Thermal-Safe Test Scheduling for Core-Based System-on-Chip Integrated Circuits*

Paul Rosinger, Bashir M. Al-Hashimi
University of Southampton
School of Electronics and Computer Science
Southampton, SO17 1BJ, UK
{pmr,bmah}@ecs.soton.ac.uk

Krishnendu Chakrabarty
Department of Electrical and Computer Engineering
Duke University
Durham, NC 27708
krish@ee.duke.edu

Abstract

Overheating has been acknowledged as a major problem during the testing of complex system-on-chip (SOC) integrated circuits. Several power-constrained test scheduling solutions have been recently proposed to tackle this problem during system integration. However, we show that these approaches cannot guarantee hot-spot-free test schedules because they do not take into account the non-uniform distribution of heat dissipation across the die and the physical adjacency of simultaneously active cores. This paper proposes a new test scheduling approach that is able to produce short test schedules and guarantee thermal-safety at the same time. Two thermal-safe test scheduling algorithms are proposed. The first algorithm computes an exact (shortest) test schedule that is guaranteed to satisfy a given maximum temperature constraint. The second algorithm is a heuristic intended for complex systems with a large number of embedded cores, for which the exact thermal-safe test scheduling algorithm may not be feasible. Based on a low-complexity test session thermal cost model, this algorithm produces near-optimal length test schedules with significantly less computational effort compared to the optimal algorithm.

*A preliminary version of this work was presented at the Design Automation and Test in Europe (DATE) Conference in March 2005.

1. Introduction

Recent reports from industry indicate that power consumption during scan testing in some designs can be significantly higher compared to the normal operation mode. A case study involving the Motorola Version 3 ColdFire processor [14] reported a minimum of 3X increase of test power over functional power. In the same experiment, there have been also cases where test power for at-speed compressed patterns was as high as 8X the functional power. A more recent industrial article [19] reports test power up to 30X higher compared to the normal operation mode. The elevated levels of power dissipation during test lead inherently to higher die temperatures compared to the normal operation. This creates a number of problems, because both soft error rates and aging increase exponentially with temperature. An undesirable consequence of overheating is thermal stress. At high temperatures, transistors fail to switch properly and many failure mechanisms, such as electro-migration, are accelerated resulting in an overall decrease in reliability or even permanent damage. These problems are exacerbated for core-based system-on-chip (SOC) designs because, quite often, several embedded cores are tested concurrently in order to reduce the overall test time. A significant amount of research has been devoted to reducing power consumption during test in order to avoid the overheating of the silicon die during test. Consequently, several low power solutions targeting core-level design-for-test(DFT), as well as system-level DFT have been recently proposed. Techniques falling in the first category include low-power scan chain architectures with gated clocks [18, 17], scan cell and test pattern reordering [3, 5], and low-transition test patterns generated by specialised ATPG algorithms [22] and low-transition TPGs [21]. The second category of techniques is mainly based on power-constrained test scheduling algorithms [2, 9, 11, 7, 6, 1, 15, 12, 13][1].

This paper focuses on avoiding overheating during test through appropriate test scheduling. The main contributions of the paper are

- propose thermal-aware test as a better alternative to power-constrained test when dealing with overheating during test
- propose a test session thermal model which will reduce the thermal simulation effort required to identify a thermal-safe test schedule

The motivation for this work is presented in Section 2. The basic ideas behind the existing power constrained test scheduling approaches are examined from the perspective of chip overheating, and it is explained why these approaches cannot guarantee thermal-safety. Section 3 proposes a new test scheduling approach that overcomes this problem. An exact algorithm that guarantees minimum test times as well as thermal-safety is presented in

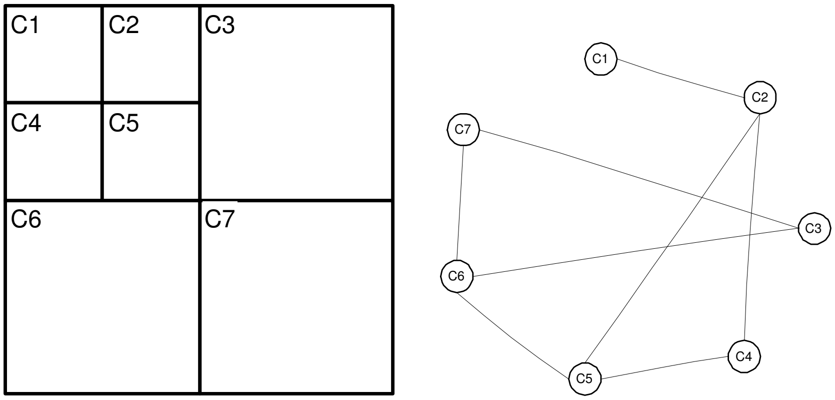
[1] It should be noted that in reference [13], the term "hot-spot" is used to denote an area of the die where the peak power consumption exceeded a given threshold. In this paper, we use the term "hot-spot" to denote areas where the die temperature exceeds the maximum allowable junction temperature

Section 3.1. It is shown through experimental results that significantly shorter test schedules can be obtained without increasing the maximum die temperature during test when compared with existing power-constrained test scheduling approaches. While this algorithm guarantees the optimal solution for a given thermal limit, it may require significant computational effort for complex systems, mainly due to the required amount of accurate thermal simulations. Therefore, a fast heuristic algorithm for thermal-aware test scheduling is proposed Section 3.3. This approach uses a low-complexity test session thermal cost model in order to speed-up the solution space exploration and reduce the thermal simulation effort required to reach an acceptable solution. The experiments show that this heuristic produces nearly-optimal test schedules (and even optimal schedules in some cases) while significantly reducing the thermal simulation effort.

2. Motivation

In this section, we examine the effectiveness of power-constrained test scheduling(PCTS) as a means of avoiding die overheating during test. The common idea behind PCTS is to impose a chip-wide maximum allowable limit on power consumption, which should not be exceeded during test application. Several recently proposed power-constrained test scheduling algorithms aim to maximise the number of tests running in parallel without exceeding this limit [2, 9, 11, 7, 6, 1, 15, 12, 13].

