VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects

Smruti R. Sarangi, Brian Greskamp, *Student Member, IEEE*, Radu Teodorescu, *Student Member, IEEE*, Jun Nakano, *Member, IEEE*, Abhishek Tiwari, *Student Member, IEEE*, and Josep Torrellas, *Fellow, IEEE*

***Abstract*—Within-die parameter variation poses a major challenge to high-performance microprocessor design, negatively impacting a processor's frequency and leakage power. Addressing this problem, this paper proposes a microarchitecture-aware model for process variation—including both random and systematic effects. The model is specified using a small number of highly intuitive parameters. Using the variation model, this paper also proposes a framework to model timing errors caused by parameter variation. The model yields the failure rate of microarchitectural blocks as a function of clock frequency and the amount of variation. With the combination of the variation model and the error model, we have *VARIUS*, a comprehensive model that is capable of producing detailed statistics of timing errors as a function of different process parameters and operating conditions. We propose possible applications of VARIUS to microarchitectural research.**

I. Introduction

As high-performance processors move into 32-nm technologies and below, designers face the major roadblock of parameter variation—the deviation of process, voltage, and temperature (PVT [1]) values from nominal specifications. Variation makes designing processors harder because they have to work under a range of parameter values.

Variation is induced by several fundamental effects. Process variation is caused by the inability to precisely control the fabrication process at small-feature technologies. It is a combination of systematic effects [2]–[4] (e.g., lithographic lens aberrations) and random effects [5] (e.g., dopant density fluctuations). Voltage variations can be caused by IR drops in the supply distribution network or by $L\,dI/dt$ noise under changing load. Temperature variation is caused by spatially and temporally varying factors. All of these variations are becoming more severe and harder to tolerate as technology scales to minute feature sizes.

Two key process parameters subject to variation are the transistor threshold voltage V_{th} and the effective length L_{eff}. V_{th} is especially important because its variation has a substantial impact on two major properties of the processor, namely the frequency it attains and the leakage power it dissipates. Moreover, V_{th} is also a strong function of temperature, which increases its variability [6].

One of the most harmful effects of variation is that some sections of the chip are slower than others—either because their transistors are intrinsically slower or because high temperature or low supply voltage renders them so. As a result, circuits in these sections may be unable to propagate signals fast enough and may suffer timing errors. To avoid these errors, designers in upcoming technology generations may slow down the frequency of the processor or create overly conservative designs. It has been suggested that parameter variation may wipe out most of the potential gains provided by one technology generation [7].

An important first step to redress this trend is to understand how parameter variation affects timing errors in high-performance processors. Based on this, we could devise techniques to cope with the problem—hopefully recouping the gains offered by every technology generation. To address these problems, this paper proposes *VARIUS*, a novel microarchitecture-aware model for process variation and for variation-induced timing errors. VARIUS can be used by microarchitects in a variety of studies.

The contribution of this paper is two-fold.

A model for process variation: We propose a novel model for process variation. Its component for systematic variation uses a multivariate normal distribution with a spherical correlation structure. This matches empirical data obtained by Friedberg *et al.* [2]. The model has only three parameters—all highly intuitive—and is easy to use. Moreover, we also model temperature variation.

A model for timing errors due to parameter variation: We propose a novel, comprehensive timing error model for microarchitectural structures in dies that suffer from parameter variation. This model is called *VATS*. It takes into account process parameters, the floorplan, and operating conditions like temperature. We model the error rate in logic structures, SRAM structures, and combinations of both, and consider both systematic and random variation. Moreover, our model matches empirical data and can be simulated at high speed.

This paper is organized as follows. Section II introduces background material and provides mathematical preliminaries; Section III presents the process variation model; Section IV presents the model of timing errors for logic and SRAM under parameter variation; Section V shows a model validation and evaluation; Section VI presents related work; and Section VII concludes the paper.

Manuscript received June 15, 2007; revised September 24, 2007. This work was supported in part by the National Science Foundation under Grants EIA-0072102, EIA-0103610, CHE-0121357, and CCR-0325603, in part by the Defense Advanced Research Projects Agency under Grant NBCH30390004, in part by the DOE under Grant B347886, and in part by gifts from IBM and Intel.

S. R. Sarangi is with Synopsis Research, Bangalore, India (e-mail: sarangi@cs.uiuc.edu).

B. Greskamp, R. Teodorescu, A. Tiwari, and J. Torrellas are with the Department of Computer Science, University of Illinois, Urbana, IL 61801 USA (e-mail: atiwari@cs.uiuc.edu; torrellas@cs.uiuc.edu).

J. Nakano is with IBM, Japan.

Digital Object Identifier 10.1109/TSM.2007.913186

II. Background

In characterizing CMOS delay under process variation, two important transistor parameters are the effective channel length L_{eff} and the threshold voltage V_{th}, both of which are affected by variation. This section presents equations that show how these two parameters determine transistor and gate speeds. It also introduces some aspects of probability theory that will feature in the following sections.

A. Transistor Equations

The equations for transistor drain current I_d using the traditional Shockley model are as follows:

$$I_d = \begin{cases} 0, & \text{if } V_{gs} \leq V_{\text{th}} \\ \beta\left(V_{gs} - V_{\text{th}} - \frac{V_{ds}}{2}\right)V_{ds}, & \text{if } V_{ds} < V_{gs} - V_{\text{th}} \\ \beta\frac{(V_{gs}-V_{\text{th}})^2}{2}, & \text{if } V_{ds} \geq V_{gs} - V_{\text{th}} \end{cases}. \quad (1)$$

Here, $\beta = \mu C_{\text{ox}} W/L_{\text{eff}}$, where μ is the mobility and C_{ox} is the oxide capacitance. In deep submicron technologies, these relationships are superseded by the alpha power law [8]

$$I_d = \begin{cases} 0, & \text{if } V_{gs} \leq V_{\text{th}} \\ \frac{W}{L_{\text{eff}}}\frac{P_c}{P_v}(V_{gs} - V_{\text{th}})^{\alpha/2}V_{ds}, & \text{if } V_{ds} < V_{d0} \\ \frac{W}{L_{\text{eff}}}P_c(V_{gs} - V_{\text{th}})^{\alpha}, & \text{if } V_{ds} \geq V_{d0} \end{cases}. \quad (2)$$

In this equation, P_c and P_v are constants and V_{d0} is given by

$$V_{d0} = P_v(V_{gs} - V_{\text{th}})^{\alpha/2}.$$

The time required to switch a logic output follows from (2). For most of the switching time, the driving transistor is in the saturation region [the last case of (2)]. The driver is trying to pull an output capacitance to a switching threshold (expressed as a fraction of V_{dd}) so that the switching time is

$$T_g \propto \frac{L_{\text{eff}}V}{\mu(V - V_{\text{th}})^{\alpha}} \quad (3)$$

where α is typically 1.3 and μ is the mobility of carriers which, as a function of temperature (T), is $\mu(T) \propto T^{-1.5}$. As V_{th} decreases, $V - V_{\text{th}}$ increases and a gate becomes faster. As T increases, V_{th} decreases and, as a result, $V - V_{\text{th}}(T)$ increases. However, $\mu(T)$ decreases [9]. The second factor dominates and, with higher T, a gate becomes slower. The Shockley model occurs as a special case of the alpha-power model with $\alpha = 2$.

