Temperature-Gradient Based Burn-In for 3D Stacked ICs

Nima Aghaee, Zebo Peng, and Petru Eles
Embedded Systems Laboratory (ESLAB), Linkoping University, Sweden
{nima.aghaee, zebo.peng, petru.eles}@liu.se

***Abstract*—3D Stacked IC fabrication, using Through-Silicon-Vias, is a promising technology for future integrated circuits. However, large temperature gradients may exacerbate early-life-failures to the extent that the commercialization of 3D Stacked ICs is challenged. The effective detection of these early-life-failures requires that burn-in is performed when the IC's temperatures comply with the thermal maps that properly specify the temperature gradients. In this paper, two methods that efficiently generate and maintain the specified thermal maps are proposed. The thermal maps are achieved by applying heating and cooling intervals to the chips under test through test access mechanisms. Therefore, no external heating system is required. The scheduling of the heating and cooling intervals is based on thermal simulations. The schedule generation is guided by functions that are derived from the temperature equations. Experimental results demonstrate the efficiency of the proposed method.**

Keywords—temperature gradients; 3D-Stacked-IC test; burn-in

I. INTRODUCTION

The trend of increasing transistor densities in order to achieve higher performance and extended functionalities has been enabled by reducing the feature size of the ICs. As the feature size approaches the size of atoms, however, keeping this trend up becomes a big challenge and, most importantly, costly. Increasing the die size in order to increase the number of transistors in an IC is not a good option. One reason is that the wire lengths and consequently the power and the delay will increase. It will also lead to low yield and large production cost. An emerging solution is therefore to go upward into the third dimension.

A promising and practical method to fabricate three dimensional integrated circuits is based on Through-Silicon Vias (TSV) [4, 6, 9]. The ICs fabricated using TSVs are commonly referred to as 3D-Stacked IC (3D-SIC) [9]. The important advantages of this technology include high inter-die interconnect densities and low inter-die interconnect wire lengths. This leads to higher operating frequencies at lower power consumptions. But there are challenges to be resolved before commercializing 3D-SIC. In this paper we address a potential problem that is related to the early-life-failures.

One way of accelerating and detecting the early life failures is burn-in which 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 go through their early life before test. The wear mechanisms that are speeded up include metal stress voiding and electromigration, metal slivers bridging shorts, and gate-oxide wearout and breakdown [14]. But 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 [13]. The experiments in [13] show 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 which is used to model electromigration, temperature gradient is present directly and also indirectly through its effect on the mechanical-stress gradient [11]. Therefore, larger temperature gradients speed up some of the wear mechanisms more effectively. Furthermore, it is more effective to speed up the formation of defects that in reality are more likely to happen and this is dependent very much on the real thermal maps that the die will experience in its functional modes.

For example, a burn-in process which has not created a realistic thermal scenario may speed up the formation of some defects that are not likely to happen during real functional operation (increased yield loss), but not speed up sufficiently the formation of some of the defects that are likely to happen during real functional operation (e.g., those that depend on large temperature gradients). Therefore, it is necessary to introduce a burn-in process that creates temperature scenarios in accordance with reality. This necessity is more urgent for 3D-SIC because of the significantly larger temperature gradients compared to 2D. The simulations in [15] shows that the thermal gradients for 3D-SIC can be three times larger as compared with 2D IC.

For 2D ICs, tests are usually performed in two stages, *wafer sort* before packaging and *final test* after packaging [14]. Similar to test, there are usually two stages for burn in, Wafer-Level Burn-In (WLBI) which is performed before packaging and Die-Level Burn-In (DLBI) which is performed after packaging [12]. For 3D-SIC it might be appropriate to have more test stages, namely pre-bond, mid-bond, post-bond and final tests [14]. Similarly, it might be cost saving to introduce burn-in at one or multiple of these stages. For example, at different test stages, different defects (e.g., defects related to TSV bonding) can be targeted based on their likelihood and considering the tests costs.

A convenient way of producing the thermal conditions that correspond to the reality might be to apply the actual functional inputs to the IC in an environment with raised temperature and voltage. This might be possible for 2D ICs, since from the functional point of view all the required circuitry is there when a die enters the test facility. In case of 3D-SIC, burn-in for the post-bond and the final test could be performed using realistic inputs, similar to 2D. But for pre-bond or mid-bond, the inputs to the die or partially stacked dies are not necessarily the inputs to the IC.

The input ports of a 3D-SIC, before the final bonding, are likely to include a number of TSVs. Unfortunately, the TSVs and test equipments are not expected to be designed to support simultaneous application of realistic signals, particularly to large number of TSVs (even though they might be designed to allow simple electrical tests for the TSV itself). Therefore, such scenario is not expected to be possible for pre-bond and mid-bond stages. But, there will be a test access mechanism

for the die [1] that could be used to create the specified thermal condition for the die.

