

An Efficient Temperature-Gradient Based Burn-In Technique for 3D Stacked ICs

Nima Aghaee, Zebo Peng, and Petru Eles

Embedded Systems Laboratory (ESLAB), Linköping University, Sweden
{nima.ghaee, zebo.peng, petru.eles}@liu.se

Abstract—Burn-in is usually carried out with high temperature and elevated voltage. Since some of the early-life failures depend not only on high temperature but also on temperature gradients, simply raising up the temperature of an IC is not sufficient to detect them. This is especially true for 3D stacked ICs, since they have usually very large temperature gradients. The efficient detection of these early-life failures requires that specific temperature gradients are enforced as a part of the burn-in process. This paper presents an efficient method to do so by applying high power stimuli to the cores of the IC under burn-in through the test access mechanism. Therefore, no external heating equipment is required. The scheduling of the heating and cooling intervals to achieve the required temperature gradients is based on thermal simulations and is guided by functions derived from a set of thermal equations. Experimental results demonstrate the efficiency of the proposed method.

I. INTRODUCTION

Burn-in is a common way of accelerating and detecting early-life failures, and should be done with low cost in a reasonably short time. For this purpose, usually the dies are operated at elevated temperature and voltage. The elevated temperature and voltage speed up the aging and wear mechanisms so that the dies experience their early life before testing. The wear mechanisms that are speeded up include metal stress voiding and electromigration, metal slivers bridging shorts, as well as gate-oxide wear-out and breakdown [11].

Recently several studies have, however, shown that some wear mechanisms are speeded up more efficiently by large temperature gradient rather than the high temperature itself. A temperature-gradient induced wear mechanism is identified in [12] which shows that a metal layer elevation happens rapidly at the points on the die that are experiencing a large temperature gradient. Moreover, in the atomic flux equation, used to model electromigration, temperature gradient is present directly and also indirectly through its effect on the mechanical-stress gradient [10]. Therefore, a burn-in process that has not created the appropriate thermal scenarios do not sufficiently speed up the formation of the defects that depend on large temperature gradients and consequently such early-life defects will go undetected. In order to prevent these test escapes, it is necessary to introduce a burn-in process that enforces appropriate temperature scenarios on the IC. This necessity is more urgent for the ICs that suffer from large temperature gradients, such as 3D-Stacked ICs (3D-SIC), which have considerably larger temperature gradients compared with 2D ICs (three times is reported in [13]). Moreover, 3D-SIC technology is one of the most promising future technologies [8]. Therefore, in this paper we focus on 3D-SICs.

3D-SIC technology, similar to other deep submicron technologies, suffers from high power densities. Additionally, power densities are considerably higher in the test mode compared to the functional mode, in particular for core-based designs [4]. The temperatures in the test mode could actually be high enough to damage the IC because of overheating [2, 9, 16]. This means that the application of test stimuli can raise the ICs' temperatures to their tolerable limits for large deep-submicron ICs and in particular 3D-SIC. This often undesirable effect is, however, utilized in this paper to heat up the IC for burn-in. The stimuli that are used to aggressively

heat up the IC are called *heating sequences*. The use of the heating sequences to heat up the IC from inside means that special equipment for heating the IC from outside are not necessary. This will lead to large reduction of cost, and also the uneven distribution of heat in different parts of an IC, thus creating temperature gradients.

The heating sequences are sent through the Test Access Mechanism (TAM) [1] that provides access to the cores in the test mode. One reason for utilizing TAM is as follows. It is likely that certain temperature gradients that must be enforced are in unusual locations and with unusual differences. Such gradients are not achievable if the IC is driven by its functional input ports, but they can be achieved if the TAM is used. The reason is that the TAM, in the test mode, provides direct access to cores while in the functional mode a core might be limited to receive inputs only from a particular core. Therefore only by using the TAM, heating could be precisely targeted toward a specific core. In this paper a technique to enforce a set of given specified temperature scenarios using available TAMs is proposed.

