2012 IEEE 21st Asian Test Symposium

Linear Programming Formulations for Thermal-Aware Test Scheduling of 3D-Stacked Integrated Circuits

Spencer K. Millican and Kewal K. Saluja
Department of Electrical and Computer Engineering, University of Wisconsin-Madison, WI
1415 University Ave., Madison, WI 53706
Tel: 715/321-0066, 608/262-6490, Fax: 608/262-1267
smillican@wisc.edu, saluja@ece.wisc.edu

***Abstract*—With technology scaling towards smaller geometries, the power density of modern integrated circuits (ICs) can potentially result into high temperatures during test, a problem further compounded by stacking dies in 3D stacked structures (3DSICs). Scheduling tests in a way to minimize the total test time becomes a key issue when temperature constraints are involved, since a more compact schedule leads to a hotter device. Unfortunately, many previous attempts at temperature-bounded scheduling either use inferior temperature models leading to under compaction, or they can only be applied to traditional single-die designs. Simple thermal models based on steady state temperatures are inadequate to schedule tests in 3DSICs due to their limitations. This paper proposes two formulations for test scheduling under thermal constraints for 3DSICs using the superposition principle, which allows for accurate thermal modeling and superior test compaction. This paper then compares them to previous formulations which use steady-state models, and also discusses the inherent limitations of the steady-state model. Results of the algorithms proposed in this paper show the superiority of the schedules obtained for testing 3DSICs.**

I. Introduction

As technology scales towards smaller feature sizes, new issues arise in the general area of testing digital circuits. One of the areas that drew considerable attention of researchers was test scheduling. Yao, Saluja, and Parmesh [1] provide an exhaustive list of references (nearly 100 papers) in this area. The initial focus revolved around hardware design and compatibility issues exclusively, and while this is an NP-hard problem, it is a problem that relies on relatively few variables. However, with smaller technology sizes and new manufacturing methods come new variables and additional testing issues.

One such new testing issue that has become important is the temperature during test. While older generations of integrated circuits did not have to cope with temperature issues, the power density of modern integrated circuits has risen to the point where the operating temperatures of devices have become a design issue which is often resolved through external cooling. The importance of this stems from the fact that if a device in question becomes too hot it may fail permanently. Testing devices in as little time as possible, in order to provide better test economy, can directly conflict with this goal. This is even more important if test compaction methods are used to reduce test data volume, since compacting individual tests can cause the power density of a device to increase to the point that may result in temperature causing permanent damage to the device under test (DUT). Therefore, it is often the goal to test a DUT not only as fast as possible, but also to stay within a given temperature bound [2]. Such constraints add to the complexity of an already NP-hard testing problem.

There have been several studies on scheduling tests with power constraints which use either generic heuristic-based algorithms or ILP-like formulations. However, using power alone to enforce a temperature constraint is inadequate. Although power and temperature are directly correlated, it is easy to show that a test profile and corresponding schedule, which has an adequate power constraint, will violate its temperature constraint. This is an important distinction since computation time for generation of accurate temperature profiles can be very large, which has resulted in more simplified temperature models being used during test generation and compaction. Bild et. al. [3] presented a temperature-constrained MILP compaction formulation based on a steady-state temperature model which can be implemented from power traces. Rosinger et. al. [4] proposed an a heuristic-based algorithm using a steady-state model, as well as a computation-intensive optimal algorithm.

The problem with many proposed solutions is that they either use a thermal model that is inaccurate, or they use computationally-intensive thermal profiling. Although using a steady-state thermal model is tempting due to its simplicity and its linearity, the accuracy of such models in the context of test scheduling, at best, is very poor. First, a steady-state model assumes that the temperature during the first clock-cycle of a test will be at it's maximum, which is obviously false. Second, the steady-state model assumes that the temperature of the processor one clock cycle after the test completes will be the same as the ambient temperature, which is also false. Third, tests often are not long enough for the blocks under tests to reach steady state temperatures in most cases. These situations can lead to under-compaction by not overlapping two tests when it is entirely possible to do so. The solution to this is to use accurate thermal simulation, but proponents of such an approach also admit the infeasibility of such an approach due to its computational complexity [4].

However, a recently proposed thermal aware scheduling method promises to relieve both of these problems. The superposition principle proposed by Yao et. al. [5] argues that for a linear system the change in temperature caused two overlapping tests is the same as the sum of the change in temperature caused by each test. This allows for temperatures during test to be accurately modeled while doing the computationally intensive temperature simulation only once for each test.

