Temperature-Gradient Based Test Scheduling 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*—Defects that are dependent on temperature-gradients (e.g., delay-faults) introduce a challenge for achieving an effective test process, in particular for 3D ICs. Testing for such defects must be performed when the proper temperature gradients are enforced on the IC, otherwise these defects may escape the test. In this paper, a technique that efficiently heats up the IC during test so that it complies with the specified temperature gradients is proposed. The specified temperature gradients are achieved by applying heating sequences to the cores of the IC under test trough test access mechanism; thus no external heating mechanism is required. The scheduling of the test and heating sequences is based on thermal simulations. The schedule generation is guided by functions derived from the IC's temperature equation. Experimental results demonstrate that the proposed technique offers considerable test time savings.**

I. Introduction

A promising technology for fabricating three dimensional integrated circuits is based on Through-Silicon Vias (TSV) [7, 9, 13]. The ICs fabricated using TSVs are commonly referred to as 3D Stacked IC (3D-SIC) [13]. 3D-SIC and other deep submicron technologies suffer from a considerably larger number of delay faults as compared with previous technologies. The causes for these delay faults include resistive bridges and vias, power droops, and cross-talk noise effects. Therefore, delay-fault testing is necessary to provide sufficient fault coverage [4, 10]. A large number of pre-bond TSV defects are resistive in nature and, moreover, the mechanical stress caused by TSVs contributes to delay faults [7, 9]. Therefore, the expected number of delay faults for 3D-SIC is larger than that of 2D ICs; thus delay-fault test is, in particular, important for 3D-SIC.

Since temperature has a significant effect on delay, its impact should be taken into account in delay-fault test. A very important effect of temperature on signal integrity throughout an IC is its effect on the clock network [6]. Delay faults usually happen because of increased clock skew and the major sources of skew for 3D-SICs are induced by temperature gradients [15]. Since propagation delays in a clock network depend on temperature, different temperatures at different places on an IC (i.e., temperature gradients) result in clock skew. Temperature gradients in an IC may reach up to 50℃ in adjacent cores for normal operation and even higher during test [6, 15]. Besides, the temperature gradients in 3D-SICs are much larger than in 2D ICs [19]. This will exacerbate temperature gradient related issues including delay faults and temperature aware delay-fault test becomes, in particular, important for 3D-SIC. Therefore, the test should be performed when the IC has proper temperature maps. A temperature map specifies the appropriate temperatures for different parts (e.g., cores) in the IC. These temperatures are to be realized simultaneously in order to enforce the proper temperature gradients. The temperature maps are given along with their corresponding tests and this paper proposes a technique to enforce these maps on the IC and then apply the corresponding tests while the maps are actively enforced.

It is well known that deep submicron integration results in high power densities that are even higher in test mode compared to the normal operational mode, in particular for core-based designs [5, 23]. Such ICs are so hot during the test that they might be overheated [2, 14, 22]. Therefore, application of test stimuli can raise the ICs' temperatures to their tolerable limits for deep-submicron ICs and in particular for 3D-SIC. This often undesirable effect is however utilized in this paper to enforce the specified temperature maps on the IC. Special stimuli that cause large switching activities and aggressively heat up the IC are used for this purpose. Such stimuli which are not necessarily actual test patterns, are called *heating sequence*.

For 2D ICs, tests are usually performed in two stages, *wafer sort* before packaging and *final test* after packaging. For 3D-SIC there are more stages, including pre-bond, mid-bond, post-bond, and final stages [18]. At these different stages, different defects can be targeted based on their likelihood and considering the test costs.

For some ICs and some temperature maps, driving the chip by applying special inputs to its functional input ports will enforce the temperature map. 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 process. In case of 3D-SIC, enforcing temperature maps for the post-bond and the final test might be possible using functional input ports, similar to 2D. But for the pre-bond or mid-bond stages, the inputs to the die or partially stacked dies are not necessarily the inputs to the IC. The input ports to the unit under test for 3D-SICs, before the final bonding, are likely to include a number of TSVs. The TSVs and test equipment are not expected to be designed to support simultaneous real-life signals, particularly to large number of TSVs (even though they might be designed to allow simple electrical tests for the TSV itself). Therefore, such a scenario is not expected to be possible for the pre-bond and mid-bond stages. To address this problem, Test Access Mechanism (TAM) could be utilized, instead; knowing that every die is equipped with at least one TAM that provides access to its cores [1].