B. Mathematical Preliminaries

Single Variable Taylor Expansion: The Taylor expansion of a function $f(x)$ about x_0 is

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \quad (4)$$

where $f^{(n)}(x_0)$ is the n^{th} derivative of f at x_0.

μ, σ of a Function of Normal Random Variables: Consider a function $Y = f(X_1, X_2, \ldots, X_n)$ of normal random variables $X_1, \ldots, X_n$ with mean $\mu_1, \ldots, \mu_n$ and standard deviation $\sigma_1, \ldots, \sigma_n$. Multivariate Taylor series expansion [10] yields the mean and standard deviation of Y as follows:

$$\mu_Y = f(\mu_1 \ldots \mu_n) + \sum_{i=1}^{n}\left[\left.\frac{\partial^2 f(x_1 \ldots x_n)}{\partial(x_i)^2}\right|_{\mu_i} \times \frac{\sigma_i^2}{2}\right]$$

$$\sigma_Y^2 = \sum_{i=1}^{n}\left[\left(\left.\frac{\partial f(x_1 \ldots x_n)}{\partial(x_i)}\right|_{\mu_i}\right)^2 \times \sigma_i^2\right]. \quad (5)$$

Maximum of n Independent Normal Random Variables: Given n independent and identically distributed normal random variables, each with cumulative distribution function (cdf) F, we are interested in the distribution of the largest variable. Define

$$\gamma = 0.577216$$
$$b = F^{-1}\left(1 - \frac{1}{n}\right)$$
$$a = F^{-1}\left(1 - \frac{1}{n\,e}\right) - b.$$

Extreme value theory [11] shows that the value of the largest variable follows a Gumbel distribution, whose mean and standard deviation are

$$\mu \approx b + a\gamma \quad \sigma \approx \frac{\pi a}{\sqrt{6}}. \quad (6)$$

III. Process Variation Model

Process variation has die-to-die (D2D) and within-die (WID) components, with the WID component further subdividing into *random* and *systematic* components. Lithographic aberrations introduce systematic variations, while dopant fluctuations and line edge roughness generate random variations. By definition, systematic variations exhibit spatial correlation and, therefore, nearby transistors share similar systematic parameter values [2]–[4]. In contrast, random variation has no spatial correlation and, therefore, a transistor's randomly varying parameters differ from those of its immediate neighbors. Most generally, variation in any parameter P can be represented as follows:

$$\Delta P = \Delta P_{\text{D2D}} + \Delta P_{\text{WID}} = \Delta P_{\text{D2D}} + \Delta P_{\text{rand}} + \Delta P_{\text{sys}}.$$

In this paper, we focus on WID variation. For simplicity, we model the random and systematic components of WID variation as normal distributions [12]. We treat random and systematic variation separately, since they arise from different physical phenomena. As described in [12], we assume that their effects are additive. If required, D2D variation can be modeled as an independent additive variable by adding a chip-wide offset to the parameters of every transistor on the die. This approach does

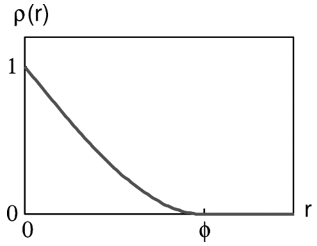

Fig. 1. Correlation of systematic parameters at two points as a function of distance r between them.

sacrifice some fidelity since, in reality, WID and D2D variations may not be statistically independent.

A. Systematic Variation

We model systematic variation using a multivariate normal distribution [10] with a spherical spatial correlation structure [13]. For that, we divide a chip into n small, equally sized rectangular sections. Each section has a single value of the systematic component of V_{th} (and L_{eff}) that is distributed normally with zero mean and standard deviation σ_{sys}, where the latter is different for V_{th} and L_{eff}. This is a general approach that has been used elsewhere [12]. For simplicity, we assume that the spatial correlation is homogeneous (position-independent) and isotropic (not depending on the direction). This means that, given two points $\vec{x}$ and $\vec{y}$ on the chip, the correlation of their systematic variation values depends only on the distance between $\vec{x}$ and $\vec{y}$. These assumptions have been used by other authors such a Xiong *et al.* [14].

Assuming position independence and isotropy, the correlation function of a systematically varying parameter P is

$$\text{corr}(P_{\vec{x}}, P_{\vec{y}}) = \rho(r) \quad r = |\vec{x} - \vec{y}|.$$

By definition, $\rho(0) = 1$ (i.e., totally correlated). Intuitively, $\rho(\infty) = 0$ (i.e., totally uncorrelated) if we only consider WID variation. To specify the behavior of $\rho(r)$ between the limits, we choose the spherical model [13] for its good agreement with Friedberg's [2] measurements. Although the correlation function Friedberg reports is not isotropic, the shape of the function (as opposed to the scale) is the same on the horizontal and vertical die axes. In both cases, the shape closely matches that of the spherical model; it is initially linear in distance and then tapers before falling off to zero. Adopting the well-studied spherical model also ensures a valid spatial correlation function as defined in [14]. Equation (7) defines the spherical function

$$\rho(r) = \begin{cases} 1 - \frac{3r}{2\phi} + \frac{r3}{2\phi 3}, & (\text{r} \leq \phi) \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

Fig. 1 plots the function $\rho(r)$. The parameter values of a transistor are highly correlated to those of transistors in its immediate vicinity. The correlation decreases approximately linearly with distance at small distances. Then, it decreases more slowly. At a finite distance ϕ that we call *range*, the function converges to zero. This means that, at distance ϕ, there is no longer any correlation between two transistors' WID variation values.

In this paper, we express ϕ as a fraction of the chip's length. A large ϕ implies that large sections of the chip are correlated with each other; the opposite is true for small ϕ. As an illustration, Fig. 2 shows example systematic V_{th} variation maps for chips with $\phi = 0.1$ and $\phi = 0.5$. These maps were generated by the geoR statistical package [15] of R [16]. In the $\phi = 0.5$ case, we discern large spatial features, whereas in the $\phi = 0.1$ one, the features are small. A distribution without any correlation ($\phi = 0$) appears as white noise.

The process parameters we are concerned with are L_{eff} and V_{th}. A former ITRS report [17] projected that the total σ/μ of L_{eff} would be roughly half that of V_{th}. Lacking better data, we make the approximation that L_{eff}'s σ_{sys}/μ is half of V_{th}'s σ_{sys}/μ. Moreover, the systematic variation in L_{eff} causes systematic variation in V_{th}. Most of the remaining V_{th} variation is due to completely random (spatially uncorrelated) doping effects. Consequently, we use the following equation to generate a value of the systematic component of L_{eff} in a chip section given the value of the systematic component of V_{th} in the same section. Let L^0_{eff} be the nominal value of the effective length and let V^0_{th} be the nominal value of the threshold voltage. We use

$$L_{\text{eff}} = L^0_{\text{eff}} \left(1 + \frac{V_{\text{th}} - V^0_{\text{th}}}{2V^0_{\text{th}}}\right). \tag{8}$$

B. Random Variation

Random variation occurs at a much finer granularity than systematic variation—at the level of individual transistors. Hence, it is not possible to model random variation in the same explicit way as systematic variation, by simulating a grid where each section has its own parameter value. Instead, random variation appears in the model analytically. We assume that the random components of V_{th} and L_{eff} are both normally distributed with zero mean. Each has a different σ_{rand}. For ease of analysis, we assume that the random V_{th} and L_{eff} values for a given transistor are uncorrelated.