Emerging 3D stacked integrated circuits (3DSICs) pose new challenges. For example, a new hindrance caused by 3DSICs is the temperature problems of technology scaling is further compounded by the stacking of several ICs on top of each other. A naive method of test scheduling such as solving a test schedule for each individual die then combining them for an entire stack is unacceptable, since it is sure to violate a temperature constraint. Another problem introduced by 3DSICs during test is that new test architectures which are designed and suitable for 3DSICs must be used. Marinissen [6] has done several studies implementing test access mechanisms (TAMs) in 3DSICs, with Bild et. al. [3] giving formal definitions of com-

1081-7735/12 $26.00 © 2012 IEEE
DOI 10.1109/ATS.2012.28

patibility. These new parameters make the compatibility calculations of current test scheduling algorithms exponentially more complex. Further, the addition of Thermal Through Silicon Vias (TTSVs) as well as the three-dimensional placement of individual dies makes the calculation of temperature conductance more difficult. Also, this new conductance information makes many previous solutions based on steady-state temperature models obsolete without significant adjustments.

In this paper we address the problem of test scheduling in 3DSICs while satisfying resource, power and thermal constraints. The contributions of this paper include:

- Several MILP formulations for thermally-constrained test time reduction, providing near optimal and pessimistic but practical solutions for 3DSICs.
- Comparing our formulations and solutions against steady-state model based formulations extended to function for 3DSICs to demonstrate the inherent inefficiencies of steady state models.
- Exploring the complexities of expanding test scheduling formulations in three dimensions.

The rest of the paper is organized as follows: Section 2 introduces several techniques for temperature modeling used in past studies and previous 3DSIC scheduling studies. Section 3 discusses the methodology used to generate the needed information for tests, as well as the use of the superposition principle. Section 4 presents a basic MILP formulation using the superposition principle and its reasoning, while Section 5 expands it to obtain a pessimistic and always valid solution. Section 6 discusses the results of using such methods against existing steady-state formulations, and the paper concludes with Section 7.

II. Past Work

The most basic technique to schedule thermally-constrained tests is to rely on power-constrained test scheduling. The rational behind such a technique is that the temperature of a device is a direct result of the power dissipated by a device. Therefore, if the power dissipated during a test is kept below a given power bound, then the temperature will stay below a proportional bound. There have been examples of generic power-constrained test scheduling such as in [7], and there have been studies in power-constrained test scheduling for the purposes of limiting device temperatures [8]. Although power and temperature are correlated, using power values alone is not adequate for determining temperature values. The actual temperature of a DUT is not just dependent on the present power being dissipated by the DUT, but also on the past power dissipation and the physical properties of the DUT. It is easy to show that a relatively low power value can raise the temperature of a well-insulated device to the point of failure. This problem is compounded by the non-linear relation between power and temperature. Although power and temperature values are related, they are not directly proportional because of the thermal capacitance of a device, which can lead to under and over-compaction by presuming the temperature of a DUT is at all times proportional to the power value.

Another technique is to directly use power-generated temperature traces while scheduling. Such techniques use power traces to generate temperature traces using a simulator, most often based on the thermal RC model, for a given potential test schedule. Such a technique was proposed in [4], where an algorithm generates a potential test schedule, which is then simulated to check for temperature constraint violations. This technique has the benefit of being extremely accurate, and therefore can potentially provide an optimal schedule as suggested above. However, such temperature simulations have a drawback of having a very large runtime that cannot be ignored. Doing a single accurate temperature simulation is a computationally-intensive, which in turn implies that simulating every possible test schedule is practically impossible, especially for DUT with numerous tests. This was also addressed in [4], which sacrificed the optimal algorithm for a heuristic-based algorithm for the sake of algorithm runtime.

An alternative to the two extremes described above has been to use steady-state values of temperatures of devices during scheduling. While some techniques attempt to model the actual cycle-accurate temperature of a test, steady-state models use the maximum possible temperature achievable by a test. Such a technique was effectively used in a formulation proposed in [3]. The advantage of steady-state formulations is the simplicity of the thermal RC model in which thermal capacitances are deleted. However, removing the capacitance from the RC temperature model is also its downfall. Since thermal capacitance is effectively removed, the actual temperature of a test will always be lower then the presumed steady-state temperature unless the test runs for a long period of time, which can be a great hindrance when scheduling short, power-intensive tests. Another drawback of the steady-state model is tests which result in a very high steady-state temperature may be impossible to schedule. One last drawback of the steady-state model is that it is not cycle-accurate, which means individual tests cannot be partially overlapped if their combined steady-state temperatures violate the temperature bound.