II. RELATED WORKS

Traditionally burn-in is performed at elevated temperature, which is achieved by special equipment (e.g., temperature chambers), and elevated voltage [11]. This approach will not be able to achieve the specified temperature gradients, especially those with large magnitudes.

Several works that are not directly related to burn-in but are, in methodology, similar to our proposed technique are briefly reviewed as follows. A thermal-aware test scheduling approach is introduced in [14] for stacked multi-chip modules, which tries to achieve a vertical uniform temperature distribution throughout the 3D IC during the test. A linear programming approach is used in [9] to generate thermally-safe test schedules for 3D-SICs.

Two different approaches for multi-core ICs are introduced in [6] and [15] to guarantee that the cores' temperatures are kept within the specified range when the corresponding tests are applied. They focus on the temperature of the individual cores that are under test and the temperatures of other cores are neglected.

Speeding up the test by carefully planning safety margins that counteract negative effects of process variation is addressed in [2, 3]. The test temperatures are kept sufficiently low by introducing cooling cycles into the test schedule. The cooling cycles are carefully planned using thermal simulations. A fast thermal simulation technique is suggested in [3].

These existing methods for controlling the chips' temperatures during test try to respect a global upper temperature limit to prevent overheating or to respect upper and lower bounds for individual cores in order to target temperature dependent defects. In both cases, the temperature bounds are defined for each core independent from other cores and therefore spatial thermal gradients cannot be planned. To our knowledge, there is no existing method for constructing the specified temperature gradients on an IC for burn-in. In this paper we present a technique to rapidly achieve the specified temperature gradients and then maintain them for a given period of time in order to achieve the intended burn-in effects.

III. PROBLEM FORMULATION

Assume that there are M modules in an IC (on one or multiple dies) and their tests could be started and stopped independently (e.g., the modules are cores with core wrappers in a core-based design). In order to enforce the specified temperature gradients, heating sequences are applied to heat up the modules. A heating sequence consists of real or dummy test stimuli with large switching activities. The average power of a heating sequence is given as a real number, denoted as p_m^{HS} for module m ($0 \leq m < M$). It is assumed that the TAM only affords W (a positive integer number) modules to be accessed simultaneously¹.

The temperature gradients that speed up the early-life of the targeted defects are specified as thermal maps. A thermal map specifies, for all modules, the temperature bounds that should be respected simultaneously in order to enforce the specified spatial temperature gradients on the IC. A thermal map is achieved when all the specified modules in the IC have the temperature values specified by the map. For certain thermal maps, the temperatures of some modules on the IC might not be important. Such modules are indicated as don't-cares. Even though they are marked as don't-cares, their temperatures must, however, be kept below the overheating limit considering a safety margin (denoted by $\theta_{overheating}$) in order to prevent damage.

The thermal maps are assumed as given by the user, who studies the typical temperature-gradient induced failure mechanisms in an IC analytically or experimentally [10, 12]. A given thermal map is specified by a low and a high temperature limit for each module and the don't-care modules are declared separately. A thermal map specifies that module m has a low temperature limit equal to θ_m^L and a high temperature limit equal to θ_m^H . There is a set of thermal maps that have to be achieved and maintained. It is, therefore, important to achieve them very fast whether starting from room temperature or from another map.

The inputs to the proposed method include thermal maps, IC's thermal model, IC's electrical model (e.g., specification of the TAM and power-related specifications), switching activities of the heating sequences, and ambient temperature ($\theta^{ambient}$). The output is a schedule that guides the application of the heating sequences to the modules so that their temperatures move into the specified ranges and stay there.