Silicon die hot spots resulted from localised overheating occur much faster than chip-wide overheating due to the non-uniform spatial on-die power distribution. Recent research supported by industrial observations suggests that spatial temperature gradients exceeding 30°C are possible even under typical operating conditions [20], which suggests that there are large variations in power density across the die. These gradients, especially between active and inactive blocks, are likely to increase during test since test power dissipation can be significantly higher compared to functional power [14, 19]. Having large variations in power densities across the die means that constraining the maximum chip-level power consumption is not an effective way for avoiding local overheating. We demonstrate this using the hypothetical system shown in Figure 2, which serves as an example of non-uniform power distribution. As shown in the test descriptions table from Figure 1(c), cores with different sizes, such as C1 and C3 for example, are assumed to consume the same amount of power during test. Let us consider two possible test sessions TS1 = {C2,C4,C5} and TS2 = {C3,C6,C7}. They are both valid in terms of test compatibility, as can be seen from the associated test compatibility graph shown in Figure 1(b). According to existing power constrained test scheduling approaches, both test sessions would be acceptable under a power constraint of 15W or more. We have run thermal simulations using the HotSpot tool presented in [20] on each of these test sessions, and found a large discrepancy in terms of maximum die temperature: 127.19°C for TS1 while only 66.47°C for TS2. This difference is mainly because the power density (power consumed per area unit) varies significantly from cores such as C2, C4 and C5 to cores such as C3, C6 and C7 (for example, the power density of core C2



(a) Floorplan

(b) Test compatibility graph

Core	Test power (W)	Test length (s)
C1	5W	0.4
C2	5W	0.5
C3	5W	0.6
C4	5W	0.6
C5	5W	0.7
C6	5W	0.1
C7	5W	0.1

(c) Tests

Figure 1. Example chip

is 4 times higher than that of C3). Moreover, the hottest cores in the two test sessions were C5 and C7. This is because of their reduced lateral heat removal paths: cores C5 and C7 have only 2 “cold” (inactive) neighbours (more specifically, on their “EAST” and “SOUTH” edges), while all other cores have 3 “cold” neighbours.

This example has shown that imposing a global power constraint during test cannot guarantee thermal-safety because it does not consider power densities across the die nor the clustering of “hot” cores, which can limit the lateral heat removal. In the following section, we present a new test scheduling approach which overcomes these issues.

3. Thermal-safe test scheduling

The mean time to failure (MTTF)—a commonly used metric in reliability models—is based on the Arrhenius equation, which shows that reliability is decreasing exponentially with the absolute junction temperature: $MTTF = Ae^{\frac{E_a}{kT}}$, where A is an empirical constant, E_a is the so-called activation energy and k is Boltzmann’s constant [20]. The semiconductor industry is currently using commonly accepted limits for the maximum tolerable operating junction temperature based on the device package type. These have been well accepted as numbers relating to reasonable device lifetimes and thus failure rates. For example, for devices fabricated in a molded package, the maximum allowable junction temperature is 150°C, while for devices assembled in ceramic or cavity DIP packages, the maximum allowable junction temperature is 175°C [10]. Based on these practices, the thermal-safe test scheduling approach proposed in this paper aims to produce solutions guaranteeing that the maximum allowable junction temperature will not be exceeded during test. Throughout this paper, the term “hot-spot” will be used to refer to cores that exceed the maximum allowable junction temperature during test. Any tests running below this critical temperature are considered to be “thermally safe”.

In the following, we propose two thermal-safe test scheduling algorithms. The first one, although computationally expensive, computes the exact solution to the problem, i.e., the shortest test schedule that meets the thermal constraint. The second proposed algorithm takes into consideration the on-chip lateral heat transfer paths in order to determine a nearly-optimal solution with less computational effort. The results obtained using this algorithm are then compared with the solutions obtained using the exact algorithm.

Both proposed test scheduling algorithms start from the set of cores (S) of the target system, the corresponding test compatibility graph (TCG), such as the one shown in Figure 1(b), and the maximum junction temperature that can be tolerated during test (T_{max}). Each core is annotated with the length of its corresponding test. The TCG captures the concurrency compatibility relationships between the system cores: each node in the TCG corresponds to a core, and an edge between two nodes means that the two corresponding cores can be tested concurrently without causing any resource conflicts. The floorplan of each system is also needed for performing thermal simulations on the generated test sessions. The algorithms return a thermal-safe test schedule as a list of test sessions, where each test session is a group of cores to be tested concurrently.

3.1. The exact algorithm

In this subsection we present an algorithm for determining an exact solution to the thermal-safe test scheduling problem (see Figure 2). The algorithm computes the shortest test schedule which guarantees that the specified T_{max} will not be exceeded during test. An outline of this algorithm is presented in the following. First, a thermal simulation is performed on each individual core and corresponding test in order to ensure they all comply with the thermal constraint. the shortest test schedule is computed using only the test compatibility relations between the cores. A thermal simulation is then performed to check whether the test schedule complies with the thermal constraint (T_{max}). If T_{max} is violated during any test sessions, these test sessions are discarded and the process is repeated until a thermal-safe test schedule is found.

We will explain the steps of the algorithm using the hypothetical system shown in Figure 2 assuming a thermal constraint Tmax = 110 °C. In the first stage (lines 1-6), it is ensured through thermal simulations that each core, when tested individually with all other cores inactive, does not exceed Tmax . None of the cores in our example system was found to violate the thermal constraint when tested individually. In case a thermal violation is detected, the designer needs to fix it by appropriate modifications to the core DFT infrastructure and/or test set. If the violation cannot be fixed, it means that Tmax is too restrictive and other means for reducing the core temperature are needed. Possible solutions include redesigning the cooling structures for the chip and reducing the test clock frequency. Once all cores have passed this initial check, the algorithm computes the clique set[4] (TCC) for TCG (line 7). For our example, the clique set for the TCG shown in Figure 1(b) is TCC = [[C2,C5,C4], [C5,C6], [C2,C1], [C3,C6,C7]]. Since the number of nodes in TCG (number of cores in a design) is reasonably low, we have used a straight-forward exhaustive search algorithm for determining TCC (the *all_cliques* function in Figure 2). Each clique in the TCC represents a maximal group of cores that can be tested concurrently without causing resource sharing conflicts. Consequently, any valid test schedule must consist only of subsets (TCS) of the cliques in TCC (line 8 in Figure 2). The shortest test schedule can be determined as the minimum weight set cover for TCS, where the weight of each test compatible subset in TCS is the length of its longest test (it is assumed that all tests in a test session start at the same time). For our example, TCS and the corresponding subset weights is [C2]=0.5, [C2, C5, C4]=0.7, [C5]=0.7, [C5, C6]=0.7, [C2, C5]=0.7, [C3]=0.6, [C7, C3]=0.6, [C4]=0.6, [C2, C4]=0.6, [C5, C4]=0.7, [C3, C6]=0.6, [C7, C6]=0.1, [C7]=0.1, [C2, C1]=0.5, [C1]=0.4, [C7, C3, C6]=0.6, [C6]=0.1 The minimum weight set cover is determined using a O-1 integer linear programming (ILP) formulation (the *formulate_min_weight_cover_lp* function in Figure 2). The ILP formulation for our example is shown below:

Constraint 1: $X_{TCS_0}+X_{TCS_1}+X_{TCS_4}+X_{TCS_8}+X_{TCS_{13}}=1$
Constraint 2: $X_{TCS_1}+X_{TCS_2}+X_{TCS_3}+X_{TCS_4}+X_{TCS_9}=1$
Constraint 3: $X_{TCS_6}+X_{TCS_{11}}+X_{TCS_{12}}+X_{TCS_{15}}=1$
Constraint 4: $X_{TCS_5}+X_{TCS_6}+X_{TCS_{10}}+X_{TCS_{15}}=1$
Constraint 5: $X_{TCS_{13}}+X_{TCS_{14}}=1$
Constraint 6: $X_{TCS_1}+X_{TCS_7}+X_{TCS_8}+X_{TCS_9}=1$
Constraint 7: $X_{TCS_3}+X_{TCS_{10}}+X_{TCS_{11}}+X_{TCS_{15}}+X_{TCS_{16}}=1$
Minimise: $0.5X_{TCS_0}+0.7X_{TCS_1}+0.7X_{TCS_2}+0.7X_{TCS_3}+0.7X_{TCS_4}$
$+0.6X_{TCS_5}+0.6X_{TCS_6}+0.6X_{TCS_7}+0.6X_{TCS_8}$
$+0.7X_{TCS_9}+0.6X_{TCS_{10}}+0.1X_{TCS_{11}}+0.1X_{TCS_{12}}$
$+0.5X_{TCS_{13}}+0.4X_{TCS_{14}}+0.6X_{TCS_{15}}+0.1X_{TCS_{16}}$

A value of 1 for the binary variable X_{TCS_i} , means TCS_i represents a test session in the final test schedule, and a value of 0 if it does not. A constraint is added to the ILP formulation for each core C_i , hence 7 constraints for our example, enforcing that each core should be covered by one and only one TCS in the final solution. The ILP objective is to minimise the sum of the weights (in this case the test lengths) of the test compatible subsets in the minimum weight set cover. The solution for this particular ILP is $X_{TCS_1} = 1$, $X_{TCS_{14}} = 1$ and $X_{TCS_{15}} = 1$. The corresponding test schedule will be [[C2, C5, C4],[C1],[C7, C3, C6]]. Once a valid test schedule has been computed, a thermal simulation is performed to check whether T_{max} is not exceeded during test. The following information is necessary for performing a thermal simulation for a test session: the chip floorplan, the test power values for the cores in the test session, and the corresponding test lengths. The thermal simulator produces temperature traces for each core for the entire duration of the test session, based on a user specified time step. The maximum temperature for each core can then be easily computed from these traces. Thermal simulations for the three test sessions in the previously computed test schedule produce the following results:

Test session	[C2,C4,C5]	[C1]	[C3,C6,C7]
Max temperature (°C)	127.19	98.83	66.47

These results show that the maximum die temperature during the first test session([C2,C5,C4]) violates the thermal constraint of 110 °C(line 16). Consequently, this test session is removed from TCS and a new test schedule is computed based on the updated TCS. The algorithm continues until a test schedule which does not violate the thermal constraint is found. For our example, the shortest thermal-safe test schedule is found after three such iterations and it consists of the following test sessions:

Test session	[C2,C4]	[C5]	[C1]	[C3,C6,C7]
Max temperature (°C)	103.2	106.54	98.83	66.47

In order to reach this result, a total of 5.8 seconds of test session time had to be thermally simulated.

```
INPUT: $S$, the core set for the target system
         TCG: the test compatibility graph
         $T_{max}$ = maximum tolerable temperature

OUTPUT: Thermal-safe schedule as a list of
          thermal-safe test sessions

1   foreach $C_i \in S$
2        thermal_simulation( $C_i$ )
3        if MaxTemp( $C_i$ ) $\geq T_{max}$
4             fix_thermal_violation( $C_i$ )
5        endif
6   endfor

7   TCC = all_cliques( TCG )
8   TCS = all_subsets( TCC )
9   violation = TRUE
10  while violation == TRUE do
11       violation = FALSE
12       LP = formulate_min_weight_cover_lp( S, TCS )
13       LPSol = lp_solve( LP )
14       foreach $TS_i \in LPSol$
15            thermal_simulation( $TS_i$ )
16            if MaxTemp( $TS_i$ ) $\geq T_{max}$
17                 remove $TS_i$ from TCS
18                 violation = TRUE
19            endif
20       endfor
22  endwhile

23  LPsol holds the optimum thermal-safe test schedule
```

Figure 2. Exact thermal-safe test scheduling algorithm

3.2. Experimental results for the exact algorithm

Design name	Power-constrained test scheduling		Thermal-aware test scheduling (optimal)			
	Test time(s)	Max temp.(°C)	Test time(s)	Sav.(%)	Iterations	Simulation length(s)
asic_z	0.32	70.81	0.28	12.69	1	0.28
kime	3.81	56.51	3.48	8.42	1	3.48
muresan_10	2.4	58.85	2.0	16.66	1	2.0
muresan_20	5.69	181.79	4.2	26.18	6	24.9
system_l	3.05	191.74	2.53	17.04	2	5.06
system_s	12.12	104.48	9.22	23.88	6	54.17

Table 1. Power-constrained test scheduling vs. optimal thermal-aware test scheduling

