System-level Process Variability Analysis and Mitigation for 3D MPSoCs*

Siddharth Garg
Carnegie Mellon University
Email: sgarg1@andrew.cmu.edu

Diana Marculescu
Carnegie Mellon University
Email: dianam@ece.cmu.edu

***Abstract*—While prior research has extensively evaluated the performance advantage of moving from a 2D to a 3D design style, the impact of process parameter variations on 3D designs has been largely ignored. In this paper, we attempt to bridge this gap by proposing a variability-aware design framework for fully-synchronous (FS) and multiple clock-domain (MCD) 3D systems. First, we develop analytical system-level models of the impact of process variations on the performance of FS 3D designs. The accuracy of the model is demonstrated by comparing against transistor-level Monte Carlo simulations in SPICE - we observe a maximum error of only 0.7% (average 0.31% error) in the mean of the maximum critical path delay distribution. Second, to mitigate the impact of process variations on 3D designs, we propose a variability-aware 3D integration strategy for MCD 3D systems that maximizes the probability of the design meeting specified system performance constraints. The proposed optimization strategy is shown to significantly outperform FS and MCD 3D implementations that are conventionally assembled - for example, the MCD designs assembled with the proposed integration strategy provide, on average, 44% and 16.33% higher absolute yield than the FS and conventional MCD designs respectively, at the 50% yield point of the conventional MCD designs.**

I. Introduction

Recently, major semiconductor companies have advocated a move toward three dimensional integrated circuit (3D IC) technologies to mitigate the growing wire delay concerns in deep sub-micron technologies [1]. While a number of techniques have been proposed for dense 3D integration, we concentrate primarily on stacked-die 3D technologies [2] which involve fabricating each active device layer on a different wafer and stacking the fabricated die on top of each other using pick-and-place techniques. Through-silicon-vias (TSV) are typically used to interconnect dies in different layers. Of specific interest in this paper is the case of application specific embedded systems, consisting of a network of processing elements (PEs) or on-chip memories, implemented using the described 3D die-stacking methodology.

While there has been significant prior research in the EDA community on tools for analyzing and optimizing the performance of 3D designs from a physical design perspective, the analysis and optimization of manufacturing process variations for 3D designs has not been addressed. Moreover, the impact of process parameter variation at the transistor- and gate-level for 2D systems has been extensively researched in the past. However, micro-architecture and system-level analysis and optimization of the impact process variations has only recently gained attention, driven by the need to address process variations as early in the design process as possible. To the best of our knowledge, this paper presents the first *analytical, system-level model* of the impact of process variations on the performance of 3D designs. Using this model, we show that process variations impact 3D designs differently as compared to an equivalent 2D design. To demonstrate the accuracy of the models, they are validated against SPICE based Monte Carlo simulations that model both within-layer and layer-to-layer critical paths along with the associated wire and TSV delays.

Since the proposed variability model indicates that FS 3D systems suffer a greater performance loss to process variations as compared to their FS 2D counterparts, we propose using a *variability adaptive, multiple clock-domain (MCD) 3D design style* as a possible solution to mitigate the performance loss. Moreover, we show that by using a **novel variability-aware die-level integration strategy for 3D MCD SoCs**, it is possible to further improve the system *performance yield* - i.e., the fraction of fabricated systems that meet a specified system-level performance constraint, such as execution latency or throughput - beyond that achievable by a 3D MCD system assembled using a conventional integration strategy.

The proposed framework can be used by system designers to compare the performance yield that they can expect, under the impact of process variations, for a range of possible 3D implementation strategies, namely: (1) a FS 3D design; (2) a 3D MCD design with conventional die-to-die integration; and (3) a 3D MCD design with the proposed yield-aware integration strategy. While we show that the performance yield improves from (1) to (3), the associated implementation costs may also increase due to the area overhead of local clock generation and additional test costs for the proposed integration strategy. Such information would be extremely useful for system designers to choose an implementation strategy that can maximize the performance yield within implementation cost budgets.

II. Related Work

Prior research in the area of system-level performance analysis and optimization of 3D MPSoCs has focused on temperature- and performance-aware floorplannning [5], design of 3D networks-on-chip (NoCs) [4], and power optimization [6]. However, none of these works consider the impact of process variations on 3D designs, which is the focus of this study.

The impact of manufacturing process variations on circuit power and performance characteristics, especially at the transistor/gate levels has received a lot of attention recently [8]. On the other hand, high-level modeling of the performance impact of process variations has only recently gained attention, and was pioneered by [3], where the widely used *generic critical path* model was presented. More recently, in [9], the authors study the impact of process variations on the system performance and show, theoretically and experimentally, that 2D MCD designs are more likely to meet performance constraints than their FS counterparts. While 3D MCD systems assembled using a conventional approach can be analyzed the same way as in [9], in this paper we propose a new methodology to utilize the unique flexibility of the post-fabrication 3D assembly process to obtain a variability-aware integration strategy for 3D MCD designs that outperforms conventional integration.

* This research was supported in part by NSF award CCR-0702451.