As an example, consider an IC with 3 modules. Assume that a thermal map is specified as $\theta_0^H = 125^\circ\text{C}$, $\theta_0^L = 115^\circ\text{C}$, $\theta_1^H = 95^\circ\text{C}$, $\theta_1^L = 85^\circ\text{C}$, $\theta_2^H = 65^\circ\text{C}$, and $\theta_2^L = 55^\circ\text{C}$, and no module is specified as don't-care. These temperature limits are shown in Fig. 1a with dashed/dotted lines. A temperature simulation is performed for this IC based on a proper periodic schedule and the simulated temperatures are shown in Fig. 1a. Starting from the ambient temperature ($\theta^{ambient} = 30^\circ\text{C}$), the modules' temperatures steadily raise until they are inside the specified ranges. As shown in this example, applying heating sequences can drive the modules of an IC into a high temperature situation.

The temperatures in one period around 6×10^4 TU point, are amplified and shown in Fig. 1b. Since the schedule is periodic, one

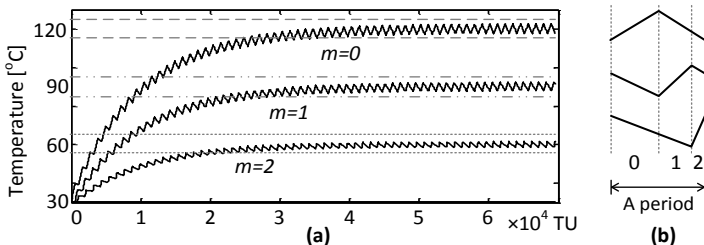


Figure 1. Temperature curves for an example.

period represents the entire schedule. In this example, the TAM provides access to one module at a time ($W = 1$), therefore in interval 0 in Fig. 1b only module m_0 receives heating sequence. Similarly, in interval 1 only m_1 is heated and the same goes for interval 2 for m_2 . The main task of our work is to design an algorithm to generate such schedules.

IV. STEADY STATE SOLUTION

Let us first analyze a simplified situation, where we assume that a steady state power could be provided for the modules. In this case, a steady state solution exists that could generate and maintain the specified thermal map. This means that the desirable steady state temperatures should be obtained so that the powers can be calculated based on them.

The desired steady state temperatures are defined at the middle of the specified ranges $\theta_m^{SS} = \frac{1}{2} \times (\theta_m^L + \theta_m^H)$ to reduce the risk of out of range temperatures due to temperature ripples. Such ripples could be seen in the temperature curves given in Fig. 1. In order to find the power values that result in the specified temperatures, the IC's thermal model should be analyzed. A widely used thermal model is the lumped element thermal model, as used in HotSpot [7]. Such a model divides an IC into elements called nodes. Each node has a heat capacitance modelling its thermal capacity. Adjacent nodes are connected through a heat resistance that models the thermal conductivity between them. A node is called active if it directly receives electrical power caused by switching activities. Detailed information of such models can be found in [5, 7].

In this section it is assumed that a module is a single active thermal node. Furthermore, it is assumed that TSV blocks are always thermally don't-care and do not dissipate heat since their drivers are placed and considered together with the corresponding modules. These restrictions will be relaxed later on in this paper.

All the characteristics of the thermal model are captured in two matrices \mathbf{A} and \mathbf{B} . The thermal behavior of an IC is captured in the following system of ordinary differential equations [3].

$$\mathbf{A} \times \frac{d}{dt} \boldsymbol{\theta} + \mathbf{B} \times \boldsymbol{\theta} = \mathbf{P} \quad (1)$$

In this equation, $\boldsymbol{\theta}$ is the temperature vector and \mathbf{P} is the power. Note that because of the heat transfer a node may heat up even if it does not have switching activities.

A thermal map could be thought as the targeted steady state temperatures (θ_m^{SS} for module m). Since θ_m^{SS} is constant for a map, the temperature derivatives are zero (no variation in time). Therefore, equation 1 could be written as

$$\mathbf{P}^{SS} = \mathbf{B} \times \boldsymbol{\theta}^{SS}. \quad (2)$$

This means that it is possible to calculate the required powers that lead to the specified thermal map. In order for the specified thermal map to be achievable, the computed steady state power values must satisfy a feasibility and a schedulability condition. The first part of the feasibility condition is that the computed steady state power for a module m (p_m^{SS}) should be larger than or equal to the stray power dissipation of the module. The stray power is an unintended part of the power that could not be independently controlled with available test controls. Its value for module m is denoted by \bar{p}_m . It consists of the leakage power in addition to the clock networks' power. The second part of the feasibility condition is that p_m^{SS} should be less than or equal to the average power of the corresponding heating sequence, p_m^{HS} , plus \bar{p}_m . The feasibility condition is, therefore, as follows:

$$\forall m, \bar{p}_m \leq p_m^{SS} \leq (p_m^{HS} + \bar{p}_m) \quad (3)$$

Usually the feasibility condition is easily met if the specified thermal map is realistic (for example the specified temperature is not lower than the ambient). Assuming that equation 3 is satisfied, the schedulability condition which is related to the limited TAM

¹The problem formulation will slightly change later on to accommodate more sophisticated TAM bandwidth limitations.

bandwidth should be verified. The challenging problem here is to create the required average power values, \mathbf{P}^{SS} , using the available TAM bandwidth. This is done by selectively applying the heating sequences to the modules.

The continuous application of the heating sequence generates an average dynamic power equal to p_m^{HS} . The desired power values, p_m^{SS} , which are smaller than $p_m^{HS} + \bar{p}_m$, are created by applying the heating sequence, p_m^{HS} , for a fraction of a time period. The average power in a period should be made equal to the required steady state power. This is done using a technique similar to Pulse-Width Modulation (PWM). The ratio of the duration of heating sequence application to the overall time period is therefore called Duty-cycle (D_m).

$$D_m = \frac{(p_m^{SS} - \bar{p}_m)}{p_m^{HS}} \quad (4)$$

These duty-cycles might not be achievable if their values are relatively large and if the TAM does not provide sufficient bandwidth. For example, assume a design with two modules, m_0 and m_1 . Assume that the duty-cycles are $D_0 = 0.6$ and $D_1 = 0.8$. This means that in a period of time equal to 1, we need access to m_0 for 60% of the time and access to m_1 for 80% of the time. Therefore, simultaneous access to more than one module ($0.6 + 0.8 = 1.4$ modules) is required. This means that the TAM should provide simultaneous access to these two modules otherwise these duty-cycles are not schedulable and the specified thermal map is not achievable. The feasibility condition and the schedulability condition can be written as:

$$\forall m, 0 \leq D_m \leq 1, \text{ and} \quad (5)$$

$$\sum_{m=0}^{M-1} D_m \leq W$$

Given a thermal map that satisfies both feasibility and schedulability conditions, it is relatively simple to develop a schedule to deliver the required duty cycles. Fig. 2a gives an illustrative example, where the available parallelism, W , provided by the TAM is represented by the number of rows that could be filled with duty-cycles, D_m s ($W = 3$). The scheduling algorithm starts by sorting the duty-cycles and then allocating them from the largest one to the smallest ones by filling the rows from the lowest one upwards. Note that a module needs to switch at most twice during a period and therefore the switching overheads are negligible. The fractions of the time period that the modules receive heating sequences are illustrated in Fig 2b. At every moment in time only three modules are receiving their heating sequences (the TAM limitation is not exceeded), and the average of applied heating sequence for a module is equal to the specified steady state power.

As mentioned before, a thermal map may leave the temperatures for some nodes unspecified (don't-care nodes). Besides, the temperatures for inactive thermal nodes (e.g., TSV blocks) are also left unspecified. On the other hand, in order to compute the steady state powers, these temperatures should also be known. The proper choice of temperatures for the don't-care nodes may help a thermal map that is otherwise not schedulable become schedulable. The problem of finding proper temperature values for don't-care nodes is

