

Reliability-Constrained Die Stacking Order in 3DICs Under Manufacturing Variability

Tuck-Boon Chan[‡], Andrew B. Kahng^{†‡} and Jiajia Li[‡]

[†]CSE and [‡]ECE Departments, UC San Diego, La Jolla, CA 92093
{tbchan, abk, jil150}@ucsd.edu

Abstract—3D integrated circuits (3DICs) with through-silicon vias (TSVs) are an important direction for semiconductor-based products and “More than Moore” scaling. However, 3DICs bring simultaneous challenges of reliability (power and temperature in stacks of thinned die) as well as variability (performance and power) in advanced technology nodes. In this paper, we study variability-reliability interactions and optimizations in 3DICs. Initial motivating studies show that in the presence of manufacturing variability, different die stacking orders can lead to as much as 2 years ($\sim 44\%$) difference in MTTF of a 3DIC stack. We study MTTF-driven die-stacking optimization with consideration of variability, and propose a “rule-of-thumb” guideline for stacking optimization to improve peak temperature as well as reliability in 3DICs. We also propose integer-linear programming (ILP) methods for reliability-driven die-stacking optimization. Our methods can achieve $\sim 7\%$ and $\sim 28\%$ improvement in average and minimum MTTF, respectively, of 3DICs; we also achieve $\sim 3\%$ improvement in performance under fixed reliability constraints. Our stacking optimizations can help improve 3DIC product yields under reliability requirements. Our research also yields the notable observation that a limited amount of manufacturing variation can “help” improve 3DIC product reliability when die-stacking optimization is applied.

I. INTRODUCTION

Stacked-die 3D integrated circuits (3DICs) using through-silicon via (TSV) technology are an emerging architecture for heterogeneous integration and More-than-Moore scaling in late-CMOS technologies. A 3DIC die stack, or simply *stack*, offers increased transistor density in a given form factor, as well as potential cost and yield benefits (multiple smaller dies versus a single larger die). However, the stacking of multiple thinned die (also referred to as *tiers*) increases power density, creating temperature management and reliability challenges. Puttaswamy et al. [19] show that 3DICs with two tiers and four tiers increase peak temperature by 17°C and 33°C , respectively, compared to planar implementations. Since current density and temperature have a significant impact on IC reliability, reliability issues are especially important in the 3DIC context [22], [23].

With technology scaling, additional challenges arise from *process variability*, with variation sources spanning dopant fluctuation, mask data preparation and OPC, line-edge roughness, misalignment in double-patterning, and a variety of across-field and across-wafer variability mechanisms [11], [12]. These process variations are present (and, uncorrelated) within a 3DIC stack; because of higher temperatures due to die stacking, the process variations can heavily affect performance as well as leakage power of the 3DIC product [7]. So that a given product can meet its performance requirements, process variations in each manufactured die are typically characterized at manufacturing time (e.g., for product binning, or to set one-time programmable tables for adaptive voltage scaling [10], [26]).

In this paper, we study reliability-variability interactions and optimizations in the context of 3DIC die stacking. Specifically, we focus on the stacking of multiple copies of logic dies (e.g., as envisioned for many-core processor die in high-performance computing architectures) [4], [20]. We use the term *stacking style* to indicate both the selection of dies which exhibit particular process variations (e.g., fast, typical, or slow dies) as well as the ordering of dies within a given 3DIC product stack. Because of inter-die process

variation, the choice of stacking styles will impact performance, power consumption and reliability of 3DICs. In our studies, the required performance for each die is predefined, and the *adaptive voltage scaling* (AVS) [6] is assumed.

There are three methods to bond dies in 3DIC fabrication: die-to-die, die-to-wafer and wafer-to-wafer. Applying different methods results in different flexibility, yield, and cost. Among these three methods, die-to-die bonding offers the highest flexibility and yield, but also incurs high cost. On the other hand, although wafer-to-wafer bonding offers the highest throughput in production, bad dies cannot be scrapped before bonding, which results in the lowest yield. Die-to-wafer bonding, which is easy to implement while offering flexibility and yield that are similar to the die-to-die method, is promising for 3DIC fabrication. Our paper applies primarily to the die-to-die and die-to-wafer bonding contexts.

We assume that all logic dies are identical (a similar assumption can be applied to memory-logic integration, where all memory dies are identical). Such a case may arise if applying the identical design to all the tiers in a stack reduces design efforts as well as manufacturing cost [8]. Based on such an assumption, dies can be used interchangeably in different tiers. Hence, we are able to change the stacking order during optimization.

Our discussion below will assume face-to-back stacking of the multiple logic-die tiers, with a heat sink (or other heat removal mechanism) adjacent to the top tier as illustrated in Figure 1. The figure shows a “STF” stacking order for a 3-tier stack, i.e., a slow-corner die on bottom, typical-corner die in the middle, and fast-corner die on top.

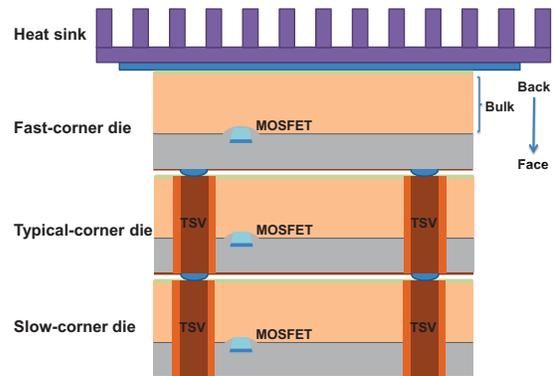


Fig. 1. “STF” stack in which a slow-corner die is located on the bottom tier, a typical-corner die in the middle, and a fast-corner die on the top tier (adjacent to the heat sink).

To motivate our present work, Figure 2 shows the *mean time to failure* (MTTF) of 3-tier stacks with different stacking styles (orders). The maximum difference in MTTF resulting from different stacking styles can be up to 2 years (44%). The study is conducted by estimating temperature using Hotspot [32] and calculating MTTF based on Black’s equation [2], with the assumption that the supply voltage of each tier in a stack is adjusted using *adaptive voltage scaling* (AVS) to meet a given fixed performance requirement.