An alternative to all these strategies and the one used in this study is the superposition principle proposed by Yao et al. [5]. The superposition principle states that the temperature offset of two or more tests running in parallel is the sum of temperature offsets generated by running those tests by themselves. Validity of this can be checked either by simulation or by mathematically manipulating the thermal RC model. The advantage of such a model is that it is cycle-accurate and its computational complexity is significantly less than other approaches with the same accuracy. The accuracy aspect is self-explanatory, since actual temperature values are being used instead of estimates, and the low computation-complexity is due to the summation of temperature offsets being a simple addition operation. Also, the superposition principle can be used in a MILP formulation since it is a linear formulation. What is different from previously discussed simulations is the temperature profile only needs to be simulated once per each test instead of once per proposed solution.

Although stacking dies has been a design technique for some time, studies into testing such designs has only been done recently. Most studies of testing revolve around optimizing Test Access Mechanisms (TAMs) for bandwidth or power consumption for 3DICs [6], [9]. However, because the thermal characteristics of individual dies change greatly after stacking, and because there are unique thermal characteristics, such as TTSVs, that do not apply to non-stacked ICs, some studies have addressed the problem of temperature-constrained test scheduling of 3DICs [8], [10]. However, past studies have not been generic enough to be applied beyond specific stacking methods, and they often make assumptions which are particular to those design methods. This study seeks to break free of these constraints by giving formulations that can be used for any kind of stacking method.

III. Methodology

Thermal simulation in this study is done through a modified version of the open-source thermal simulator, Hotspot [11]. Hotspot uses an RC thermal model to generate the temperature profile of a given IC with a given power trace. Modifications had to be done to Hotspot

in order to simulate TTSV behavior, which is not natively supported by Hotspot.

Power traces for a given test are generated using a pseudo-random Markov Chain technique [12]. Ideally, tests would be given in the form of inputs and outputs to actual ICs. However, due to the proprietary nature of IC manufacturing, such information is unobtainable. It is safe to assume that the maximum power of the IC is limited by some constant factor which is reflective of current technology scaling [13]. We also model the conditions where the power consumption of a test is known, such as high during scan-in and scan-out stages or low during idle stages. Although it may be sufficient to claim that the power dissipation is constant throughout a test for the purposes of scheduling, the Markov model provides a more accurate and realistic behavior of a test.

The modeling of test compatibility is a more complicated issue given the nature of 3DICs. Compatibility within dies is modeled based on the locality of the modules under test, with modules that close to another being more likely to be incompatible than if they were farther away from each other. This assumption is based on the observation that TAMs used between adjacent modules are more likely to be shared between each other, thus making them incompatible. Compatibility between dies is a more complicated issue that resolves itself to a simple solution for the sake of this study. In this study, compatibility between dies is considered to be universally compatible. Although this is not a realistic assumption to make, it will not take away from the validity of this study since the purpose is to evaluate how different scheduling methods perform handling temperature conflicts and not hardware conflicts. Making compatibility between dies always compatible should not give one testing method an advantage over another.

As with compatibility generation, the placement of TTSVs must be done somewhat randomly for lack of information. Although it is tempting to randomly place TTSVs across all dies, it is much more reasonable to follow some guidelines found in other research. For one, instead of placing several small TTSVs randomly, fewer larger TTSVs are used as suggested in [14]. It is also sensible not to "stagger" TTSVs placements between layers, instead having TTSVs run continuously throughout the stack as suggested in [15]. Although there have been studies into how TTSVs should be placed [15], [16] or how modules should be placed around TTSVs [17], for the purposes of testing it is reasonable to assume that the actual placement of TTSVs was left to the designer's discretion, whether they be good choices or not. From this assumption, in this study general TTSV locations are placed randomly, with TTSVs penetrating entirely though the stack with a randomly-generated clustering factor.

IV. Optimistic MILP Formulation

A. Importing Variables from [3]

Bild et al. in [3] proposed a formulation for scheduling tests with constraints based on overlapping tests. We use portable variables and constraints from their work and these are listed and explained below and are used for subsequent formulations.