The necessity to utilize the TAM has yet another reason that is not specific to 3D. The thermal gradients in some maps might be placed in locations that cannot be properly stimulated through functional input ports. Such thermal maps can often be enforced if the TAM is used. The reason is that the TAM, in the test mode, provides direct access to cores; while in the normal operational mode, a core might be limited to receive inputs only from an adjacent core. Therefore, heating could be targeted toward a specific core using the TAM. In this paper a technique to construct the specified temperature maps using available TAMs is proposed.

II. Related Works

Several related works in temperature aware test have been reported. A thermal-aware test scheduling technique for network-on-chip based ICs using multiple clock rates is proposed in [17]. The objective is to find a thermally safe test schedule with a minimal test application time. A simulated annealing metaheuristic is used to search alternative cores' placements, clock frequencies, and other decision variables.

A thermal-aware test scheduling is introduced in [20] for stacked multi-chip modules which tries to achieve a uniform temperature distribution throughout the 3D IC during the test. The proposed heuristic is based on simplifications of the thermal model and focuses on vertical thermal distribution.

Linear programming is used in [14] in order to generate thermally-safe test schedules for 3D-SICs. The proposed method uses a super-position based temperature simulator.

Two different approaches for multi-core ICs are introduced in [11] and [21] to guarantee that the cores' temperatures are kept within specified ranges when the corresponding tests are being applied. The problem specifies the temperature of the individual cores that are under test and the test for a core can be

978-1-4799-2452-3/13/$31.00 ©2013 IEEE

correctly performed disregarding the temperatures of other cores. Keeping the temperatures within the specified range is achieved by introducing heating sequences and cooling intervals (when no stimuli are applied) into the test schedule.

Obtaining a short test application time considering process variations is addressed in [2, 3]. The test temperatures are kept low by introducing cooling cycles. A fast thermal simulation technique that speeds up iterative simulations by introducing an initialization phase is suggested in [3].

The 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, independent of others, 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 specified. To our knowledge, there is no existing method to enforce the specified thermal map on an IC and perform the tests under that specified temperature map.

III. Preliminaries

The thermal behavior of a 3D-SIC can be captured using a well-known thermal modelling technique used in HotSpot [12] and extended in [8] to model 3D ICs. According to this technique, the characteristics of the thermal model are captured in two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, and the thermal behavior is described by 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 temperature vector and $\boldsymbol{P}$ is the power vector. The modelling technique divides the IC into tiny elements called *thermal nodes*. Each node has a heat capacitance that models its thermal capacity. Adjacent nodes in the model 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 activity (in this case, there is also a leakage power). It is assumed that there are a total of N nodes in the thermal model.

Our objective is to obtain a short test application time while the constraints captured by the temperature maps are satisfied. There are usually a number of maps corresponding to a number of tests. A thermal map specifies, for all nodes, the temperature bounds that should be respected simultaneously in order to construct the specified spatial temperature gradients. Therefore, a map specifies the *high temperature limit* (θ_n^H) and *low temperature limit* (θ_n^L) for the temperature of node n ($0 \leq n < N$) during the application of the corresponding tests. There might also be some nodes such that their temperatures are not relevant with regard to the defects that the temperature map is targeting. Such locations are separately specified as don't-cares. In order to prevent damage, the temperature of don't-care nodes are kept below the overheating limit considering a safety margin ($\theta^{overheating}$). Furthermore, there are usually constrains imposed by TAM limitations or test dependencies. Therefore, before a module's test starts, it should be verified that it is entitled to run the test. The heating sequences are also subject to TAM limitations, but cooling intervals could be used whenever required. A cooling interval is when no test or heating is applied.

The dynamic power consumed by tests and heating sequences are given as inputs. Assuming that there are a total of M modules in the IC, there will be M sets of test sequences each corresponding to a module. When a module is receiving stimuli all of its nodes may receive active power, therefore the power values for tests and for heating sequences are provided individually for each active thermal node as input to our problem. The dynamic power of the heating sequence that targets node n is denoted by p_n^{HS}. When this heating power is being applied to node n, node k in the same module, also, receives dynamic power denoted by $p_{n,k}^{HS}$. Besides these power dissipations that could be stopped or started using test controls, a thermal node is impacted by the stray power (denoted by $\overline{\boldsymbol{P}}$) which includes the leakage power and the power dissipated by the clock network. The details of stray power are specified as input and its value may depend on temperature (taken care of, in this paper).