Definition: The *reliability* ($R(t)$) is defined as the probability that a device (or a die) operating under specified conditions shall perform satisfactorily for a given period of time (t).

The reliability can be calculated as [27]

$$R(t) = e^{-\lambda \cdot t} \quad (2)$$

Based on (2) and a constant λ during the useful lifetime, the MTTF of a die (i.e., expectation of the time to failure) can be calculated as

$$MTTF = \int_0^{\infty} R(x) \cdot dx = \int_0^{\infty} e^{-\lambda \cdot x} \cdot dx = \frac{1}{\lambda} \quad (3)$$

Note that according to (3), the value of λ can be calculated using Black's equation (1).

Furthermore, since any failure of any die in a 3DIC can cause the 3DIC to fail, the failure rate of a 3DIC can be evaluated as

$$\lambda_{stack} = \prod_{i=1}^L \lambda_{die_i} \quad (4)$$

where λ_{stack} is the failure rate of the 3DIC, and λ_{die_i} ($i = 1, 2, \dots, L$) is the failure rate of the i^{th} die in the stack. Based on (3) and (4), the MTTF of a 3DIC is

$$MTTF_{stack} = \frac{1}{\prod_{i=1}^L \frac{1}{MTTF_{die_i}}} \quad (5)$$

where $MTTF_{stack}$ is the MTTF of the 3DIC, and $MTTF_{die_i}$ ($i = 1, 2, \dots, L$) is the MTTF of the i^{th} die in the stack.

In our MTTF calculations reported below, we use Black's equation (1) to estimate the MTTF for each die in a 3DIC based on temperature and current density information. We then apply (5) to calculate the MTTF of a given 3DIC.

B. Process Variation

Given an arbitrary number of dies, each exhibiting different process variation (e.g., characterized during manufacturing test [10], [26]), the number of possible stacking styles in 3DICs composed of these dies can be quite large. For example, if there are 2000 distinct (in terms of process variation) manufactured dies, and the 3DIC to be produced has 5 tiers, then the number of distinct stacking styles is $P(2000, 5) = 2000 \cdot 1999 \cdot \dots \cdot 1996$. Stacking the 2000 dies into 400 5-tier stacks would have an even more unmanageable solution space, wherein figuring out the optimal (set of) stacking styles is intractable. (As noted above, the previous work of [7] shows that the stacking optimization problem for certain objectives is NP-hard.)

In our work, we classify dies into a *constant* number of (i.e., $O(1)$) process *bins* according to the speed of dies. Dies are classified into the same bin if they have similar process variations, and to make the stacking optimization tractable, we assume the same process variation characteristics for all dies that are classified into a given bin. This bin-based model assumption greatly reduces the number of distinct stacking styles as well as the solution space for stacking optimization. (E.g., for the same example of 2000 manufactured input dies and a 5-tier stack, if we classify the dies into 3 process bins, the number of feasible stacking styles is reduced to 3^5 .)

Taking advantage of such a bin-based model, we are able to explore the reduced solution space and determine optimal stacking styles when given a small number of bins. When instantiating each distinct 3DIC stack (e.g., each of the 400 5-tier stacks to be made out of 2000 manufactured dies), we randomly select dies from corresponding bins to make up the stack.¹ As discussed below, when

¹For example, in a "FTTTS" 5-tier stack, we would successively pick one random die from the Fast bin, three random dies from the Typical bin, and one random die from the Slow bin, and stack them bottom-up in this order.

the number of process bins is sufficiently large, results from stack optimization flows that apply the bin-based models can be near-optimal.

C. A Rule-of-Thumb

For EM reliability, peak temperature is the main determinant of a given die's reliability. Additionally, the die with the weakest reliability in a stack determines the reliability of the entire stack. Thus, to optimize reliability of a 3DIC, we seek to minimize the peak temperature among all stacked dies in a 3DIC. It is not difficult to realize that two factors have significant impacts on the temperature of dies in a 3DIC stack: process variation and stacking order.

As previewed in Section I above, we assume that the same performance requirement is applied to all dies in a 3DIC, and that to compensate for interdie process variation, AVS is deployed.² In this context, individual dies will have different supply voltages, corresponding to process variation. Slow dies require higher supply voltages than fast dies in order to satisfy the performance requirements. Such high supply voltages can lead to high power consumption on slow dies, which increases temperature. Hence, as a consequence of process variation and deployment of AVS, slow dies will have higher temperature than fast dies.

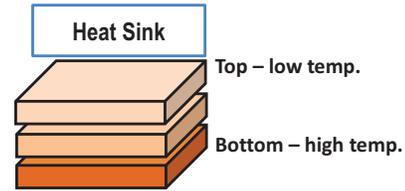


Fig. 4. Temperature gradient. The top-tier die is directly contacted to the heat sink, and thus has the lowest temperature. Due to intervening dies that block thermal conduction to the heat sink, dies in bottom tiers have higher temperature.

The stacking order can also affect the temperature distribution of dies. We assume that a vertical temperature gradient always exists in the 3DIC stack, because only the top-layer die is directly contacted to the heat sink (a cartoon is shown in Figure 4). For dies in lower tiers, the thermal dissipation through the heat sink, which is the primary mechanism for thermal dissipation in 3DICs, is blocked by dies in the upper tiers. Hence, higher temperatures are observed in bottom-tier dies. Moreover, heat generated from dies in adjacent tiers exacerbates thermal issues for any individual die in the stack. Figure 5 shows a simulated 5-tier 3DIC, where all dies in the stack are assumed to exhibit the same process variation. In this example, the maximum temperature difference between the bottom-layer die and the top-layer die is 35°C .³

Based on the above analysis, considering effects of process variation as well as stacking order on temperature distribution, we expect that if the same performance is required from dies which exhibit different process variations, the worst-case peak temperature among feasible stacking styles will occur when we locate slower dies in lower tiers (e.g., the slowest die is located in the bottom tier, the second-slowest die is located in the next-to-bottom tier, and so on). Furthermore, such worst-case peak temperature will likely correspond to the minimum MTTF among all stacking styles, that is, the worst-case of reliability of the 3DIC. On the other hand, if we locate slow dies on top, by taking advantage of thermal

²Indeed, for nearly all low-power consumer SOCs in advanced nodes today, sensor-based AVS is the norm; it is the only available mechanism to recover power from a chip that has been overdesigned due to large model guardbanding.