The only other work on the impact of process variations on 3D designs was published very recently by Ferri *et al.* [10]. As opposed to that work, we provide *analytical* models of the impact of parameter variations on 3D circuits that can be used by designers to make variability aware system-level design decisions early in the design process. In contrast, [10] uses simulation based models for parameter variations, which cannot be utilized for the same purpose. Furthermore, we propose and evaluate the use of 3D MCD designs with variability-aware integration as opposed to [10] that concentrates only on FS systems. Finally, as opposed to [10] that can only handle 3D systems with two device layers, the proposed integration strategy works for an arbitrary number of device layers and, more importantly, we do *not* assume that unpackaged bare die can be perfectly speed-binned, since it has been shown to be prohibitively expensive [13]. Instead, as will be explained later in the paper, we use quiescent leakage current test data to approximately *predict* the operating frequency of the bare die before assembly.

III. Paper Contributions

As compared to previous research, the work proposed herein makes the following novel contributions:

- We propose and experimentally verify an *analytical* system-level model for the impact of process variations on the performance of FS 3D designs.
- We propose and evaluate the use of variation-aware MCD 3D designs, in which each device layer lies in a separate clock domain, as a way to mitigate the loss in performance.
- We propose a novel, efficient die-level integration strategy to further increase the performance yield of 3D MCD systems *beyond* that achievable by a conventional integration strategy. The integration strategy works for an arbitrary number of device layers and does not assume perfect prior knowledge of frequency bins.

IV. System-level Variability Modeling

The authors of [3] have shown that the impact of process variations on FMAX (maximum clock speed) can be captured by two micro-architectural parameters: n_{cp}, the number of logic stages in a critical path in the circuit, and N_{cp}, the total number of critical paths in the circuit. If $T_{max,2D}$ is a random variable (RV) that represents the worst-case critical path delay of a 2D system under the impact of process variations, T_{D2D} is a RV that represents the variation in delay of a critical path due to the impact of D2D variations, T^i_{WID} is the RV that represents the variation in path delay for the i^{th} critical path in the circuit ($1 \leq i \leq N_{cp}$), and $T_{cp,nom}$ is the nominal delay of a critical path, we can write:

$$T_{max,2D} = T_{cp,nom} + T_{D2D} + \max_{i \in (1,N_{cp})} T^i_{WID} \quad (1)$$

Using this equation, the authors show that the probability density function (*pdf*) of maximum critical path delay, $f_{T_{max,2D}}(t)$ can be written as:

$$f_{T_{max,2D}}(t) = f_{T_{D2D}}(t') * \{N_{cp} F_{T_{WID}}(t')^{N_{cp}-1}\} f_{T_{WID}}(t') \quad (2)$$

where $*$ represents *convolution*, $t' = t - T_{cp,nom}$, $F_X(.)$ represents the cumulative distribution function (*cdf*) of R.V. X and $f_X(.)$ represents its *pdf*. Furthermore, if σ_{D2D} and σ_{WID} are the standard deviations of path delay due to D2D and WID variations respectively, we can write:

$$\sigma_{WID} = \sqrt{n_{cp}}\sigma_{WID,gate} \quad \sigma_{D2D} = n_{cp}\sigma_{D2D,gate} \quad (3)$$

where $\sigma_{WID,gate}$ ($\sigma_{D2D,gate}$) refers to the standard deviation of *gate* delay due to WID (D2D) variations.

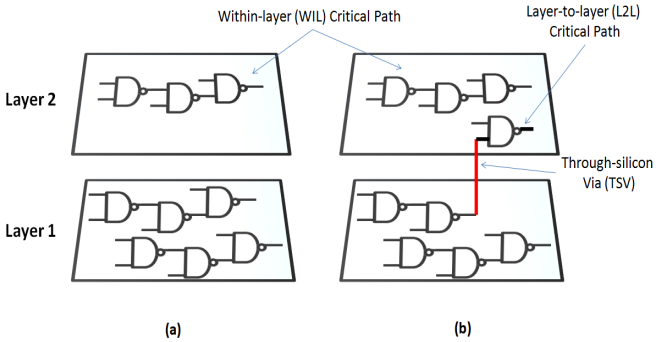

Fig. 1. (a) 3D design with only WIL critical paths. (b) 3D design with both WIL and L2L critical paths.

A. FS 3D Architectures

As opposed to 2D systems, there are two possible types of critical paths in a 3D system as shown in Figure 1 - **Within-Layer (WIL) paths** are fully contained within one of the device layers in the system while **Layer-to-Layer (L2L) paths** utilize TSVs to cross from one layer to another. We note that the number of L2L critical paths in a 3D system is likely to be smaller than the number of WIL paths since the available layer-to-layer TSV based routing resources are significantly fewer than the available within-layer routing resources, especially for stacked 3D designs in which each module (either a PE or embedded memory), and thereby all the critical paths within the module, lies completely within one of the device layers [2] . Therefore, in the proposed model, we start by assuming a 3D system with only WIL critical paths. We then introduce a simple approximation to account for any L2L critical paths in the system. As we will show in the results section, the approximation provides extremely accurate results even when as many as 50% of the critical paths in the system are L2L paths.