As with any test scheduling algorithm or formulation, the primary goal is to minimize the overall testing time so as to optimize test economy. We presume that the DUT is a core-based design, such that each core $c \in C$ has its own unique test. Given this, the goal of the formulation is straightforward.

$$\text{minimize } t_{finish}$$

$$\forall c \in C : t_{finish} \geq t_f(c)$$

$$\forall c \in C : t_f(c) = t_s(c) + E(c)$$

The first constraint is straightforward, since the goal in any test scheduling algorithm or formulation is to minimize overall test time. The second constraint is an implementation of a "maximum" function, since the time to finish is effectively the time at which the last test finishes. The last constraint above gives the relation between the starting time of a test, $t_s(c)$, and the finishing time of a test, $t_f(c)$, relative to the constant execution time of the test, $E(c)$. In order to represent two tests running in parallel, both for the sake of compatibility and temperature constraint, a binary variable $\eta(c_1, c_2)$ is introduced.

$$\eta(c_1, c_2) = \begin{cases} 1 & \text{if the test for core } c_1 \text{ finishes} \\ & \text{before the test for core } c_2 \text{ begins} \\ 0 & \text{otherwise} \end{cases}$$

Defining $\eta(c_1, c_2)$ in such a way can be done using the relationship of $t_s(c)$ and $t_f(c)$ for two separate tests. Note that $\eta(c_1, c_2)$ and $\eta(c_2, c_1)$ are distinctly different from each other, and only one of them can be 1 at a given time. Below, λ is a large constant greater than or equal to the longest possible test length, or the time of all tests running serially.

$$\forall c_1 \in C, \forall c_2 \in C : t_f(c_1) \leq t_s(c_2) + (1 - \eta(c_1, c_2)) \cdot \lambda$$

$$\forall c_1 \in C, \forall c_2 \in C : t_s(c_2) \leq t_f(c_1) + \eta(c_1, c_2) \cdot \lambda$$

Now that $\eta(c_1, c_2)$ is defined, overlap and compatibility can be defined. Given the definition of $\eta(c_1, c_2)$, it can be stated that if both $\eta(c_1, c_2)$ and $\eta(c_2, c_1)$ are zero, then the two tests must overlap. From this, compatibility and overlapping are defined.

$$\Gamma(c_1, c_2) = \begin{cases} 1 & \text{if the test for core } c_1 \text{ is incompatible} \\ & \text{with the test for core } c_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\forall c_1 \in C, \forall c_2 \in C : \eta(c_1, c_2) + \eta(c_2, c_1) \leq 1$$

$$\forall c_1 \in C, \forall c_2 \in C : \eta(c_1, c_2) + \eta(c_2, c_1) \geq \Gamma(c_1, c_2)$$

B. Superposition-Based Optimistic Formulation

The implementation we present here is a simpler formulation that acts as a counter-part to formulations that use steady-state models, such as the formulation in [3]. The assumptions made by this formulation are similar to those made by steady-state formulations. It is presumed that a test has no effect on the temperature of the DUT before the test starts and after the test is finished, which is not necessarily true and will be addressed in the next formulation. It is also presumed that if a maximum temperature of a given test exists for a given interval, then the temperature will never go above that point during that interval for the test, whereas steady-state formulations presume that the maximum temperature achieved is the steady-state temperature.

Given the assumptions made by this formulation, this can be translated in terms of overlapping tests. For every core c_2, the sum of temperature offsets where the test of core c_1 overlaps with in every test of core c_3 must be below a given temperature bound. For the sake of simplicity, a new variable $\Omega(c_1, c_2)$ is introduced. This variable does not increase the complexity of the problem, since this variable is simply a substitution for existing variables.

$$\Omega(c_1, c_2) = \begin{cases} 1 & \text{if test } c_1 \text{ overlaps with } c_2 \\ 0 & \text{otherwise} \end{cases}$$

$$\Omega(c_1, c_2) = 1 - \eta(c_1, c_2) + \eta(c_2, c_1)$$

$$\forall c_1 \in C, \forall c_2 \in C : \sum_{\forall c_3 \in C} \Omega(c_1, c_3) T_M(c_3, c_2) \leq T_{bound} - T_{amb}$$

Here, $T_M(c_3, c_2)$ is the maximum temperature offset from ambient temperature achieved by test of core c_3 in the core c_2, which due to the nature of the superposition principle is a constant.