C. Values for σ and ϕ

Since the random and systematic components of V_{th} and L_{eff} are normally distributed and independent, the total WID variation is also normally distributed with zero mean. The standard deviation is as follows:

$$\sigma_{\text{total}} = \sqrt{\sigma^2_{\text{rand}} + \sigma^2_{\text{sys}}}. \tag{9}$$

For V_{th}, the 1999 ITRS [17] gave a design target of $\sigma_{\text{total}}/\mu = 0.06$ for year 2005 (although no solution existed); however, the projection has been discontinued since 1999. On the other hand, it is known that ITRS variability projections were too optimistic [18], [19]. Consequently, for V_{th}, we use $\sigma_{\text{total}}/\mu = 0.09$. Moreover, according to empirical data from [20], the random and systematic components are approximately equal in 32-nm technology. Hence, we assume that they have equal variances. Since both components are modeled as normal distributions, (9) tells us that their standard deviations σ_{rand} and σ_{sys} are equal to $9\%/\sqrt{2} = 6.3\%$ of the mean. This

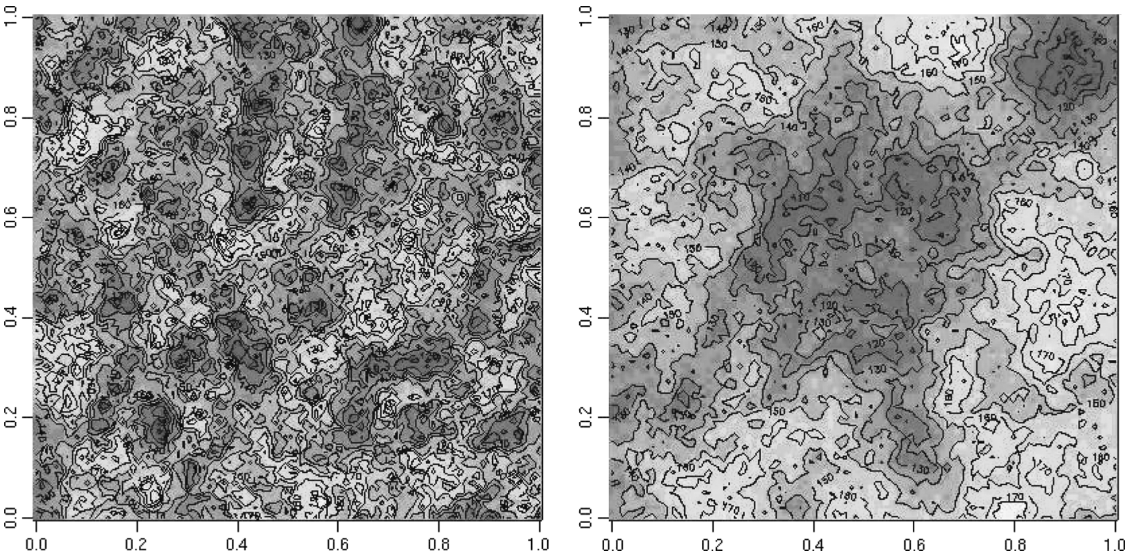

Fig. 2. Systematic V_{th} variation maps for chip with $\phi = 0.1$ (left) and $\phi = 0.5$ (right).

value for the random component matches the empirical data of Keshavarzi *et al.* [21].

As explained before, we set L_{eff}'s $\sigma_{\mathrm{total}}/\mu$ to be half of V_{th}'s. Consequently, it is 4.5%. Furthermore, assuming again that the two components of variation are more or less equal, we have that σ_{rand} and σ_{sys} for L_{eff} are equal to $4.5\%/\sqrt{2} = 3.2\%$ of the mean.

To estimate ϕ, we note that Friedberg *et al.* [2] experimentally measured the gate-length parameter to have a range close to half of the chip length. Hence, we set $\phi = 0.5$. Through (8), the same ϕ applies to both L_{eff} and V_{th}.

D. Impact on Chip Frequency

Through (3), process variation in V_{th} and L_{eff} induces variation in the delay of gates and, therefore, variation in the delay of critical paths. Unfortunately, a processor structure cannot cycle any faster than its slowest critical path can. As a result, processors are typically slowed down by process variation. To motivate the rest of the paper, this section gives a rough estimation of the impact of process variation on processor frequency.

Equation (3) approximately describes the delay of an inverter. Substituting (8) into (3) and factoring out constants with respect to V_{th} produces

$$T_g \propto \frac{V\left(1 + V_{\mathrm{th}}/V_{\mathrm{th}}^0\right)}{(V - V_{\mathrm{th}})^{\alpha}}. \tag{10}$$

Empirically, we find that (10) is nearly linear with respect to V_{th} for the parameter range of interest. Because V_{th} is normally distributed and a linear function of a normal variable is itself normal, T_g is approximately normal.

Assuming that every critical path in a processor consists of n_{cp} gates, and that a modern processor chip has thousands of critical paths, Bowman *et al.* [7] compute the probability distribution of the longest critical path delay in the chip $(\max\{T_{\mathrm{cp}}\})$. Then, the processor frequency can be estimated to be the inverse of the longest path delay $(1/\max\{T_{\mathrm{cp}}\})$.

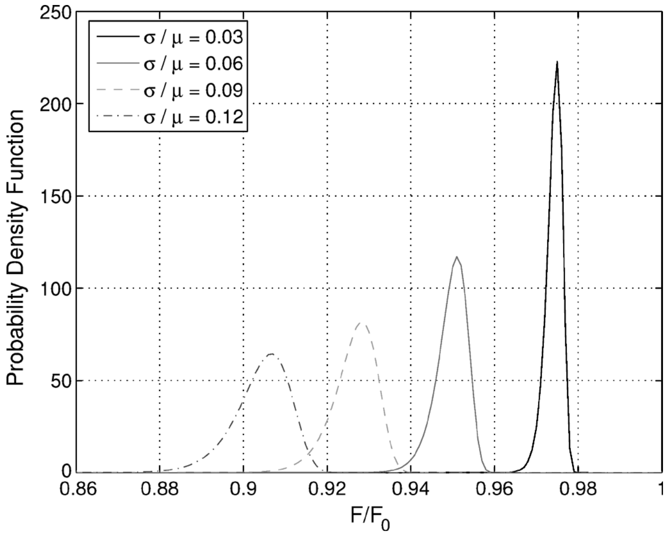

Fig. 3. Probability distribution of relative chip frequency as a function of V_{th}'s $\sigma_{\mathrm{total}}/\mu$. We use $V_{\mathrm{th}}^0 = 0.150$ V at 100 °C, 12 FO4s in the critical path, and 10 000 critical paths in the chip.

Fig. 3 shows the probability distribution of the chip frequency for different values of V_{th}'s $\sigma_{\mathrm{total}}/\mu$. The frequency is given relative to a processor without V_{th} variation (F/F_0). The figure shows that, as $\sigma_{\mathrm{total}}/\mu$ increases: 1) the mean chip frequency decreases and 2) the chip frequency distribution gets more spread out. In other words, given a batch of chips, as V_{th}'s $\sigma_{\mathrm{total}}/\mu$ increases, the mean frequency of the batch decreases and, at the same time, an individual chip's frequency deviates more from the mean.

