PAPER

Thermal-Aware Test Access Mechanism and Wrapper Design Optimization for System-on-Chips

Thomas Edison YU[†a)], ***Nonmember***, **Tomokazu YONEDA**[†], ***Member***,
Krishnendu CHAKRABARTY[††], ***Nonmember***, ***and*** **Hideo FUJIWARA**[†], ***Fellow***

SUMMARY Rapid advances in semiconductor manufacturing technology have led to higher chip power densities, which places greater emphasis on packaging and temperature control during testing. For system-on-chips, peak power-based scheduling algorithms have been used to optimize tests under specified power constraints. However, imposing power constraints does not always solve the problem of overheating due to the non-uniform distribution of power across the chip. This paper presents a TAM/Wrapper co-design methodology for system-on-chips that ensures thermal safety while still optimizing the test schedule. The method combines a simplified thermal-cost model with a traditional bin-packing algorithm to minimize test time while satisfying temperature constraints. Furthermore, for temperature checking, thermal simulation is done using cycle-accurate power profiles for more realistic results. Experiments show that even a minimal sacrifice in test time can yield a considerable decrease in test temperature as well as the possibility of further lowering temperatures beyond those achieved using traditional power-based test scheduling.

key words: *SoC testing, test architecture design, test scheduling, thermal constraint*

1. Introduction

As feature sizes and frequencies of newer System-on-Chips scale much faster than operating voltages, not only power densities but also heat densities will experience a considerable increase. Furthermore, the problem of overheating becomes much larger during testing when beyond normal switching activities occur due to the need for concurrently testing cores to shorten test time. Overheating can lead to problems such as increased leakage power and even permanent chip damage. For every 20°C rise in temperature, there is approximately a 5-6% increase in interconnect delay timing [15]. These timing uncertainties can result in further yield loss. Traditionally, simply using better packaging and cooling methods would suffice but this has become increasingly difficult and expensive. To reduce packaging cost, packages have increasingly been designed for the worst case *typical* application [12], [13] and the cost of cooling during test application has become very prohibitive.

For SoCs, test planning usually involves the design of a test data delivery method (TAM: Test Access Mechanism), and the use of *wrappers* which isolate cores under test. While several approaches to optimize wrapper designs for single frequency embedded core test [1], [2] have been proposed, Iyengar et al. [3], [4] integrated the process into one wrapper and TAM co-optimization algorithm. Up to now, limiting power consumption during test has been the main method of temperature control, and test scheduling under power constraints have been considered in [4]–[7].

Because of the non-uniform spatial power distribution across the chip, limiting the maximum chip-level power dissipation is not effective in reducing and avoiding localized heating (called hot spots) which occurs faster than chip-wide heating [9], [12], [13] as shown in Table 1. In Table 1, the maximum test temperatures, $maxT$, do not scale with power constraints P_{max} for the SoC p93791 using the power-constrained method in [4]. Furthermore, power-constrained test scheduling does not allow further exploration of schedule variations with the same test peak power. For the benchmark SoC d695 with a layout shown in Fig. 1, two schedules having the same peak power value can have different peak temperature, as shown in Fig. 2. The peak temperatures for the three hottest cores, c5, c6, and c10 are indicated and the maximum temperature for c5 varies from 89.6°C to 77.2°C simply by changing its allocated TAM width from 32 to 31 bits.

Table 1 Max. temperatures of p93791 under various power constraints.

p93791	$TAM = 32$		$TAM = 64$	
P_{max}	$maxT(^oC)$	$TAT(cycles)$	$maxT(^oC)$	$TAT(cycles)$
13000	121.43	1105893	115.24	634685
17000	115.44	1033179	127.91	566076
21000	143.78	994803	110.66	538301
25000	127.33	975528	130.09	517541
∞	157.25	955989	123.49	523730

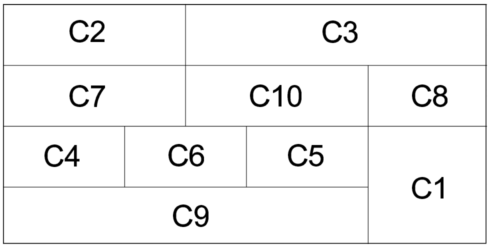


Fig. 1 Hand-crafted layout of SoC d695.

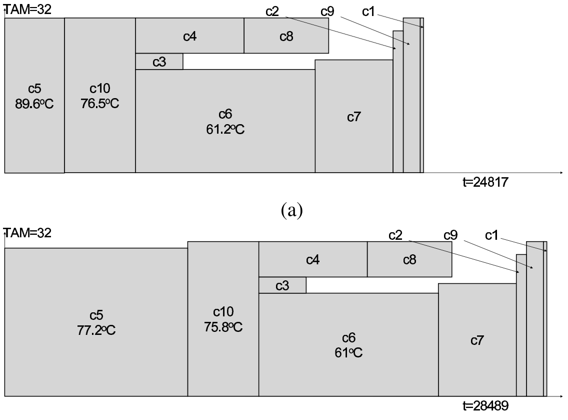
Manuscript received March 21, 2008.
Manuscript revised June 4, 2008.

[†] The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.

[††] The author is with the Department of Electrical and Computer Engineering, Duke University, Box 90291, 130 Hudson Hall, Durham, NC, 27708 USA.

a) E-mail: tomasu-y@is.naist.jp

DOI: 10.1093/ietisy/e91–d.10.2440



(a)

(b)

Fig. 2 Two possible schedules at peak power=1600 switches, (a) $maxT = 89.6°C$, (b) $maxT = 77.2°C$.