Without any loss of generality, we assume that the 3D system is a vertical stack of L dies ($L > 1$) and each layer i has N^i_{cp} ($1 \leq i \leq L$) critical paths. As in the previous case, we define T^i_{D2D} ($1 \leq i \leq L$) to be the RV corresponding to the variation in critical path delay due to D2D variations in layer i, and $T^{i,j}_{WID}$ ($1 \leq i \leq L, 1 \leq j \leq N^i_{cp}$) to be the RV corresponding to the variation in delay of critical path j in layer i due to WID variations. Since the entire system is driven by a single global clock, we can write the maximum (i.e., worst-case) critical path delay, $T_{max,3D}$ as:

$$T_{max,3D} = T_{cp,nom} + \max_{i \in [1,L]} (T^i_{D2D} + \max_{j \in [1,N^i_{cp}]} T^{ij}_{WID}) \quad (4)$$

Now, since the die for each layer in a 3D system comes from a different wafer [2], the D2D RVs for the device layers in a system can be assumed to be independent. Therefore, using Equation 4, we can write the *pdf* of the maximum critical path delay for a FS 3D system as:

$$\begin{aligned} f_{T_{max,3D}}(t) \quad = \quad & L[F_{T_{D2D}}(t') * (F_{T_{WID}}(t'))^{N'_{cp}}]^{L-1} \\ & \times [f_{T_{D2D}}(t') * N'_{cp} F_{T_{WID}}(t')^{N'_{cp}-1} f_{T_{WID}}(t')] \end{aligned} \quad (5)$$

where $\times$ represents *multiplication*, and the rest of the notation is consistent with Equation 2. To simplify the analytical form of the equation, we assumed that the number of critical paths in each layer of the system is the same, i.e., $N^i_{cp} = N^j_{cp} = N'_{cp}$ for every pair of layers i and j. However, an analytical solution, although a more complex one, is as easily derived for the general case where the number of critical paths in each layer is different. The important thing to note in Equation 5 is that along with a dependence on the

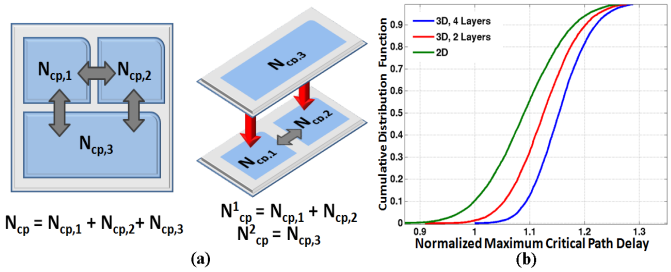

Fig. 2. (a) 2D and 3D implementations of an MPSoC with three PEs. (b) *cdf* of normalized maximum critical path delay for FS 2D and 3D systems.

number of critical paths per layer (as in the 2D case), the maximum critical path delay *pdf* also has a strong *exponential* dependence on the number of layers, L, in the system.

L2L Critical Paths The model proposed above is, at least theoretically, perfectly accurate for 3D systems without any L2L critical paths. Now, to model the case in which there are N_{cp}^{L2L} layer-to-layer critical paths in the system, we *assume* that each L2L critical path impacts *FMAX* in the same way as a WIL path. Therefore, to account for L2L paths, the number of critical paths in layer i ($1 \leq i \leq L$), N_{cp}^{i}, from Equation 4 is modified to a new value, $N_{cp,approx}^{i}$ as:

$$N_{cp,approx}^{i} = N_{cp}^{i} + \frac{N_{cp}^{L2L}}{L} \tag{6}$$

In the experimental results section we demonstrate that this approximation works extremely well in practice, even for the case of 3D systems with a large fraction of L2L paths.

B. FS 2D Vs. FS 3D Systems

One application of the proposed model is to compare the impact of process variations on the clock frequency of FS 2D and stacked 3D implementation options for an MPSoC. Figure 2(a), for example, shows a 2D implementation and a two layer 3D implementation of a three processor system. As it can be seen, the number of critical paths in each layers can simply be computed by summing up the contribution from all the PEs in that layer. Now, using Equations 2, 5 and 6, we can compare various implementation options from a variability perspective. As an example, we use the proposed model to compare three implementations for a system with 10,000 critical paths: a FS 2D design, a FS 3D design with two layers and a FS 3D design with four layers. The *cdfs* of maximum critical path delay for the three systems are shown in Figure 2(b) and we observe that, while the mean of maximum critical path delay for a 2D design is only 8% worse than the nominal, it is 14% worse for a 3D design with two layers and 17% worse for a 3D design with four layers. In fact, we prove theoretically that, under certain assumptions, 3D systems *always* lose more performance to process variations than their 2D counterparts.