The schedule for heating sequences, tests, and cooling intervals is the output generated by our algorithm. It is assumed that the high and low temperature limits (θ_n^H, θ_n^L) are designed so that the thermal effects due to process variation are tolerable. It is also assumed that the safety margins used to avoid overheating for don't-care nodes do not have an important impact on the test application time. Note that the application of the proposed technique is not limited to delay-fault test and it could also be used for other tests. The proposed method could be used as a complement to existing test techniques to detect defects that depend on temperature levels and gradients. While these defects are more problematic for 3D-SIC, the proposed technique could also be used for 2D ICs if the need arises.

IV. Proposed Technique

The test schedules are generated by comparing simulated temperatures with three different temperature levels. These temperature levels are the decision variables that impact the test application time. They consist of stop boosting temperature (θ_n^{SB}), heating trigger temperature (θ_n^{HT}), and testing trigger temperature (θ_n^{TT}), for each node and for each temperature map. In the following we focus on only one map and therefore these variables are only indexed with the corresponding node, n. In experiments multiple maps and therefore multiple of these decision variables are present. The ICs' temperatures are simulated based on the thermal model introduced in section III. The decisions to introduce a heating sequence, cooling interval or perform tests are made based on the simulated temperatures as described in the following. An illustrative example is given in Fig. 1.

When a new map has to be enforced on the IC, if a node is changing from a lower temperature in the previous map (or initial/ambient temperature for the very first map) to a higher temperature in the next map, it is heated up to the stop boosting temperature, θ_n^{SB}, that can be higher than the high temperature limit, θ_n^H. This is helpful; as an example assume that the node is initially heated beyond θ_n^H (Fig. 1a). Then the node does not need heating for a while (Fig. 1b) and this leaves the TAM available for other nodes. Meanwhile, the temperature keeps decreasing (naturally) and just before all other nodes are in their specified temperature ranges, the temperature drops below the high temperature limit. Heating a node up to θ_n^{SB} at the very beginning of a new map (Fig. 1a) is called boosting. While boosting, the temperatures are kept below $\theta^{overheating}$.

The nodes' temperatures will naturally decrease when there are no stimuli (Fig. 1b,d,g) or when there is a low power test (Fig. 1c). These temperatures should not fall below the low temperature limit, θ_n^L. Unlike cooling, heating is not always feasible immediately when the need is recognized since the TAM might be fully occupied (delivering heating sequences to other nodes that have higher priorities). Therefore the need for

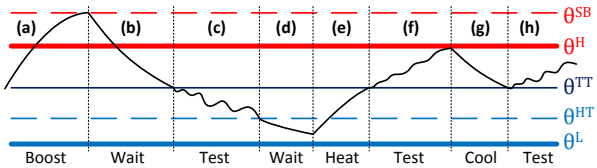

Figure 1. The temperature of a node as the schedule generated using the proposed technique is being applied (curves are illustrative).

heating is recognized when the temperature falls below heating trigger temperature, θ_n^{HT} that is higher than θ_n^L. The time needed for the node to naturally cool down from θ_n^{HT} to θ_n^L provides the node with opportunity to queue up and wait for access to the TAM without falling out of range (Fig. 1d). In order to start heating, all nodes covered by the module should be colder than their high temperature limits, θ_n^H, since heating for one node is very likely to inject power to other nodes in the same module.

Heating stops when the temperature reaches the testing trigger temperature, θ_n^{TT} ($\theta_n^{HT} < \theta_n^{TT} < \theta_n^H$), as in Fig. 1e. If the previous temperature is higher than the temperature required by the current map, or if the test has raised the temperature up to θ_n^H (Fig. 1f), a cooling interval is introduced (Fig. 1g). The cooling interval continues until the temperature is dropped below θ_n^{TT} and then testing may resume (Fig. 1h). Both heating and cooling stop at the testing trigger temperature, θ_n^{TT}, and consequently it is likely that testing starts/resumes when the temperature is close to θ_n^{TT} (Fig. 1f,h). A proper θ_n^{TT} value should be selected so that the temperature changes during test (caused by the changes in the test power) rarely fall below θ_n^{HT} or exceed θ_n^H. Every time that θ_n^{HT} or θ_n^H are violated, the test must be interrupted and a heating or cooling interval must be introduced, respectively. Since these are time consuming, a proper θ_n^{TT} value helps to obtain a short test application time by reducing the number of interrupts.