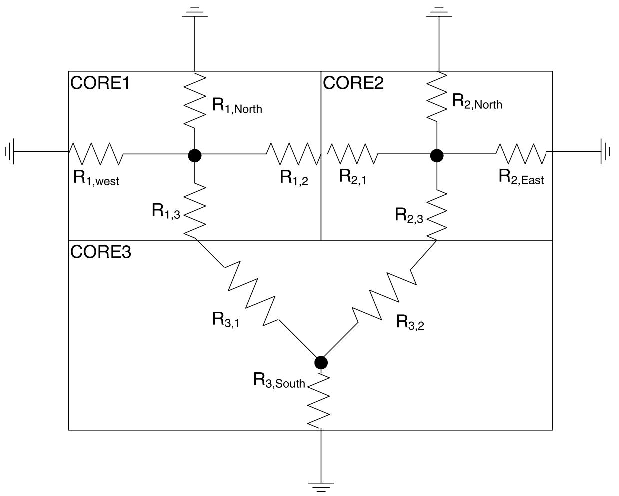


Fig. 3 Lateral thermo-resistive model [9].

In this paper, we propose a design framework which integrates the TAM/wrapper co-optimization process with a thermal-aware test scheduling algorithm. The main contributions of this paper are as follows:(1)consider a different cycle-accurate power profile per wrapper configuration for more realistic results, (2)present a simplified thermal cost function and develop a test scheduling algorithm to minimize the overall test time while satisfying temperature constraints, (3)show that for the ITC'02 SoC benchmarks [8], even a small increase in test time can yield a considerable decrease in test temperature as well as the possibility of further lowering temperatures beyond those achieved using traditional power-based test scheduling.

The rest of this paper is organized as follows. A review of related works is given in Sect. 2. The motivation for this work is discussed in Sect. 3. Section 4 discusses the proposed TAM/wrapper co-optimization algorithm and the proposed test scheduling algorithm. Section 5 gives the experimental results while Section 6 concludes this paper.

2. Related Work and Motivation

Rosinger et al. [9] first proposed using a thermal model as a guide to test scheduling instead of a chip-level power constraint. They used the RC-equivalent micro-architecture thermal model from [12]–[14] which in turn makes use of the well-known duality between heat transfer and electrical phenomena: *heat can be described as a current passing through a thermal resistance and leading to a temperature difference analogous to a voltage* [12]. More specifically, [9] only considered the lateral flow of heat away from an active core by reducing a chip into a network of thermal resistances as shown in Fig. 3. The same thermal resistance network model is used in this work. The proposed test scheduling algorithm in [9] uses a test compatibility graph as its basis and cores are grouped into test sessions which are applied sequentially.

In [10], Liu et al. defines a "hot spot" as a core whose temperature is substantially higher than the average temperature over all cores. They proposed two algorithms which try to spread heat more evenly over a chip via layout information and a progressive weighting function, respectively. For this work, we define "hot spot" as any core which exceeds the thermal constraint during test. Thus, a core can be scheduled even if its temperature is much higher than its surrounding cores unlike in [10].

Table 2 Max. temperatures of d695 under various power constraints using different power models.

d695	$TAM = 24$		
P_{max}	$T_{real}(^oC)$	$T_{pavg}(^oC)$	$T_{peak}(^oC)$
1600	99.64	90.14	345.68
1800	103.80	91.15	409.58
2000	106.84	93.52	424.86
2200	111.74	103.75	479.03
2400	104.94	93.60	421.48

In [11], 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 a single fixed power value per core was considered and steady-state temperatures were used as temperature upper bounds [9]. However, this is not realistic, as shown in Table 2 where the peak temperature of test schedules using static average power T_{pavg} during thermal simulation are usually less than cycle-accurate values T_{real}, while maximum temperatures using peak power values T_{peak} are usually much higher and can be considered pessimistic.

Furthermore, the choice of using a single fixed power profile per core is also not realistic. From our experiments, we found that higher TAM widths (therefore, shorter test time) can yield lower maximum temperatures despite having higher peak power values. This is shown in Fig. 4, where the temperature profile, as well as the peak temperature of core 5 of d695 varies with TAM width. This can be attributed the varying power profile per TAM configuration [7] as well as the RC characteristic of temperature rise: if a test can finish before the temperature curve reaches steady state, the

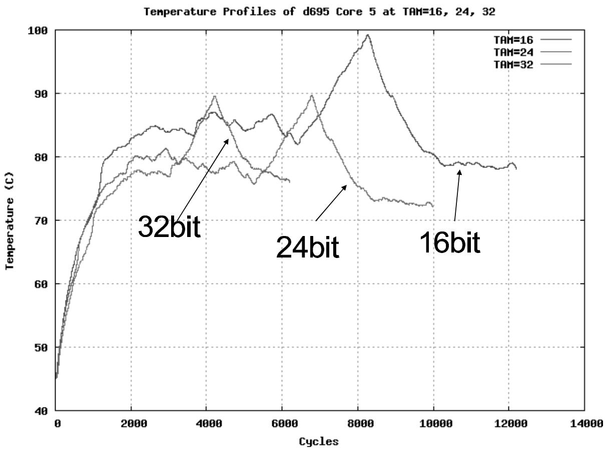


Fig. 4 Varying temperature profiles per TAM configuration for core 5 of d695.

capacitance can have a “filtering” effect on the maximum temperature values. Thus, test time must also be considered when deriving a thermal model or thermal cost function as discussed in the next section.

Finally, flexible TAM-width and partitioned testing were also outside the scope of [9] and [11]. To the best of our knowledge, this is the first work which attempts to integrate TAM/wrapper co-optimization and test scheduling under a thermal constraint using cycle-accurate power profile per wrapper configuration for more realistic temperature simulation.

3. TAM/Wrapper Co-optimization and Test Scheduling

In this section, we formally present the TAM/Wrapper co-optimization and test scheduling problem P_{TWOP}.