³In the simulation, we assume that the thermal resistivity for silicon is 100mK/W , die thickness is $50\mu\text{m}$, ambient temperature is 45°C , and that there is a heat sink on top of the stack.

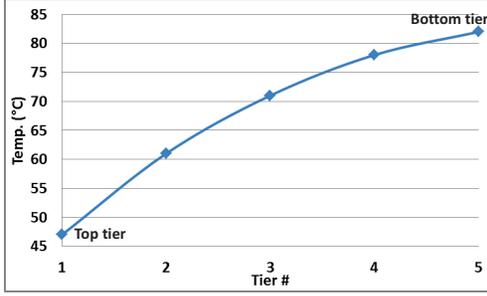


Fig. 5. Example simulated temperature gradient in a 5-tier 3DIC stack. The difference between the peak temperatures in the bottom-tier die and the top-tier die can reach 35°C.

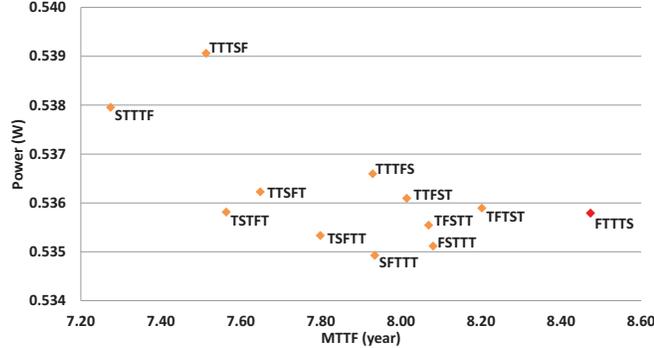


Fig. 6. QoR metrics (MTTF, power) of stacks with different stacking orders. Placing slow dies close to the heat sink helps achieve large MTTF of stacks.

dissipation through the heat sink, high temperature caused by high supply voltages can be relieved. Hence, the peak temperature will decrease, and the MTTF of the stack will increase. Note that even when the thermal gradient is small, the vertical thermal distribution is still monotonic, so that placing slow dies on top still results in improved MTTF.

The experimental results shown in Figure 6 confirm our expectation. Figure 6 shows QoR metrics (MTTF and power) of 5-tier stacks implemented with different stacking orders. We observe in the experimental results that placing slow dies close to the heat sink helps improve the MTTF of the stack. We conclude this part of our discussion with the following *rule-of-thumb*.

Rule-of-thumb: To optimize reliability of a 3DIC, the slowest dies should be located closest to the heat sink in the stack.

The rule-of-thumb can further reduce the complexity of the stacking optimization problem, since for a stack with fixed composition, the reliability-aware optimal stacking order can be fixed according to the rule-of-thumb. In other words, for a stack whose input dies are given, instead of enumerating all permutations (stacking orders), the optimal stacking style is defined by the rule-of-thumb. Therefore, in a case where input dies are classified into K bins and output stacks are assumed to have L tiers, the number of stacking styles that need to be considered for reliability-driven stacking optimization can be reduced from K^L to $\binom{K+L-1}{L}$.

III. PROBLEM FORMULATION

Given N dies which are classified into K bins, we want to determine the optimal stacking style for each output stack that contains L tiers. Our experimental results show that power consumption mainly depends on composition of the stack. We observe that for a particular number of given input dies, the power consumption of output stacks exhibits only slight differences ($<1\%$) across different stacking orders, while the difference in MTTF can be up to 16% for 5-die implementations. Therefore, we only focus on optimization for reliability in our studies. Three exemplary

reliability-driven stacking optimization problems are formulated as follows.

Formulation 1: *OPT_MTTF*.

One objective of reliability-driven 3D stacking optimization is to maximize the sum of MTTFs of output stacks ($MTTF_{sum}$), where a required frequency (f_{req}) is predefined as a constraint for dies in a stack. AVS is applied to achieve the same performance across dies. In other words, in a 3DIC, due to interdie process variation, each die has a particular supply voltage corresponding to its process variation. The problem that searches for the optimal stacking style of each stack can be formulated as follows.

OPT_MTTF: Given N dies, each of which is classified into one of the K process bins

Maximize $MTTF_{sum}$

such that frequency of each die in a stack = f_{req}

Formulation 2: *OPT_YIELD*.

We may also optimize the minimum MTTF ($MTTF_{min}$) among all output stacks to improve the yield of 3DICs with respect to a particular reliability (MTTF) requirement. In this scenario, MTTF constraints are predefined for 3DICs, and when constraints are not satisfied, the failed 3DICs are scrapped. The objective for optimization is to maximize the number of good stacks.

OPT_YIELD: Given N dies, each of which is classified into one of the K process bins

Maximize Number of good stacks

such that frequency of each die in a stack = f_{req}

MTTF of each good stack $\geq MTTF_{req}$

Note that we can maximize the minimum MTTF over all stacks by performing binary search over $MTTF_{req}$, until the number of good stacks equals to the number of all stacks (i.e., N/L).

Formulation 3: *OPT_PERFORMANCE*.

We also formulate a reliability-driven stacking optimization problem to improve the performance (f_{stack}) of 3DICs where reliability constraints are applied, e.g., by setting a lower bound MTTF ($MTTF_{req}$) on 3DICs.

OPT_PERFORMANCE: Given N dies, each of which is classified into one of the K process bins

Maximize f_{stack}

such that MTTF of each stack $\geq MTTF_{req}$

IV. METHODOLOGY

A. ILP-Based Method

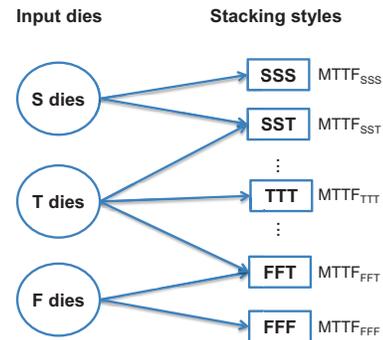


Fig. 7. Allowed assignments in ILP-based stacking optimization method.