Table 1 compares the results obtained using the proposed algorithm with those obtained using the power constrained test scheduling approach presented in [7]. We have chosen the approach presented in [7] for comparison since it is very recent, has been applied to large designs and performs well in comparison with other existing power constrained test scheduling approaches. Details such as floorplan information and realistic test power and time values had to be added or modified in the original design descriptions in order to provide all necessary information for the proposed thermal safe test scheduling algorithms. The modified design descriptions used in our experiments can be found at [16]. Some of the physical constants used for thermal simulations performed with the HotSpot tool presented in [20] are reported in Table 4. The second column shows the test times corresponding to the power-constrained test schedules. Columns 4 to 7 show the results corresponding to the proposed thermal-aware test scheduling algorithm. For each design, the temperature limit T_{max} was set to the maximum temperature of the power constrained test schedule in order to see whether shorter test schedules could be obtained within the same thermal limits. Columns 4 and 5 show the test times and relative savings obtained using the thermal-safe test scheduling algorithm when compared to the algorithm presented in [7]. The experimental data shows that the proposed algorithm was able to produce up to 26% shorter test schedules without increasing the maximum die temperature during test. It should be noted that the proposed thermal-aware test scheduling approach does not pose any constraints on the overall power dissipation, hence it is possible that the resulting test schedules may exhibit higher overall power compared to the power-constrained test schedules. However, this falls outside the goal of the proposed test scheduling approach, which is only to keep the die temperature within safe limits. The last two columns show the number of iterations and the cumulated length of the thermal simulations required in each case to find a thermal-safe test schedule. One iteration consists of the process of computing a test schedule and checking if it meets the thermal constraint. For example, for *system_s*, 6 test schedules needed to be computed until a suitable (thermal-safe) solution was found. It should be noted that the most time consuming part of the algorithm is represented by the thermal simulations, which could take up to a couple of minutes per design depending on the chosen time step. Computing the clique set and solving the ILP for the minimum weight set cover were taking under one second of CPU time on a Pentium IV @ 1.8GHz system.

3.3. Heuristic algorithm

Although the algorithm presented in the previous section computes the optimal solution to the thermal-safe test scheduling problem, it requires significant computational effort, especially because it requires a large amount of thermal simulations. This is mainly because no knowledge of the heat transfer paths is used while computing the test schedule and the thermal compliance check is performed only in a post-scheduling phase. This implies that for tight thermal constraints (such as in the case of *system_s* shown in Table 1), several iterations, and thus several thermal simulation runs, are required until a valid solution is found. The thermal simulation effort required to identify thermal-safe test schedules can be reduced by exploiting the knowledge of the on-chip heat transfer paths. There are two predominant paths for heat transfer out of the integrated circuit package. The first one is from the die to the surrounding package material, then to the package lead frame and on to the printed circuit board, and finally to the ambient air. The second path is from the package to the heat spreader to the heat sink and then to the ambient air. Local die temperature is strongly dependent on the proximity with other heat sources because close heat sources means more heat has to flow through the same paths. Therefore, keeping simultaneous heat sources as far apart as possible reduces the probability of hot-spots.

In order to capture the thermal interactions between different cores that are tested concurrently, we have derived a thermo-resistive model for the test sessions. The basic idea is to derive some quantitative measure of the lateral heat removal paths for a core by taking into account the thermal interactions with active neighbouring cores.

The duality between the electrical and thermal domains, illustrated in Table 2, offers a convenient basis for an architecture-level thermal model. According to this duality relationship, heat flow can be described as a "current" passing through a thermal resistance, leading to a temperature difference analogous to a "voltage". Thermal resistance R_{th} is directly proportional to the thickness of the material (t) and inversely proportional to the cross-sectional area across which the heat is being transfered (A):

$$R_{th} = \frac{t}{kA} \tag{1}$$

where k is the thermal conductivity of the material per volume unit (100W/mK for silicon and 400W/mK for copper at 85°C).

In order to clarify how the lateral thermal resistances are computed, consider the two adjacent cores CORE1 and CORE2 shown in Figure 3. The chip thickness is t and the core dimensions are (L1,W1) and (L2, W2) respectively. The lateral resistance R_{th21} is the thermal resistance from the centre of Block 2 to the shared edge of cores 1 and 2. In this case, the heat is constricted from CORE1 to CORE2 via the surface areas defined by L1*t

Thermal domain	Electrical domain
P, heat flow, power (W)	I, current flow (A)
T, temperature difference (K)	V, voltage (V)
R_{th}, thermal resistance (K/W)	R, electrical resistance (Ω)

Table 2. The duality between the thermal and electrical domains

and L2*t . The constriction thermal resistance can be calculated by assuming the heat source area to be L1*t , the silicon bulk area that accepts the heat to be L2*t , and the thickness of the bulk to be W2/2. With these values found, the spreading/constriction resistance can be computed using the formulas given in [8]. The resistance is of the spreading type if the lateral area of the source is smaller than the bulk lateral area, and it is of the constriction type otherwise. When computing lateral thermal resistances, each core is assumed to present a thermal resistance towards each neighbouring core.

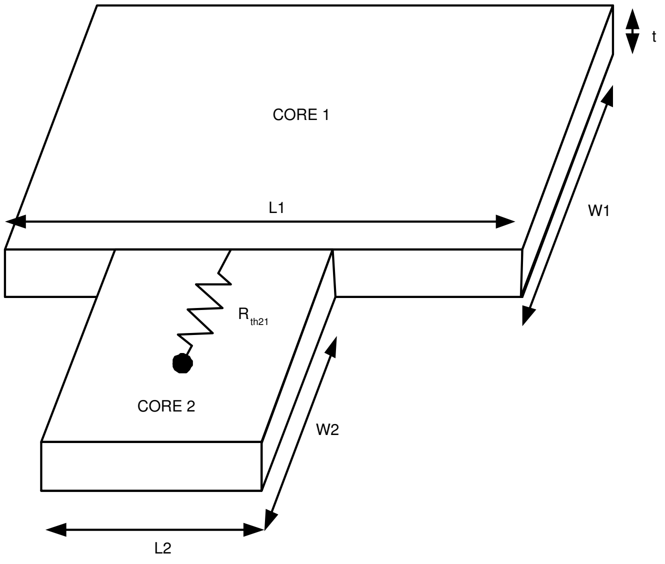


Figure 3. Lateral thermal resistance between neighbouring cores

The lateral thermo-resistive representation for the example floorplan system shown in Figure 1(a) according to the thermo-resistive model thermal model presented in [20], is shown in Figure 4. In the following we are proposing a test session thermal cost model aiming to capture the thermal effects due to the physical proximity of simultaneously active cores, since these effects can be controlled by the choice of cores that are to be active at the same time. The proposed test session thermal cost model is derived from this lateral thermo-resistive model presented using the following simplifying assumptions:

1. Only steady-state temperatures are considered as they represent upper bounds for the transient thermal profiles of individual cores. Therefore, only the thermal resistances of the generic RC model are used.