Lemma: The probability that the maximum critical path delay of a 2D system with $N_{cp,2D}$ critical paths is less than a value τ is always greater than the probability for a 3D system with L layers and $N_{cp,3D}^{i}$ critical paths in layer i, assuming that $N_{cp,2D} = \sum_{i=1}^{L} N_{cp,3D}^{i}$ and that the magnitude of D2D and WID process variation for the 2D and 3D systems is the same.

Proof: The proof is based on the theorem presented in [7] on the comparing the maximum of two Gaussian Random Vectors. While a detailed exposition is excluded due to space constraints, we note that comparing Equation 1 and Equation 4 using this theorem completes the desired proof. □

V. Variability Aware MCD 3D Systems

The primary reason for the susceptibility of FS 3D designs to process variations is that the clock speed of the design is limited by the slowest die layer in the stack, i.e., the layer that is the worst hit by D2D variations, even if the other layers can support higher clock speeds. By allowing each clock domain to run independently at its optimal frequency, MCD architectures have previously been shown to provide increased variability tolerance [9] in the context of 2D systems. As we show in the experimental results, the same argument holds for 3D MCD designs as well, i.e., we see an increase in variability tolerance while moving from a 3D FS to a 3D MCD design. **However, the 3D die-to-die assembly process offers an additional degree of flexibility that is not available in the 2D case** - i.e., the ability to decide *post-fabrication* which die should be combined together so that the maximum number of assembled 3D systems meet the desired performance specification. We formulate computing the optimal post-fabrication variability-aware integration strategy as an integer program which we then solve using a novel linearization and relaxation approach. Before discussing the proposed solution in greater detail, we briefly overview the implementation details of 3D MCD systems assumed in the paper. Specifically we assume that each layer in the system is implemented as a separate clock-domain with its own local clock generator equipped with fine-grained frequency control, for example, a digital PLL or voltage-controlled ring oscillator. Since the clock-domains are asynchronous with respect to each other, we assume that communication between clock-domains occurs via point-to-point mixed-clock FIFOs [11] between communicating PEs. Finally, we assume that there exists an on-chip module in each die-layer to sense the impact of process variations on the maximum critical path delay of that layer and correspondingly set the frequency of its local clock generator at its maximum possible value [12].

A. Variability-aware Die-level Integration

We begin by assuming that for a 3D design with L device layers, we obtain N_{total} bare dies for each layer in the design after fabrication. The bare dies are then typically functionally tested, assembled and packaged to create N_{total} 3D systems. Now, due to process variations, each die actually will have a different maximum operating frequency (*FMAX*); however, in a **conventional integration** scheme the bare dies are assembled without any knowledge of the $FMAX$ of each bare die. The **variability-aware 3D integration scheme** takes advantage of the fact that if the $FMAX$ of each bare die can be estimated before assembly, the assembly can be performed to maximize the number of assembled 3D systems that meet a certain performance specification.

We first assume that there exists an *oracle* that correctly bins each fabricated bare die into one of F frequency bins and later relax this unrealistic assumption. Once each die has been allocated to its respective bin, we can represent the number of dies in layer i that are allocated to frequency bin j as N_{ij}, ($1 \leq i \leq L, 1 \leq j \leq F$). Clearly, the total number of 3D systems that can be fabricated, N_{total} can be written as $N_{total} = \sum_j N_{ij}, \forall i \in [1, L]$, which is the sum of the number of dies in each frequency bin for any given layer. Given this information, the proposed variability-aware integration strategy can be conceptually described as follows:

- Die-level integration occurs in L steps, where L is the number of active device layers. In each step, a new layer is added to the system using pick-and-place techniques, starting from the bottom to the top.

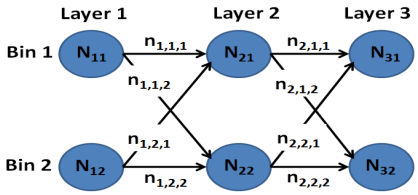

Fig. 3. Graphical depiction of a 3D die-level integration strategy.

- At step i ($2 \leq i \leq L$) , $n_{i,j,k}$ 3D systems with their top-most layer (i.e., layer $i-1$) lying in frequency bin j are selected and dies from layer i lying in frequency bin k are stacked on top of them, thereby creating $n_{i,j,k}$ systems with i layers. This process is repeated for all $j,k : 1 \leq j,k \leq F$.

This approach is graphically depicted in Figure 3, where each vertex in the graph represents the number of dies in a given bin for a given layer.

We can represent each assembled 3D system with an L dimensional vector $\underline{f} = (f_1, f_2 \ldots f_L)$, where f_i is the frequency bin of layer i in the system. We assume that there exists a function $P(\underline{f})$ that maps the frequency bin allocations of each active die layer to a real valued measure of system performance (for example, throughput or worst-case execution latency), and that the performance constraint that the design is expected to meet is represented by P_{constr}. Therefore, the goal of the optimal integration strategy is to choose the variables $n_{i,j,k}$ to maximize the number of assembled 3D systems that meet the performance constraint P_{constr}.