We propose an ILP-based method for reliability-driven stacking optimization. As mentioned in Section III, inputs of such optimization are N dies that are classified into K bins, while outputs are stacks such that each stack has L tiers. We create matching relationships between input dies and feasible stacking styles of output stacks. This is conceptually shown in Figure 7.

Vertices on the left part of the bipartite graph indicate the available pools/populations of input dies which are classified into process bins. Feasible stacking styles are enumerated on the right part of the bipartite graph. In the graph, all input dies classified into a particular process bin are connected to the stacking styles containing dies belonging to that bin. The relationships between input dies and stacking styles define the assignment constraints in the ILP formulation. Such constraints indicate that each die can be used exactly once. During the assignments, the process bins should be consistent between the composition of stacking styles and the dies used (e.g., a die belonging to the Slow bin cannot be assigned to the stacking style “FFT”). Each stacking style corresponds to a MTTF estimated from simulation. In the ILP-based method, we optimally assign input dies to stacking styles to maximize the sum of MTTFs of output stacks. We give the notations and formulate the ILP as follows.

Notations:

- 1) Die_i ($i = 1, 2, \dots, N$): input dies
- 2) $Style_j$ ($j = 1, 2, \dots, M$): feasible stacking styles (based on the rule-of-thumb, $M = \binom{K+L-1}{L}$)
- 3) Bin_q ($q = 1, 2, \dots, K$): process bins
- 4) X_q ($q = 1, 2, \dots, K$): number of input dies that are classified into Bin_q , such that $\sum_{1 \leq q \leq K} X_q = N$
- 5) $Y_{q,j}$ ($q = 1, 2, \dots, K; j = 1, 2, \dots, M$): number of dies that are classified to Bin_q contained in $Style_j$, such that $\forall j \sum_{1 \leq q \leq K} Y_{q,j} = L$
- 6) $MTTF_j$ ($j = 1, 2, \dots, M$): MTTF of the stack implemented with $Style_j$
- 7) C_j ($j = 1, 2, \dots, M$): number of output stacks implemented with $Style_j$, where $L \cdot \sum_{1 \leq j \leq M} C_j = N$.

ILP formulation ($OPT_MTTF, OPT_PERFORMANCE$):

$$\text{Maximize } \sum_{1 \leq j \leq M} MTTF_j \cdot C_j \quad (6a)$$

$$\text{Such that } \sum_{1 \leq j \leq M} C_j \cdot Y_{q,j} = X_q, \forall q \quad (6b)$$

$$C_j \geq 0, \forall j \quad (6c)$$

This formulation is used to solve the OPT_MTTF and $OPT_PERFORMANCE$ problems. In the formulation, (6a) gives the objective of maximizing the sum of MTTFs of output stacks; (6b) are the assignment constraints, which indicate that each input die should be used exactly once in the stacking implementation, consistent with its process bin; and (6c) are the non-negativity constraints which indicate that the number of output stacks implemented with stacking style $Style_j$ cannot be negative. An additional loop, which searches for the maximum frequency, is applied to solve the $OPT_PERFORMANCE$ problem.

To solve the OPT_YIELD problem, we set up the following ILP.

ILP formulation (OPT_YIELD):

$$\text{Maximize } \sum_{1 \leq j \leq M} C_j \quad (7a)$$

$$\text{Such that } \sum_{1 \leq j \leq M} C_j \cdot Y_{q,j} = X_q, \forall q \quad (7b)$$

$$C_j \geq 0, \forall j \quad (7c)$$

$$C_j \cdot MTTF_j \geq C_j \cdot MTTF_{min} \quad (7d)$$

where (7a) is the objective function which maximizes the number of good dies; (7b) are the assignment constraints; (7c) are the non-negativity constraints; and (7d) are the lower-bound constraints on MTTF, in which $MTTF_{min}$ indicates the lower bound. Note that

the factor C_j in (7d) eliminates the constraints on the MTTF for stacking styles which are not implemented (i.e., when $C_j = 0$, the MTTF of $Style_j$ does not affect $MTTF_{min}$). We determine the maximum value of $MTTF_{min}$ by doing binary search. The binary search terminates when the change in $MTTF_{min}$ is less than 0.01 year.

B. Greedy Method

We also study a greedy method, based on process binning, for reliability-driven stacking optimization. We evaluate MTTFs of all stacking styles. Then, we select the stacking style with maximum MTTF for each stack, one at a time. A stacking style is valid only if the numbers of dies required by the stacking style are less than or equal to the remaining dies in process bins.

C. “Zig-zag” Heuristic Method

The rule-of-thumb proposed in Section II suggests that slower dies should be located closer to the heat sink. Based on the rule-of-thumb, we propose a heuristic method which stacks dies in a “zig-zag” manner as shown in Figure 8.

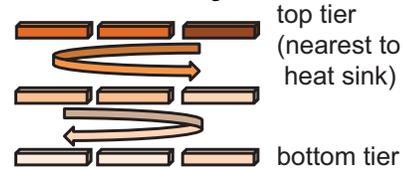


Fig. 8. Zig-zag method: stack dies from slow to fast, from top tiers to bottom tiers.

Given the input dies, we sort them according to their performance (as measured at manufacturing test). Then, we assign the sorted dies (starting from the slowest die) one at a time, from top tiers to bottom tiers. For die assignment in each tier, we record the sequence of die assignment. The sequence is reversed when we start the assignment for the next tier. In this way, all output stacks satisfy the rule-of-thumb proposed in Section II. The time complexity of this method is $O(n \cdot \log n)$ (n indicates the number of input dies), which is required for die sorting. As we will discuss in Section V, the zig-zag stacking method offers similar or even better QoR compared to the ILP-based method.