Problem P_{TWOP}: For an SoC S, given:
W_{ext}: external TAM width allocated to the SoC
N_C: Number of cores
$Temp_{max}$: maximum allowed temperature during test

For each core $C_i (1 \leq i \leq N_C)$ of SoC S
- $Wset_i$: number of usable wrapper configurations
- For each wrapper configuration $w_{ij} (1 \leq j \leq Wset_j)$

  - TAM_{ij}: alloted TAM width
  - P_{ij}: power profile
  - TAT_{ij}: test application time

Determine the following output:
For each core $C_i (1 \leq i \leq N_C)$ of SoC S
- w_{fi}: assigned final wrapper configuration

  - TAM_{fi}: final alloted TAM width
  - t_{starti}: test start time
  - t_{endi}: test end time

and minimize the overall test time of S such that the total number of TAM used at any given time does not exceed W_{ext} and temperatures do not exceed $Temp_{max}$.

Since we cannot ignore per-cycle power values and their effects on temperature, each wrapper configuration is given a different power profile as explained in the previous section.

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 width and height represents test application time and TAM width, 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 overall test time (e.g. bin width) is minimized. For scheduling under a power constraint, it can be extended into a restricted 3-D bin packing problem where the length, width and height represent total test time, peak power and TAM width, respectively, for an SoC core. 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 bin packing problem is *NP-Hard*, this paper proposes a heuristic algorithm to solve the problem.

3.1 Thermal Cost Function

Since temperature cannot be handled in the same simplistic and direct way as power(i.e. simple superposition is inapplicable), we need a thermal model and cost function which can effectively and simply express the heating phenomena without the need for data from thermal simulations.

The results in [9] prove that there exists a positive correlation between heat and heat dissipation paths represented by lateral thermal resistances. Thus, we have chosen to use lateral thermal resistance as one of the basis for our model and cost function, with necessary modifications of assumptions from previous works so the model can better approximate heating patterns during testing. From Eq. 1, the thermal resistance R_{TH} between two adjacent bodies is directly proportional to the thickness of the heat source t and inversely proportional to the cross-sectional area A of the destination across which the heat is being transferred and the thermal conductivity k of the material per volume unit.

$$R_{TH} = t/kA \tag{1}$$

First, similar to [9], it is assumed that heat transfer between two cores tested concurrently is negligible and thermal resistances between these cores are removed as shown in Fig. 5, where we are left with lateral resistances in parallel for core 1 and core 2. Since the thermal resistances of a core C_i are in parallel to each other, the total thermal resistance $Rth_{TOT,i}$ can be computed as follows:

$$Rth_{TOT,i} = \frac{1}{\sum_{j=1}^{N} 1/R_{i,j}} \tag{2}$$

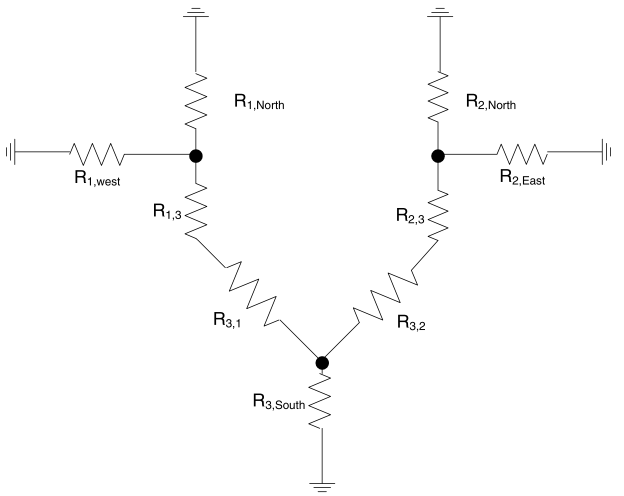


Fig. 5 Thermal resistance network when cores 1 and 2 are concurrently tested.

where *N* is the total number of thermal resistances $R_{i,j}$ of C_i. Note that removing a thermal resitance from the network increases the total thermal resitance, which reflects the fact that there are fewer paths for heat to escape to and this reflects a higher maximum temperature for the core.

The assumption made in [9] that inactive cores are thermally grounded and do not heat up is not realistic unless ample time is given for tested cores to cool down before the next test session, as shown in Fig. 6, where c5 can increase the temperatures of its inactive peripheral cores by as much as 10°C for c10. Obviously this is not practical because of the required increase in idle time. Furthermore, our experiments show that the temporal dimension, more specifically, the test length as well as the order in which cores are tested can greatly affect the maximum temperature of the next core to be tested as shown in Fig. 7 where the peak temperature of core 5 increases by 7°C when core 10 is tested right before it (Fig. 7 (b)) compared to the opposite sequence (Fig. 7 (a)). Thus, when a core is about to be tested, the lateral resistances to cores whose test has just ended are also removed from the total lateral resistance. For example, if core 2 is tested right after core 1 in Fig. 3, then $R_{2,1}$ is removed.

Furthermore, the time dependence of temperature and the power consumption must also be considered. As a rule, we want to test hot cores with large power densities as short as possible and minimize their effects on other cores (avoid concurrency and immediate precedence with cores in immediate physical periphery of the hot spot core).