It might be desirable to specify temperature maps aiming at particular defects that correspond to corner cases (e.g., for applications that require very high reliability). In this case, the magnitude of the thermal gradients might be much larger than normal or they might be placed in unusual locations. Such thermal maps might not be easily achievable if the IC is driven from its normal input ports, but they can often be achievable if the test access mechanism is used. This could be possible since test access mechanism provides direct access to cores and consequently heating could be precisely targeted toward a specific core. In this paper a technique to create the specified temperature scenarios using available test access mechanisms is proposed.

The desired temperature maps should be specified so that the early-life defects that depend on temperature gradients are targeted. The specified temperature maps usually correspond to realistic functional temperatures but are not necessarily identical to them. They should be designed to maximize the effectiveness of the burn-in process. Since it is important that the temperature gradients have the correct value and location, the proper temperature condition is best described as a temperature map for the dies. There might be some locations on the dies such that their temperatures are not important regarding the targeted defects. Such locations are indicated as don't-cares. Even though they are marked as don't-cares, the overheating protection should be provided for them. Otherwise, the dies may be damaged due to overheating.

Creating and maintaining some kind of thermal condition during the test has been an active field of research. A thermal-aware test scheduling is introduced in [16] for stacked multi-chip modules which tries to achieve a uniform temperature distribution throughout the 3D IC during the test. The heuristic proposed in [16] is based on analytical simplifications of the thermal model and focuses on vertical thermal distribution. A linear programming approach is used in [10] in order to generate thermally-safe test schedules for 3D-SICs. The method proposed in [10] uses a super-position based thermal simulator.

Two different approaches for multi-core ICs are introduced in [7] and [17] to guarantee that the cores' temperatures are kept within the specified range when the corresponding tests are being applied. They focus on the temperature of the individual cores that are under test and the temperatures of other cores are neglected. Keeping the temperatures within the specified range is achieved by introducing heating sequences (high-power stimuli are applied) and cooling intervals (no stimuli are applied) into the test schedule.

Minimizing the damages caused by overheating for different process variation situations is addressed in [2, 3]. The test temperatures are kept sufficiently low by introducing cooling cycles. For time-invariant temperature variations, the cooling cycles are planned before the actual test using thermal simulations [2]. But for the time-variant temperature variations the test schedule is adapted to the current thermal situation based on the on-chip temperature sensor readouts [3].

The existing methods for controlling the chips' temperatures under test try to respect a global upper temperature limit to prevent overheating [10] or they try to respect local upper and lower bounds in order to maximize test effectiveness [7, 17]. But to our knowledge, there is no existing method to create the specified thermal maps and temperature gradients for burn-in or for delay-fault test. In this paper we present methods to create and maintain the specified temperature maps for burn-in and for test.

The rest of the paper is organized as follows. Section II addresses the temperature-gradient based burn-in. Section IV presents experimental results. Section V gives the conclusion.

II. Temperature-Gradient Based Burn-In

A thermal map specifies the temperature values of the different locations (e.g., cores) in a die. It corresponds to some particular temperature condition of an IC, such as large temperature differences between adjacent cores (i.e., large temperature gradients), that can accelerate aging for early life failures so that they can be tested for, later on. A temperature map is achieved if the specified locations (e.g., cores) on the die have the temperature values specified by the map.

In a burn-in process, we would like to achieve the specified temperature map quickly and maintain the temperature for a given period of time to achieve the intended effect. There are usually many temperature maps that we would like to achieve and maintain. It is therefore important to achieve them rapidly whether the dies start from room temperature or from another map.

Traditionally, burn-in is usually achieved at elevated temperature, by physical means (e.g., temperature chambers), and perhaps elevated voltage. This approach will usually not be able to achieve most of the temperature maps, especially those with large temperature gradients. In our approach, the temperature map will be achieved by using special input stimuli sent through the test access mechanism. It is assumed that no test is applied when an IC is kept under a temperature map in burn-in. The problem formulation for a simplified situation is given in the next section. This will be extended to the realistic situation in section II.B.

A. Steady-State Solution

Assume that there are M modules in the IC (on one or multiple dies) that their test could be started and stopped independently (e.g., cores with core wrappers in a core-based design). In order to create the specified thermal maps, *heating sequences* are used to heat up the modules. A heating sequence is a real or dummy test stimulus that results in very high switching activities. An easy way of obtaining a heating sequence is to extract a part of the test stimuli that has the largest power (switching activity). The average power of the heating sequence is a real number represented by p_m^{HS} for module m $(0 \leq m < M)$. It is assumed that the test access mechanism only affords W (a positive integer number) modules to be tested simultaneously.

The desired thermal map is specified by a *low temperature limit* and a *high temperature limit* for each module and the don't-care modules are declared separately. For example, 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. Assuming that a steady state power could be provided for modules, a steady state solution exists that could generate and maintain the specified thermal map. This means that desirable steady state temperatures should be obtained so that the powers can be calculated based on them.

Providing steady state powers simultaneously for all modules is, however, very likely to be impossible mainly due to test access mechanism limitations. Therefore, the best that can be achieved is a stimulus sequence that has constant average power with small ripples. In order to reduce the risk of out of range temperatures due to ripples in input power, the

desired steady state temperatures are defined at the middle of the specified ranges $\theta_m^{SS} = \frac{1}{2} \times (\theta_m^L + \theta_m^H)$.