Based on the integration strategy in Figure 3, the expected number of 3D systems obtained after integration with a given frequency vector $\underline{f}$, $a_{\underline{f}}$ can be written as:

$$a_{\underline{f}} = n_{1,f_1,f_2} \prod_{i=2}^{L-1} \frac{n_{i,f_i,f_{i+1}}}{\sum_{j=1}^{F} n_{i,f_i,f_j}} = n_{1,f_1,f_2} \prod_{i=2}^{L-1} \frac{n_{i,f_i,f_{i+1}}}{N_{if_i}} \quad (7)$$

We can now write an optimization problem that tries to maximize the number of systems with performance greater than the constraint P_{constr} by appropriately selecting the values of the $n_{i,j,k}$ variables as follows:

$$\max_{n} \sum_{\forall \underline{f}} w_{\underline{f}} n_{1,f_1,f_2} \prod_{i=2}^{L-1} \frac{n_{i,f_i,f_{i+1}}}{N_{if_i}} \quad (8)$$

where:

$$w_{\underline{f}} = \begin{cases} 1 & \text{if } P(\underline{f}) \geq P_{constr} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

$$\sum_{1 \leq j \leq F} n_{i,j,k} \leq N_{(i+1)k} \quad \forall i \in [1, L-1], \forall k \in [1, F] \quad (10)$$

$$\sum_{1 \leq k \leq F} n_{i,j,k} \leq N_{ij} \quad \forall i \in [1, L-1], \forall j \in [1, F] \quad (11)$$

$$n_{i,j,k} \in \mathbb{Z} \quad (12)$$

Relaxation, i.e., removing the constraints for the variables to lie in the set of integers, is a commonly used first step towards solving Integer Programming problems efficiently. Unfortunately, in this case, relaxing the problem (removing the constraint in Equation 12) is not useful since the objective function (Equation 8) is the maximization of a convex *posynomial* function, and therefore, does not admit an efficient solution [15]. However, we observe that the $n_{i,j,k}$ variables can be expressed as linear functions of the $a_{\underline{f}}$ variables, i.e.,:

$$n_{i,j,k} = \sum_{\forall \underline{f} : f_i = j, f_{i+1} = k} a_{\underline{f}} \quad (13)$$

Equation 8 can now simply be re-written as:

$$\max_{a,n} \sum_{\forall \underline{f}} w_{\underline{f}} a_{\underline{f}} \quad (14)$$

such that:

$$a_{\underline{f}} \in \mathbb{R}, \forall \underline{f} \quad (15)$$

Based on the observation in Equation 13, we now have a standard mixed Integer Linear Programming (mILP) problem with Equation 14 as an objective function and Equations 10, 11, 12, 13 and 15 as constraints. The mILP problem can now be *relaxed* to yield a standard Linear Programming (LP) problem that can be efficiently solved. However, the solution obtained from the LP, n^{LP}, needs to be converted back to an integral solution, n^* without violating any other constraints. This can be done using a simple floor operation:

$$n^*_{i,j,k} = \lfloor n^{LP}_{i,j,k} \rfloor \quad \forall i,j,k \quad (16)$$

Note that because we relaxed the original ILP to an LP problem and took the *floor* of the resulting solution to obtain the final integer solution, we cannot guarantee global optimality. However, we are able to derive a guaranteed upper bound on the difference between the optimal yield obtained from the mILP problem, Y^{mILP}, and the yield obtained from the proposed relaxation/flooring method Y^*.

Lemma: The value of $Y^{mILP} - Y^*$ can be no greater than $\frac{F^2}{N_{total}}$.

Proof: If Y^{LP} denotes the optimal (unachievable) yield obtained from the LP relaxation of the mILP, we know that $Y^{LP} \geq Y^{mILP} \geq Y^*$. Therefore we can write $Y^{mILP} - Y^* \leq Y^{LP} - Y^*$. Moreover, using some algebraic manipulation and optimization, the maximum value of $Y^{LP} - Y^*$ can be shown to be bounded by F^2/N_{total}, and therefore, $Y^{mILP} - Y^* \leq F^2/N_{total}$. □

The F^2/N_{total} bound proven above is extremely tight for realistic scenarios - for example, for F=8 frequency bins and $N_{total} = 10,000$ manufactured systems, the maximum possible yield loss between the proposed and optimal solution is 0.064%.

B. Frequency Bin Prediction

Having described the optimization procedure assuming an *oracle* that can correctly predict the frequency bin of each bare die before 3D assembly (as assumed by [10]), we now consider a practical scenario in which such information may not be readily available. As mentioned before, at-speed testing of bare die before 3D assembly can be prohibitively expensive [13]. On the other hand, burn-in testing and I_{DDQ} or quiescent leakage current tests are routinely performed on bare die and are an integral part of the Known Good Die test methodology [14].