V. EXPERIMENTS

A. Implementation Tools

Our experiments use RTL design *JPEG* obtained from the OpenCores website [28] as the logic die. The design is implemented using 65nm NVT, LVT and HVT libraries. The RTL is synthesized using *Synopsys Design Compiler vC-2009.06-SP2* [29] and then placed and routed using *Cadence SoC Encounter vEDI10.1* [31]. We characterize all libraries at different corners (SS, TT and FF), for a range of voltages (0.8V-1.2V) and temperatures (45°C-165°C) using *Cadence Library Characterizer vEDI9.1* [31]. Timing analyses and power estimation are performed using *Synopsys PrimeTime C2009.6* [30]. We estimate the temperature of stacks using *Hotspot 5.02* [32] and solve ILPs using *lp_solve 5.5* [33].

B. Hotspot Configuration

In Hotspot, we set the chip thickness as 50μm, convection capacitance as 140.4J/K, convection resistance as 0.7K/W, ambient temperature as 60°C, the thickness of heat spreader and heat sink as 1mm and 6.9mm respectively [21], [32]. Based on number of I/O pins (~100 per die), we set the spreader side and the heat sink side as 15mm and 30mm, respectively, for 5-tier stacks. We model TSVs by changing the thermal resistivity of *thermal interface material* layers [17]. The thermal resistivity of such a layer is set as 0.2mK/W.

C. Estimation of Stacks' MTTF

We implement a flow deploying voltage-temperature feedback loops to estimate the MTTF of an output stack or a stacking style. A change in temperature will change performance. Thus, voltages are altered to retain the required performance. This in turn results in a change in the temperature, and again affects frequencies. Taking such a “chicken-egg” chain into consideration, an accurate estimation of MTTF requires a feedback loop in the analysis flow, as illustrated in Figure 9. The inputs to the flow are stacking styles, the required frequency and an initial temperature (ambient temperature). Then, a voltage-temperature feedback loop is applied to each tier. To avoid large execution time resulting from running simulation for power and timing analysis in each loop, we build lookup tables and apply interpolation to estimate the supply voltage and power consumption.

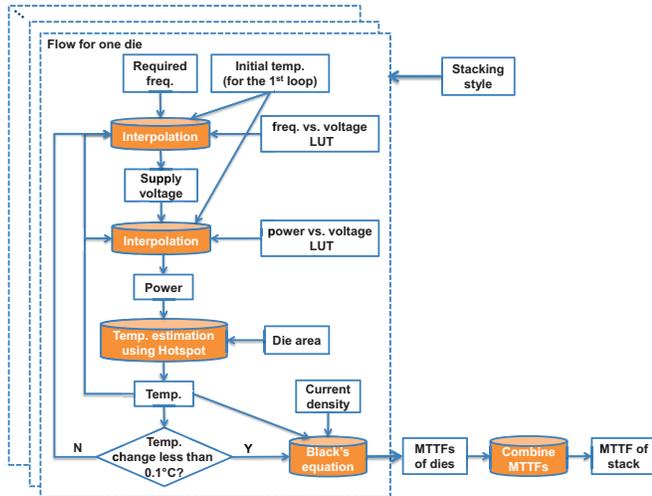


Fig. 9. The flow of MTTF estimation.

First, based on the required frequency and temperature either from input (for the first loop) or from the Hotspot simulation, we estimate the required supply voltage.

Second, based on the supply voltage we estimate the power consumption of each die. According to the estimated power and area of each die, *Hotspot* is used to estimate each die's temperature. Then, we check the temperature change with respect to the previous loop. If the change in temperature is less than 0.1°C , the loop converges. Otherwise, the impact of temperature change on frequencies is estimated, and another loop is applied. Based on the output temperature from the voltage-temperature feedback loop, together with the current density, we use Black's equation (1) to evaluate the MTTF of each die. We calculate the MTTF of the entire stack using the model proposed in Section II.

D. Design of Experiments

We implement experiments on the *JPEG* circuit [28] in TSMC 65nm technology. We assume that the process variation distributions of input dies are Gaussian, where the SS corner and FF corner of TSMC 65nm technology are at $\pm 3\sigma$. For the bin-based model of process variation, we implement 30 trials of picking dies randomly from process bins to stack 3DICs. We observe that the variation in results are small ($< 1\%$), so we only show the average results of the 30 trials in the following discussion.

In the experiments, we compare QoR of four different stacking methods including the ILP-based method (ILP), the zig-zag heuristic method (Zig-zag), the Greedy method (Greedy) and a reference case where no stacking optimization is applied (Random). Three problems formulated in Section III are studied.

OPT_MTTF: We implement four methods to optimize the sum of MTTFs of output stacks (Cases 1-6 in Table I). The average MTTF of stacks resulting from four methods are compared.

OPT_YIELD: We implement four methods to optimize the minimum MTTF of output stacks. We apply different MTTF limitations on output stacks and compare the number of good stacks resulting from different optimization methods. Such experiments are implemented on Case 5 shown in Table I.

OPT_PERFORMANCE: We implement stacking optimization to improve performance of stacks under the reliability constraints. For 3-tier cases (Case 1 in Table I), the lower bound on MTTF is set as 12 years; for 4-tier and 5-tier cases (Cases 2-6 in Table I), such constraints are set as 10 years.

TABLE I
EXPERIMENT DESIGN FOR RELIABILITY-DRIVEN STACKING OPTIMIZATION.

Case	# of dies	# of tiers	σ	μ	# of bins
1	1200	3	1.0	0.0	9
2	1600	4	1.0	0.0	9
3	2000	5	0.2	0.0	9
4	2000	5	0.6	0.0	9
5	2000	5	1.0	0.0	9
6	2000	5	1.4	0.0	9

VI. RESULTS

A. Results for Optimization Problems

Results for OPT_MTTF. We study the impacts of bin-based modeling and the number of input dies on QoR of the ILP-based method to solve the *OPT_MTTF* problem.

Figure 10 shows the average MTTF of output stacks resulting from the ILP-based method modeled with different number of bins. We observe that as the number of bins increases, better MTTF is achieved. With certain number of bins (e.g., 13 in this case), the solution approaches optimality and noise occurs afterwards (e.g., number of bins ≥ 13 in this case).