2. The heat transfer between two cores tested concurrently is considered to be negligible, hence the thermal resistance between those cores is ignored for the current test session. This is a valid assumption because the amount of exchanged heat depends on the temperature difference, which is low for cores tested at the same time.

3. Inactive cores are assumed to be thermally grounded, i.e. their temperature is assumed to be equal to the ambient temperature and fixed for the entire duration of the test session.

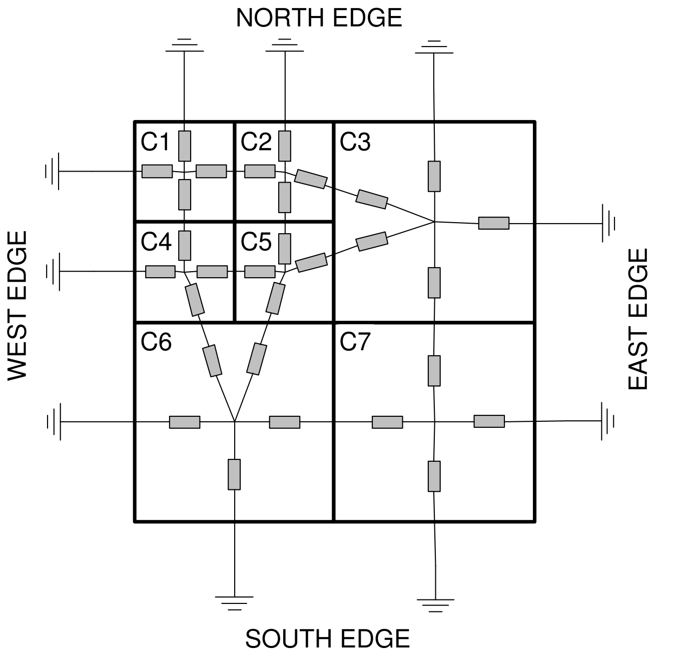
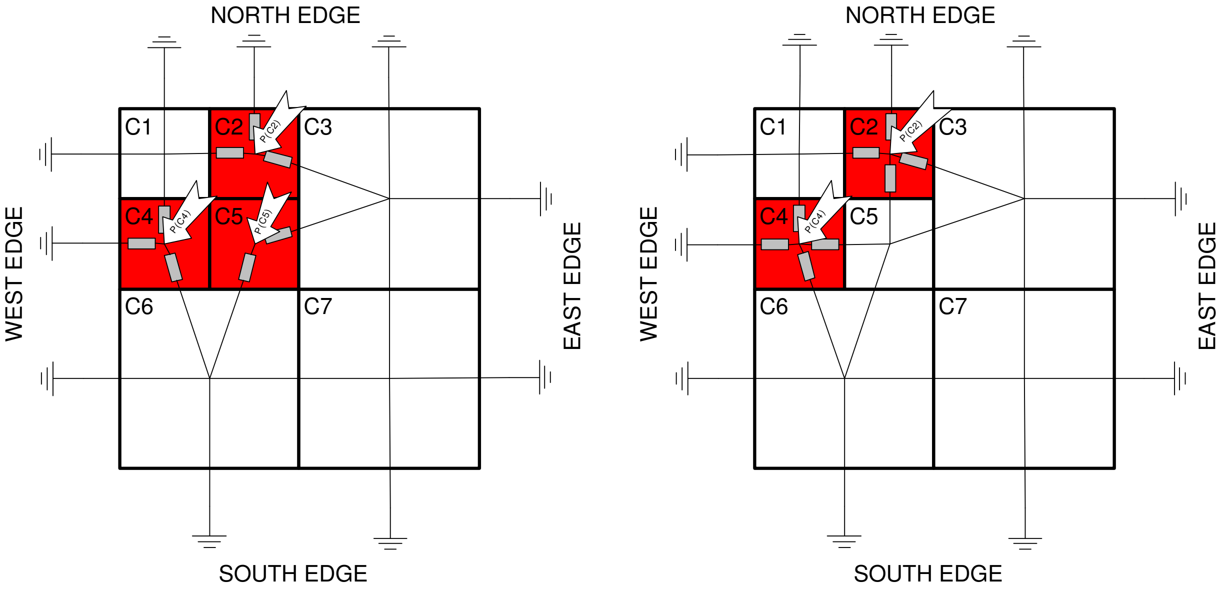


Figure 4. Lateral thermo-resistive model

Let us consider a test session consisting of cores C2,C4 and C5 from our example shown in Figure 2. The lateral thermo-resistive model derived for this test session according to the previous assumptions assumptions is shown in Figure 5(a). The white arrows pointed to the centre of the active cores signify the power dissipated by each of the cores, which is pumped away from the core through the lateral thermal resistances. As it can be observed, the thermal resistances between pairs of nodes corresponding to active cores(such as [C2,C5] and [C4,C5]) are omitted (assumption 2), while all remaining thermal resistances connect the active core nodes to the ambient, i.e. thermal ground (assumption 3). According to this model, the heat transfer paths from an active core to its cooler surroundings appear as a number of thermal resistances in parallel. For example, core C4 in Figure 5(a) has three lateral heat removal paths towards cores C1, C6 and the left chip edge (WEST EDGE). C5 does not represent a heat removal path for C4 since it is itself an active, and thus ”hot”, core. A small equivalent lateral spreading resistance



(a) (C2,C4,C5)

(b) (C2,C4)

Figure 5. Lateral thermo-resistive models for two test sessions

associated with an active core represents good heat exchange between the core and the ambient, consequently it predicts a lower core temperature during test. On the other hand, a large lateral thermal spreading resistance means poor heat exchange with the ambient, therefore it signals a potential hot-spot during test for cores with high power consumption. This can be seen by comparing the lateral thermo-resistive models shown in Figure 3.3. Each active core in Figure 5(a) has only three lateral heat removal paths, represented by three thermal resistors. Cores C2 and C4 shown in Figure 5(b) have both gained an additional lateral heat removal path through the removal of core C5 from the test session. The equivalent lateral thermal resistances of cores C2 and C4 is lower in this case compared to the scenario for test session [C2,C4,C5] shown in Figure 5(a). Thermal simulations performed on the two test sessions shown in Figure 3.3 yielded a 103.20 °C maximum temperature for [C2,C4] and a 127.19 °C maximum temperature for [C2,C4,C5], which supports our earlier observations.