Such frequency loses may be reduced if the processor is equipped with ways of tolerating some variation-induced timing errors. As a possible first step in this direction, the rest of the paper presents a model of variation-induced timing errors in a processor. In future work, we will examine how such errors can be tolerated. In the rest of the paper, we do not use Bowman *et al.*'s [7] critical path model any more.

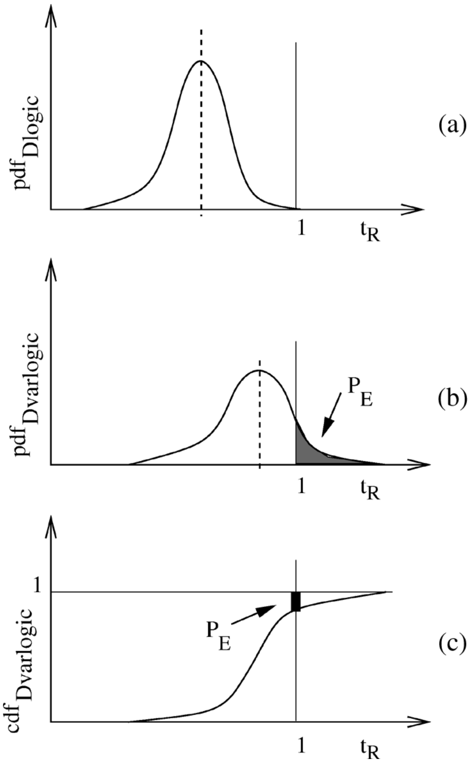

Fig. 4. Example critical path delay distributions (a) before variation in pdf form and after variation in (b) pdf and (c) cdf form. Dark parts show error rate.

IV. Timing Error Model

This section presents VATS, a novel model of variation-induced timing errors in processor pipelines. In the following, we first model errors in logic and then in SRAM memory.

A. General Approach

A pipeline stage typically has a multitude of paths, each one with its own time slack—possibly dependent on the input data values. This work makes two simplifying assumptions about the failure model.

Assumption 1: A path causes a timing fault if and only if it is exercised and its delay exceeds the clock period. Note that this fault definition does not account for any architectural masking effects. However, architectural vulnerability factors (AVFs) [22] could be applied to model these masking effects if desired.

Assumption 2: Each stage is tightly designed so that, in the absence of process variation, at least one path has a delay equal to the clock period t_0. This provides a prevariation base case against which to make delay comparisons.

In the following, path delay is normalized by expressing it as a fraction t_R of t_0. Our model begins with the probability density function (pdf) of the normalized path delays in the pipeline stage. Fig. 4(a) shows an example pdf before variation effects. The right tail abuts the $X = 1$ abscissa and there are no timing errors.

As the pipeline stage paths suffer parameter variation, the pdf changes shape: the curve may change its average value and its spread [e.g., Fig. 4(b)]. All the paths that have become longer than 1 generate errors. Our model estimates the probability of error (P_E) as the area of the shaded region in the figure. Alternatively, we can efficiently compute P_E using the cdf of the normalized path delays by taking the difference between 1 and the value of the cdf as shown in Fig. 4(c). In general, if we clock the processor with period t_R, the probability of error is

$$P_E(t_R) = 1 - \text{cdf}(t_R).$$

In the event that race-through errors are also a concern, $\text{cdf}(t_h)$ gives the probability of violating the minimum hold time t_h. However, we will not consider hold-time violations in the rest of the paper.

B. Timing Errors in Logic

We start by considering a pipeline stage of only logic. We represent the logic critical path delay in the absence of variation as a random variable D_{logic}, which is distributed in a way similar to Fig. 4(a). Such delay is composed of both wire and gate delay. For simplicity, we assume that wire accounts for a fixed fraction k_w of total delay. This assumption has been made elsewhere [23]. Consequently, we can write

$$\begin{aligned} D_{\text{logic}} &= D_{\text{wire}} + D_{\text{gates}} \\ D_{\text{wire}} &= k_w \; D_{\text{logic}} \\ D_{\text{gates}} &= (1 - k_w) D_{\text{logic}}. \end{aligned} \tag{11}$$

We now consider the effects of variation. Since variation typically has a very small effect on wires, we only consider the variation of D_{gates}, which has a random and a systematic component. For each path, we divide the systematic variation component $(\Delta D_{\text{gates\_sys}})$ into two terms: 1) the average value of it for all the paths *in the stage* $(\overline{\Delta D_{\text{gates\_sys}}})$—which we call the stage systematic mean—and 2) the rest of the systematic variation component $(\Delta D_{\text{gates\_sys}} - \overline{\Delta D_{\text{gates\_sys}}})$—which we call intrastage systematic deviation.

Given the high degree of spatial correlation in process (P) and temperature (T) variation, and the small size of a pipeline stage, the intrastage systematic deviation is small. Indeed, in Section III-C, we suggested a value of ϕ equal to 0.5 (half of the chip length). On the other hand, the length of a pipeline stage is less than, say, 0.1 of the length of a typical four-core chip. Therefore, given that the stage dimensions are significantly smaller than ϕ, the transistors in a pipeline stage have highly correlated systematic V_{th} and systematic L_{eff} values. Using Monte Carlo simulations with the parameters of Section III-C, we find that the intrastage systematic deviation of D_{gates} has a $\sigma_{\text{intrasys}} \approx 0.004 \times \mu$, while the variation of $\overline{\Delta D_{\text{gates\_sys}}}$ across the pipeline stages of the processor has a $\sigma_{\text{intersys}} \approx 0.05 \times \mu$. Similarly, T varies much more across stages than within them.

The random component of D_{gates}'s variation is estimated from the fact that we model a path as n FO4 gates connected with short wires. Each gate's random component is independent. Consequently, for a whole n-gate path, D_{gates}'s σ_{rand} is $\sqrt{n} \times \sigma_{\text{rand\_D}_{\text{FO4}}}$, where $\sigma_{\text{rand\_D}_{\text{FO4}}}$ is the standard deviation of the delay of one FO4 gate. If we take $n = 12$ as representative of high-end processors, the overall variation is small. It can be shown that D_{gates}'s $\sigma_{\text{rand}} \approx 0.01 \times \mu$. Finally, T has no random component.

We can now generate the distribution of D_{logic} with variation (which we call D_{varlogic} and show in Fig. 4(b)) as follows. We model the contribution of $\overline{\Delta D_{\text{gates\_sys}}}$ in the stage as a factor η that multiplies D_{gates}. This factor is the average increase in gate delay across all the paths in the stage due to systematic variation. Without variation, $\eta = 1$.