An advantage to such a formulation is its simplicity. A steady-state counterpart such as proposed in [3] requires another binary variable for each two-test combination to represent power dissipation during test, as well as variables and constraints to represent the temperature during test. With this superposition-pasted formulation, such variables and constraints are not needed. It is true that a time-consuming temperature simulation is needed for every single test, but this simulation need only be performed once for each test before computing the formulation. Another advantage of this model over steady-state models is that it can cope with short tests and power-intensive tests that steady-state formulations can not. In a steady-state formulation equivalent to the one above, $T_{MAX}(c_3, c_2))$ would not be the maximum temperature achieved by the test, but the maximum possible temperature achieved given that the test length is long enough. This means power-intensive tests that run for short periods of time can create a formulation which may indicate non-existence of a schedule even though one exists, since the steady-state temperature may be higher than the maximum temperature bound. Also, short tests who do reach their steady-state temperature will still have their maximum temperature overestimated.

There are disadvantages of this superposition-based formulation that are shared with steady-state formulations: both formulations are pessimistic since they both over-estimate temperatures for the test, especially during the beginning of the test. This leads to under-exploitation of tests that can be partially overlapped without violating temperature constraints. This superposition-based optimistic formulation has another major shortcoming. It is a fact that the temperature change caused by a test when it finishes is not zero, but instead it takes time for the temperature offset to cool back to zero, an effect that is not modeled by this formulation. This can lead to over-compaction by scheduling too many "hot tests" back to back. This error is avoided by a steady-state formulation, since the maximum temperature that can be achieved during a test is the steady-state temperature, no mater what the temperature is at the start of the test. The given temperature bound may be exceeded if several power-intensive tests are scheduled consecutively in sequence rather then separated by power-conservative tests. In our approach after the schedule is found, it is verified for temperature violation by thermal simulation and the results in Section VI show that this violation is rarely encountered.

V. Pessimistic MILP Formulation

The goal of this extended formulation is to correct the discrepancies of the previous optimistic formulation. As stated before, a flaw in the previous formulation is that it does not take into account temperature differentials caused after a test is completed. Although these "trailing temperatures" are not accounted for in a steady-state model either, the maximum temperature achievable is the steady-state temperature regardless of what tests happen immediately before it, but this is not the case with superposition-based formulations.

Before modeling these trailing temperatures, it is important to understand their nature. It is understandable that the temperature offsets of a test after the test is complete will fall back to zero (ambient temperature), but it is not instantaneous as it takes time to fall back to zero at an exponential rate. The challenge of modeling this decay is creating a formulation that models it efficiently, and

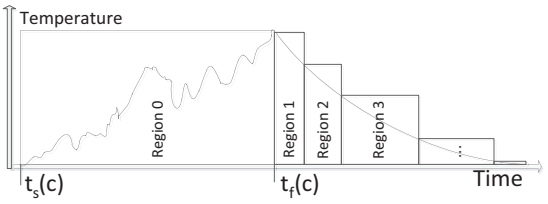

Fig. 1. An example of partitioning a single test into regions.

with a reasonably high degree of accuracy. This exponential decay means that the trailing temperature offset will never be zero, but at the same time one can argue that in time it will be close enough to zero resulting in a negligibly small error. Our approach to modeling this effect efficiently is to assume the duration of the decay time to be significantly long but not unduly long. A very long duration will provide an accurate model but will take much longer to find a solution, where as a very short duration will have the same disadvantage of optimistic formulation. In other words, the temperature offset of a test at the finishing time of the last region should be virtually zero. Therefore as explained below we divide the temperature decay into regions and model each region with a constant temperature. Although there are many ways to model these "trailing temperatures", it would be beneficial to model these trailing temperatures using already-existing variables and constraints. Because of the nature of exponential decay, the time of the first region new will be smaller than the second, which will be smaller than the third, and so on with each region having its own max temperature (see Figure 1). To model these regions in the previous formulations, they are modeled as tests themselves. Unlike actual tests which can occur at any time, these regions must explicitly occur after the test they represent. This conditions is embedded in the following:

$$\forall c \in C : t_{finish} \geq t_f(c, 0)$$

$$\forall c \in C, \forall r = 1 \ldots R : t_s(c, r) = t_f(c, r-1)$$

$$\forall c \in C, \forall r = 0 \ldots R : t_{finish} = t_s(c, r) + E(c, r)$$

Here, R is the number of extra regions modeling the trailing region of each test. Essentially, each test expands into R more regions, with each region specifically starting when the previous region finishes, or in the case of the first new region, when the original test finishes (here, region zero). Note that the overall finishing time only considers the actual test region (region zero), since the trailing regions are not an actual part of the test.