In order to find the power values that will result in the specified temperatures for a single die or for partially or fully stacked dies, their thermal model should be known. A widely used thermal model is the lumped element thermal model, as used in HotSpot [8]. Such a model is composed of a number of heat capacitance elements and heat resistance elements, connected together in a network configuration, similar to an electric circuit. The temperatures correspond to voltages and the heat dissipation corresponds to a current source. An example is given in Fig. 1 for a thermal model with only one active node, *X*. A node is called active if it directly receives electrical power during the test.

On a layout or floor plan of a die, the main blocks (e.g., logic and memory) are placed in a certain distance relative to TSVs to avoid undesirable effects such as high mechanical stress. Usually a Keep-Out-Zone (KOZ) is defined that the designer should avoid placing devices in there [4, 6]. It is assumed that in order to overlap the KOZ of different TSVs and save area on the die, the TSVs are not spread all over the die, but are packed in TSV blocks. Another advantage of packing TSVs in dedicated blocks is the possibility of a more convenient TSV test technique that connects TSVs together and tests them collectively [4].

In this section, for simplicity reasons, 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 (are passive thermal nodes) since their drivers are placed together with the corresponding modules. These assumptions will be relaxed in the next section.

The characteristics of the thermal model are captured in two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The thermal behavior of the IC is captured in the following system of ordinary differential equations.

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

In the above equation, $\boldsymbol{\theta}$ is the temperatures vector and $\boldsymbol{P}$ is the powers vector. The specified thermal map consists, in fact, of the steady state temperatures that the IC should be kept at for a while. The thermal map could be thought as the targeted steady state temperatures, $\boldsymbol{\theta}^{SS}$, that is thus composed of the desired steady state temperatures for each module, θ_m^{SS}. Since the specified thermal map is in this case equivalent to the steady state temperatures, which are considered constant (for a certain amount of time), its derivatives are zero (no variation in time). Therefore, equation 1 could be written as

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

This means that it is possible to calculate the required powers that lead to the specified temperature map. In order for the specified temperature map to be achievable, two necessary conditions on the computed steady state power values exist. To distinguish between these two conditions, the first one is called feasibility condition and the second one is called schedulability condition.

The feasibility condition is composed of the following two parts. The computed steady state power for a module m should be larger than or equal to the stray power dissipation of the module. The stray power, $\overline{\boldsymbol{P}}$, is a fraction of the total consumed power that is unintended and could not be independently controlled with available test controls. The stray power, $\overline{\boldsymbol{P}}$, includes the leakage power and the power of the clock network. The second part in the feasibility condition is that the computed steady state power should be less than or

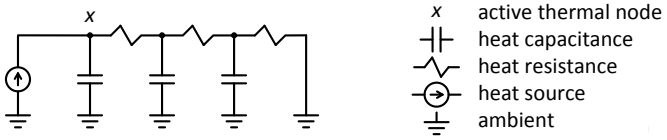

Figure 1. An example of the thermal model.

equal to the average power of the corresponding heating sequence, p_m^{HS}, plus the already existing stray power. The two parts of the feasibility condition are put together as

$$\forall m, \overline{p_m} \le p_m^{SS} \le (p_m^{HS} + \overline{p_m}). \quad (3)$$

Usually the feasibility condition (equation 3) is easily met if the specified temperature map is realistic (an example for a non-realistic map is a map that asks for a temperature lower than ambient temperature). Assuming that the equation 3 is satisfied, the schedulability condition which is related to the limited test access mechanism bandwidth should be studied. The challenging problem here is to create the required average power values, $\boldsymbol{P}^{SS}$, using a limited access through the test access mechanism. This is done by selectively applying the heating sequences to the modules.

It is known that the continuous application of the given heating sequence for the module m generates an average dynamic power equal to p_m^{HS}. The desired power values, p_m^{SS}, which are smaller than p_m^{HS}, 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), in this paper. The ratio of the duration of heating sequence application to the overall time period is therefore called Duty-cycle (D_m) and is computed as shown below.

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

These duty-cycles might not be achievable if their values are relatively large and if the test access mechanism provides insufficient bandwidth. For example, assume a design with two modules. 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 module 0 for 60% of the time and access to module 1 for 80% of the time. Therefore, we need simultaneous access to more than one module ($0.6 + 0.8 = 1.4$ modules). This means that the test access mechanism should provide parallel access to these two modules otherwise these duty-cycles are not schedulable and the specified temperature map is not achievable.

Moreover, D_m can be divided into pieces; for example $D_1 = 0.8$ could be implemented by first applying the heating sequence for a duration equal to $D_{1,0} = 0.3$ at the beginning of the period and then for a duration of $D_{1,1} = 0.5$ at the end of the same period. Therefore, the feasibility and schedulability conditions could be written as follows.

$$\begin{aligned} &\forall m, 0 \le D_m \le 1, \text{and} \\ &\textstyle\sum_{m=0}^{M} D_m \le W. \end{aligned} \quad (5)$$

In fact, the first line in equation 5 is identical to the feasibility condition in equation 3, which is written in terms of the duty cycles. The second line in equation 5 is the schedulability condition.