We model the contribution of the intrastage systematic deviation and of the random variation as D_{extra}, a small additive normal delay perturbation. Since D_{extra} combines D_{gates}'s intrastage systematic and random effects, $\sigma_{\text{extra}} = \sqrt{\sigma^2_{\text{intrasys}} + \sigma^2_{\text{rand}}}$. For our parameters, $\sigma_{\text{extra}} \approx 0.011 \times \mu$. Like η, D_{extra} should multiply D_{gates} as shown in (12). However, to simplify the computation and because D_{logic} is clustered at values close to one, we prefer to approximate D_{extra} as an additive term as in

$$\begin{aligned} D_{\text{varlogic}} &= (\eta + D_{\text{extra}})D_{\text{gates}} + D_{\text{wire}} \\ &= (1-k_w)(\eta + D_{\text{extra}})D_{\text{logic}} + k_w D_{\text{logic}} \qquad (12) \\ &\approx (1-k_w)(\eta D_{\text{logic}} + D_{\text{extra}}) + k_w D_{\text{logic}}. \qquad (13) \end{aligned}$$

Once we have the D_{varlogic} distribution, we numerically integrate it to obtain its $\text{cdf}_{D_{\text{varlogic}}}$ [Fig. 4(c)]. Then, the estimated error rate P_E of the stage cycling with a relative clock period t_R is

$$P_E(t_R) = 1 - \text{cdf}_{D_{\text{varlogic}}}(t_R). \qquad (14)$$

1) How to Use the Model: To apply (13), we must calculate k_w, η, D_{extra}, and D_{logic} for the prevailing variation conditions. To do this, we produce a gridded spatial map of process variation using the model in Section III-A and superimpose it on a high-performance processor floorplan. For each pipeline stage, we compute η from the pipeline stage's temperature and the systematic L_{eff} and V_{th} maps. Moreover, by subtracting the resulting mean delay of the stage from the individual delays in the grid points inside the stage, we produce the intrastage systematic deviation. We combine the latter distribution with the effect of the random process variation to obtain the D_{extra} distribution. D_{extra} is assumed normal.

Ideally, we would obtain a per-stage k_w and D_{logic} through timing analysis of each stage. For our general evaluation, we assume that the LF adder in [24] is representative of processor logic stages and set $k_w = 0.35$ [23]. Additionally, we derive $\text{pdf}_{D_{\text{logic}}}$ using experimental data from Ernst *et al.* [25]. They measure the error rate P_E of a multiplier unit as they reduce its supply voltage V. By reducing V, they lengthen path delays. Those paths with delays longer than the cycle time cause an error. Our aim is to find the $\text{pdf}_{D_{\text{logic}}}$ curve from their plot of $P_E(V)$ [a curve similar to that shown in Fig. 5(a)].

Focusing on (13), Ernst's experiment corresponds to an environment with no parameter variation, so $D_{\text{extra}} = 0$. Each V corresponds to a new average $\eta(V)$ and, therefore, a new $D_{\text{varlogic}}(V)$ distribution. We compute each $\eta(V)$ using the alpha-power model (3) as the ratio of gate delay at V and gate delay at the minimum voltage in [25] for which no errors were detected.

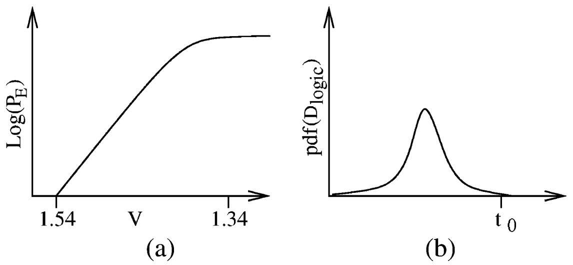

Fig. 5. (a) Error rate versus voltage curve from [25] and (b) corresponding $\text{pdf}_{D_{\text{logic}}}$.

At a voltage V, the probability of error is equal to the probability of exercising a path with a delay longer than one clock cycle. Hence, $P_E(V) = P(D_{\text{varlogic}}(V) > 1)$. If we use (13) and define $g(V) = 1/(k_w + \eta(V) \times (1 - k_w))$, we have $D_{\text{varlogic}}(V) = D_{\text{logic}}/g(V)$. Therefore

$$\begin{aligned} P_E(V) &= P\left(D_{\text{varlogic}}(V) > 1\right) \\ &= P\left(D_{\text{logic}}/g(V) > 1\right) \\ &= P\left(D_{\text{logic}} > g(V)\right) \\ &= 1 - \text{cdf}_{D_{\text{logic}}}\left(g(V)\right). \qquad (15) \end{aligned}$$

Letting $y = g(V)$, we have $\text{cdf}_{D_{\text{logic}}}(y) = 1 - P_E(V)$. Therefore, we can generate $\text{cdf}_{D_{\text{logic}}}$ numerically by taking successive values of V_i, measuring $P_E(V_i)$ from Fig. 5(a), computing $y_i = g(V_i)$, and plotting $(y_i, 1 - P_E(V_i))$, which is $(y_i, \text{cdf}_{D_{\text{logic}}}(y_i))$. After that, we smooth and numerically differentiate the resulting curve to find the sought function $\text{pdf}_{D_{\text{logic}}}$. Finally, we approximate the $\text{pdf}_{D_{\text{logic}}}$ curve with a normal distribution, which we find has $\mu = 0.849$ and $\sigma = 0.019$ [a curve similar to that shown in Fig. 5(b)].

Strictly speaking, this $\text{pdf}_{D_{\text{logic}}}$ curve only applies to the circuit and conditions measured in [25]. To generate $\text{pdf}_{D_{\text{logic}}}$ for a different stage with a different technology and workload characteristics, one would need to use timing analysis tools on that particular stage. In practice, Section V-A shows empirical evidence that this method produces $\text{pdf}_{D_{\text{logic}}}$ curves that are usable under a range of conditions, not just those under which they were measured.

Finally, since D_{logic} and D_{extra} are normally distributed, D_{varlogic} in (13) is also normally distributed.

C. Timing Errors in SRAM Memory

To model variation-induced timing errors in SRAM memory, we build on the work of Mukhopadhyay *et al.* [26]. They consider *random* V_{th} variation only and use the Shockley current model. We extend their work to account for random and systematic variation of both L_{eff} and V_{th} and use the more accurate alpha-power current model. Additionally, we describe the access time distribution for an entire multiline SRAM array rather than for a singe cell.

Mukhopadhyay *et al.* [26] describe four failure modes in the SRAM cell of Fig. 6: Read failure, where the contents of a cell are destroyed when the cell is read; Write failure, where a write is unable to flip the cell; Hold failure, where a cell loses its state; and Access failure, where the time needed to access the cell is too long, leading to failure. The authors provide analytical equations for these failure rates, which show that for the standard

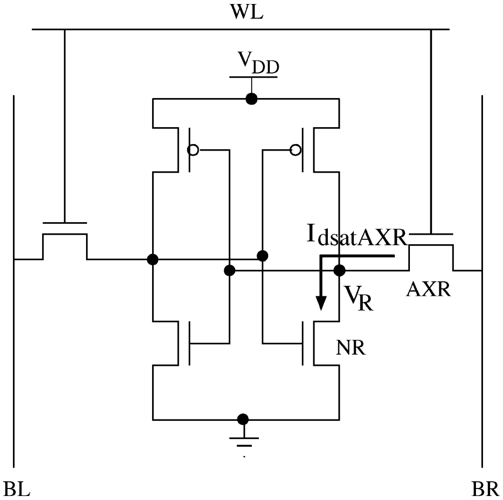

Fig. 6. Read from 6T SRAM cell, pulling right bitline low.

deviations of V_{th} considered here, Access failures dominate and the rest are negligible. Because Access failures are the dominant errors and have no clear remedy, they are our focus. According to [26], the cell access time under variation on a read is

$$\begin{aligned} D_{\text{varcell}} &\propto \frac{1}{I_{\text{dsatAXR}}} \\ &= h(V_{\text{thAXR}}, V_{\text{thNR}}, L_{\text{AXR}}, L_{\text{NR}}) \end{aligned} \tag{16}$$

where V_{thAXR} and L_{AXR} are the V_{th} and L_{eff} of the AXR access transistor in Fig. 6, and V_{thNR} and L_{NR} are the same parameters for the NR pull-down transistor in Fig. 6. We now discuss the form of this function h, first using the Shockley-based model of [26] and then using our extension that uses the alpha-power current model. Finally, we use h to develop the delay distribution D_{varmem} for a read to a variation-afflicted SRAM structure containing a given number of lines and a given number of bits per line.

