frequency has a best case scenario of 50% power reduction without sacrificing test time. The overall power reduction can lead to a significant drop in temperature during test.

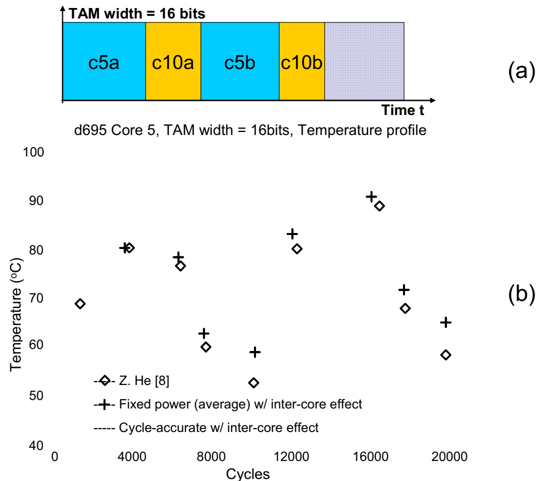

Figure 3. (a) Interleaved schedule for core 5 and 10, (b) thermal profile of core 5

3. TAM DESIGN AND TEST SCHEDULING

In this section, we formally present the TAM design and test scheduling problem P_{SKED}.

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

W_{ext}: TAM width allotted to the SoC

N_C: a set of cores belonging to S

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

For each core c_i $(1 \le i \le |N_C|)$ of SoC S

- $Wset_i$: number of usable wrapper configurations
- NP_{maxi}: maximum number of test partitions allowed
- For each wrapper configuration w_{ij} $(1 \le j \le Wset_i)$
  - TAM_{ij}: allotted TAM width

- P_{ij}: power profile
- TAT_{ij}: test application time

Our goal is to determine the following:

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 belonging to b_i
  - For each core $c_j \in C_i$,
    - 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 the test application time is minimized.

3.1 Basic Strategy

Our basic strategy for TAM design and test scheduling involves four main steps.

1. During the initialization step, the algorithm determines an initial TAM design and optimal test schedule for the SoC under no thermal constraint and determines the hottest possible core.
2. During test reshaping, the schedule is rearranged to minimize the temperature of the hotspot core. The new schedule undergoes another thermal simulation.
3. If the temperature constraint is not satisfied in Step 2, test partitioning is performed on the hottest core. Steps 2 and 3 are repeated until the test for the hottest core can no longer be partitioned. Note that interleaving is done during the reshaping stage.
4. If the previous step fails, the partitions of test for the hottest core are recombined and bandwidth matching circuitry is inserted on the TAM where the hotspot core belongs. Steps 2 to 4 are repeated until the constraint is satisfied or the virtual TAM width limit is reached.

Note that cycle-accurate thermal simulation is performed to check the test temperature every time the schedule is reshaped. In reality, this accounts for almost all the processing time. Note also that exploration of all possible schedule arrangements, partitioning and virtual TAM configurations is virtually impossible. Thus, we propose a simplified thermal cost function which will give us an idea of the heating phenomena during test without resorting to thermal simulation. This also serves as the basis for the heuristic test scheduling algorithm to minimize the thermal simulation effort and overall computation time.

3.2 Thermal Cost Function

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 during the scheduling step. 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. We assume that the thermal contribution of core c_j on core c_i for a given schedule depends on the following three parameters: 1) the average power consumption of the core c_j, 2) the thermal resistance between c_j and c_i proposed in [6], and 3) the relative test times between c_j on core c_i. It was established in [6] that there exists a positive correlation between heat and heat dissipation paths represented by lateral thermal resistances, shown in Figure 3. Thermal resistance is directly proportional to the thickness of the material and inversely proportional to the cross-sectional area across which the heat is being transferred [9]. For this work, we express the thermal contribution of core c_j on core c_i for a test schedule as the thermal cost function below:

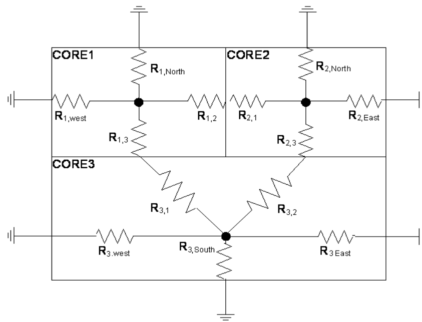

Figure 4. Lateral thermo-resistive model [6]

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

where R_{ji} is the lateral thermal resistance from core c_j to c_i (R_{ii}=0), $R_{TOT,j}$ is the total lateral resistance from core c_j, and $Pavg_j$ is the average power dissipation of c_j. Moreover, the parameter $Trel_{ji}$ is defined as follows:

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

where TAT_i is the test application time of c_i, $Tstart_i$ is the test start time of c_i, and $Tend_i$ is the test end time of c_i. In Equation (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 less heat flowing through each lateral resistance. Therefore, we divide the cost by $R_{TOT,j}$. The parameter $Trel_{ji}$ expresses the weight we give on how the relative test times between the 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 [10], it has been shown that using fixed average power values, instead of peak power values, for thermal simulation gives a closer approximate of the thermal profile curve derived from cycle-accurate 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 in Equation (2), where N is the total number of cores of the SoC. The main idea is to use this information 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.

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

3.3 Heuristic TAM Design and Test Scheduling Algorithm

Each step of the proposed TAM design and test scheduling algorithm is explained in detail in the following sub-sections.

Step 1: Initial TAM Design, Bin Sorting, and Initial Scheduling

Among TAM design algorithms, TR-Architect [1] has been shown to be one of the most efficient algorithms for determining TAM partition and core assignments. For this work, we utilize this algorithm to determine an initial TAM design and core assignment that minimizes the test application time without any power or thermal constraints during the initialization step. Then, the algorithm makes sure that each core wrapper configuration satisfies the thermal constraint $Temp_{max}$. Each core c_i has a minimum thermal cost $cost_max_i$ (initially set to 0) 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 assumed in Equation (3) that the core with the highest power density and/or longest test time has the potential to be the hottest core during test. Each core is represented as a rectangle, where the height represents allotted TAM width and the length represents

test application time. The rectangles are then sorted in descending order from the core with the highest *cost_tmp*.

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

During initial scheduling, an empty bin, whose height and width represent total external TAM width and overall test time, respectively, is first divided into $|B|$ sub-bins representing each TAM partition, b_i $(1 \le i \le |B|)$. 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_{HOT}, and the time when the temperature is equal to $Temp_{max}$, t_{HOT}, using the HotSpot simulator developed in [9]. The algorithm ends if the hottest core does not exceed $Temp_{max}$. Otherwise, it proceeds to Step 2.

Step 2: Cost-constrained Schedule Reshaping

In this step, the algorithm rearranges the current test schedule to minimize the cost of the current hottest core, c_{HOT}, without increasing the costs of previous hotspot cores (called reference cores) belonging to the set C_{ref}, which is initially empty.

For each TAM partition designed in Step 1, the cores are sorted in descending order of $cost_tmp_i$ to determine potential hot spot cores. It then looks for the core c_{target} with highest *cost_tmp* value but with the smallest thermal contribution to c_{HOT}. For each core c in C_{ref}, if the new cost of c, $Tcont_{TOT}(c)$ after packing c_{target} does not exceed its maximum cost constraint, we pack c_{target} into its assigned TAM partition. If no core can be found, revert to the schedule at the beginning of this step and go to Step 3. Otherwise, continue this step until all cores have been packed. In Figure 4, when packing the new target core, c2, the new costs of previous hot spot cores, c3, c4, and c7 must be checked and they must not exceed their *cost_max* values.

After packing all cores, thermal simulation is performed to determine the new hottest core c_{new}. The algorithm finishes if the temperature of c_{new} satisfies $Temp_{max}$. If the temperature of c_{new} does not satisfy the thermal constraint and c_{new} already belongs to $C_{ref} \cup \{c_{HOT}\}$, then we proceed to Step 3. Otherwise, add c_{HOT} to C_{ref}, and set $c_{HOT} = c_{new}$, and we repeat Step 2.