Due to the localized nature of hot spots as well as the effects of layout and varying thermal resistance configurations, the core with the highest thermal cost does not always mean that it is hotter than cores with lower thermal costs. Thus, we define the following thermal cost for each core C_i with respect to its wrapper configuration w_{ij} and time *t* as shown below:

$$Cost_i(w_{ij}, t) = p_{ij} \times \left(R_{THi}(t) + TAT_{ij}\right) \tag{3}$$

TAT_{ij} is the test application time and p_{ij} is the average power computed from power profile P_{ij} for wrapper configuration w_{ij}. The lateral resistance R_{THi} is expressed as a function of time because it changes according to when core C_i is scheduled and what cores are tested before as well concurrently with it. In our experiments, the average power dis-

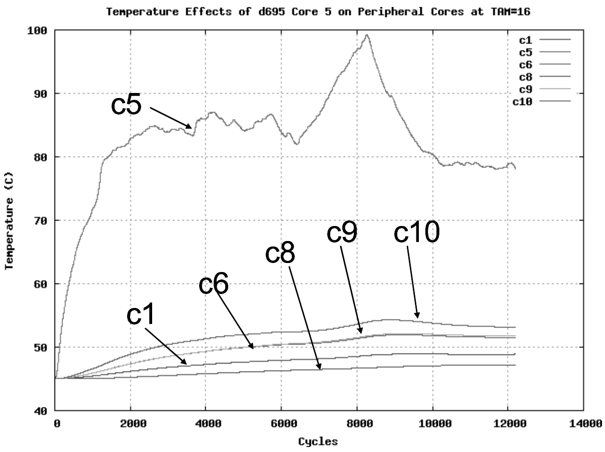
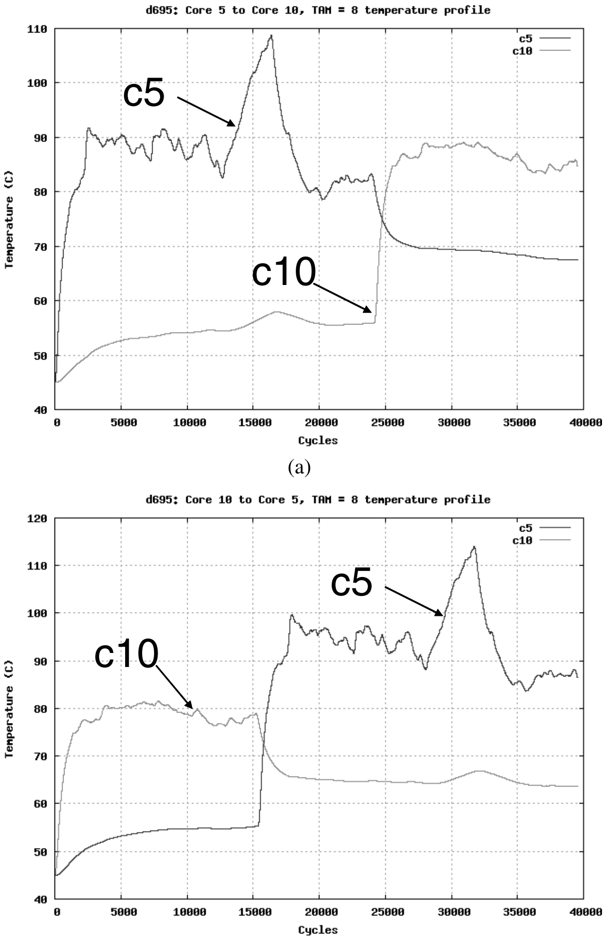


Fig. 6 Thermal effect of d695 core 5 on peripheral cores.



(a)

(b)

Fig. 7 Effects of test order on peak temperature, (a) core 5 before core 10, (b) core 10 before core 5.

sipation was found to give 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 which vary with respect to w_{ij} to greatly simplify cost calculations. The main idea is to pick out hot spot cores, determine an upper limit to their thermal cost, $cost\_max_i$, and gradually decrease this limit until the thermal constraint is satisfied. Furthermore, a thermal cost minimum is computed which represents the worst case configuration of a core to be packed. It inevitably leads to the core being tested alone regardless of time frame, and not preceded by any immediate peripheral cores as given by the equation below:

$$cost\_min_i = \min_{1 \le j \le W_{ext}} (Cost_i(w_{ij}, NULL)) \tag{4}$$

where $Cost_i(w_{ij}, NULL)$ denotes the cost of unscheduled core C_i with wrapper configuration w_{ij} and no thermal resistance is removed in equation 3, denoted by $NULL$ time.

3.2 Test Scheduling Algorithm

The pseudo-code for our proposed algorithm is shown in Fig. 8.

Init: Optimal Wrapper Configuration Creation
The initialization steps (lines 1-5 of Fig. 8) first makes sure that a configuration for each core can be found which satisfies the thermal constraint $Temp_{max}$. Initially, the highest cost $cost\_max$ is set to infinity, and the minimum cost $cost\_min$ is computed for each core (line 4). It then uses a selection process introduced in [4] where Pareto-optimal points of the TAM vs. Test time graph are chosen as optimal wrapper configurations (w_{iopt}) in line 5. When choosing optimal wrapper configurations, the thermal cost must always satisfy both cost constraints.

Priority 1: Packing Rectangles with Optimal Wrapper Configuration
Before packing, the algorithm takes note of the current time in the schedule, denoted by the variable $current\_t$. In line 8, we try to pack as many cores using optimal TAM widths while $available\_TAM \neq 0$. Each core C_i is examined in order of decreasing thermal cost when using their optimal wrapper configurations, denoted by $Cost_i(w_{iopt}, NULL)$, since potential hot spot cores should be scheduled as early and as quickly as possible to minimize their effects on subsequent cores.