Importantly, since there exists a strong correlation between the variability in leakage power dissipation of a die and the variability its maximum frequency [16], it is possible to use leakage measurements from the I_{DDQ} tests to predict the frequency bin of a bare die. Specifically, we assume that there exists prior data, either from previous fabrication runs or from statistical circuit/gate level simulation, in the form of frequency bin and leakage measurement for M instances of the design; i.e., 2-tuples of the form $(f^{train}_m, I^{train}_m)$ for $1 \leq m \leq M$, where f^{train}_m and I^{train}_m are the frequency bin and leakage power of the m^{th} training sample respectively. Given a new bare die with measured leakage power dissipation I^{test}, we predict its frequency bin, f^{test} using a simple one nearest neighbor search, i.e.,:

$$f^{test} = f^{train}_{m^*} \quad (17)$$

$$m^* = \min_m |(I^{train}_m - I^{test})| \quad (18)$$

Once the bare die have been binned using the proposed technique, they are assembled using the optimization strategy described in the previous section, *optimistically assuming that the predicted frequency bin values are correct*. After assembly and packaging, the on-chip speed testing modules set the frequency of each layer in every 3D

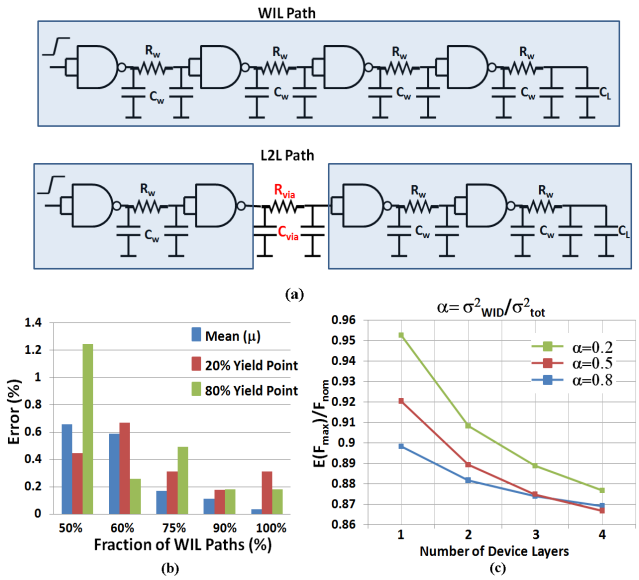

Fig. 4. (a) SPICE modeling of WIL and L2L critical paths. (b) % Error of analytical model with respect to MC SPICE simulations (c) Predicted impact of number of device layers on ratio of mean *FMAX* to nominal frequency

system to the maximum frequency that the die in that layer can safely run at.

VI. Experimental Results

A. Fully Synchronous 3D Architectures

We begin by validating the accuracy of the analytical models for FS systems developed in Section 4 against SPICE based Monte Carlo simulations. To be consistent with the methodology described in [3], each critical path in the SPICE netlist is modeled as a chain of two-input NAND gates in a *90 nm* PTM technology. Wire and via delay are inserted using a standard π model as shown in Figure 4(a). The RC parameters associated with the wire models are computed using the average dimensions for Metal 2 wires reported in [21], while vias are assumed to be $1.2\mu m \times 1.2\mu m$, with a $2.4\mu m$ pitch and $20\mu m$ length as reported in [22]. Finally, variations in process parameters are modeled by introducing both D2D and random WID variations in gate length, each with a σ of 5% of the nominal value. In Figure 4(b), we plot the error between MC SPICE simulations and the proposed analytical model in the mean, 20% yield point and 80% yield point of maximum critical path delay obtained for a two layer design with 200 critical paths. To test the robustness of the proposed model, the fraction of L2L critical paths in the design is varied from 0% to 50%. As it can be seen, even for the case in which 50% of the paths are L2L, the error in the mean critical path delay is only 0.7%, and drops to 0.1% for the design with 10% L2L paths. Similarly, the average errors in the 20% yield point and 80% yield point are only 0.4% and 0.5% respectively.

To investigate the impact of increasing number of device layers, we conducted an experiment in which we varied the number of layers in a 3D system from 1 to 4 and swept the variance of the WID delay distribution from 20% to 80% of the total delay variance, for both wafer-level and die-level integration. N_{cp} was assumed to be $10,000$ for the entire design. The results are graphed in Figure 4(c), where the *y-axis* represents the mean of the *FMAX* distribution normalized to the nominal frequency in the absence of variability. From the plot, we can see that, as expected, the mean FMAX decreases *significantly* with increasing number of layers, and that the decrease is more pronounced when D2D variations are a large contributor to total variability.

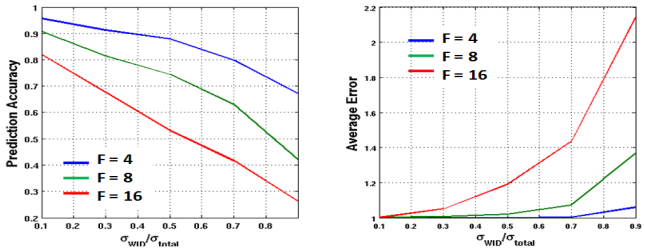

Fig. 5. Accuracy of the proposed frequency-bin prediction technique for $F = \{4, 8, 16\}$ (a) The fraction of dies correctly binned by the proposed technique. (b) The average number of bins by which the prediction technique is off, given that it mispredicts.