Step 3: Test Partitioning and Interleaving

The algorithm takes note of the time t_{HOT} when the temperature of the hottest core c_{HOT} is equal to $Temp_{max}$. Then, the test of c_{HOT} is partitioned at t_{HOT} into two tests (creating two new virtual cores c_{HOT1}, c_{HOT2}) as long as the number of test partitions for c_{HOT} does not exceed NP_{max}. The algorithm updates the core list and returns to Step 2, but this time with an added precedence constraint that the partition c_{HOT2} can only be scheduled after finishing the test of c_{HOT1}. Furthermore, the schedules of all other cores that were active on or before t_{HOT} remain unchanged so that the temperature profile up to this time is preserved. If the test of c_{HOT} can no longer be partitioned, the algorithm proceeds to Step 4. In Figure 5, core c3 is partitioned and c1 and c2 are inserted between the two partitions during scheduling during reshaping. Also, the schedule of c4 remains unchanged since it was active at time t_{HOT}.

Step 4: Bandwidth Matching Circuitry Insertion

In this step, bandwidth matching circuitry is added to the TAM partition where the target hot spot core found in Step 1, c_{HOT}, is assigned. Before doing so, all the cores are reset to their un-partitioned configuration, and all their cost values are reset to their initial values. For this step, the algorithm tries to reduce the power consumption of the target core by half by halving the TAM partition frequency. Increase in total test application time for the target TAM partition is minimized by doubling the assigned virtual TAM width. The algorithm then re-computes *cost_tmp* for all cores, repeats Steps 1 to 4.

4. EXPERIMENTAL RESULTS

The experiments were carried out using d695 and p22810 SoCs from the ITC'02 SoC Benchmark suite [5]. For thermal simulation, cycle-accurate power profiles provided by the authors of [4] 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 scaling them to reflect realistic 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. All temperature values were obtained using the HotSpot temperature simulator from [9].

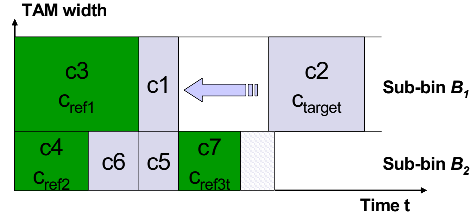

Figure 4. Core c2 is scheduled if reference cores c3, c4 and c7 satisfy their cost constraints

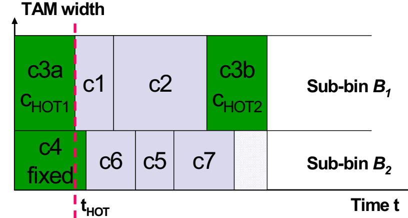

Figure 5. Core c3 is partitioned during step 3

Since the original SoC benchmarks did not include layout information, we handcrafted the layout of the SoCs. Experiments were conducted for TAM widths 16, 24, 32 and 64 bits. Furthermore, each core can only be partitioned 3 times and the maximum virtual TAM width for each TAM partition is set to 64 bits.

The experimental results for d695 and p22810 are shown in Tables 1 and 2, respectively. We set the thermal constraint, $Temp_{max}$, at the initial value of the actual maximum temperature of the schedule, *maxT*, when the constraint is at infinity and decrease it by 5°C and 10°C intervals for d695 and p22810, respectively, each time recording the test application time (*TAT*), and peak power value (P_{max}) given as number of switches. We also computed the gains in temperature (*dT*) with respect to the original base temperature as well as the differences in TAT (*dTAT*) compared to the unconstrained TAT. Grayed-out values indicate results achieved using a combination of reshaping, partitioning, and bandwidth matching while unmarked values were obtained using reshaping alone. The effectiveness of the reshaping and partitioning steps can be seen when temperature drops were achieved without any increase in TAT and/or drastic decrease in power dissipation, as can be seen from $Temp_{max}$=104.3°C to 80.59°C for TAM width of 64 bits for d965 in Table 2, and $Temp_{max}$=167.71°C to 147.55°C for TAM width of 16 bits for p22810 in Table 2. Note that as TAM width increases, more TAM partitions can be formed and fewer cores are placed in each partition, resulting in a higher probability of hot cores being tested concurrently and reducing the ability of the algorithm to separate their test instances via interleaving, which is indicated by overall higher minimum temperatures for larger TAM widths.