Here and in all subsequent steps, the thermal costs for all active cores are computed and checked with their upper and lower limits before packing since they change whenever a new core is scheduled. The *Assign()* function in Fig. 9 invoked after every priority step updates the parameters of the core to be scheduled (lines 1-3), recomputes the thermal costs of all the scheduled cores (line 4), updates the remaining core list (line 5) and available TAM (line 6). As the algorithm iterates further, hotspot cores are gradually separated from each other during scheduling due to the imposition of cost limits.

```
Function Schedule(S, $W_{ext}$, $Temp_{max}$)
1   Do thermal simulation for each $w_{ij}$ configuration of core $C_i \in$ S
2       If no configuration that satisfies $Temp_{max}$, terminate scheduling;
3       Set $available\_TAM = W_{ext}$, $current\_t = 0$, $maxT = \infty$;
4   For each $C_i \in$ S, compute $cost\_min_i$, set $cost\_max_i = \infty$,
5       Find $w_{iopt}$ (from[4]) such that
        $Cost_i(w_{iopt}, NULL) \leqq cost\_max_i$, then end For
6   While S≠Ø
7       If $available\_TAM > 0$
8       (Priority 1)
            If there exist a core $C_i \in$ S such that $TAM_{iopt} \leqq$
            $available\_TAM$ AND $Cost_i(w_{iopt}, NULL)$ is maximum AND
            $Cost_j \leqq cost\_max_j$ for all scheduled cores $C_j$ when $C_i$ is
            scheduled at $current\_t$ with $TAM_{iopt}$
                Then, Assign($C_i$, $TAM_{iopt}$) and go to line 6;
9       (Priority 2)
            Else If there exist a core $C_i \in$ S such that $TAM_{iopt} \leqq$
            ($available\_TAM + \alpha$) AND $TAM_{iopt}$ is minimum AND
            $Cost_j \leqq cost\_max_j$ for all scheduled cores $C_j$ when $C_i$ is
            scheduled at $current\_t$ with $available\_TAM$
                Then, Assign($C_i$, $available\_TAM$) and go to line 6;
10      (Priority 3)
            Else If there exist a scheduled $C_i$ with assigned wrapper $w_{fi}$
            such that $tstart_i = current\_t$ AND has maximum decrease in
            test application time if $TAM_{fi} = TAM_{fi} + available\_TAM$
            AND $Cost_j \leqq cost\_max_j$ for all scheduled cores $C_j$ when
            $C_i$ is scheduled at $current\_t$ with $TAM_{fi} + available\_TAM$
                Then, Assign($C_i$, $TAM_{fi} + available\_TAM$) and go to
                line 6;
11      Else, update $current\_t$ to the earliest test end time among
        currently scheduled cores, reset $available\_TAM$, return to line 6;
12  End While
13  (Updating and cost adjustment)
    Do thermal simulation of finished schedule and compute $maxT$
    If $maxT \leqq Temp_{max}$, Then terminate scheduling
14  Else, Find the hottest core $C_{hot}$
      If $cost\_max_{hot} = \infty$ Then compute $cost\_max_{hot}$
15        If ($cost\_max_{hot}$ * $adjust\_factor$) $\geqq cost\_min_{hot}$, Then
          $cost\_max_{hot}$ = ($cost\_max_{hot}$ * $adjust\_factor$) and determine a
          new $w_{hotopt}$ as done in line 5 and go to line 6;
16        Else If next hottest core exists, let it be $C_{hot}$, go to line 15;
17        Else terminate scheduling (no adjustable cores exists);
```

Fig. 8 Pseudo code of proposed test scheduling algorithm.

```
Function Assign($C_i$, $TAM_{ij}$)
1   $TAM_{fi} = TAM_{ij}$;
2   $t_{starti} = current\_t$;
3   $t_{endi} = current\_t + TAT_{ij}$;
4   Update thermal cost $Cost_k$ for all scheduled cores $C_k$
5   S = S – {$C_i$};
6   $available\_TAM = available\_TAM - TAM_{ij}$;
```

Fig. 9 Pseudo code for the core Assign() function.

Priority 2: Insertion of Rectangles into Idle Space
If no rectangle can be packed in their optimal configuration, the algorithm looks for a core C_i whose optimal tam-wdith TAM_{iopt} is less than or equal to $available\_TAM + \alpha$ where $(1 \le \alpha \le 4)$ in line 9. In Fig. 10, the wrapper configuration

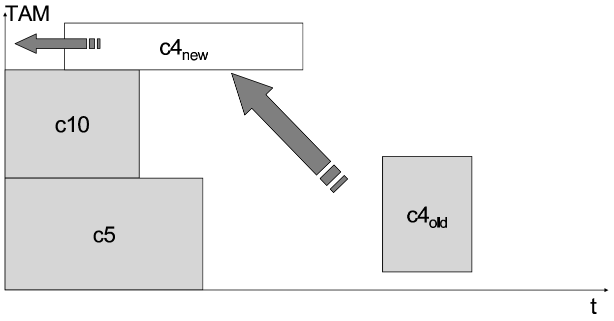


Fig. 10 Inserting core 4 into idle TAM space by reducing assigned TAM width.

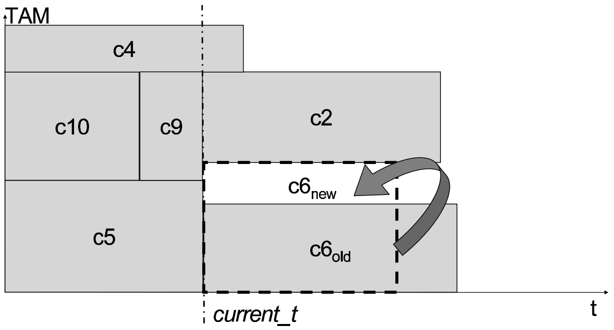


Fig. 11 Allotting more TAM wires to core 6.

of core 4 was changed to a non-optimal configuration ($c4_{old}$ to $c4_{new}$) and inserted into the idle space of the schedule.

Priority 3: Filling Idle Space by Increasing TAM width
The algorithm checks among the currently scheduled cores whose start times *tstart* equal *current_t* and determines which core would have the largest gain in test time if given the unused TAM lines and packs this core in line 10. In Fig. 11, the alloted TAM width to the scheduled core c6 is changed to minimize wasted TAM wires. In line 11, *current_t* is updated when *available_TAM* becomes zero or when no cores can be scheduled in lines 8-10.