In the worst case, D_m is divided into not more than two pieces. In order to demonstrate this, an illustrative example is given in Fig. 2. The available parallelism, W, provided by test access mechanism is represented by the number of rows that could be filled with duty-cycles, D_ms, as shown in Fig. 2a ($W = 3$). The scheduling starts by sorting the duty-cycles and

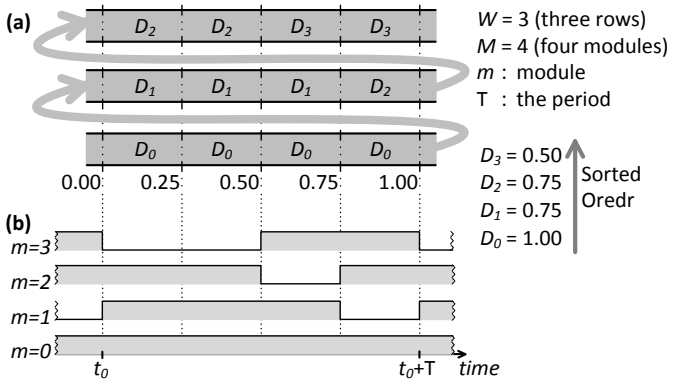

Figure 2. An example for scheduled duty-cycles.

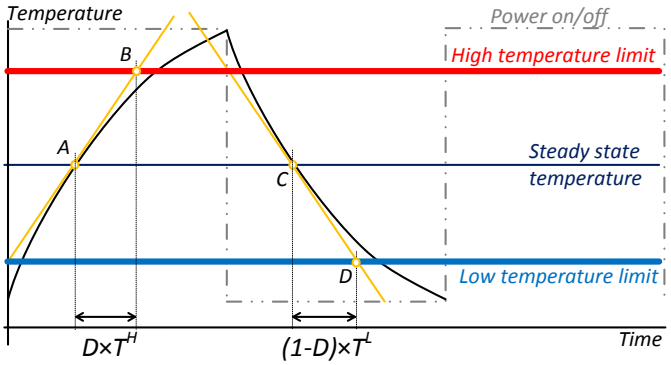

Figure 3. An example for the computation of the safe period so that the temperature limits are not violated. In this example the computed period will be much smaller than the period used in the above image.

then allocating them from the largest one to the smaller ones. Since $D_m \leq 1$, if a duty-cycle is already divided and the first part is allocated in the lower row, the remaining part on the upper row will not reach the end of the row and therefore, there will not be further divisions. The maximal number of divided parts is two and this is desirable since the number of changes is small and the overheads associated with switching are negligible. The fractions of the time period that modules receive heating sequences are illustrated in Fig 2b. At every moment in time only three modules are receiving their heating sequences (the test access mechanism limitation is not exceeded), and the average of applied heating sequence for a module in a period is equal to the specified steady state power.

The period should be short enough so that the fluctuations in the input power do not cause the IC's temperatures to violate the specified temperature limits. On the other hand, a longer period is desirable because it minimizes the overheads associated with switching. Therefore, we shall find the longest period that keeps the fluctuations in temperature inside the specified ranges. In order to estimate the maximal period for the time that the heating sequence is being applied (e.g., the second half of the period for module 3 in Fig. 2b) equation 1 is re-written around the steady state temperature for the heating sequence power as equation 6a. For a no-power interval (e.g., the first half of the period for module 3 in Fig. 2b) equation 6b is used, instead.

$$\left(\frac{d}{dt}\boldsymbol{\theta}\right)^H = \boldsymbol{A}^{-1} \times (\boldsymbol{P}^{HS} + \overline{\boldsymbol{P}} - \boldsymbol{B} \times \boldsymbol{\theta}^{SS}) \quad (6a)$$

$$\left(\frac{d}{dt}\boldsymbol{\theta}\right)^L = \boldsymbol{A}^{-1} \times (\overline{\boldsymbol{P}} - \boldsymbol{B} \times \boldsymbol{\theta}^{SS}) \quad (6b)$$

An example for such derivatives around the steady state temperature for a single module is shown in Fig. 3. The derivatives are then used to estimate the desired value for the period (linear approximation) for the upper limit (high temperature limit). Assuming that T_m^H is the amount of time that will result in a near violation situation for the heating interval for module m, the estimation is

$$\frac{\theta_m^H - \theta_m^{SS}}{D_m \times T_m^H} \cong \left(\frac{d}{dt}\boldsymbol{\theta}\right)_m^H. \quad (7)$$

Now, the amount of time that will result in a near violation situation is computed for each module as

$$T_m^H \cong (\theta_m^H - \theta_m^{SS}) / \left(D_m \times \left(\frac{d}{dt}\boldsymbol{\theta}\right)_m^H\right). \quad (8)$$

The values for $\left(\frac{d}{dt}\boldsymbol{\theta}\right)_m^H$ are obtained from the right hand term in equation 6a and consequently the values for T_m^H are computed using equation 8. For example in Fig. 3, when the power is on, the derivative that is represented by a straight line is tangential to the temperature curve on the steady state temperature (at point A) and later on intersects with high temperature limit (at point B). The safe period, T_m^H, is then calculated based on the time difference between A and B.