The other constraint that is explicitly different for this formulation is compatibility. The compatibility for all new trailing regions is not tied to compatibility of the original test or any other test. Instead new trailing regions are explicitly compatible with all other regions (trailing and original) over every other test. This is important, for it allows trailing regions to overlap with any other region as long as this overlapping does not violate any temperature constraints. Hence:

$$\forall c_1, c_2 \in C, \forall r_1 = 0 \ldots R, \forall r_2 = 1 \ldots R : \Gamma(c_1, r_1, c_2, r_2) = 0$$

Since all other variables and constraints are expanded solely based on adding regions to each test, they will not be explicitly stated here.

The inherent advantage of such a formulation is its correctness over the more basic formulation and its compacting ability over steady-state formulations. The ability to compact more than a steady-state formulation comes form modeling maximum temperatures as actual maximum temperatures, not steady-state temperatures which

can be higher than maximum temperatures, and unlike the previously presented optimistic formulation, this formulation will not violate temperature constraints for trailing temperatures.

However, this extended formulation has a weakness in its trade-off between correctness and complexity. As more trailing regions are added, the more room there is for overlapping tests since temperatures will be modeled more accurately. This is true up to the point where the number of regions for each tests is equal to the number of time steps in the trailing region, at which point the optimal overlapping of trailing regions can be found. However, adding a single extra region is equivalent to doubling the number of tests to schedule, adding another is equivalent to tripling, and so forth, thus causing an exponential growth in complexity with the growth in the number of regions. On the other side of the spectrum, adding only a single new region will generate greatly pessimistic schedules, since it will be presumed that the temperature after testing will be the same as the temperature during testing for the length of the original test. Hence, it is in the scheduler's best interest to find a balance point between the two extremes.

If there is one extension to this formulation that would be most beneficial, it would be to split the original test into multiple regions as well. This would allow for overlaps that would not be available under steady-state or with the current pessimistic or optimistic superposition formulations.

VI. Results

All formulations in this study were evaluated using the ITC02 benchmarks. Since the benchmarks themselves do not contain specific geometry information, test length, or power consumption data, this information is generated using the information about the number of pins of each module to estimate the relative sizes of each module in a benchmark. Power profiles for a test of each module are generated using the Markov model described in Section III. 3DSICs are formed for this study by stacking benchmarks on top of each other, as shown in Table I which includes dies (benchmarks) creating the stack, the total number of cores/modules in the stack, the maximum test length in milliseconds, and the maximum temperature achieved by a single test in degrees Celsius. Dies to be stacked are chosen randomly as to be unbiased, and stack heights never exceed three dies so as to emulate reasonable and practical designs in modern context. Also note that some "stacks" consist of only a single die (first seven designs in Table I), which is effectively modeling a regular single-die test. Both the optimistic and pessimistic formulations are implemented using IBM's CPLEX. The steady-state formulation from [3] is implemented for comparison as well, which is extended to work for 3DSICs. Each benchmark stack is evaluated using each formulation with its total runtime in seconds, total schedule time in milliseconds, and temperature slack in degrees Celsius with a temperature bound of 145 ^{o}C. General results for each of these stacks are shown in Table II, which includes the runtime in seconds and the temperature slack in degrees Celsius. The scheduled times of each stack for each formulation are shown in Figure 2. A firsthand evaluation of the optimistic formulation shows that although it gives the best scheduled time in the smallest amount of computational time in all cases, it gives results that can violate the temperature constraint such as in stack4, which is marked an invalid schedule (IS) in the table as it has a slack of -4.56. The pessimistic formulation, on the other hand, shows promising results especially when compared to the steady-state formulation in [3]. Despite the pessimistic nature of the formulation, it achieves better compaction results in less time. However, in fairness this time does not account for the time of temperature simulation, which in this study can take between thirty minutes to three hours if each temperature trace for every test is generated in parallel. However, this is a constant overhead time for each scheduling problem which gives a large improvement in the final result. It is also worthwhile to note that although the optimistic formulation can produce invalid schedules, in most cases it gives better results then the more pessimistic formulation in shorter computation time. It is possible to still use this formulation with success given that the result is properly checked. We must also point out that the steady-state formulation used in [3] failed to generate a schedule in many cases. As stated in earlier sections, steady-state formulation cannot schedule any test that has a steady-state temperature higher than the given temperature bound. Since stacking dies on each other greatly increases the thermal resistance to ambient of many modules, it is not uncommon to see the steady-state temperatures rise greatly, even with high TTSV coverage. This leads to a high rate of failure for steady-state formulations, especially with power-intensive tests, no matter how long the tests are. A final point to make is that two benchmark stacks (stack13 and stack16) failed to achieve a schedule under any formulation. These stacks are included