B. 3D MCD Architectures

We now investigate the performance of the proposed multiple-clock domain 3D architectures compared to FS 3D designs on a set of embedded system benchmarks. We experimented with the two benchmarks from the E3S benchmark suite [17] - the *telecom* and *consumer* benchmarks - and the *software-defined radio* (SDR) benchmark used in [18]. For the SDR benchmark, each task is separately mapped on to an separate Hitachi SuperH core, resulting in a 12 PE system that is implemented in four device layers, with cycle counts obtained using the Sunflower tool suite [20]. For the E3S benchmarks, we assumed a 12 (*consumer*) and 28 (*telecom*) PE design, again with four layers and an equal number of PEs per layer. Each PE is assumed to be an AMD ElanSC520 embedded processor, and the cycle counts and communication volumes for the tasks are taken from the pre-characterized data for the AMD ElanSC520 processor included with the E3S suite.

For all the experiments in this section, we assume a total standard deviation in gate delay to be 10% of its nominal value, and unless otherwise mentioned, assume that D2D and WID variations contribute equally to the total gate delay variations. Furthermore, in the absence of pre-characterized critical path data for the PEs, we assumed the number of logic stages, $n_{cp} = 9$, as suggested by [19], and the number of critical paths, $N_{cp} = 10,000$, as suggested by [3]. Finally, we use the models proposed by [18] to account for the communication latency of the point-to-point inter-layer mixed-clock FIFOs.

We begin with results for the frequency-bin prediction algorithm presented in Section V-B. Since we do not have access to the gate-level net lists of the processors used, we generate training samples for frequency bin prediction using 1,000 runs of Monte-Carlo (MC) simulation on a synthetically generated gate-level net list that consists of the same number of critical paths and logic stages as each die in the design, and record the leakage and frequency information for each MC sample. In Figure 5(a), we plot the accuracy of the prediction algorithm, i.e., fraction of correctly binned dies, as a function of the number of frequency bins F, and as a function of the contribution of WID variations to the total gate delay variation. As expected, the accuracy decreases as the WID variation increases (correlation between leakage and frequency decreases) and as F increases. In Figure 5(b), we plot the average number of bins by which the prediction differs from its correct value when a mis-prediction occurs. We can see that even though the frequency-bin prediction accuracy can be as low as 52% for $F = 16$, i.e., almost half the dies are binned incorrectly, the mis-predicted frequency bins are, on average, only 1.2 bins away from their correct assignments.

Under these assumptions, we studied the performance (we use *worst-case execution time*, or *latency*, as a performance measure) of the three benchmarks for four 3D designs: (1) **MCD-PER**, A 3D MCD architecture assembled using the proposed variability-aware

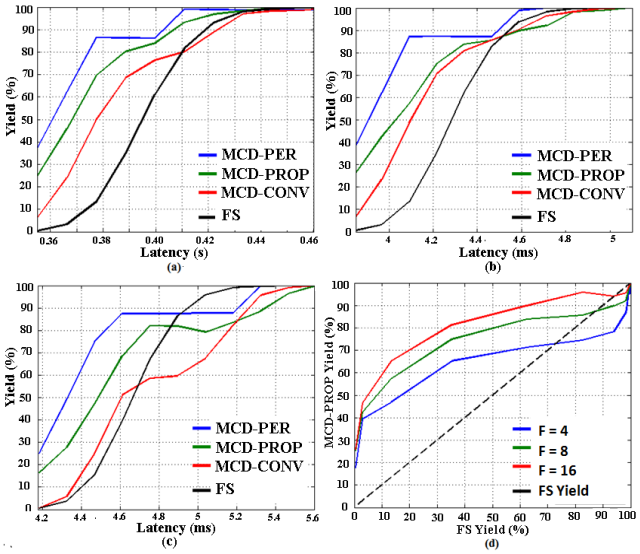

Fig. 6. Yield as a function of latency constraint for four different implementation choices for the *telecom*, *consumer* and SDR benchmarks (F=8 in each case). (d) Yield of the MCD-PROP architecture for $F = 4, 8, 16$ as a function of the FS yield for the *telecom* benchmark.

die-level integration strategy with *perfect knowledge* of the frequency of each bare die (though this is an unrealistic assumption, it serves as an upper bound on the quality of the solution); (2) **MCD-PROP**, A 3D MCD architecture assembled using the *proposed variability-aware die-level integration strategy* in which the prediction algorithm from Section V-B was used to determine the frequency of each die; (3) **MCD-CONV**, A 3D MCD architecture using *conventional die-level assembly* with no prior knowledge of frequencies of the bare dies before assembly; (4) **FS**, A fully-synchronous 3D design assembled using conventional integration.

For each MCD architecture, we considered three cases - $F = 4$, $F = 8$ and $F = 16$. Finally, so as to not unfairly skew the results in favor of the MCD designs, the fully-synchronous design (FS) is allowed to choose from a *continuous* range of frequency values, and therefore, represents the *upper limit* of performance yield that is achievable by any FS 3D design.