Updating and Cost Adjustment
When all cores have been scheduled, thermal simulation using HotSpot tool is performed using cycle-accurate power profiles in line 13. The peak chip-wide temperature *maxT* is then compared to the thermal constraint. If it is satisfied, then the program ends. If not, then cost adjustment is performed on the hottest core C_{hot} in lines 14-15 and $cost\_max_{hot}$ is updated. Line 16 looks for the next hottest core to adjust when the current hot spot core's cost can no longer be adjusted. The program ends when the thermal constraint is satisfied or no more cores can be adjusted. The adjustment factor, *adjust_factor*, can be any value from 0-1. For this work, a constant factor of 0.90 is used.

4. Experimental Result

Table 3 Experimental results for d695.

TAM = 16					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	$(cycles)$	$(switches)$	$(\%)$	$(\%)$
∞	101.54	43504	1598	N/A	N/A
96.54	92.79	46873	1598	8.62	-7.74
91.54	N/A	N/A	N/A	N/A	N/A
TAM = 24					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	$(cycles)$	$(switches)$	$(\%)$	$(\%)$
∞	122.42	30879	1713	N/A	N/A
117.42	109.53	31490	1624	10.53	-1.98
112.42	109.53	31490	1624	10.53	-1.98
107.42	96.88	32516	1598	20.86	-5.30
:	:	:	:	:	:
92.42	91.49	34250	1630	25.27	-10.92
87.42	N/A	N/A	N/A	N/A	N/A
TAM = 32					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	$(cycles)$	$(switches)$	$(\%)$	$(\%)$
∞	105.16	22837	1650	N/A	N/A
100.16	89.58	24817	1598	14.82	-8.67
:	:	:	:	:	:
85.16	81.41	28489	1598	22.58	-24.75
80.16	77.15	28489	1598	26.64	-24.75
75.16	N/A	N/A	N/A	N/A	N/A
TAM = 64					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	$(cycles)$	$(switches)$	$(\%)$	$(\%)$
∞	92.76	12696	1689	N/A	N/A
87.76	84.71	15343	1620	8.68	-20.85
82.76	N/A	N/A	N/A	N/A	N/A

The experiments were done using three SoCs from the ITC'02 SoC Benchmark suite [8], d695, p22810, and p93791. For thermal simulation, cycle-accurate power profiles provided by the authors of [7] 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, 200, and 500 for d695, p22810, and p93791, respectively, to reflect power dissipation during test. The test data for d695, upon thermal simulation, reveals that the total test time under TAM configurations used for this experiment (16, 24, 32, 64) are too short to show any significant heating of the chip. Therefore, when necessary, we have increased the length of the sampling interval during thermal simulation to allow the effects of heat to show. This is reasonable if we consider that tests for delay faults are normally 2-4 times larger than stuck-at-fault test sets. Since the test application time per core is normally much larger in magnitude compared to lateral resistance, we scaled the test time values for each SoC such that their magnitudes are within acceptable range of each other (in this work, both total lateral resistance and test time was adjusted to not exceed 100) when computing for the thermal costs. Experiments were done using an HP ProLiant Workstation with 4 Opteron CPU's operating at 2.4 GHz with 32 GB of memory.

Since the original SoC benchmarks did not include layout information, we handcrafted the layout of each SoC. The scheduling and thermal simulation results for d695, p22810 and p93791 are shown in Tables 3 to 5. Before applying

Table 4 Experimental results for p22810.

TAM = 16					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	170.94	467362	8488	N/A	N/A
165.94	152.47	472762	7226	10.80	-1.16
:	:	:	:	:	:
150.94	133.02	511441	6006	22.18	-9.43
:	:	:	:	:	:
130.94	N/A	N/A	N/A	N/A	N/A
TAM = 24					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	166.37	324723	8104	N/A	N/A
161.37	154.07	338267	8054	7.39	-4.17
156.37	154.07	338267	8054	7.39	-4.17
151.37	148.98	345661	8048	10.45	-6.45
146.37	145.06	357802	7258	12.81	-10.19
141.37	135.45	359907	6986	18.59	-10.84
136.37	135.45	359907	6986	18.59	-10.84
131.37	113.89	396397	6166	31.54	-22.07
:	:	:	:	:	:
111.37	110.1	390905	6006	33.82	-20.38
106.37	N/A	N/A	N/A	N/A	N/A
TAM = 32					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	155.5	241403	9222	N/A	N/A
150.5	149.25	254660	7898	4.02	-5.49
145.5	109.36	263916	6184	29.67	-9.33
:	:	:	:	:	:
105.5	N/A	N/A	N/A	N/A	N/A
TAM = 64					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	138.81	149604	9936	N/A	N/A
133.81	129	145417	9974	7.07	2.80
128.81	113.79	153146	8542	18.02	-2.37
:	:	:	:	:	:
108.81	107.25	185614	6010	22.74	-24.07
103.81	N/A	N/A	N/A	N/A	N/A

Table 5 Experimental results for p93791.