TABLE I
Benchmarks Composing Different Stacks

Stack	ITC02 Benchmark	Cores	Longest Test	Max Temp
stack1	d281	8	9.63	93.42
stack2	f2126	4	4.16	69.91
stack3	d695	10	5.13	84.28
stack4	p22810	28	1.67	123.56
stack5	h953	8	9.49	82.10
stack6	p34392	19	8.99	126.26
stack7	g1023	14	8.74	95.61
stack8	d281, g1023	22	9.63	109.87
stack9	f2126, h953	12	9.47	70.19
stack10	d695, p34392	29	8.99	118.11
stack11	d281, f2126	12	9.63	105.31
stack12	h953, p22810	36	9.47	92.40
stack13	g1034, p34392	33	8.99	170.90
stack14	g1023, d281	22	9.63	105.96
stack15	d281, f2126, d695	22	9.63	135.19
stack16	h953, p34392, g1023	41	9.47	183.90

TABLE II
Benchmark Results Under Different Formulations

	Formulation from [3]		Optimistic Formulation		Pessimistic Formulation	
Stack	Runtime	Slack	Runtime	Slack	Runtime	Slack
stack1	0.81	26.16	0.76	10.26	0.56	10.26
stack2	0.05	50.87	0.01	49.51	0.07	49.51
stack3	6.02	38.21	0.43	32.86	0.66	34.26
stack4	X	X	12.29	**IS**	631.12	4.51
stack5	0.42	31.35	0.04	31.35	0.17	31.35
stack6	1046	20.26	1.34	13.37	683.55	20.26
stack7	10.79	23.68	0.2	19.26	9.44	19.26
stack8	612.02	15.84	0.62	9.62	205.55	15.84
stack9	6.39	49.68	0.06	49.68	0.54	49.68
stack10	902.52	1.12	2.17	0.53	600.21	1.12
stack11	3.28	14.68	0.11	11.22	1.19	14.68
stack12	X	X	7.24	25.54	600.27	31.26
stack13	X	X	X	X	X	X
stack14	X	X	1.99	16.65	124.45	16.65
stack15	1282.46	4.54	1.45	3.84	160	4.02
stack16	X	X	X	X	X	X

IS indicates an invalid solution due to negative slack.

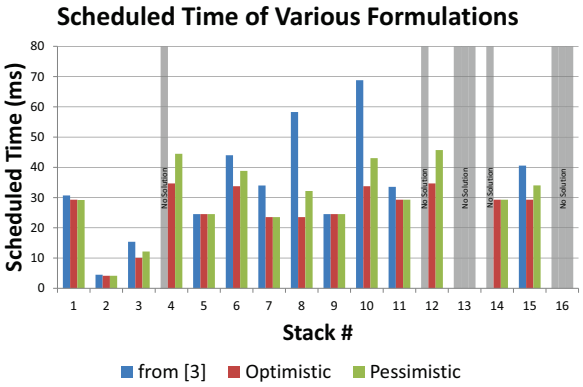

Fig. 2. Scheduled times of various formulations.

to demonstrate that some tests, even under realistic conditions, may be impossible to schedule under a given temperature constraint. This situation is especially true for 3DICs. Although outside the scope of this study, the only way to resolve this issue is to change the original test itself. Though the scheduler may not be able to change the individual vectors of a test, the scheduler may have the ability to divide a large test into smaller tests. This can allow any hot test to effectively have its maximum temperature reduced to the maximum temperature of the smaller tests. However, this approach will not be effective using a steady-state model, since the power values of these tests will remain the same and therefore have the same steady-state temperatures as before. Given how stacking dies drastically increases thermal resistance, this approach of dividing large tests into smaller ones appears inevitable and remains to be a problem for our future research.

VII. Conclusions

MILP formulations based on the superposition principle have been shown to provide superior schedules compared to previous formulations, as well as having faster computation times. This decrease in overall test time comes from the ability to use actual temperature traces instead of approximations like steady-state models, and the faster computation times comes form the linear nature of the superposition principle.

However, we discovered in this paper that in some cases any formulation discussed in the paper will fail, especially with 3DSICs. This fact, as pointed out in previous sections, is due to the condition that single test's temperature profile has a maximum temperature that will violate the given constraint by itself, making it impossible to schedule that test without partitioning it into several smaller tests. This observation is even more important because it implies that the steady-state model will not be able to address this issue, since dividing a test into smaller tests does not decrease steady-state temperatures. This problem becomes magnified with 3DSICs, since when individual dies are stacked on top of each other the thermal characteristics become more difficult to deal with. When dies are stacked, the power during test must be reduced from the single-die case. Since it is often undesirable to change the tests (it may not be possible provided the desired fault coverage), dynamically partitioning the test during scheduling is a more practical solution. Although outside of the scope of this study, this subject is primed for future work.