When the temperatures for all of the thermal nodes covered by module m are between their heating trigger and their high temperature limit, $[\theta_n^{HT}, \theta_n^H]$, testing may start. All other nodes should be within their temperature limits $[\theta_n^L, \theta_n^H]$. Testing continues until the temperature of at least one of the nodes goes beyond θ_n^H or falls below θ_n^L (below θ_k^{HT} for the nodes that are running their tests).

First the nodes that require heating (not the tests) are given access to TAM. This helps to keep the temperatures most of the time within the specified limits and thus keep the flow of the tests uninterrupted. Note that if only one node falls out of its specified range, all tests must be interrupted until the map is achieved again. This will waste a lot of time, since the tests for the modules that are in their specified range should also be interrupted. The priorities for the nodes that require heating is determined based on the *regional need for heating* (denoted by d_n around a node n). It is designed to represent the need of a node for receiving heating power and it is obtained using the following procedure that is activated when the node needs heating. Node n needs heating if $\theta_n < \theta_n^{HT}$ after being boosted and if $\theta_n < \theta_n^{SB}$ before being boosted. In order to obtain the regional need for heating, equation 1 is estimated as

$$\boldsymbol{A} \times \frac{(\boldsymbol{\theta}^{HT} - \boldsymbol{\theta})}{T} + \boldsymbol{B} \times \boldsymbol{\Theta} = \boldsymbol{D} \times \boldsymbol{P}^{HS} + \overline{\boldsymbol{P}}, \qquad (2)$$

for a small period of time equal to T. Heating trigger temperature, θ_n^{HT}, is a good reference point, since the colder a node is compared with θ_n^{HT}, the more heating it needs and the warmer a node is compared with θ_n^{HT}, the less heating it needs. $\boldsymbol{\Theta}$ is the current temperatures vector. Having the current temperatures and the temperatures that are targeted to be reached after T (this is assumed just to compute d_n), the required powers could be computed. Large power value for a node roughly indicates a large need for heating. But it is appropriate to take the stray powers, $\overline{\boldsymbol{P}}$, into account since their values are different for different nodes. The remaining parts of the required powers are supplied by the heating sequences $\boldsymbol{P}^{HS}$. Since heating sequence powers are different for different nodes, different nodes will require different amounts of heating sequences. These amounts of heating sequences are regional-need-for-heatings, d_ns, that constitute vector $\boldsymbol{D}$. Equation 2 is then solved for the nodes that need heating as follows.

$$d_n = \frac{\frac{\sum_{k=0}^{N-1} a_{n,k} \times (\theta_k^{HT} - \theta_k)}{T} + \sum_{k=0}^{N-1} b_{n,k} \times \theta_k - \overline{p_n}}{p_n^{HS}} \qquad (3)$$

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 a node adjacent to node 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 heat leakage between nodes is taken into account automatically, since equation 3 is derived from the thermal model (equation 1) and includes the thermal conductivities from matrix $\boldsymbol{B}$ ($b_{n,k}$).

The priorities when the nodes are competing to receive their heating sequences for boosting are computed in a similar manner by replacing θ_n^{HT} with θ_n^{SB} (e.g., in equations 2 and 3). In any case, the priority for using the TAM 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.

If the TAM is left with some available bandwidth after the heating sequences are scheduled, the modules that are thermally qualified may resume their tests. The priority is given to the modules that are expected to offer long *test endurance*. The test endurance is denoted by e_m for module m, and is defined as

$$e_m = r_m \times tt_m . \qquad (4)$$

The test endurance is directly proportional with the *remaining test size* (denoted by r_m for module m). The larger the remaining test size, the longer the test endurance. The *thermal tolerance*, denoted by tt_m for module m, is the other contributor to the test endurance. High thermal tolerance, tt_m, indicates that the module is capable of receiving tests for a long time without violating the specified thermal limits. Therefore, a module with large thermal tolerance may remain under test for a relatively long time. The thermal tolerance is defined as

$$tt_m = \min_k \{\Delta_k\} . \qquad (5)$$

In equation 5, it is assumed that module m covers a total of K active nodes. Δ_k $(0 \le k < K)$ denotes the expected thermal distance to a temperature limit for node k and is defined as