In Figure 6, we plot the results from our experiments. Parts (a), (b) and (c) of Figure 6 clearly indicate that the MCD-PROP architecture is able to significantly outperform MCD-CONV when the performance constraints are stringent - for example, at a latency constraint that provides 50% yield for MCD-CONV, the MCD-PROP yield is 21%, 8% and 18% higher for the *consumer*, *telecom* and SDR benchmarks respectively. At the same latency constraint MCD-PROP provides 58%, 43% and 30% higher yield than the FS design for the same three benchmarks. We note that the curves for MCD-PER demonstrate that the potential for further improving the yield of MCD-PROP is substantial if more precise die frequency information were available before 3D assembly.

Finally, we study the impact of varying the number of frequency bins, F, on the performance yield of the proposed method by plotting in Figure 6(d) the MCD-PROP yield as a function of the FS yield with varying number of frequency bins, i.e., $F = \{4, 8, 16\}$. From the plot, it is clear that though the MCD-PROP yield increases with F, though there seems to be a trend of diminishing returns.

The results presented in this paper can be used by system-level designers to determine, early in the design cycle, the architecture, integration strategy and the number of frequency bins that provide acceptable performance yield, while minimizing design complexity.

VII. Conclusions and Future Work

In this paper, we presented an analytical, system-level model for the impact of process variation on the maximum frequency of FS 3D designs. The model was experimentally validated against circuit-level Monte Carlo simulations in SPICE and predicts that FS 3D designs suffer larger performance degradation due to process variations than their 2D counterparts. Furthermore, the extent of performance degradation depends on the number of layers in the design and the 3D integration strategy used. Next, we proposed a variability-aware multiple clock-domain architecture for 3D MPSoCs, and developed a novel variability-aware integration strategy that maximizes the number of assembled 3D systems that satisfy a specified performance constraint. Our results indicate that the proposed technique can provide significant performance yield improvement over 3D MCD systems assembled using conventional system integration strategies, especially for stringent performance constraints.

As future work, we plan to better model L2L paths and to include the impact of leakage variations on 3D designs.

References

[1] R. Ho et al., "The future of wires," in Proc. of the IEEE, Apr. 2001.
[2] Eric Beyne, "3D integration," in *Workshop on Heterogeneous Systems Integration*, DATE 2008.
[3] K. Bowman et al., "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," in *IEEE JSSC*, Feb. 2002.
[4] F. Li et al., "Design and management of 3D chip multiprocessors using network-in-memory," in *Proceedings of ISCA*, Jun. 2006.
[5] J. Cong et al., "A thermal-driven floorplanning algorithm for 3D ICs," in *Proceedings of ICCAD*, Nov. 2004.
[6] J. Joyner and J. Meindl, "Opportunities for reduced power dissipation using three-dimensional integration," in *Proceedings of the ITC*, 2002.
[7] R. Adler., "An Introduction to Continuity Extrema and Related Topics for General Gaussian Processes," in *Institute of Mathematical Statistics Lecture Notes*, 1990.
[8] X. Li et al., "Statistical performance modeling and optimization," in *Foundations and Trends in Electronic Design Automation*, Sep. 2006.
[9] D. Marculescu and S. Garg, "System-level process-driven variability analysis for single and multiple voltage-frequency island systems," in *Proceedings of ICCAD*, Nov. 2006.
[10] C. Ferri et al., "Strategies for improving the parametric yield and profits of 3D ICs," in *Proceedings of ICCAD*, Nov. 2007.
[11] T. Chelcea and S. Nowick, "Robust interfaces for mixed-timing systems," in *IEEE TVLSI*, Aug. 2004.
[12] A. Raychowdhury et al., "A novel on-chip delay measurement hardware for efficient speed-binning," in *Proceedings of IOLTS*, 2005.
[13] Y. Deng and W. Maly, "2.5-dimensional VLSI system integration," in *IEEE Trans. Very Large Scale Integr. Syst.*, June 2005.
[14] J. Hagge and R. Wagner, "High-yield assembly of multichip modules through known-good IC's and effective test strategies," in *Proceedings of the IEEE*, Dec. 1992.
[15] S. Boyd et al., "A tutorial on geometric programming," in *Optimization and Engineering*, Apr. 2007.
[16] A. Keshavarzi et al., "Multiple-parameter CMOS IC testing with increased sensitivity for i/sub ddq/," in *IEEE TVLSI*, Oct. 2003.
[17] http://www.ece.nonhwestem.edu/-dickrp/e3s.
[18] K. Niyogi and D. Marculescu, "System level power and performance modeling of GALS point-to-point communication interfaces," in *Proceedings of ISLPED*, Aug. 2005.
[19] A. Hartstein and T. Puzak, "The optimum pipeline depth for a microprocessor," in *Proceedings of ISCA*, June 2002.
[20] http://www.sunflowersim.org
[21] http://www.eas.asu.edu/ ptm
[22] S. Gupta et al., "Techniques for Producing 3D ICs with High-Density Interconnect," in *Proceedings of VMIC*, Sept. 2004.