Further, we believe that there is a need for a set of benchmarks circuits for research to continue in this area of thermal aware test scheduling. Availability of benchmarks will promote development of even more efficient algorithms and researchers will be able to fairly compare various algorithms. We are currently developing such benchmarks which will be released within a year.

References

[1] C. Yao, K. K. Saluja, and P. Ramanathan, "Test Scheduling for Deep Submicron Technologies," in *International Conference on VLSI Design*, 2011.

[2] P. Tadayon, "Thermal Challenges During Microprocessor Testing," *Intel Technology Journal*, vol. 4, no. 3, pp. 1–8, 2000.

[3] D. R. Bild, S. Misra, T. Chantemy, P. Kumar, R. P. Dick, X. S. Huy, and A. Choudhary, "Temperature-aware test scheduling for multiprocessor systems-on-chip," in *2008 IEEE/ACM International Conference on Computer-Aided Design*, pp. 59–66, IEEE, Nov. 2008.

[4] P. Rosinger, B. Al-Hashimi, and K. Chakrabarty, "Thermal-Safe Test Scheduling for Core-Based System-on-Chip Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, pp. 2502–2512, Nov. 2006.

[5] C. Yao, K. K. Saluja, and P. Ramanathan, "Power and Thermal Constrained Test Scheduling Under Deep Submicron Technologies," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 317–322, Feb. 2011.

[6] E. Marinissen, "Testing TSV-based three-dimensional stacked ICs," *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010*, pp. 1689–1694, 2010.

[7] R. Chou, K. K. Saluja, and V. Agrawal, "Scheduling tests for VLSI systems under power constraints," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 5, pp. 175–185, June 1997.

[8] N. Vinay, I. Rawaty, E. Larssonz, M. Gaurx, and V. Singh, "Thermal aware test scheduling for stacked multi-chip-modules," in *2010 East-West Design & Test Symposium (EWDTS)*, pp. 343–349, IEEE, Sept. 2010.

[9] B. Noia, K. Chakrabarty, and S. Goel, "Test-Architecture Optimization and Test Scheduling for TSV-Based 3-D Stacked ICs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 11, pp. 1705–1718, 2011.

[10] F. A. Hussin, T. E. C. Yu, T. Yoneda, and H. Fujiwara, "RedSOCs-3D: Thermal-safe test scheduling for 3D-stacked SOC," in *2010 IEEE Asia Pacific Conference on Circuits and Systems*, pp. 264–267, IEEE, Dec. 2010.

[11] K. Skadron, M. Stan, W. Huang, and D. Tarjan, "Temperature-aware microarchitecture," in *30th Annual International Symposium on Computer Architecture, 2003. Proceedings.*, pp. 2–13, IEEE Comput. Soc, 2003.

[12] C. Yao, K. K. Saluja, and P. Ramanathan, "Thermal-Aware Test Scheduling Using On-chip Temperature Sensors," in *2011 24th Internatioal Conference on VLSI Design*, pp. 376–381, IEEE, Jan. 2011.

[13] P. Gschwandtner, T. Fahringer, and R. Prodan, "Performance Analysis and Benchmarking of the Intel SCC," in *2011 IEEE International Conference on Cluster Computing*, pp. 139–149, IEEE, Sept. 2011.

[14] M. Ni, Q. Su, Z. Tang, and J. Kawa, "An Analytical Study on the Role of Thermal TSVs in a 3DIC Chip Stack," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2011.

[15] J. Cong, L. Guojie, and S. Yiyu, "Thermal-aware cell and through-silicon-via co-placement for 3D ICs," *Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE*, pp. 670–675, 2011.

[16] Y. Chen, E. Kursun, D. Motschman, C. Johnson, and Y. Xie, "Analysis and mitigation of lateral thermal blockage effect of through-silicon-via in 3D IC designs," in *IEEE/ACM International Symposium on Low Power Electronics and Design*, pp. 397–402, IEEE, Aug. 2011.

[17] P. Ghosal, H. Rahaman, and P. Dasgupta, "Thermal Aware Placement in 3D ICs," in *2010 International Conference on Advances in Recent Technologies in Communication and Computing*, pp. 66–70, IEEE, Oct. 2010.