To further show the effectiveness of the proposed algorithm, the results obtained using the method in [10] is compared with the results using the proposed algorithm for d695 under the same thermal constraints in Table 3, where *diffTAT* represents the difference in TAT. Before applying any thermal constraints, we used the scheduling algorithm in [10] to create a base schedule under no constraints. The results show that for TAM widths of 24, 32 and 64 bits, the proposed algorithm yields shorter overall test application time than [10] under the same thermal constraints, with a maximum difference of 26% at TAM width of 64 bits. Furthermore, our method allows us to generate results at lower thermal constraints that exceed those in [10].

The minimum temperatures and the respective test times for each SoC achieved using the algorithm in [10] and our proposed algorithm are shown

Table 1. Results using proposed algorithm for d695

d695					
TAM width: 16 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	111.28	45363	1646	N/A	N/A
106.28	106.19	45363	1625	4.57	0.00
101.28	80.73	51485	1647	27.45	-13.50
:	:	:	:	:	:
76.28	76.25	51485	1624	31.48	-13.50
71.28	71.28	51485	1647	35.95	-13.50
66.28	64.16	63863	1633	42.34	-40.78
61.28	61.28	63863	810	44.93	-40.78
56.28	55.80	118375	820	49.86	-160.95
51.28	N/A	N/A	N/A	N/A	N/A
TAM width: 24 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	103.47	29122	1855	N/A	N/A
98.47	98.46	29122	1863	4.84	0.00
93.47	78.27	34189	1779	24.35	-17.40
:	:	:	:	:	:
73.47	73.46	34189	1771	29.00	-17.40
68.47	67.15	35871	1724	35.10	-23.17
63.47	63.08	35871	1667	39.04	-23.17
58.47	58.46	49995	888	43.50	-71.67
53.47	N/A	N/A	N/A	N/A	N/A
TAM width: 32 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	118.02	22543	2023	N/A	N/A
113.02	89.09	22543	1840	24.51	0.00
:	:	:	:	:	:
88.02	86.13	22619	1764	27.02	-0.34
83.02	69.91	23851	1130	40.76	-5.80
:	:	:	:	:	:
68.02	N/A	N/A	N/A	N/A	N/A
TAM width: 64 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	104.30	11358	1811	N/A	N/A
99.30	80.59	11358	1769	22.73	0.00
:	:	:	:	:	:
79.30	N/A	N/A	N/A	N/A	N/A

in Table 4 for TAM widths of 16, 24, 32 and 64 bits. *diffT* shows the difference in minimum temperatures while *diffTAT* represents the difference in TAT. For d695, the proposed algorithm enabled us to lower the test temperature much further compared to [10], with a minimum temperature approx. 40% lower at TAM width of 16 bits (92.79°C compared to 55.8°C), albeit with a 152% increase in TAT. While this increase might seem large, the algorithm at least offers the option to trade-off TAT for further decrease in temperature when needed. On average, the minimum temperatures for d695 were 22% lower using the proposed algorithm. The results are similar for p22810, were the biggest temperature difference was 41% at TAM width of 16 bits and the average temperature difference is 24%. Note that the proposed algorithm is especially effective at very narrow TAM widths, as shown by generally lower minimum temperatures at narrower TAM widths. This is the situation where we can benefit most from reshaping, partitioning, inter-leaving and frequency throttling, as we have fewer TAM partitions more cores are assigned to each of them.