In a similar manner values for the cooling intervals, T_m^L, are calculated based on equation 6b. Since the temperatures should not violate the specified limits, the shortest T_m ($T_m = \min\{T_m^H, T_m^L\}$) is selected as the acceptable period for module m. The actual period, T, is the smallest period among acceptable periods for modules ($T = \min_m\{T_m\}$).

This solution achieves the specified temperature maps by focusing on the steady state solution, but it ignores the transient response. It means that we should wait until these new temperatures are built up (and the initial temperatures are faded away). This implies that the burn-in time is very long. The *burn-in time* is defined as the time required for bringing the IC into a thermal situation that complies with the first temperature map and then to the next map, until all maps are applied. For example assume that there is only one thermal node and two temperature maps. The first map requires that the node's temperature is around 50°C and the second map requires that the node's temperature is around 70°C. Assume that the ambient temperature is around 30°C. The burn-in time is, therefore, equal to the time required to heat up the node from 30°C to 50°C plus the time required to heat up the node from 50°C to 70°C.

The steady state solution results in an excessively long burn-in time. In order to make this clear, the analytic solution to equation 1 is given below.

$$\boldsymbol{\theta}^t = \boldsymbol{\alpha} \times \boldsymbol{\theta}^0 + \boldsymbol{\beta} \times \boldsymbol{P} \quad (9)$$

In the above equation, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are matrices that are computed based on $\boldsymbol{A}$ and $\boldsymbol{B}$, and for a duration of time equal to t. The initial temperatures are expressed by $\boldsymbol{\theta}^0$ and the temperatures after time t are denoted by $\boldsymbol{\theta}^t$; $\boldsymbol{P}$ is the power vector (assumed to be constant for the time interval t). An intuitive explanation of equation 9 is that $\boldsymbol{\alpha}$ determines how fast the initial temperatures fade away and $\boldsymbol{\beta}$ determines how fast the input powers affect the temperatures. The matrix $\boldsymbol{\alpha}$ ensures that the initial conditions (i.e., $\boldsymbol{\theta}^0$) will eventually fade away (heat will flow out of the IC). In order to reduce the burn-in time by pushing more power into the die at the beginning, the initial conditions should be taken into consideration. Otherwise risks are high that the excessive power overheats the chip. It is likely that a very large number of temperature maps that correspond to different operational modes and/or are targeting different defects are specified. Therefore, achieving a new temperature map in a short time is crucial and this temperature transition should be performed as fast as possible. Once the IC's temperatures have converged to

the specified temperature map, they can be maintained using the steady state powers, $\boldsymbol{P}^{SS}$.

Assume that the higher power that is used to speed up the warm up process is denoted by $\boldsymbol{P}^{B}(\forall m, p_m^B > p_m^{SS})$. Therefore, equation 9 could be written as

$$\boldsymbol{P}^{B} = \boldsymbol{\beta}^{-1} \times (\boldsymbol{\Theta}^{SS} - \boldsymbol{\alpha} \times \boldsymbol{\Theta}^{0}). \quad (10)$$

The proposed approach for the steady state solution was to solve the steady state equation (equation 2) in order to obtain the power values. Extending the "steady state solution" approach to the complete thermal equation (equation 10) in order to find the schedulable power values that result into the shortest burn-in time, results in the following problem. The problem is to find the shortest warm up time, t ($\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ depend on t) such that the resulted powers, $\boldsymbol{P}^{B}$, are schedulable. This problem can be solved using a numerical method (i.e., an iterative approach) that tries different alternatives for t. This extended approach will be very slow in design time (very long CPU time) since the calculation of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is excessively time consuming.

Extending the "steady state solution" to include the transient response is, therefore, not practical and a new approach that avoids successive calculations of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ is necessary. Such an approach is proposed in the next section based on a heuristic that is fast. This proposed approach is capable of handling a more realistic problem formulation that the simple extension of the "steady state solution" would not have been capable of. This is another reason for the necessity of the transient-based heuristic.

B. Transient-Based Heuristic

As mentioned before, a more aggressive approach is required to take the initial temperatures into account and start a new temperature map by injecting higher powers in order to speed up the convergence to it and therefore shorten the overall burn-in time. Besides, a realistic problem formulation should be supported (e.g., support for high resolution thermal model, non-equal test access bus widths, and active TSV blocks).

Since our goal is to create the specified temperature-map with temperature gradients in their exact locations, the thermal model should be sufficiently precise. Therefore, the thermal model may require a higher number of nodes; the number of active nodes is represented by a positive integer N ($M \leq N$). This means that a single physical module in a die may map on multiple active thermal nodes. For example, module 1 that is shown in Fig. 4a is mapped to two thermal nodes (node 2 and node 3) as shown in Fig. 4b.