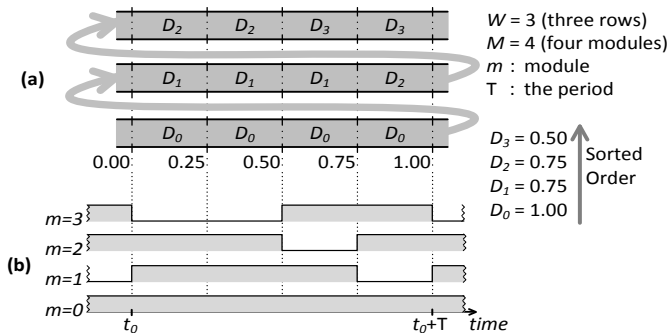


Figure 2. An example for scheduled duty-cycles.

formulated as a Linear Programming (LP) problem. Since we are more interested in knowing the duty cycles than the temperatures, the problem formulation is, then, written with the duty cycles as decision variables, as shown in Fig. 3. If the LP solver finds a feasible solution, then the thermal map is achievable and the duty cycles are returned by the LP solver.

The period should be short enough so that the fluctuations in the temperatures do not violate the specified limits. On the other hand, a longer period is desirable to minimize switching. An algorithm based on linear estimation is used in this paper to calculate the proper periods. An example for the results could be seen in Fig. 1. After the temperatures have completed their transitions to their new values (after 4×10^4 TU), the proper choice of the period keeps them inside the specified ranges, albeit relatively large fluctuations caused by relatively low number of switching actions in the schedule.

V. TRANSIENT-BASED HEURISTIC

A transient-based heuristic is designed so that active nodes/modules are no longer obliged to have individual access to TAM. Consequently, a module can be further divided into a number of thermal nodes. The overall number of nodes is represented by a positive integer N ($M \leq N$). The desired thermal map is specified for thermal nodes (instead of modules in previous section). Consequently, the thermal map specifies that node n has low temperature limit equal to θ_n^L and high temperature limit equal to θ_n^H .

In this new approach, the switching activities for heating sequences are more specific and provide information concerning the power breakdown among active thermal nodes. For example, instead of only one heating sequence for module m , there are two heating sequences corresponding to two active thermal nodes n and o (if module m is divided into nodes n and o). The average power of a heating sequence for active node n is represented by p_n^{HS} . Other active nodes of that module (e.g., node o) may also receive power, denoted by $p_{n,o}^{HS}$. Therefore, when trying to heat up node n with p_n^{HS} , node o is also heated by $p_{n,o}^{HS}$.

Furthermore, power dissipation for TSV blocks is supported (TSV drivers/buffers may be placed in TSV blocks) and their desired temperatures might also be specified in the thermal maps. Up till now it was assumed that the TAM only affords W modules to be tested simultaneously. But in this section we only need to know that at each moment which modules have access to TAM.

The proposed technique is based on applying the power in two different modes, a *thermal boost* mode which is followed by a *thermal rest* mode. During the boost, the temperatures can be outside the specified ranges. When the temperatures reach inside the specified ranges, the thermal rest mode takes over. Since a very short transition time is desirable, the highest possible power should be continuously applied during boost (assuming that the new map's temperature is higher). The proposed method works as follows: Boosting of an active node stops when the node reaches the Stop Boosting temperature, θ_n^{SB} . The stop boosting temperatures may be higher than the high temperature limit, θ_n^H . The temperatures in the boost mode are kept below $\theta_{overheating}$. Moreover, the duration of the boost mode is very short. Therefore, boost mode is thermally safe and it has no significant effect on the wear mechanisms.

Decision variables:	$D_m ; m = 0, 1, \dots, M - 1$
Objective:	<i>find a feasible solution</i>
Constraints:	$\begin{cases} \theta_m = \theta_m^{SS}, m \text{ is specified} \\ \theta_{ambient} \leq \theta_m < \theta_{overheating}, m \text{ is don't care or inactive} \\ D_m = 0, m \text{ is inactive} \\ 0 \leq D_m \leq 1, m \text{ is active} \\ \sum_m D_m \leq W ; m = 0, 1, \dots, M - 1 \end{cases}$

Figure 3. Linear programming formulation.