1) $\text{I}_{\text{dsatAXR}}$ *Using Shockley Model:* The model in [26] uses the traditional Shockley long channel transistor equations. Consider the case illustrated in Fig. 6: a read operation where the bitline BR is being driven low. Transistor AXR is in saturation and transistor NR is in the linear range. Equating the currents using Kirchoff's current law

$$\begin{aligned} I_{\text{dsatAXR}} &= \frac{K_1}{L_{\text{AXR}}}(V_{DD} - V_R - V_{\text{thAXR}})^2 \\ &= \frac{K_2}{L_{\text{NR}}}(V_{DD} - V_{\text{thNR}} - 0.5V_R)V_R. \end{aligned} \tag{17}$$

In the Shockley model (1), we have replaced β with K/L_{eff}, where K is a constant and L_{eff} is the effective length of the respective transistor. Equation (17) is a quadratic equation in V_R. We can thus find I_{dsat} and subsequently the function h.

2) $\text{I}_{\text{dsatAXR}}$ *Using Alpha-Power Model:* We now use the more accurate alpha power law [8] to find I_{dsatAXR}. By equating currents as in (17), we have

$$\begin{aligned} I_{\text{dsatAXR}} &= \frac{K_1}{L_{\text{AXR}}}(V_{DD} - V_R - V_{\text{thAXR}})^{\alpha} \\ &= \frac{K_2}{L_{\text{NR}}}(V_{DD} - V_{\text{thNR}})^{\alpha/2}V_R. \end{aligned} \tag{18}$$

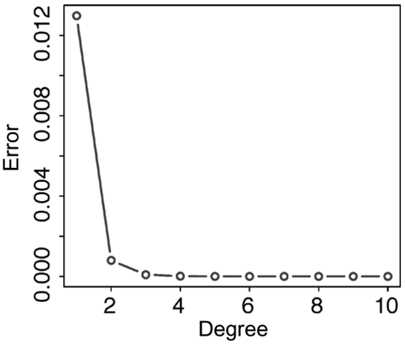

Fig. 7. Error versus degree of expansion of $\mathbf{z}$.

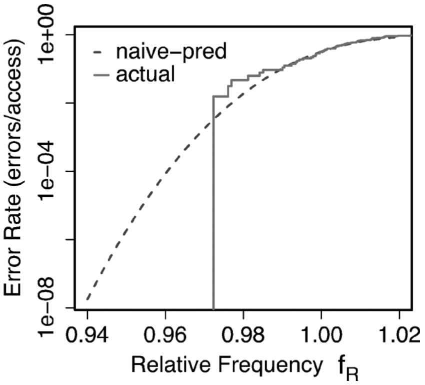

Fig. 8. Error-rate for example 64-line SRAM structure assuming continuous model (dashed line) or discrete one with fixed read latencies (solid line).

As in (17), constants have been folded into K_1 and K_2. To solve for V_R, perform the following transformation:

$$\begin{aligned} (V_{DD} - V_R - V_{\text{thAXR}})^{\alpha} &= (V_{DD} - V_{\text{thAXR}})^{\alpha} \\ &\quad \times \left(1 - \frac{V_R}{V_{DD} - V_{\text{thAXR}}}\right)^{\alpha}. \end{aligned} \tag{19}$$

Let $z = V_R/(V_{DD} - V_{\text{thAXR}})$ and expand $(1-z)^{\alpha}$ using the Taylor series (4). Typical values of z are near 0.25, so we compute the expansion about that point. Fig. 7 plots the error versus the degree of the expansion. Depending on the accuracy desired, we can choose the appropriate number of terms, but for most practical purposes, a degree of 2 is sufficient, making (18) a quadratic equation in V_R

$$(1-z)^{\alpha} \approx 1 - \alpha z + \alpha(\alpha - 1)\frac{z^2}{2}.$$

Now, we can easily solve for V_R and find a closed form analytic expression for I_{dsatAXR}.

3) Error Rate Under Process Variation: We now have an analytic expression for the access time D_{varcell} of a single SRAM cell under variation using (16). It is a function of four variables: V_{thAXR}, V_{thNR}, L_{AXR}, and L_{NR}. A six-transistor memory cell is very small compared to the correlation range ϕ of V_{th} and L_{eff} (Section III-A). Therefore, we assume that the systematic component of variation is the same for all the transistors in the cell and even for the whole SRAM bank. Now, using multivariate Taylor expansion (5), the mean μ_{Dvarcell} and standard deviation σ_{Dvarcell} of D_{varcell} can be expressed as a function of the μ and σ of each of these four variables.

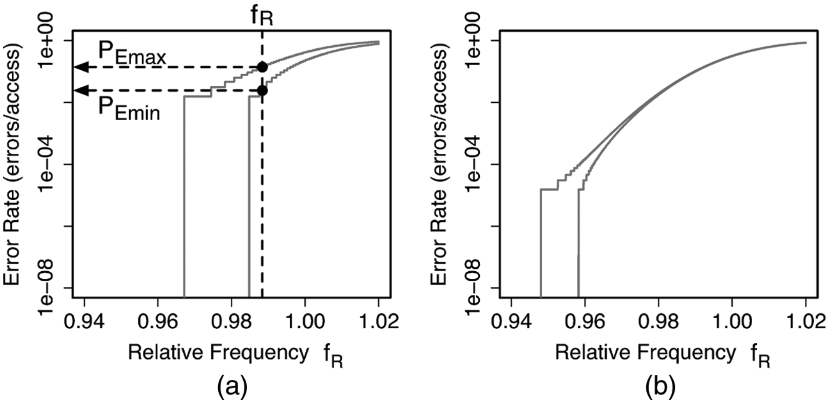

Fig. 9. The 90% confidence intervals for P_E in (a) 64-line SRAM and in (b) 64K-line SRAM as a function of relative frequency f_R.

In reality, an SRAM array access does not read only a single cell but a line—e.g., 8–1024 cells. The time required to read an entire line (D_{varline}) is then the maximum of the times required to read its constituent cells. To compute this maximum, we use (6), which gives us the mean and standard deviation of the line access time D_{varline} in terms of the cell access time distribution. D_{varline} follows the Gumbel distribution, but we approximate it with a normal distribution.

The access to the memory array itself takes only a fraction a of the whole pipeline cycle—logic structures such as sense amplifiers, decoders, and comparators consume the rest. Section IV-B has already shown how to model the logic delays. Consequently, the total delay to read a line from an SRAM in the presence of variation (D_{var_rd}) is the sum of the normal distributions of the delays in the memory array and in the logic. It is distributed normally with

$$\mu_{var\_rd} = a\,\mu_{\text{varline}} + (1-a)\mu_{\text{varlogic}} \tag{20}$$

$$\sigma_{var\_rd} = \sqrt{a^2\,\sigma^2_{\text{varline}} + (1-a)^2\sigma^2_{\text{varlogic}}}. \tag{21}$$

Then, the estimated error rate of a memory stage cycling with a relative clock period t_R is

$$P_E(t_R) = 1 - \text{cdf}_{Dvar\_rd}(t_R). \tag{22}$$

Note that this model is only an approximation, since it provides a curve for $P_E(t_R)$ that is continuous. In reality, an SRAM structure has relatively few paths and, as a result, a stepwise error curve is more accurate. For example, assume that we have a 64-line SRAM structure where the slowest line fails at some period t_0. If we assume that all lines are accessed with equal frequency, the probability of error jumps instantaneously from 0 to 1/64 at t_0. Fig. 8 shows the P_E curve for accesses to a 64-line SRAM as a function of $f_R = 1/t_R$. The dashed curve corresponds to the model of (22); the solid line corresponds the case when we consider that each line has a different read latency and assume that it is fixed. We have generated these latencies by sampling the D_{var_rd} distribution.