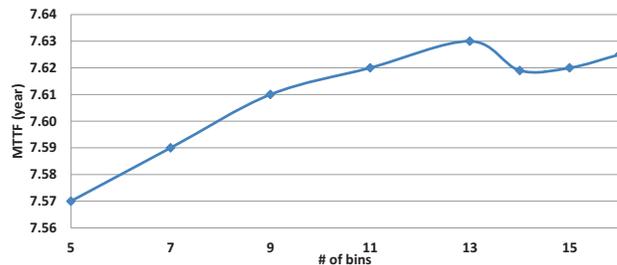


Fig. 10. As the number of process bins increases, MTTF of stacks increases. The results approach optimality when the number of bins is equal to 13, noise appears after that.

Table II shows the QoR of the ILP-based method with different numbers of input dies and the number of process bins equal to 9. We observe that as the number of dies increases, the average MTTF of output stacks increases. This is because the degradation induced by discretization in bin-based modeling reduces as number of dies increases. We also use the zig-zag heuristic method as reference and observe that (for this experiment, where the number of bins is 9) the zig-zag heuristic method always performs better than the ILP-based method.

From the results in Table III we observe that the ILP-based and the zig-zag heuristic methods offer $\sim 7\%$ improvement in the average MTTF ($MTTF_{avg}$) compared to the random method where no optimization is applied.

Results for OPT_YIELD. Table III shows that the ILP-based and the zig-zag heuristic methods achieve $\sim 28\%$ improvement in the minimum MTTF ($MTTF_{min}$). In addition, the ILP-based and zig-zag heuristic methods also reduce the variation in MTTFs of stacks,

TABLE II
IMPACT OF NUMBER OF DIES ON QOR OF THE ILP-BASED METHOD.

# of dies	QoR	ILP	Zig-zag
200	MTTF (year)	7.57	7.58
	Power (mW)	533.6	533.6
	Execution time	86min	<1sec
500	MTTF (year)	7.58	7.59
	Power (mW)	533.4	533.4
	Execution time	86min	<1sec
2000	MTTF (year)	7.61	7.63
	Power (mW)	530.9	531.0
	Execution time	86min	<1sec
10000	MTTF (year)	7.63	7.65
	Power (mW)	530.6	530.6
	Execution time	86min	<1sec
100000	MTTF (year)	7.63	7.65
	Power (mW)	530.8	530.8
	Execution time	86min	<1sec

TABLE III
QOR OF OUTPUT STACKS FROM DIFFERENT METHODS.

Case		ILP	Zig-zag	Greedy	Random
1	$MTTF_{avg}$ (year)	11.20	11.20	10.37	10.31
	$MTTF_{min}$ (year)	10.42	10.78	6.02	7.20
	Power (mW)	319.4	319.4	318.8	318.8
	f_{max} (MHz)	975.0	975.4	943.4	943.4
	Execution time	11min	<1sec	11min	-
2	$MTTF_{avg}$ (year)	9.85	9.88	9.29	9.23
	$MTTF_{min}$ (year)	9.47	9.66	5.91	7.11
	Power (mW)	424.8	424.9	424.2	424.2
	f_{max} (MHz)	993.7	995.4	966.4	966.5
	Execution time	33min	<1sec	33min	-
3	$MTTF_{avg}$ (year)	7.30	7.30	7.22	7.22
	$MTTF_{min}$ (year)	7.23	7.27	6.71	6.98
	Power (mW)	527.7	527.7	527.4	527.4
	f_{max} (MHz)	860.3	862.3	857.1	856.0
	Execution time	86min	<1sec	86min	-
4	$MTTF_{avg}$ (year)	7.47	7.47	7.22	7.20
	$MTTF_{min}$ (year)	7.30	7.39	5.72	6.40
	Power (mW)	528.9	528.9	528.1	528.2
	f_{max} (MHz)	867.3	869.7	854.7	853.1
	Execution time	86min	<1sec	86min	-
5	$MTTF_{avg}$ (year)	7.61	7.63	7.21	7.16
	$MTTF_{min}$ (year)	7.34	7.51	4.61	5.88
	Power (mW)	530.9	531.0	530.1	530.2
	f_{max}	875.1	876.7	851.8	849.2
	Execution time	86min	<1sec	86min	-
6	$MTTF_{avg}$ (year)	7.78	7.80	7.21	7.12
	$MTTF_{min}$ (year)	7.29	7.62	3.16	5.05
	Power (mW)	533.3	533.4	532.5	532.7
	f_{max} (MHz)	884.0	886.0	849.1	844.3
	Execution time	86min	<1sec	86min	-

which is illustrated in Figure 11. Among the four methods, the greedy method leads to large variation in MTTFs of output stacks.

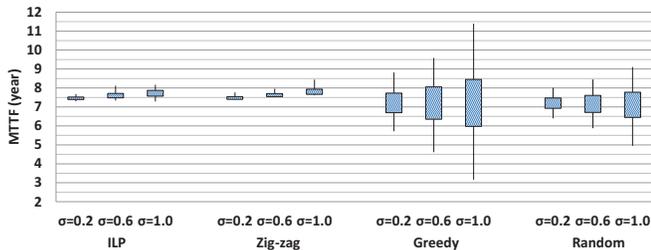


Fig. 11. Stacking optimization using the ILP-based method and the zig-zag method helps increase the minimum MTTF of output stacks, while reducing the variation in MTTFs.

Figure 12 shows the yield of stacks constrained by different $MTTF_{req}$ using the four methods. We observe that the improvement in yield can be up to 300% (when MTTF limitation = 7.5 years) by using the zig-zag heuristic method, compared to the random case.

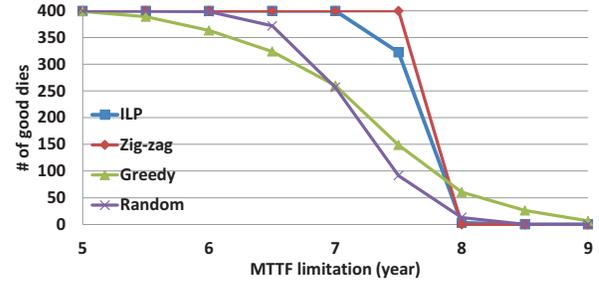


Fig. 12. Yield decreases with MTTF limitation. The ILP-based and the zig-zag heuristic methods help increase the yield of 3DICs compared to the random case.