Furthermore, it is assumed that the TSV blocks are able to dissipate power (TSV drivers/buffers might be placed in TSV blocks) and their desired temperatures might also be specified in the thermal maps (not always don't-care as been assumed in section II.A). In this section, the desired thermal map is specified for thermal nodes (unlike section II.A where it was specified for modules). For example, the temperature map specifies that node n has low temperature limit equal to θ_n^L and high temperature limit equal to θ_n^H.

Therefore, the switching activities for heating sequences which serve as inputs to our problem should be more specific and provide information concerning the power breakdown among the active thermal nodes. As a result, the heating sequences are extracted for different active thermal nodes. For example, instead of only one heating sequence for module 1 in Fig. 4a, there are two heating sequences corresponding to two active thermal nodes (nodes 2 and 3 in Fig. 4b). The average power of a heating sequence for active node n is represented by p_n^{HS}. We should note that, other active nodes of that module (e.g., node o) may also receive some power, denoted by $p_{n,o}^{HS}$. For example, when trying to heat up node 3 in Fig. 4b with p_3^{HS}, node 2 is also heated by $p_{3,2}^{HS}$. Moreover, node 1 will also warm up a little bit because of the lateral heat leakage. The heat leakage phenomenon is modeled in the thermal model and it means that a node can heat up even if it is not receiving electrical active power (switching activity).

An important shortcoming of the steady-state solution (section II.A) is explained here with an example. Assume that a heating sequence for thermal node $n = 3$ is being applied. The active power received by node $n = 2$ (part of the same module) is probably greater than zero. It means that unlike section II.A, the power value for a node that is calculated using equation 2 cannot be realized independent from other nodes. The technique suggested in section II.A is not able to provide a fast warm up and besides it is not able to handle a precise thermal model that is based on N thermal nodes ($N > M$). In the following, a fast technique is proposed to address these shortcomings.

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 (making sure, of course, that the chip is not damaged) and when all temperatures are placed inside the specified ranges, the thermal rest takes over. Since a very short boosting time is desirable, the highest possible power should be applied during boost. According to the proposed method, 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 ($\theta_n^{SB} \geq \theta_n^H$). This possibility could be helpful, for example to achieve the following desirable scenario. Assume that the node is initially heated beyond θ_n^H. Then the node does not need to receive active power and this leaves the test access mechanism available for other nodes. Meanwhile, the temperature keeps decreasing (naturally) and just before the end of the boosting time (the moment that all other nodes are in their specified temperature ranges), the temperature drops below the high temperature limit. The temperatures in boost mode are kept below the overheating temperature (taking the safety merging into account). Moreover, the duration of boost mode is very short. Therefore, boost mode is thermally safe and it has no significant effect on wear mechanisms.

In thermal-rest mode, initially all nodes' temperatures are in the specified ranges. The nodes' temperatures will naturally decrease, but they 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 is 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 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. In a situation

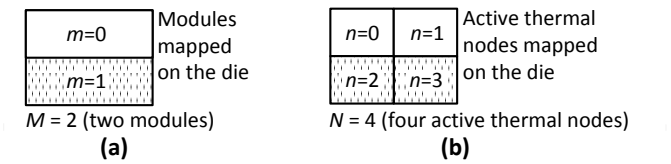

Figure 4. An example of modules and active thermal nodes mapped on a die.

that a module covers multiple active thermal nodes, the heating sequence could only be applied if all of those thermal nodes have temperatures lower than their high temperature limit.

The heating trigger, θ_n^{HT}, that is greater than the low temperature limit ($\theta_n^{HT} > \theta_n^L$) works as follows. Assume that a node's temperature just falls below θ_n^{HT} and therefore it is recognized to need heating. Assume that the test access mechanism is not available at that moment. The difference between θ_n^{HT} and θ_n^L provides sufficient time for the node to wait for gaining access to the test access mechanism without its temperature falling below θ_n^L. Therefore, a possible violation of the temperature map is avoided this way.

The nodes that simultaneously require heating should be accommodated within the available bandwidth of the test access mechanism. This bandwidth might not be sufficient for all of them and therefore some of them should be prioritized. The priorities for using the test access mechanism are determined based on the regional need for heating around a node n which is denoted by d_n. It is similar to the duty cycles in section II.A and it is analytically obtained using the following procedure. This procedure is activated when the node needs heating ($\theta_n < \theta_n^{HT}$ in the thermal rest mode and if the node has not been boosted and $\theta_n < \theta_n^{SB}$ in the thermal boost mode). In the following the regional need for heating is introduced for the thermal rest mode. Equation 1 could be estimated as

$$\frac{\boldsymbol{A}\times(\boldsymbol{\theta}^{HT}-\boldsymbol{\theta})}{T} + \boldsymbol{B}\times\boldsymbol{\theta} = \boldsymbol{D}\times\boldsymbol{P}^{HS} + \overline{\boldsymbol{P}}. \quad (11)$$

The values for regional-need-for-heatings, d_ns, constitute the $\boldsymbol{D}$ vector. The equation is written for one test cycle (the period is T) that is assumed to be a small time. Equation 11 is then solved for the active nodes that need heating as follows.