TAM = 16					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	137.59	1842004	19690	N/A	N/A
132.59	119.74	1868602	12540	12.97	-1.44
:	:	:	:	:	:
117.59	115.03	1875576	12540	16.40	-1.82
112.59	N/A	N/A	N/A	N/A	N/A
TAM = 24					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	107.02	1261748	12540	N/A	N/A
102.02	N/A	N/A	N/A	N/A	N/A
TAM = 32					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	139.82	946416	27890	N/A	N/A
134.82	126.9	969552	20675	9.24	-2.44
129.82	126.9	969552	20675	9.24	-2.44
124.82	115.37	1030210	16350	17.49	-8.85
119.82	115.37	1030210	16350	17.49	-8.85
114.82	107.93	1141742	12930	22.81	-20.64
109.82	107.93	1141742	12930	22.81	-20.64
104.82	103.78	1153424	12930	25.78	-21.87
99.82	96.96	1207921	12545	30.65	-27.63
94.82	94.63	1157587	12540	32.32	-22.31
89.82	N/A	N/A	N/A	N/A	N/A
TAM = 64					
$Temp_{max}$	$maxT$	TAT	P_{max}	dT	$dTAT$
(^oC)	(^oC)	($cycles$)	($switches$)	(%)	(%)
∞	142.25	483680	36930	N/A	N/A
137.25	113.38	527141	20935	20.30	-8.99
:	:	:	:	:	:
112.25	100.3	585385	19545	29.49	-21.03
:	:	:	:	:	:
97.25	92.53	631314	12885	34.95	-30.52
92.25	91.59	656079	12885	35.61	-35.64
87.25	N/A	N/A	N/A	N/A	N/A

any thermal constraints, we used our scheduling algorithm to create a base schedule without any constraints. From the non-constrained schedule, we determine its maximum temperature, $maxT$, and use it as the thermal constraint, $Temp_{max}$. We gradually decreased the constraint by 5 degree steps, each time recording the actual maximum temperature ($maxT$), 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 base temperature as well as the differences in TAT ($dTAT$).

In Table 3 for d695, a maximum temperature gain of 26.64% was achieved with a modest 24.75% increase in TAT (TAM = 32, $Temp_{max}$ = 80.16°C). For as little as 5.30% increase in TAT, we can get a relatively large gain of 20.86% in temperature reduction (TAM = 24, $Temp_{max}$ = 107.42°C). The limitations of global peak-power based approaches becomes apparent when we consider the results for TAM = 32 in Table 3. For most of the temperature variations, the peak power value remained constant at 1598. When such a power constraint is applied, the temperatures of the generated schedule can vary within the range of 89.58°C – 77.15°C and our algorithm makes sure that the thermal constraint is indeed satisfied.

For p22810 in Table 4, a maximum temperature reduction of 33.82% can be had for a 20.38% increase in TAT (TAM = 24, $Temp_{max}$ = 111.37°C). At TAM = 32, the algorithm was able to decrease the temperature from 155.5°C to a manageable 109.36°C with just a 9.33% sacrifice in TAT. Similar results were obtained for p93791 in Table 5, where there is a maximum temperature reduction of 35.61% with a 35.64% increase in TAT at TAM = 64. Note that at TAM=24, there was no further gain in temperature from temperature at $Temp_{max} = \infty$ and is mainly a result of the lack of flexibility due to the limited usable TAM width. Overall, the algorithm works well for designs with many cores and exploits the availabilty of wider TAMs.

5. Conclusion

In this paper, we have presented a TAM/Wrapper co-optimization framework for system-on-chips that ensures thermal safety while still optimizing the test schedule. 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 while limiting increases in test application time. 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 Assoc. Prof. Erik Larsson of Linköping University, Sweden, for providing the power profiles for the benchmark SoCs, Dr. Paul Rosinger for providing the preliminary benchmark circuit data, and Prof. Michiko Inoue, Asst. Prof. 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). The work of K. Chakrabarty was supported in part by the US National Science Foundation under grant no. OISE-0403217.

References

[1] E.J. Marinissen, S.K. Goel, and M. Lousberg, “Wrapper design for embedded core test,” Proc. 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. IEEE International Test Conference (ITC), pp.529–538, 2002.

[3] V. Iyengar, K. Chakrabarty, and E.J. Marinissen, “Test wrapper and test access mechanism co-optimization for system-on-chip,” J. Electron. Test., Theory Appl., vol.18, pp.213–230, April 2002.

[4] 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. Comput., vol.52, no.12, pp.1619–1632, Dec. 2003.

[5] Y. Xia, M. Chrzanowska-Jeske, B. Wang, and M. Jeske, “Using a distributed rectangle bin-packing approach for core-based SoC test scheduling with power constraints,” Proc. International Conference on Computer-Aided Design (ICCAD), pp.100–105, 2003.

[6] Y. Huang, S.M. Reddy, W. Cheng, P. Reuter, N. Mukherjee, C. Tsai, O. Samman, and Y. Zaidan, “Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm,” Proc. IEEE International Test Conference (ITC), pp.74–82, 2002.

[7] S. Samii, E. Larsson, K. Chakrabarty, and Z. Peng, “Cycle-accurate test power modeling and its application to SoC test scheduling,” Proc. IEEE International Test Conference (ITC), pp.1–10, 2006.

[8] E.J. Marinissen, V. Iyengar, and K. Chakrabarty, “A set of benchmarks for modular testing of SoCs,” Proc. IEEE International Test Conference (ITC), pp.519–528, 2002.

[9] P. Rosinger, B. Al-Hashimi, and K. Chakrabarty, “Rapid generation of thermal-safe test schedules,” Proc. Design, Automation, and Test in Europe (DATE), pp.840–845, 2005.

[10] C. Liu, K. Veeraraghavan, and V. Iyengar, “Thermal-aware test scheduling and hot spot temperature minimization for core-based systems,” Proc. 20th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT’05), pp.552–562, 2005.

[11] Z. He, Z. Peng, P. Eles, P. Rosinger, and B.M. Al-Hashimi, “Thermal-aware SoC test scheduling with test set partitioning and interleaving,” Proc. 21st IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT’06), pp.477–485, 2006.

[12] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture,” Proc. 30th Annual Int’l Symposium on Computer Architecture (ISCA’03), pp.2–13, 2003.

[13] K. Skadron, M.R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, “Temperature-aware microarchitecture: Extended diiscussion and results,” Tech. Rep. CS-2003-08, University of Virginia, Department of Computer Science, April 2003.