The thermal cost model we are proposing for a core is basically the value of the equivalent thermal resistance towards cooler surroundings weighted by the power dissipated by that core, as shown in Equation 2. This is necessary in order to account for the actual power density of the core as well as for the lateral heat removal paths.

$$ThCost(C_i, TS) = R_{th}(C_i, TS) \times P(C_i), C_i \in TS \tag{2}$$

In order to asses the impact of the lateral heat exchange on the core temperature, we have performed the following experiment. We randomly generated a number of test sessions. For each core in each test session we computed its thermal cost according to Equation 2, and the worst-case (i.e. maximum) values were correlated with the maximum temperature reached during the execution of each test session. The high correlation coefficients obtained for several designs, shown in Table 3, suggest that lateral heat spreading has a significant influence on the maximum core temperature. The maximum core temperature during test was determined through thermal simulations using the HotSpot tool [20].

Design name	**Correlation coefficient**
asic_z	0.98
kime	0.82
muresan_10	0.94
muresan_20	0.74
system_l	0.98
system_s	0.98
Average	0.91

Table 3. Correlation between the test session thermal cost and maximum core temperature

Based on the results of the previous experiment, we are extending our thermal cost model to test sessions as follows:

$$ThCost(TS) = \max ThCost(C_i, TS), C_i \in TS \tag{3}$$

In the following, we are presenting a fast heuristic algorithm for computing thermal-safe test schedules which uses the proposed test session thermal cost model in order to reduce the required amount of thermal simulations(see Figure 6). As in the exact algorithm presented in Section 3.1, the heuristic algorithm starts by checking whether individual cores comply with the maximum allowable temperature limit Tmax (lines 1-6). Once all cores have passed this test, they are marked as available (line 7) and arranged in the descending order of their test lengths (line 8). While there still are unscheduled cores, the algorithm tries to assign them to the current test session TS. The test session TS is initially empty (line 11) and cores are added to it until no core can be added due to resource conflicts (lines 12-22). A thermal simulation is performed on TS to verify if it complies with Tmax . In our experiments we have used the Hotspot tool presented in [20], however any other thermal simulator could be used for this purpose. Consequently, the accuracy of the results are dependent on the chosen thermal simulator. According to the data reported in [20], the simulation accuracy error of Hotspot is at most 5.8% with respect to FloTherm, the commercial thermal simulator from FloWorks(http://www.floworks.com). If the maximum temperature for TS complies with Tmax , the test session is added to the test schedule and the process is repeated for the unscheduled cores. If the maximum temperature during TS exceeds Tmax , the flag GotCostLimit is set to TRUE, and a thermal cost limit is computed based on the thermal cost of TS and the fraction by which Tmax was exceeded (lines 34-37). The thermal cost of TS is computed according to Equation 3. The test session cost limit is computed as the thermal cost of TS, scaled down linearly by the fraction by which Tmax was exceeded. The thermal cost adjustment factor is computed as follows:

$$ThCostAdjust(MaxTemp(TS), T_{max}) = \frac{(MaxTemp(TS) - T_{max})}{T_{max}} \times K + 1 \qquad (4)$$

where $K \in (0, 1]$ is a user specified constant used to relax the thermal cost limit (ThCostLimit). In our experiments, we have used $K = 0.5$. Once a thermal cost limit had been computed, a core is added to the current test session only if does not increase the test session thermal cost over ThCostLimit (lines 23-38). This way, it is ensured that once a thermal violation has been detected, test sessions with similar or worse lateral heat exchange capabilities are avoided without requiring lengthy thermal simulations.

We are illustrating the steps of this algorithm using the example system shown in Figure 2. The same thermal constraint Tmax = 100 °C used for the exact algorithm in Section 3.1 will be used here as well. After the initial core check, the *Available* array is initialised with all cores in the system arranged in the descending order of their test lengths (Line 7):

$$Available = [C5, C3, C4, C2, C1, C6, C7] \qquad (5)$$

GotCostLimit is set to FALSE (Line 8) and an empty test session TS is created(Line 11). The first core added to the TS is C5. The next available core, C3, cannot be added to TS because it is not test compatible with C5 (see Figure 1(b)) (Line 15). This process continues until no more cores can be added to TS. At this moment TS = [C5,C4,C2]. A thermal simulation is performed on TS (Line 31), to determine the maximum temperature reached during TS, in this case 127.19 °C. This violates the thermal constraint of 110 °C, therefore the algorithm proceeds by setting *GotCostLimit* to TRUE and computing the thermal cost limit based on the maximum temperature reached during TS and the thermal cost value of TS: *ThCostLimit* = 35.43 (line 36). The algorithm continues by discarding TS and building a new test session, this time checking also that the thermal cost of the test session does not exceed the previously computed thermal cost limit. The first core added to TS is C5. At this point, the thermal cost for TS is 26.88. This is below the imposed limit, therefore the algorithm continues to add test compatible cores to TS (Line 26). The next core to be added is C4, which rises the thermal cost of TS to 32.01. This is still below the imposed limit, so a new core, C2 is added to TS. The thermal cost of TS becomes 39.57 which is over the imposed limit. Consequently C2 is removed from TS. Since no more cores can be added to TS, a thermal simulation is performed on TS (Line 31). The maximum temperature is found to be 117.04 °C, which violates the thermal constraint. The thermal cost limit is re-adjusted to ThCostLimit = 30.45 (Line 36) and TS is discarded. C5 is added to a new empty test session, rising it's thermal cost to 26.88. C4,C5 and C6 cannot be added to TS since the thermal cost for TS would exceed the new thermal cost limit. A thermal simulation is run for TS