Results for $OPT_PERFORMANCE$. Table III shows that the ILP-based and the zig-zag heuristic methods offer $\sim 3\%$ improvement in performance compared to the random case.

B. Suboptimality of the Zig-zag Heuristic Method

Although experimental results show that the zig-zag heuristic method performs better than other methods, it is still suboptimal for an adversarial example. Given 6 input dies ($die_{\{1..6\}}$), and each output stack has 3 tiers. Without loss of generality, we assume that die_i is faster than die_j , when $i > j$. Further, we assume large performance difference between die_5 and die_6 . In this example, the output stacks resulting from the zig-zag heuristic method are “ $die_6 die_3 die_2$ ” and “ $die_5 die_4 die_1$ ”. Due to the large performance difference between die_5 and die_6 , the bottom two tiers of stack “ $die_5 die_4 die_1$ ” generate more heat than the bottom two tiers of stack “ $die_6 die_3 die_2$ ”. If we swap die_1 and die_2 , the $MTTF_{min}$ of output stacks is higher. On the other hand, due to the nonlinear relationship between temperature and MTTF, the stacks “ $die_5 die_3 die_1$ ” and “ $die_6 die_4 die_2$ ” can achieve better $MTTF_{avg}$ compared to stacks resulting from the zig-zag heuristic method, at the cost of having larger MTTF variation for output stacks.

C. Variability Helps

The experimental results show that when no stacking optimization is applied, the MTTF of output stacks decreases as process variation increases. However, when stacking optimization is applied, MTTF increases with process variation. This trend is illustrated in Figure 13, in which the solid lines indicate the average MTTF and the dotted lines indicate the minimum MTTF of output stacks with different process variation distributions. When the σ of process variation distribution changes from 0.2 to 0.6, the improvement in the average MTTF changes from 1.1% to 9.6%, while the improvement in the minimum MTTF changes from 4.2% to 50.9%, where the zig-zag heuristic method is applied. A similar benefit from process variation is observed in [9], where process variation with a proposed matching solution helps to reduce clock skew in 3DICs. The benefit from process variation disappears when the variation exceeds a certain amount. This is because supply voltages of slow dies can exceed the maximum voltage allowed by the package as the process variation keeps increasing. Figure 14 shows the increase (decrease) of the maximum (minimum) supply voltage with process variation. If the package can only tolerate up to 1.3V supply voltage, the help of variability in stacking optimization will stop when σ is close to 1.7. Therefore, we conclude that a limited amount of manufacturing variation can “help” improve reliability of die stacks when stacking optimization is applied. In other words, the reliability benefit of stacking optimization depends on the magnitude of die-to-die process variation.

ACKNOWLEDGMENTS

Research support from Sandia National Labs, Qualcomm, Samsung, NSF, SRC and the IMPACT (UC Discovery) center is gratefully acknowledged.

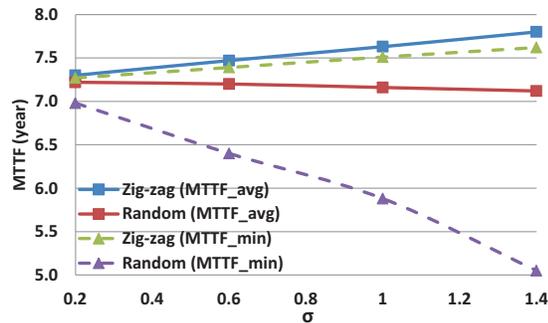


Fig. 13. The solid lines and dotted lines indicate the average and the minimum MTTF of stacks, respectively. As the variation increases, stacking without optimization leads to worse results. However, manufacturing variation helps improve the average MTTF and the minimum MTTF with the zig-zag stacking optimization.

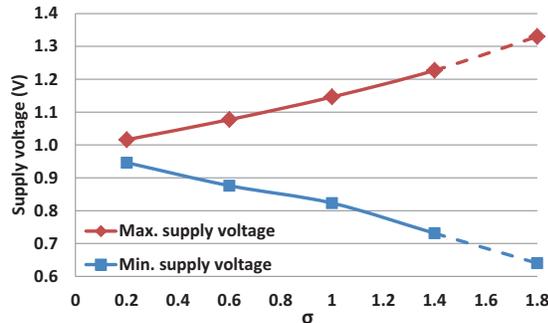


Fig. 14. The maximum supply voltage of stacks increases with process variation, while the minimum voltage decreases. The solid line corresponds to our experimental results. The dashed line is an extrapolation of the trend.

VII. CONCLUSIONS

In this paper, we study variability-reliability interactions and optimizations in 3DIC. We propose a “rule-of-thumb” guideline for stacking optimization to reduce the peak temperature and increase the MTTF of 3DICs. An ILP-based method and an $O(n \log n)$ zig-zag heuristic method for reliability-driven stacking optimization achieve $\sim 7\%$, $\sim 28\%$ and $\sim 3\%$ improvement in average MTTF, minimum MTTF and performance (under reliability constraints) of 3DICs, respectively, compared to the case where no optimization is applied. Optimization of yield of 3DICs under reliability requirements is also implemented. Interestingly, our studies show that a limited amount of variability can “help” to improve reliability of stacks when stacking optimization is applied.