5. CONCLUSION

In this paper, we have presented a thermal-aware TAM deign and test scheduling algorithm for system-on-chips with fixed-width TAMs that ensures thermal safety while minimizing 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, interleaving and bandwidth matching techniques. 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.

Table 2. Results using proposed algorithm for p22810

p22810					
TAM width: 16 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	167.71	482480	7902	N/A	N/A
157.71	157.18	482480	7869	6.28	0.00
147.71	147.55	482480	7876	12.02	0.00
137.71	121.64	568077	4934	27.47	-17.74
:	:	:	:	:	:
117.71	112.34	568077	4864.5	33.02	-17.74
107.71	104.58	568077	4575.5	37.64	-17.74
97.71	97.71	783823	3105.25	41.74	-62.46
87.71	82.69	783823	2464.25	50.69	-62.46
77.71	77.71	783823	2453	53.66	-62.46
67.71	N/A	N/A	N/A	N/A	N/A
TAM width: 24 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	193.67	307039	8557	N/A	N/A
183.67	167.89	307039	9055	13.31	0.00
:	:	:	:	:	:
163.67	148.66	332611	6110.5	23.24	-8.33
:	:	:	:	:	:
143.67	143.67	332611	6110.5	25.82	-8.33
133.67	115.66	355157	5152.5	40.28	-15.67
:	:	:	:	:	:
113.67	107.347	355157	5407.5	44.57	-15.67
103.67	102.29	363043	3967.5	47.18	-18.24
93.67	N/A	N/A	N/A	N/A	N/A
TAM width: 32 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	166.17	230136	9744	N/A	N/A
156.17	151.46	293501	6863	8.85	-27.53
146.17	146.17	293501	6734	12.04	-27.53
136.17	136.17	293501	7271	18.05	-27.53
126.17	122	295015	6228.5	26.58	-28.19
116.17	115.78	295015	6802.5	30.32	-28.19
106.17	106.17	474807	5688	36.11	-106.32
96.17	94.64	474807	4846.75	43.05	-106.32
86.17	86.17	474807	4650.5	48.14	-106.32
76.17	76.08	474807	3343.75	54.22	-106.32
66.17	N/A	N/A	N/A	N/A	N/A
TAM width: 64 bits					
$Temp_{max}$	***maxT***	***TAT***	***P_{max}***	***dT***	***dTAT***
(°C)	*(°C)*	*(cycles)*		*(%)*	*(%)*
∞	130.44	133404	10300	N/A	N/A
120.44	118.89	165909	9619.5	8.85	-24.37
110.44	105.11	165909	6652.5	19.42	-24.37
100.44	100.23	223385	5811	23.16	-67.45
90.44	N/A	N/A	N/A	N/A	N/A

REFERENCES

[1] 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.

[2] 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.

[3] 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.