In reality, the random component of variation affects the read latency of each of the N_l lines of the structure. Consequently, given a relative clock period t_R, we cannot readily compute the number of lines $l(t_R)$ that have a $D_{var\_rd} \le t_R$. However, suppose that we are able to determine that any one individual line has a probability $p(t_R)$ to have $D_{var\_rd} \le t_R$. This is $\text{cdf}_{Dvar\_rd}(t)$. In this case, we can compute a confidence interval to bound $l(t_R)$. Specifically, the number of lines $l(t_R)$ that have $D_{var\_rd} \le t_R$ follows the binomial distribution $\mathbf{B}(N_l, p(t_R))$. Let us call its cdf $B(n)$.

Taking the inverse of the binomial cdf provides a confidence interval for $l(t_R)$. For example, the following gives a 90% confidence interval:

$$B^{-1}(0.05) \le l(t_R) \le B^{-1}(0.95). \tag{23}$$

This means that the number of lines in the SRAM that can be accessed without error $(l(t_R))$ is between $B^{-1}(0.05)$ and $B^{-1}(0.95)$ with 90% probability. These two boundaries are numbers between 0 and N_l.

The expression $l(t_R)/N_l$ is the fraction of lines in the SRAM that can be accessed without error at t_R. Assuming that all lines are accessed with equal frequency, this is the probability of error-free execution of an SRAM read at t_R. We define this function as $\text{cdf}_{\text{Dvarmem}}(t_R)$. The bounds for $\text{cdf}_{\text{Dvarmem}}(t_R)$ for a 90% confidence interval are then

$$\frac{B^{-1}(0.05)}{N_l} \le \mathit{cdf}_{\text{Dvarmem}}(t_R) \le \frac{B^{-1}(0.95)}{N_l}. \tag{24}$$

The estimated error rate of the memory stage cycling with a relative clock period t_R is then

$$P_E(t_R) = 1 - \mathit{cdf}_{\text{Dvarmem}}(t_R). \tag{25}$$

Fig. 9 shows P_E for a 90% confidence interval as a function of $f_R = 1/t_R$. Charts (a) and (b) correspond to an SRAM with 64 lines and 65 536 lines, respectively. In both cases, the line has 64 bits. Each chart has two curves, which bound the 90% confidence interval. For example, in Chart (a), if we select a given f_R, the intersections to the two curves (P_{Emin} and P_{Emax}) give the 90% confidence interval for P_E at this f_R.

The figure shows that the confidence interval of P_E is narrow for large SRAMs. Consequently, for large SRAMs, it may make sense to discard this interval-based computation altogether and, instead, use the continuous $\text{cdf}_{Dvar\_rd}(t_R)$ to approximate P_E.

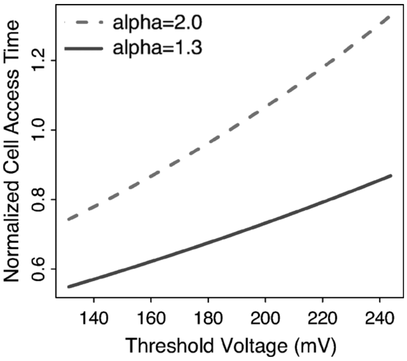

Fig. 10. Relative mean access time ($\boldsymbol{\mu}_{\mathbf{varcell}}$) for $\boldsymbol{\alpha}$ equal to 1.3 and 2.0. Latter corresponds to Shockley model.

This is accomplished by explicitly enforcing an instantaneous transition from $P_E = 0$ to $P_E = 1/N_l$

$$P_E = 1 - cdf_{\text{Dvarmem}}(t_R) \approx \begin{cases} 1 - cdf_{Dvar\_rd}(t_R), & 1 - cdf_{Dvar\_rd}(t_R) \geq \frac{1}{N_l} \\ 0, & \text{otherwise} \end{cases} . \quad (26)$$

4) Comparing Shockley and Alpha-Power Models: In Fig. 10, we plot the mean access time (μ_{varcell}) for the Shockley model (dotted line) and for the alpha-power model (solid line). Access times are normalized to the one given by the Shockley model at 85 °C. From the figure, we see that the mean access time differs significantly for the two values of α. More importantly, it can be shown that σ_{varcell} is around 3.5% of the mean for the Shockley model and around 2% of the mean for the alpha-power model. Consequently, with decreasing α, the mean and standard deviation of the access time decrease.

V. Evaluation

A. Empirical Validation

To partially validate the VATS model, we use it to explain some error rate data obtained empirically elsewhere. We validate both the logic and the memory model components. For the former, we use the curves obtained by Das *et al.* [27], who reduce the supply voltage V of the logic units in an Alpha-like pipeline and measure the error rate in errors per cycle. They report curves for three different T: 45 °C, 65 °C, and 95 °C. Their curves are shown in solid pattern in Fig. 11.

To validate our model, we use the 65 °C curve to predict the other two curves. We first determine D_{logic} from the 65 °C curve through the procedure of Section IV-B1. Recall that we generate the pdf$_{D_{\text{logic}}}$ numerically and then fit a normal distribution. We then use D_{logic} to predict the 95 °C and 45 °C curves as follows. We generate a large number of V_i values. For each V_i, we compute $\eta(V_i)$ as discussed in Section IV-B1. Process variation is small in the dataset—since the latter corresponds to a 180-nm process. Consequently, we set D_{extra} to zero. Knowing the D_{logic} distribution, we use (13) for each $\eta(V_i)$ to compute the $D_{\text{varlogic}}(V_i)$ distribution. Finally, we plot the $(V_i, P_E(V_i))$ pairs from our model as dashed lines in Fig. 11 along with the measured values (solid lines). From the figure, we see that the predicted curves track the experimental data closely. One source of the disagreement between the two is the normal approximation of D_{logic}, which is assumed for simplicity.

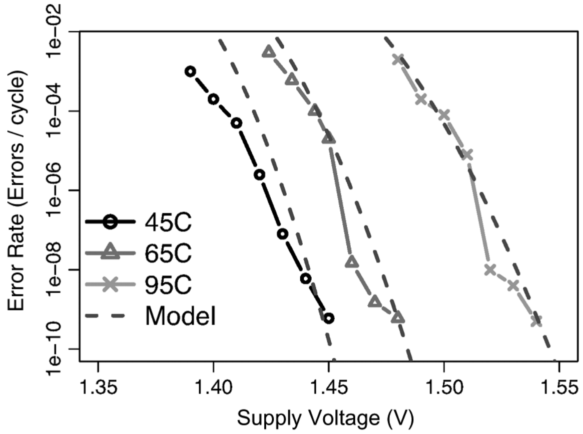

Fig. 11. Validating logic model by comparing measured and predicted number of errors per cycle.