$$d_n = \frac{\frac{\sum_{k=0}^{N} a_{n,k}\times(\theta_k^{HT}-\theta_k)}{T} + \sum_{k=0}^{N} b_{n,k}\times\theta_k - \overline{p_n}}{p_n^{HS}} \quad (12)$$

The regional need for heating, d_n, depends on the required heating for node n (consider the summations when k is equal to n), on the required heating that is related to the adjacent nodes (consider the summations when k denotes an adjacent node to n), and on the average power of the corresponding heating sequence, p_n^{HS}. The elements of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, ($a_{n,k}$ and $b_{n,k}$) are so that the regional need for heating for a node has the highest dependency on the node itself, and then a relatively high dependency on the adjacent nodes. The effects of other nodes located far from the targeted node are small.

The heat leakage between nodes is taken into account automatically, since equation 12 is derived from the thermal equation (equation 1) and includes the thermal conductances from $\boldsymbol{B}$ matrix, $b_{n,k}$. Efficient values for heating triggers, θ_k^{HT}, for each map are found using an optimization metaheuristic.

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 11 and 12). The priority for using the test access mechanism is given to the regions that need longer heating time (e.g., larger ($\theta_n^{HT} - \theta_n$) and smaller p_n^{HS}). Furthermore, the locality of this heuristic is helpful because adjacent nodes are likely to be in the same module and therefore these nodes will receive some unintended active heating power ($p_{n,k}^{HS}$) or leaked heat.

The inputs to the methods proposed in section II include thermal maps, IC's thermal model, IC's electrical model (e.g., specification of the test access mechanism and power-related specifications), heating sequence switching activities, and ambient temperature. The output for the steady state solution (section II.A) is a periodic offline schedule and therefore producing a small periodic schedule could be counted as an advantage for this method. The output for the transient-based heuristic (section II.B) could be a non-periodic offline schedule that has the advantage of offering a reduced burn-in time. Moreover, the proper values for the heating trigger, θ_n^{HT}, and stop boosting temperatures, θ_n^{SB}, that result in a reduced burn-in time could also be considered as the outputs of this method that provide a basis for an online scheduling scenario (not further discussed in this paper).

III. Experimental Results

The proposed temperature gradient-based methods are evaluated for twelve experimental ICs with one to three layers, as detailed in Table I, columns 2, 3, and 4. The one layer (one storey) experimental ICs (row 1 to 4 in Table I) are bare dies and could represent pre-bond test stage. The ICs that have two layers (row 5 to 8) could represent mid-bond test stage. The ICs with three layers (row 9 to 12) could represent post-bond test stage. There are two, four, eight, and sixteen physical modules per layer for different dies, resulting in the total number of modules ranging from two to forty eight, as given in column 3. The TSVs are supposed to be located in dedicated blocks on the die. There are one, two, and three TSV blocks per layer on the dies, resulting in the total number of TSV blocks given in column 4. 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 used in a variant of HotSpot [8] that is extended in [5] for 3D Stacked IC (3D-SIC). The heating patterns' switching activities are generated using Markov chains, similarly as in [18]. The thermal maps specify the valid temperature ranges for nodes in the thermal model. The valid ranges are randomly selected from six different temperature ranges (35–45℃, 45–55℃, 55–65℃, 65–75℃, 75–85℃, 85–95℃) and some modules/nodes are randomly selected to be don't-care. Only temperature maps that can be achieved in practice are considered. An example for a temperature map that cannot be achieved is one that requires a central node with very low temperature and its adjacent nodes with very high temperature. In this case the thermal gradient is huge and it probably will require negative power (active cooling) for the central node and excessively huge power for the adjacent nodes.

The transient-based heuristic (section II.B) is evaluated and compared with the steady-state solution (section II.A). The transient-based method is capable of handling thermal models having multiple nodes per module, while the steady state method only supports one thermal node per module. In order to have comparable experiments, the thermal model that

TABLE I. Experimental Results

IC Number	IC Specifications			Percentage change in burn-in time
	Number of layers	*Number of modules*	*Number of TSV blocks*	
1	1	2	1	-97.82
2	1	4	1	-73.05
3	1	8	2	-69.95
4	1	16	3	-62.63
5	2	4	2	-68.37
6	2	8	2	-65.94
7	2	16	4	-63.82
8	2	32	6	-55.14
9	3	6	3	-97.18
10	3	12	3	-93.17
11	3	24	6	-95.87
12	3	48	9	-94.52
Average				-78.12

is supported by the steady state method is used in the experiments. The CPU time to generate the schedules for the transient-based method for all of the twelve experimental ICs together is about 12 minutes while the steady state method completes in 2 seconds. The time required to bring the IC into a thermal situation that complies with the first temperature map and then to the next map until all maps are applied is defined as the burn-in time in this work. The percentage change in burn-in time offered by the transient-based method, compared with the steady state solution, is given in column 5 of Table I. Considerable speed up (78% in average) is achieved by the transient-based method.