Table 3. Comparison of results using [10] and proposed algorithm for d695

d695					
TAM width: 16 bits					
	Using [10]		**Proposed algorithm**		
Temp$_{max}$	***maxT***	***TAT***	***maxT***	***TAT***	***diffTAT***
(°C)	***(°C)***	***(cycles)***	***(°C)***	***(cycles)***	***(%)***
∞	101.54	43504	111.28	45363	-4.27
96.54	92.79	46873	80.73	51485	-9.84
:	N/A	N/A	:	:	N/A
76.54	:	:	76.53	51485	N/A
71.54	:	:	71.54	51485	N/A
66.54	:	:	64.17	63863	N/A
61.54	:	:	61.54	63863	N/A
56.54	:	:	55.80	118375	N/A
51.54	:	:	N/A	N/A	N/A
TAM width: 24 bits					
Temp$_{max}$	***maxT***	***TAT***	***maxT***	***TAT***	***diffTAT***
(°C)	***(°C)***	***(cycles)***	***(°C)***	***(cycles)***	***(%)***
∞	122.42	30879	103.47	29122	5.69
117.42	109.53	31490	103.47	29122	7.52
112.42	109.53	31490	103.47	29122	7.52
107.42	96.88	32516	103.47	29122	10.44
102.42	96.88	32516	102.41	29122	10.44
97.42	96.88	32516	97.41	29122	10.44
92.42	91.49	34250	78.27	34189	0.18
:	N/A	N/A	:	:	N/A
77.42	:	:	77.40	34189	N/A
72.42	:	:	72.41	34189	N/A
67.42	:	:	67.15	35871	N/A
62.42	:	:	62.41	35871	N/A
57.42	:	:	N/A	N/A	N/A
TAM width: 32 bits					
Temp$_{max}$	***maxT***	***TAT***	***maxT***	***TAT***	***diffTAT***
(°C)	***(°C)***	***(cycles)***	***(°C)***	***(cycles)***	***(%)***
∞	105.16	22837	118.02	22543	1.29
100.16	89.58	24817	89.09	22543	9.16
:	:	:	:	:	:
85.16	81.41	28489	85.16	22619	20.60
80.16	77.15	28489	69.91	23851	16.28
:	N/A	N/A	:	:	N/A
65.16	:	:	N/A	N/A	N/A
TAM width: 64 bits					
Temp$_{max}$	***maxT***	***TAT***	***maxT***	***TAT***	***diffTAT***
(°C)	***(°C)***	***(cycles)***	***(°C)***	***(cycles)***	***(%)***
∞	92.76	12696	104.30	11358	10.54
87.76	84.71	15343	80.59	11358	25.97
82.76	N/A	N/A	80.59	11358	N/A
77.76	:	:	N/A	N/A	N/A

[4] 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.

[5] 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.

[6] P. Rosinger, B. Al-Hashimi and K. Chakrabarty, "Thermal-safe test scheduling for core-based system-on-chip integrated circuits," *IEEE Trans. on Computer Aided Design*, vol. 25, no. 11, pp. 2502-2512, Nov. 2006.

[7] 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.

[8] Z. He, Z. Peng and P. Eles, "A heuristic for thermal-safe SoC test scheduling," *Proc. of IEEE International Test Conference (ITC)*, pp. 1-10, 2007.

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

[10] 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.

[11] T. Yoneda, K. Masuda and H. Fujiwara, "Power-constrained test scheduling for multi-clock domain SoCs," *Proc. Design, Automation and Test in Europe (DATE) Conf.*, pp. 297-302, 2006.

[12] A. Ajami et al., "Analysis of non-uniform temperature-dependent interconnect performance in high performance ICs," *Proc. of Design Automation Conference*, pp. 567–572, 2001.

Table 4. Comparison of minimum temperature results using [10] and proposed algorithm

	TAM width: 16 bits					
	[10]		**Proposed Method**			
SoC	***maxT (°C)***	***TAT (cycles)***	***maxT (°C)***	***TAT (cycles)***	***diffT (%)***	***diffTAT (%)***
d695	92.79	46873	55.80	118375	39.86	-152.54
p22810	133.02	511441	77.71	783823	41.58	-53.26
	TAM width: 24 bits					
	[10]		**Proposed Method**			
SoC	***maxT (°C)***	***TAT (cycles)***	***maxT (°C)***	***TAT (cycles)***	***diffT (%)***	***diffTAT (%)***
d695	91.49	34250	58.46	49995	36.10	-45.97
p22810	110.1	390905	102.29	363043	7.09	7.13
	TAM width: 32 bits					
	[10]		**Proposed Method**			
SoC	***maxT (°C)***	***TAT (cycles)***	***maxT (°C)***	***TAT (cycles)***	***diffT (%)***	***diffTAT (%)***
d695	77.15	28489	69.91	23851	9.38	16.28
p22810	109.36	263916	71.17	523139	34.92	-98.22
	TAM width: 64 bits					
	[10]		**Proposed Method**			
SoC	***maxT (°C)***	***TAT (cycles)***	***maxT (°C)***	***TAT (cycles)***	***diffT (%)***	***diffTAT (%)***
d695	84.71	15343	80.59	11358	4.86	25.97
p22810	107.25	185614	92.79	223385	13.48	-20.35