$$\Delta_k = \begin{cases} \theta_k^H - \theta_k & upcoming\_tests\_power > p_k^M \\ \theta_k - \theta_k^{HT} & upcoming\_tests\_power \le p_k^M \end{cases} \qquad (6)$$

p_k^M is the power that results in a temperature equal to $\theta_k^M = (\theta_k^L + \theta_k^H)/2$. It means that if the upcoming tests have relatively high average power, then it is likely that this thermal node will exceed the high temperature limit and therefore the difference between the current temperature, θ_k, and the high temperature limit, θ_k^H, is a good measure for thermal tolerance. Similarly, for a relatively low power test, it is more likely that the temperature falls below the heating trigger in the future. Therefore, the difference between the current temperature, θ_k, and the heating trigger temperature, θ_k^{HT}, is a good measure for thermal tolerance. Thermal tolerance, tt_m, is defined as the smallest Δ_k $(k = 0, 1, \dots, K-1)$ since as soon as a single node is out of the specified range $[\theta_n^L, \theta_n^H]$, disregarding of the temperatures of the other nodes, test should be interrupted (if only θ_k^{HT} is exceeded, only module m's testing must be paused, otherwise all tests). Proper values for stop boosting, heating trigger, and testing trigger temperatures (θ_n^{SB}, θ_n^{HT}, and θ_n^{TT}, respectively) for each and all maps are found using Particle Swarm Optimization (PSO), an iterative population based optimization technique [16].

TABLE I. EXPERIMENTAL RESULTS (PERCENTAGE CHANGE IS REPORTED IN COMPARISON WITH THE BASELINE METHOD)

IC Number		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	***Average***
IC Specifications	***Number of layers***	1	1	1	1	2	2	2	2	3	3	3	3	
	Number of modules	2	4	8	16	4	8	16	32	6	12	24	48	
Percentage change in test application time		-16.97	-39.69	-63.35	-94.77	-8.70	-60.80	-78.17	-95.04	-75.90	-84.81	-87.08	-94.72	***-66.67***

V. EXPERIMENTAL RESULTS

The proposed technique is evaluated for twelve experimental ICs with one to three layers, as detailed in Table I, row 2. The one layer (one story) experimental ICs (number 1 to 4) are bare dies and represent the pre-bond test stage. The ICs that have two layers (number 5 to 8) represent mid-bond test stage. The ICs with three layers (number 9 to 12) 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 row 3. The dies are assumed to be stacked in a face to back configuration.

The thermal models are extracted using an approach similar to [8]. A fast temperature simulation scheme similar to [3] is used. The heating patterns' switching activities are generated using Markov chains, similar to [22]. The high and low temperature limits for nodes and don't-care nodes are appointed randomly. 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 surrounded by nodes with very high temperatures, such that the central node requires negative power in order to satisfy the map.

For a small IC example, the temperatures for all nodes are simulated for the best candidate solution (candidate values for θ_n^{SB}, θ_n^{HT}, and θ_n^{TT}) in the first PSO iteration and the curves are presented in Fig. 2a. Their counterparts for the final solution are shown in Fig. 2b. The beginning of the curves corresponds to warming up from the initial room temperature and the ending part corresponds to testing. Proper values for θ_n^{SB}, θ_n^{HT}, and θ_n^{TT} result in a short schedule that efficiently mixes the heating sequences and cooling intervals with tests, as in Fig. 2b.

In order to have a meaningful evaluation, the proposed technique is compared with a baseline method. The baseline method is similar to the proposed technique, but heating sequences are scheduled separately in a number of temperature construction segments, isolated from tests that are performed in a number of test segments. The advantages of this approach include its simplicity and its ability to accommodate third party test schedules inside its segments.

The total time required to bring the IC into a thermal situation that complies with the specified thermal maps and maintain this desired thermal situation while the tests are being applied, including the time spent applying the tests is called test application time. The percentage change in test application time offered by the proposed technique compared with the baseline method is given in row 4 of Table I, which shows that considerable speed up (67% in average) is achieved. The percentage change in CPU time required to generate the schedules by the proposed technique compared with the baseline method is −36%. In spite of being more complicated, the proposed technique is faster than the simpler baseline method. The schedule length is an important contributor to the CPU time, since longer schedules require longer thermal simulations and thermal simulation is, per se, very time consuming. Therefore, when a new approach reduces the test application time drastically, it is not unexpected that the CPU time reduces as well.