IV. CONCLUSIONS

A promising technology for the manufacturing of the future generations of electronics is 3D stacked IC using Through-Silicon-Vias (TSV). Early-life failures that are affected by large temperature-gradients might be a challenge for commercialization of 3D-SIC. The problem is that some defects rapidly develop and cause early-life failures when the IC is working with certain temperature maps (with large temperature gradients).

In order to effectively detect these defects, it is necessary to create and maintain the specified thermal maps during burn-in. The methods proposed in this paper utilize the available test access mechanisms in order to do so. The specified temperature maps are achieved and maintained by selectively applying dummy high-power test patterns to the ICs (used solely for heating). Therefore, there is no need for expensive equipments to heat up the chip externally.

First, a steady state solution is introduced that is fast to generate the schedules, but the schedules are slow to achieve the specified temperatures. A schedule in this case consists of a single periodic schedule for each map. Then, the transient-based method is proposed. The transient-based method supports a more precise thermal model, and offers a shorter burn-in time by generating schedules that rapidly bring the IC to the specified temperature conditions. The experiments indicate that this method is 78% faster than the steady state solution in realizing the specified temperature maps.

REFERENCES

[1] S. Adham and E. J. Marinissen, IEEE P1838 web site. http://grouper.ieee.org/groups/1838/.

[2] N. Aghaee, Z. He, Z. Peng, and P. Eles, "Temperature-aware SoC test scheduling considering inter-chip process variation," Asian Test Symposium (ATS), 2010, pp. 395–398.

[3] N. Aghaee, Z. Peng, and P. Eles, "Adaptive temperature-aware SoC test scheduling considering process variation," Digital System Design (DSD), 2011, pp. 197–204.

[4] K. Chakrabarty, S. Deutsch, H. Thapliyal, and F. Ye, "TSV defects and TSV-induced circuit failures: The third dimension in test and design-for-test," International Reliability Physics Symposium (IRPS), 2012, pp. 5F.1.1–5F.1.12.

[5] A. K. Coskun, J. L. Ayala, D. Atienza, T. S. Rosing, and Y. Leblebici, "Dynamic thermal management in 3D multicore architectures," Design, Automation & Test in Europe (DATE), 2009, pp. 1410–1415.

[6] S. Deutsch, K. Chakrabarty, S. Panth, and S. K. Lim, "TSV stress-aware ATPG for 3D stacked ICs," Asian Test Symposium (ATS), 2012, pp. 31–36.

[7] Z. He, Z. Peng, and P. Eles, "Multi-temperature testing for core-based system-on-chip," Design, Automation and Test in Europe (DATE), 2010, pp. 208–213.

[8] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," Design Automation Conference (DAC), 2004, pp. 878–883.

[9] E. J. Marinissen, "Challenges and emerging solutions in testing TSV-based $2\frac{1}{2}$D- and 3D-stacked ICs," Design, Automation and Test in Europe (DATE), 2012, pp. 1277–1282.

[10] S. K. Millican and K. K. Saluja, "Linear programming formulations for thermal-aware test scheduling of 3D-stacked integrated circuits," Asian Test Symposium (ATS), 2012, pp. 37–42.

[11] J. Pak, M. Pathak, S. K. Lim, and D. Z. Pan, "Modeling of electromigration in through-silicon-via based 3D IC," Electronic Components and Technology Conference (ECTC), 2011, pp. 1420–1427.

[12] O. Semenov, A. Vassighi, M. Sachdev, A. Keshavarzi, and C. F. Hawkins, "Effect of CMOS technology scaling on thermal management during burn-in," IEEE Transactions on Semiconductor Manufacturing, 2003, vol. 16, no. 4, pp. 686–695.

[13] T. Smorodin, J. Wilde, P. Alpern, and M. Stecher, "A temperature-gradient-induced failure mechanism in metallization under fast thermal cycling," IEEE Transactions on Device and Materials Reliability, 2008, vol. 8, no. 3, pp. 590–599.

[14] M. Taouil, S. Hamdioui, K. Beenakker, and E. J. Marinissen, "Test impact on the overall die-to-wafer 3D stacked IC cost," Journal of Electronic Testing (JETTA), 2012, vol. 28, no. 1, pp. 15–25.

[15] G. Van der Plas, et al., "Verifying electrical/ thermal/ thermo-mechanical behavior of a 3D stack - Challenges and solutions," Custom Integrated Circuits Conference (CICC), 2010, pp. 1–4.

[16] N. S. Vinay, I. Rawat, E. Larsson, M. S. Gaur, and V. Singh, "Thermal aware test scheduling for stacked multi-chip-modules," East-West Design and Test Symposium (EWDTS), 2010, pp. 343–349.

[17] C. Yao, K. K. Saluja, and P. Ramanathan, "Temperature dependent test scheduling for multi-core system-on-chip," Asian Test Symposium (ATS), 2011, pp. 27–32.

[18] C. Yao, K. K. Saluja, and P. Ramanathan, "Thermal aware test scheduling using on-chip temperature sensors," VLSI Design (VLSID), 2011, pp. 376–381.