A node's temperature will naturally decrease if no power or little power is applied to it, but it should not fall below the low temperature limit. Therefore, a heating sequence should be applied at some point, before the temperature falls out of range. This point is marked with a temperature value named *Heating Trigger* and denoted by θ_n^{HT} for active thermal node n ($\theta_n^{HT} > \theta_n^L$). The heating sequence should be applied when the temperature of node n falls below θ_n^{HT} . The difference between θ_n^{HT} and θ_n^L provides sufficient time for the node to wait for gaining access to the TAM without its temperature falling below θ_n^L . The heating should stop when the temperature reaches the high temperature limit. The time that it takes to get back to the low temperature limit, could be utilized to heat up other nodes that need heating.

The nodes that simultaneously require heating should be accommodated within the available bandwidth of the TAM. This bandwidth might not be sufficient for all of them and therefore some of them should be prioritized. The priorities for using the TAM are determined based on the *regional need for heating* (denoted by d_n) around a node n ($d_n \in \mathbf{D}$). It is similar to the duty cycles in the previous sections and it is obtained using the following procedure. In the following the regional need for heating is introduced for the thermal rest mode. Equation 1 could be estimated as

$$\frac{\mathbf{A} \times (\theta^{HT} - \theta)}{T} + \mathbf{B} \times \theta = \mathbf{D} \times \mathbf{P}^{HS} + \bar{\mathbf{P}}. \quad (6)$$

The equation is written for one cycle (the period is T) that is assumed to be small. Equation 6 is then solved for the nodes that need heating as follows.

$$d_n = \frac{\sum_{k=0}^{N-1} a_{n,k} \times (\theta_k^{HT} - \theta_k)}{T} + \frac{\sum_{k=0}^{N-1} b_{n,k} \times \theta_k - \bar{p}_n}{p_n^{HS}} \quad (7)$$

d_n depends on (1) the required heating for node n , (2) the required heating that is related to the adjacent nodes, and (3) the average power of the corresponding heating sequence, p_n^{HS} . The elements of matrices \mathbf{A} and \mathbf{B} , ($a_{n,k}$ and $b_{n,k}$) are so that the regional need for heating has the highest dependency on the node itself, and a smaller dependency on the adjacent nodes.

The priorities in thermal boost mode are computed in a similar manner by replacing θ_n^{HT} with θ_n^{SB} (e.g., in equations 6 and 7). Efficient values for stop boosting and heating trigger temperatures for each map are found using an optimization metaheuristic similar to [2].

The output for the steady state solutions is a periodic offline schedule and therefore producing a small periodic schedule is one of its advantages. A periodic schedule means that there is a constant average power for each module, despite the fact that a higher or lower average power might be suitable for different periods. The transient-based heuristic addresses this issue by generating a non-periodic offline schedule that facilitates the heating for the nodes that need it the most. Furthermore, the introduction of the boost mode helps to reduce the switching overheads in the schedule.

VI. EXPERIMENTAL RESULTS

The proposed methods are evaluated for twelve experimental ICs with one to three layers, as detailed in row 1 Table I. There are two, four, eight, and sixteen physical modules per layer, resulting in the total number of modules ranging from two to forty eight, as given in row 2. The dies are assumed to be stacked in a face to back configuration.

The thermal models are extracted using an approach similar to the method proposed in [5] for 3D-SIC. The heating patterns' switching activities are generated using Markov chains, similarly as in [16]. The valid ranges in thermal maps are randomly selected

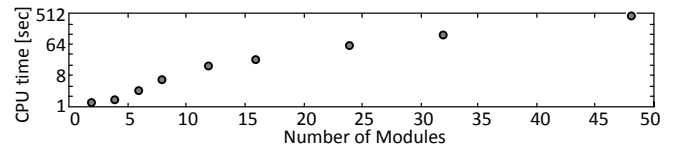


Figure 4. CPU time versus number of modules.

between 35°C and 95°C. Only thermal maps that can be achieved in practice are considered.