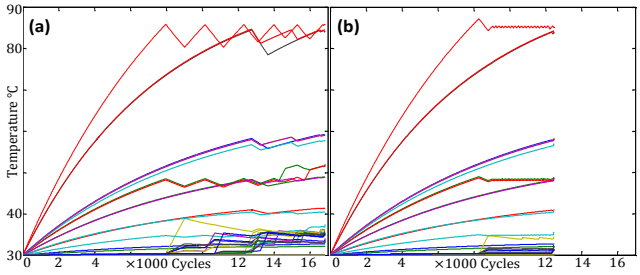

Figure 2. Temperature curves corresponding to (a) best solution in the first PSO iteration and (b) the final solution that offers a much shorter test application time.

VI. CONCLUSIONS

Delay faults that are dependent on temperature-gradients introduce a challenge for achieving an efficient delay-fault test process. The negative effects of temperature-gradients are more pronounced for 3D-SIC technology, since their magnitude is much larger. The challenge is that some defects cannot be detected unless the IC has a certain temperature map. Therefore, it is necessary to construct and maintain the specified thermal maps during delay-fault test. The technique proposed in this paper utilizes the available test access mechanisms in order to apply high-power stimuli (used solely for heating) that are planned to enforce such temperature maps. Therefore, there is no need for expensive equipment to heat up the chip externally. To our knowledge, this is the first technique to achieve thermal maps for test without any external heating mechanism. The proposed technique schedules the heating and cooling intervals mixed with the tests. Therefore, the test application time offered by this method has been substantially reduced.

REFERENCES

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

[2] N. Aghaee, Z. Peng, and P. Eles, "Process-variation and temperature aware SoC test scheduling using particle swarm optimization," IDT 2011.

[3] N. Aghaee, Z. Peng, and P. Eles, "Process-variation and temperature aware SoC test scheduling technique," JETTA 2013.

[4] N. Ahmed, M. Tehranipoor, and C. P. Ravikumar, "Enhanced launch-off-capture transition fault testing," ITC 2005.

[5] Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Test power: a big issue in large SOC designs," DELTA 2002.

[6] S. A. Bota et al., "Within die thermal gradient impact on clock-skew: a new type of delay-fault mechanism," ITC 2004.

[7] K. Chakrabarty et al., "TSV defects and TSV-induced circuit failures: The third dimension in test and design-for-test," IRPS 2012.

[8] A. K. Coskun et al., "Dynamic thermal management in 3D multicore architectures," DATE 2009.

[9] S. Deutsch et al., "TSV stress-aware ATPG for 3D-SICs," ATS 2012.

[10] D. Goswami et al., "At-speed scan tests: reality or fantasy?" ITC 2007.

[11] Z. He, Z. Peng, and P. Eles, "Multi-temperature testing for core-based system-on-chip," DATE 2010.

[12] W. Huang et al., "Compact thermal modeling for temperature-aware design," DAC 2004.

[13] E. J. Marinissen, "Challenges and emerging solutions in testing TSV-based 2.5D- and 3D-stacked ICs," DATE 2012.

[14] S. K. Millican et al., "Linear programming formulations for thermal-aware test scheduling of 3D-stacked integrated circuits," ATS 2012.

[15] M. Mondal et al., "Thermally robust clocking schemes for 3D integrated circuits," DATE 2007.

[16] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization, An overview," Swarm Intell. 2007.

[17] H. Salamy and H. M. Harmanani, "Thermal-aware test scheduling using network-on-chip under multiple clock rates," Int. J. Electronics, 2013.

[18] M. Taouil, S. Hamdioui, K. Beenakker, and E. J. Marinissen, "Test impact on the overall die-to-wafer 3D stacked IC cost," JETTA 2012.

[19] G. Van der Plas et al., "Verifying electrical/ thermal/ thermo-mechanical behavior of a 3D stack - Challenges and solutions," CICC 2010.

[20] N. S. Vinay, I. Rawat, E. Larsson, M. S. Gaur, and V. Singh, "Thermal aware test scheduling for stacked multi-chip-modules," EWDTS 2010.

[21] C. Yao, K. K. Saluja, and P. Ramanathan, "Temperature dependent test scheduling for multi-core system-on-chip," ATS 2011.

[22] C. Yao, K. K. Saluja, and P. Ramanathan, "Thermal aware test scheduling using on-chip temperature sensors," VLSID 2011.

[23] Y. A. Zorian, "Distributed BIST control scheme for complex VLSI devices," VTS 1993.