=[C5], and TS is added to the test schedule and removed from the available cores since it's maximum temperature of 106.54 °C meets the thermal constraint. The GotCostLimit is set to FALSE, and the algorithm continues to schedule the remaining cores. In the next iterations, the algorithm adds [C3,C6,C7], [C2,C4] and [C1] to the test schedule. This matches the test schedule produced by the exact algorithm, however the thermal simulation effort was reduced from 5.8 seconds of test session time to 3.7 seconds. The complexity of computing a test session using this approach is $O(N^3)$, where N is the number of cores. However, the thermal simulation (line 31 in Figure 6), which is the most computationally expensive part of the algorithm is performed in the outer-most loop, which has only a complexity of $O(N)$. This is a considerable improvement in terms of the required computational effort over the exact algorithm described in Section 3.1, which has exponential complexity due to the NP-hard nature of the optimisation problem.

3.4. Experimental results for the heuristic algorithm

Chip thickness	3mm
Heatsink side	50mm
Heatsink thickness	6.9mm
Spreader side	30mm
Spreader thickness	0.5mm
Ambient temperature	45°C

Table 4. Chip physical constants

A number of experiments have been performed in order to assess the performance of the proposed test scheduling heuristic. The first set of experiments were used to compare the proposed heuristic with the power-constrained test scheduling approach presented in [7]. The results of these experiments are reported in Table 5. The maximum temperature during test corresponding to the power constrained test schedules was used as a thermal constraint (T_{max}) for the proposed test scheduling algorithm. This way, it is guaranteed that the resulting test schedules will not lead to higher temperatures than that using the power-constrained approach. From the 4th and 5th columns, it can be observed that the proposed heuristic algorithm outperformed the power-constrained test scheduling approach for all designs, producing up to 24% shorter test schedules. Moreover, for 3 out of the 6 designs considered, the heuristic algorithm produced the same test schedules as the exact algorithm presented in Section 3.1. The last column in Table 5 shows significant reductions in terms of thermal simulation effort when compared to the exact algorithm. For example, the simulation length was reduced from 54 seconds to 18 seconds for *system_s*.

Another set of experiments was performed to analyze the effect of different maximum temperature limits on the test time and simulation effort. From Table 6, it can be observed that, as expected, both test time and the simulation effort decrease as the thermal constraint becomes more relaxed. For example, test time is reduced by 25% when the thermal constraint is increased by only 5°C from 187.25°C for *muresan_20*, and it is reduced from

```
INPUT: S, the core set for the target system
       TCG: the test compatibility graph
       $T_{max}$ = maximum tolerable temperature

OUTPUT: Thermal-safe schedule as a list of
        thermal-safe test sessions

1  foreach $C_i \in S$
2      thermal_simulation( $C_i$ )
3      if MaxTemp( $C_i$ ) $\geq T_{max}$
4          fix_thermal_violation( $C_i$ )
5      endif
6  endfor

7  Available = { $C_i | C_i \in S$ }
8  sort_descending( Available )
9  Hsol = $\emptyset$
10 GotCostLimit = FALSE
10 while Available $\neq \emptyset$ do
11     TS = $\emptyset$
12     foreach CandidateCore $\in$ Available do
13         IsCompatible = TRUE
14         foreach $C \in$ TS do
15             if C not compatible with CandidateCore
16                 IsCompatible = FALSE
17                 gotoline 20
18             endif
19         endfor
20         if IsCompatible = TRUE
21             if GotCostLimit = FALSE
22                 addCandidateCore to TS
23             else
24                 ThCost = computeThCost( TS U {CandidateCore} )
25                 if ThCost $\leq$ ThCostLimit
26                     add CandidateCore to TS
27                 endif
28             endif
29         endif
30     endfor

31     thermal_simulation( TS )
32     if MaxTemp(TS) $\leq T_{max}$
33         add TS to Hsol
34     else 35     GotCostLimit = TRUE
36         ThCostLimit = computeThCost( TS ) $\times$ ThCostAdjust( MaxTemp(TS), $T_{max}$ )
37     endif
38 endwhile

39 Hsol holds the thermal-safe test schedule
```

Figure 6. Heuristic thermal-safe test scheduling algorithm

9.22 seconds to 8.44 seconds when the thermal constraint is increased from 109.85°C to 114.85°C for *system_s*. Even more reductions are obtained in terms of simulation effort. For example, for *system_s*, increasing the thermal constraint from 109.85°C to 114.85°C reduced the simulation effort by half, from nearly 18 seconds to less than 8.5 seconds.

Table 7 compares the proposed heuristic and the exact thermal-safe test scheduling algorithms. As mentioned

Design name	Power-constrained test scheduling		Thermal-aware test scheduling (heuristic)			
	Test time(s)	Max temp.(°C)	Test time(s)	Sav.(%)	Violations	Simulation length(s)
asic_z	0.32	70.81	0.28	12.69	0	0.28
kime	3.81	56.51	3.48	8.42	0	3.48
muresan_10	2.4	58.85	2.0	16.66	1	2.4
muresan_20	5.69	181.79	4.89	14.05	1	6.0
system_l	3.05	191.74	2.87	5.09	0	2.87
system_s	12.12	104.48	9.22	23.88	1	17.67

Table 5. Power constrained test scheduling vs. heuristic thermal-aware test scheduling

Design name	Max temp.(°C)	Test time(s)	Simulation length(s)
asic_z	71.15	0.28	0.28
	76.15	0.28	0.28
	81.15	0.28	0.28
	86.15	0.28	0.28
	91.15	0.28	0.28
kime	56.84	3.48	3.48
	61.84	3.48	3.48
	66.84	3.48	3.48
	71.84	3.48	3.48
	76.84	3.48	3.48
muresan_10	59.18	2.0	2.4
	64.18	2.0	2.0
	69.18	2.0	2.0
	74.18	2.0	2.0
	79.18	2.0	2.0
muresan_20	182.25	4.89	6.0
	187.25	4.89	6.0
	192.25	4.49	4.49
	197.25	4.49	4.49
	202.25	4.49	4.49
system_l	194.90	2.87	2.87
	199.90	2.87	2.87
	204.90	2.87	2.87
	209.90	2.87	2.87
	214.90	2.87	2.87
system_s	104.85	9.22	17.67
	109.85	9.22	17.67
	114.85	8.44	8.44
	119.85	8.44	8.44
	124.85	8.44	8.44

Table 6. Test times for different temperature constraints

earlier, the heuristic determines the optimum solution for 4 out of the 6 designs considered. In only one case, the required thermal simulation effort exceeded that required by the exact algorithm, while in all other cases up to 75% reductions have been obtained.