The CPU time to generate the schedules for the steady state method for all of the twelve experimental ICs together is about 2 seconds while the transient-based heuristic completes in about 12 minutes. The percentage changes in burn-in times are given in rows 3 of Table I. Considerable speed up (78% in average) is achieved by the transient-based heuristic.

CPU times for the transient-based heuristic for different number of modules are given in Fig. 4. Even though the CPU time grows rapidly with the increase in the number of modules, for an IC with 48 modules it is still relatively short (480 sec).

VII. CONCLUSIONS

Early-life failures that depend on temperature-gradients introduce a challenge for achieving an efficient burn-in process, in particular for 3D-SIC. In order to properly detect these defects, some specific thermal gradient must be enforced on the IC as part of the burn-in process. The technique proposed in this paper utilizes the available test access mechanisms in order to selectively apply high-power stimuli to the IC. Therefore, there is no need for expensive equipment to heat up the chip from outside and it makes it possible to generate large temperature gradients, which are otherwise impossible to produce. The experimental results show that the proposed technique generates an efficient schedule in a reasonably short time. To our knowledge, this is the first technique to achieve burn-in without any external heating mechanism.

REFERENCES

- [1] S. Adham and E. J. Marinissen. <http://grouper.ieee.org/groups/1838/>.
- [2] N. Aghae, Z. Peng, and P. Eles, "Process-variation and temperature aware SoC test scheduling using particle swarm optimization," IDT 2011.
- [3] N. Aghae, Z. Peng, and P. Eles, "Process-variation and temperature aware SoC test scheduling technique," JETTA 2013.
- [4] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Test power: a big issue in large SOC designs," DELTA 2002.
- [5] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," DATE 2009.
- [6] Z. He, Z. Peng, and P. Eles, "Multi-temperature testing for core-based system-on-chip," DATE 2010.
- [7] W. Huang et al., "Compact thermal modeling for temperature-aware design," DAC 2004.
- [8] E. J. Marinissen, "Challenges and emerging solutions in testing TSV-based 2.5D- and 3D-stacked ICs," DATE 2012.
- [9] S. K. Millican et al., "Linear programming formulations for thermal-aware test scheduling of 3D-stacked integrated circuits," ATS 2012.
- [10] J. Pak et al., "Modeling of electromigration in through-silicon-via based 3D IC," ECTC 2011.
- [11] O. Semenov, A. Vassighi, M. Sachdev, A. Keshavarzi, and C. F. Hawkins, "Effect of CMOS technology scaling on thermal management during burn-in," *IEEE Trans. Semicond. Manuf.*, 2003, vol. 16, no. 4.
- [12] T. Smorodin, J. Wilde, P. Alpern, and M. Stecher, "A temperature-gradient-induced failure mechanism in metallization under fast thermal cycling," *IEEE Trans. Device Mater. Rel.*, 2008, vol. 8, no. 3.
- [13] G. Van der Plas et al., "Verifying electrical/ thermal/ thermo-mechanical behavior of a 3D stack - Challenges and solutions," CICC 2010.
- [14] N. S. Vinay, I. Rawat, E. Larsson, M. S. Gaur, and V. Singh, "Thermal aware test scheduling for stacked multi-chip-modules," EWDTS 2010.
- [15] C. Yao, K. K. Saluja, and P. Ramanathan, "Temperature dependent test scheduling for multi-core system-on-chip," ATS 2011.
- [16] C. Yao, K. K. Saluja, and P. Ramanathan, "Thermal aware test scheduling using on-chip temperature sensors," VLSID 2011.

TABLE I. PERCENTAGE CHANGE IN BURN-IN TIME: TRANSIENT-BASED HEURISTIC COMPARED WITH STEADY-STATE SOLUTION.

IC Specifications	Number of layers	1	1	1	1	2	2	2	2	3	3	3	3	Average
	Number of modules	2	4	8	16	4	8	16	32	6	12	24	48	
Percentage change in burn-in time		-97.82	-73.05	-69.95	-62.63	-68.37	-65.94	-63.82	-55.14	-97.18	-93.17	-95.87	-94.52	-78.12