[14] M.R. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, and S. Velusamy, “Hotspot: A dynamic compact thermal model at the processor-architecture level,” Microelectron. J.: Circuits and Systems, vol.34, pp.1153–1165, Dec. 2003.

[15] A.H. Ajami, K. Banerjee, M. Pedram, and L.P. van Ginneken, “Analysis of non-uniform temperature-dependent interconnect performance in high performance ICs,” Proc. Design Automation Conference, pp.567–572, 2001.

Thomas Edison Yu received his B.S. degree in Physics from Ateneo de Manila University, Philippines in 2000. He received his B.S. degree in Computer Engineering from the same university in 2001. In 2006, he received his M.E. degree in Information Science from the Nara Institute of Science and Technology, Japan and is currently pursuing a doctorate degree at the same institute. His research interests include SoC, embedded core based system, and low power system design and testing. He is also a student member of IEEE.

Tomokazu Yoneda received the B.E. degree in information systems engineering from Osaka University, Osaka, Japan, in 1998, and M.E. and Ph.D. degrees in information science from Nara Institute of Science and Technology, Nara, Japan, in 2001 and 2002, respectively. Presently he is an assistant professor in Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include VLSI CAD, design for testability, and SoC test scheduling. He is a senior member of IEEE.

Krishnendu Chakrabarty received the B. Tech. degree from the Indian Institute of Technology, Kharagpur, in 1990, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1992 and 1995, respectively, all in Computer Science and Engineering. He is now Professor of Electrical and Computer Engineering at Duke University. Prof. Chakrabarty is a recipient of the National Science Foundation Early Faculty (CAREER) award (1999) and the Office of Naval Research Young Investigator award (2001). His current research projects include: testing and testing and design-for-testability of system-on-chip integrated circuits; digital microfluidic biochips; nanotechnology circuits and systems based on DNA self-assembly; delay-tolerant wireless networks. Prof. Chakrabarty has authored five books Microelectrofluidic Systems: Modeling and Simulation (CRC Press, 2002), Test Resource Partitioning for System-on-a-Chip (Kluwer, 2002), Scalable Infrastructure for Distributed Sensor Networks (Springer, 2005), Digital Microfluidics Biochips: Synthesis, Testing, and Reconfigutaion Techniques (CRC Press, 2006), and Adaptive Cooling of Integrated Circuits using Digital Microfluidics (Artech House, April 2007) and edited the book volumes SOC (System-on-a-Chip) Testing for Plug and Play Test Automation (Kluwer, 2002) and Design Automation Methods and Tools for Microfluidics-Based Biochips (Springer, 2006). He has contributed 15 invited chapters to book volumes, published 270 papers in archival journals and refereed conference proceedings, and delivered over 110 keynote, plenary, and invited talks. He holds a US patent in built-in self-test and is a co-inventor of a pending US patent on sensor networks. He is a recipient of best paper awards at the 2007 IEEE International Conference on VLSI Design, the 2005 IEEE International Conference on Computer Design, and the 2001 IEEE Design, Automation and Test in Europe (DATE) Conference. He is also a recipient of the Humboldt Research Fellowship, awarded by the Alexander von Humboldt Foundation, Germany in 2004, and the Mercator Visiting Professorship for 2000–2002, awarded by the Deutsche Forschungsgemeinschaft, Germany. Prof. Chakrabarty served as a Distinguished Visitor of the IEEE Computer Society for 2005–2007 and as a Distinguished Lecturer of the IEEE Circuits and Systems Society for 2006–2007. Since 2008, he is serving as an ACM Distinguished Speaker. He is an Associate Editor of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on VLSI Systems, IEEE Transactions on Biomedical Circuits and Systems, and ACM Journal on Emerging Technologies in Computing Systems, and an Editor of IEEE Design & Test of Computers, and Journal of Electronic Testing: Theory and Applications (JETTA). He recently completed his term as Associate Editor of IEEE Transactions on Circuits and System I (2006–2007). In the recent past, he has also served as Associate Editor for IEEE Transactions on Circuits and System II. Prof. Chakrabarty is a recipient of Duke University's 2008 Dean's Award for Excellence in Mentoring. He served as Program Chair for the IEEE Asian Test Symposium in 2005 and the CAD, Design, and Test Conference for the 2007 IEEE Symposium on Design, Integration, Test, and Packaging of MEMS/MOEMS, and he is the Program Vice Chair for the 2008 International Mixed-Signals, Sensors, and Systems Test workshop. Prof. Chakrabarty is a Fellow of IEEE, a Senior Member of ACM, and a Member of Sigma Xi. He received a Meritorious Service Award from the IEEE Computer Society in 2008.

Hideo Fujiwara received the B.E., M.E., and Ph.D. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1969, 1971, and 1974, respectively. He was with Osaka University from 1974 to 1985 and Meiji University from 1985 to 1993, and joined Nara Institute of Science and Technology in 1993. Presently he is a Professor at the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan. His research interests are logic design, digital systems design and test, VLSI CAD and fault tolerant computing, including high-level/logic synthesis for testability, test synthesis, design for testability, built-in self-test, test pattern generation, parallel processing, and computational complexity. He is the author of Logic Testing and Design for Testability (MIT Press, 1985). He received many awards including Okawa Prize for Publication, IEEE CS (Computer Society) Meritorious Service Awards, IEEE CS Continuing Service Award, and IEEE CS Outstanding Contribution Award. He served as an Editor and Associate Editors of several journals, including the IEEE Trans. on Computers, and Journal of Electronic Testing: Theory and Application, and several guest editors of special issues of IEICE Transactions of Information and Systems. Dr. Fujiwara is a fellow of the IEEE, a Golden Core member of the IEEE Computer Society, and a fellow of the IPSJ.