REFERENCES

- [1] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. H. Loh, D. McCauley, P. Morrow, D. W. Nelson, D. Pantuso, P. Reed, J. Rupley, S. Shankar, J. Shen and C. Webb, “Die Stacking (3D) Microarchitecture”, *Proc. International Symposium on Microarchitecture*, 2006, pp. 469-479.
- [2] J. R. Black, “Electromigration – A Brief Survey and Some Recent Results”, *IEEE Transactions on Electronic Devices* 16(4) (1969), pp. 338-347.
- [3] C. B. Cho, W. Zhang and T. Li, “Thermal Design Space Exploration of 3D Die Stacked Multi-core Processors Using Geospatial-Based Predictive Models”, *Proc. SPEC Benchmark Workshop on Computer Performance Evaluation and Benchmarking*, 2009, pp. 102-120.
- [4] R. L. Clay, “Exascale Computing Systems R&D at Sandia”, talk at Texas A&M University, September 27, 2011, <http://www.cs.tamu.edu/tref/clay>
- [5] B. D. Cory, R. Kapur and B. Underwood, “Speed Binning with Path Delay Test in 150-nm Technology”, *IEEE Design and Test of Computers* 20(5) (2003), pp. 41-45.
- [6] M. Elgebaly and M. Sachdev, “Variation-Aware Adaptive Voltage Scaling System”, *IEEE Transaction on VLSI Systems* 15(5) (2007), pp. 560-571.
- [7] C. Ferri, S. Reda and R. I. Bahar, “Parameter Yield Management for 3DICs: Models and Strategies for Improvement”, *ACM Journal on Emerging Technologies in Computing Systems* 4(4) (2008), pp. 19:1-19:22.
- [8] P. D. Franzon, W. R. Davis, M. B. Steer, H. Hao, S. Lipa, S. Luniya, C. Mineo, J. Oh, A. Sule and T. Thorolfsson, “Design for 3D Integration and Applications”, *Proc. International Symposium on Signals, Systems and Electronics*, 2007, pp. 263-266.
- [9] T. Y. Kim and T. Kim, “Post Silicon Management of On-Package Variation Induced 3D Clock Skew”, *Journal of Semiconductor Technology and Science* 12(2) (2012), pp. 139-149.
- [10] M. W. Kuemerle, S. K. Lichtensteiger, D. W. Douglas, and I. L. Wemple, “Integrated Circuit Design Closure Method for Selective Voltage Binning”, *U.S. Patent No. US7475366B2*, January 2009.
- [11] K. J. Kuhn, “Reducing Variation in Advanced Logic Technologies: Approaches to Process and Design for Manufacturability of Nanoscale CMOS”, *Proc. International Electron Device Meeting*, 2007, pp. 471-474.
- [12] K. J. Kuhn, “CMOS Transistor Scaling Past 32nm and Implications on Variation”, *Proc. Advanced Semiconductor Manufacturing Conference*, 2010, pp. 241-246.
- [13] J. Lienig, “Introduction to Electromigration-Aware Physical Design”, *Proc. International Symposium on Physical Design*, 2006, pp. 39-46.
- [14] L. Liu, I. Ganusov, M. Burtscher and S. Tiwari, “Bridging the Processor-Memory Performance Gap with 3D IC Technology”, *IEEE Design and Test of Computers* 22(6) (2005), pp. 556-564.
- [15] G. H. Loh, “3D-Stacked Memory Architecture for Multi-core Processors”, *Proc. ISCA*, 2008, pp. 453-464.
- [16] G. Loi, B. Agarwal, N. Srivastava, S. Lin, T. Sherwood and K. Banerjee, “A Thermally-Aware Performance Analysis of Vertically Integrated (3-D) Processor-Memory Hierarchy”, *Proc. Design Automation Conference*, 2006, pp. 991-996.
- [17] J. Meng, K. Kawakami and A. K. Coskun, “Optimizing Energy Efficiency of 3-D Multicore Systems with Stacked DRAM under Power and Thermal Constraints”, *Proc. Design Automation Conference*, 2012, pp. 648-655.
- [18] J. T. Pawlowski, “Hybrid Memory Cube: Breakthrough DRAM Performance with a Fundamentally Re-architected DRAM Subsystem”, *HOT Chips* 23, 2011, http://www.hotchips.org/wp-content/uploads/hc_archives/hc23/Hc23.18.3-memory-FPGA/Hc23.18.320-HybridCube-Pawlowski-Micron.pdf
- [19] K. Puttaswamy and G. H. Loh, “Thermal Analysis of a 3D Die-Stacked High-Performance Microprocessor”, *Proc. ACM Great Lakes Symposium on VLSI*, 2006, pp. 19-24.
- [20] R. Radojicic, “Roadmap for Design and EDA Infrastructure for 3D Products”, *Electronic Design Processes Workshop*, April 2012.
- [21] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan and D. Tarjan, “Temperature-Aware Microarchitecture”, *Proc. International Symposium on Computer Architecture*, 2003, pp. 2-13.
- [22] C. M. Tan and F. He, “3D Circuit Model for 3DIC Reliability Study”, *Proc. International Conference on Thermal, Mechanical and Multiphysics Simulation and Experiments in Micro-Electronics and Micro-Systems*, 2009, pp. 1-7.
- [23] K. N. Tu, “Reliability Challenges in 3D IC Packaging Technology”, *Microelectronics Reliability* 51(3) (2011), pp. 517-523.
- [24] M. H. Woods, “MOS VLSI Reliability and Yield Trends”, *Proc. of the IEEE* 74(12) (1986), pp. 1715-1729.
- [25] J. Xie, V. Narayanan and Y. Xie, “Mitigating Electromigration of Power Supply Networks Using Bidirectional Current Stress”, *Proc. ACM Great Lakes Symposium on VLSI*, 2012, pp. 299-302.
- [26] V. Zolotov, C. Visweswariah and J. Xiong, “Voltage Binning Under Process Variation”, *Proc. IEEE/ACM ICCAD*, 2009, pp. 425-432.
- [27] “Guidelines to Understanding Reliability Prediction.” <http://www.epsma.org>
- [28] “OpenCores.” <http://opencores.org>
- [29] “Synopsys Design Compiler User Guide.” <http://www.synopsys.com/Tools/Implementation/RTLsynthesis/DCUltra/pages/default.aspx>
- [30] “Synopsys PrimeTime User’s Manual.” <http://www.synopsys.com>
- [31] “Cadence SOC Encounter User Guide.” http://www.cadence.com/products/di/first_encounter/pages/default.aspx
- [32] “Hotspot User Guide.” <http://lava.cs.virginia.edu/HotSpot/index.htm>
- [33] “LP_Solve Manuals.” <http://lpsolve.sourceforge.net/5.5/>