Design name	Test time increase(%)	Simulation effort reduction(%)
asic_z	0	0
kime	0	0
muresan_10	0	-20
muresan_20	16.4	75.9
system_l	13.4	43.28
system_s	0	67.38

Table 7. Comparison between the exact and the heuristic thermal-aware test scheduling algorithms

4. Conclusions

Overheating has been acknowledged as a major problem during the testing of complex system-on-chip (SOC) integrated circuits. In this paper, we have outlined the need for thermal-safe testing and explained that existing power-constrained test scheduling approaches cannot guarantee thermal safety during test. Next, we have proposed a new test scheduling approach that produces short test schedules and guarantees thermal-safety during test at the same time. Two possible algorithms have been developed for the proposed thermal-safe test scheduling approach. The first proposed algorithm, although computationally expensive, provides an optimal solution to the thermal-safe test scheduling problem. The second algorithm uses a fast heuristic based on a low-complexity test session thermal model in order to reduce the required computational effort while producing optimal or near-optimal test schedules. Experimental results show that up to 24% shorter test schedules can be obtained using the proposed approach without increasing the maximum temperature during test application, when compared to power constrained test scheduling approaches. The proposed approach provides an effective solution to the problems arising from chip overheating during test.

5. Acknowledgements

P. Rosinger and B. M. Al-Hashimi acknowledge the Engineering and Physical Sciences Research Council (EPSRC) for funding this work under grant no. GR/S05557. The work of K. Chakrabarty was supported in part by the US National Science Foundation under grant no. CCR-0204077. The authors wish to acknowledge Erik Larsson from Linkoping University, Sweden for providing the code and designs used for the work presented in reference [7].

References

[1] K. Chakrabarty. Design of system-on-a-chip test access architectures under place-and-route and power constraints. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pages 432–437, 2000.

[2] R. Chou, K. Saluja, and V. Agrawal. Scheduling tests for VLSI systems under power constraints. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2):175–184, June 1997.

[3] P. Flores, J. Costa, H. Neto, J. Monteiro, and J. Marques-Silva. Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation. In *12th International Conference on VLSI Design*, pages 37–41, 1999.

[4] A. Gibbons. *Algorithmic graph theory*. Cambridge University Press, 1985.

[5] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac. Reducing power consumption during test application by test vector ordering. In *Proc. International Symposium on Circuits and Systems (ISCAS)*, pages 296–299, 1998.

[6] V. Iyengar and K. Chakrabarty. System-on-a-chip test with precedence relationships, preemption and power constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21:1088–1094, September 2002.

[7] E. Larsson, K. Arvidsson, H. Fujiwara, and Z. Peng. Efficient test solutions for core-based designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(5):758–775, May 2004.

[8] S. Lee, S. Song, V. Au, and K. Moran. Constricting/spreading resistance model for electronics packaging. In *ASME/JSME Thermal Engineering Conference*, pages 199–206, 1995.

[9] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu. A comparison of classical scheduling approaches in power-constrained block-test scheduling. In *Proc. IEEE International Test Conference (ITC 2000)*, pages 882–891, 2000.

[10] National Semiconductor. *Understanding Integrated Circuit Package Power Capabilities*, April 2000. http://www.national.com/ms/UN/UNDERSTANDING_INTERGRATED_CIRCUIT_PACKAGE_POWER_CA.pdf.

[11] N. Nicolici and B. Al-Hashimi. Power conscious test synthesis and scheduling for BIST RTL data paths. In *Proc. IEEE International Test Conference (ITC 2000)*, pages 662–671, October 2000.

[12] M. Nourani and J. Chin. Power-time trade off in test scheduling for SoCs. In *Proc. IEEE International Conference on Computer Design(ICCD)*, pages 548–553, October 2003.

[13] M. Nourani and J. Chin. Test scheduling with power-time tradeoff and hot-spot avoidance using MILP. *IEE Proceedings - Computers and Digital Techniques*, 151(5):341–355, September 2004.

[14] B. Pouya and A. Crouch. Optimization trade-offs for vector volume and test power. In *International Test Conference (ITC)*, pages 873–881, 2000.

[15] C. P. Ravikumar, G. Chandra, and A. Verma. Simultaneous module selection and scheduling for power-constrained testing of core based systems. In *13th International Conference on VLSI Design*, pages 462–467, 2000.

[16] P. Rosinger. Sample designs for validating thermal-aware test solutions. 2005. http://www.ecs.soton.ac.uk/ pmr/thermal_test_sample_designs.zip.

[17] P. Rosinger, B. Al-Hashimi, and N. Nicolici. Scan architecture with mutually exclusive scan segment activation for shift and capture power reduction. *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, pages 1142–1154, 2004.

[18] J. Saxena, K. M. Butler, and L. Whetsel. An analysis of power reduction techniques in scan testing. In *IEEE International Test Conference(ITC)*, pages 670–677, 2001.

[19] C. Shi and R. Kapur. How power aware test improves reliability and yield. In *EETimes*. September, 15 2004. http://www.eetimes.com/news/design/features/showArticle.jhtml?articleId=47208594&kc=4235.

[20] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *International Symposium on Computer Architecture (ISCA)*, pages 2–13, 2003.

[21] S. Wang and S. K. Gupta. DS-LFSR: A new BIST TPG for low heat dissipation. In *Proc. IEEE International Test Conference*, pages 848–857, 1997.

[22] S. Wang and S. K. Gupta. ATPG for heat dissipation minimization during test application. *IEEE Transactions on Computers*, 47(2):256–262, February 1998.