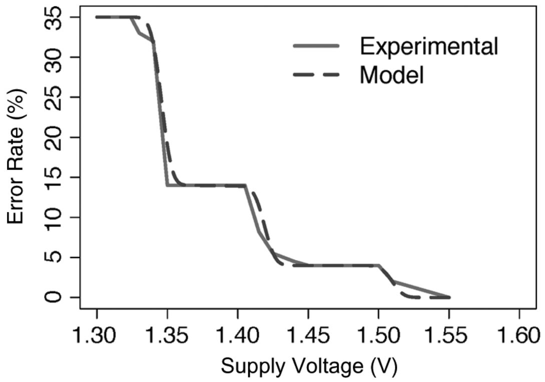

Fig. 12. Validating memory model by comparing measured and predicted fraction of accesses that fail.

To validate the memory model, we use experimental data from Karl *et al.* [28]. They examine a 64-KB SRAM with 32-bit lines comprising four different-latency banks and measure the error rate as the supply voltage V changes. We assume that all cells have the same value of the systematic process variation. Using the measured $P_E(V)$ for each bank, we find $D_{var\_rd}(t_R)$ using the method of (20) and (21) in Section IV-C3. The original data is shown in solid pattern in Fig. 12, and the prediction is displayed as a dashed line. From the figure, we see that the predicted and measured error rate are close.

B. Example Error Curves

As one example of the uses of our model, we apply it to estimate the error rate of the logic and memory stages of an AMD Opteron processor as we increase the frequency. After generating a V_{th} and L_{eff} variation map according to our variation model, we apply the timing error model to compute the error rate versus frequency for each pipeline stage. A stage is classified as either memory dominated or logic dominated. For the logic-dominated stages (e.g., the decoder and functional units), we use the error model of Section IV-B. For the memory-dominated stages (e.g., the caches), we use (26) of the noncontinuous model in Section IV-C3. Because we do not have actual

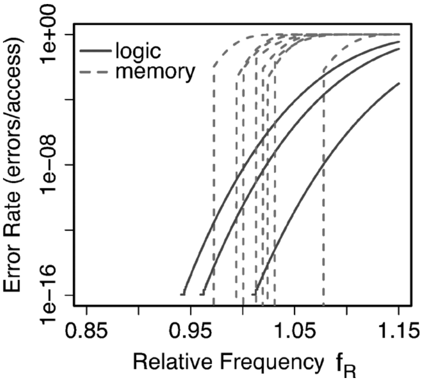

Fig. 13. Estimated error rates of memory and logic pipeline stages in AMD Opteron.

net-level data for the microprocessor, the critical path distribution of each logic stage is assumed to match that of the multiplier in [25]. Fig. 13 shows the results, where the frequency is normalized to the one that the processor without process variation can deliver.

In the figure, each line corresponds to one pipeline stage. We see that memory stages have steeper error curves than the logic ones. This is due to the small number of lines in the structures; when the clock frequency exceeds the speed of the slowest line, the error rate undergoes a step change from zero to a relatively high number. On the other hand, logic error onset is more gradual. We envision a situation where architects and circuit designers will use such error curves to design processors that can tolerate timing errors.

C. Tradeoffs in Model

Perhaps the main shortcoming of VATS is the loss of precision due to two main simplifications: 1) the use of normal approximations and 2) the assumption that wire delay is not affected by variation and accounts for a fixed fraction k_w of logic delay. Section V-A has argued that the loss of accuracy is small in practice. The approximations in VATS make it easier to apply it in the early stages of design, when architects must estimate variation effects at a high level.

VI. Related Work

Agarwal *et al.* [29] proposed a simple correlation model for systematic variation based on quad-tree partitioning. The model is widely used [12], [30]. It is computationally efficient, but no analytical form for the correlation structure is given, and it is not clear how well the model matches measured correlation data. The spherical correlation function used in this paper has been chosen to match empirical measurements but has the disadvantage that generating random instances for Monte Carlo simulation is more computationally intensive.

Mukhopadhyay *et al.* [26] proposed models for timing errors in SRAM memory due to random V_{th} variation. They consider several failure modes. As part of the VATS model, we extended their model of Access time errors by: 1) also including systematic variation effects; 2) also considering variation in L_{eff}; 3) modeling the maximum access time of a *line* of SRAM rather than a single cell; and 4) using the alpha-power model that uses an α equal to 1.3.

Memik *et al.* [31], [32] modeled errors in SRAM memory due to crosstalk noise as they overclock circuits. They use high degrees of overclocking—twice the nominal frequency and more. In the less than 25% overclocking regime that we consider, such crosstalk errors are negligible. For very small feature-size technologies, however, the situation may change.

Ernst *et al.* [25] and Karl *et al.* [28] measured the error rate of a multiplier and an SRAM circuit, respectively, by reducing the voltage beyond safe limits to save power. They plot curves for error rate versus voltage. In this paper, we outlined a procedure to extract the distribution of path delays from these curves and validated parts of our model by comparing it against their curves.

VII. Conclusion

Parameter variation is the next big challenge for processor designers. To gain insight into this problem from a microarchitectural perspective, this paper made two contributions. First, it developed a novel model for process variation. The model uses three intuitive input parameters and is computationally inexpensive. Second, the paper presented VATS, a novel model of timing errors due to parameter variation. The model is widely usable, since it applies to logic and SRAM units and is driven with intuitive parameters. The model has been partially validated with empirical data. The resulting combined model, called VARIUS, has been used to estimate timing error rates for pipeline stages in a processor with variation.

References

[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, "Parameter variations and impact on circuits and microarchitecture," in *Proc. Design Automation Conf.*, Jun. 2003.

[2] P. Friedberg, Y. Cao, J. Cain, R. Wang, J. Rabaey, and C. Spanos, "Modeling within-die spatial correlation effects for process-design co-optimization," in *Proc. Int. Symp. Quality Electronic Design*, Mar. 2005.

[3] M. Orshansky, L. Milor, and C. Hu, "Characterization of spatial intrafield gate CD variability, its impact on circuit performance, and spatial mask-level correction," *IEEE Trans. Semiconduct. Manuf.*, vol. 17, no. 1, pp. 2–11, Feb. 2004.

[4] B. Stine, D. Boning, and J. Chung, "Analysis and decomposition of spatial variation in integrated circuit processes and devices," *IEEE Trans. Semiconduct. Manuf.*, vol. 10, no. 1, pp. 24–41, Feb. 1997.

[5] S. Borkar, T. Karnik, and V. De, "Design and reliability challenges in nanometer technologies," in *Proc. Design Automation Conf.*, Jun. 2004.

[6] Y. Taur and T. H. Ning, *Fundamentals of Modern VLSI Devices*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[7] K. Bowman, S. Duvall, and J. Meindl, "Impact of die-to-die and within-die parameter fluctuations on the maximum clock frequency distribution for gigascale integration," *IEEE J. Solid State Circuits*, vol. 37, no. 2, pp. 183–190, Feb. 2002.

[8] T. Sakurai and R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.

[9] K. Kanda, K. Nose, H. Kawaguchi, and T. Sakurai, "Design impact of positive temperature dependence on drain current in sub-1-V CMOS VLSIs," *IEEE J. Solid-State Circuits*, vol. 36, no. 10, pp. 1559–1564, Oct. 2001.

[10] A. Papoulis, *Probability, Random Variables and Stochastic Processes*. New York: McGraw-Hill, 2002.

[11] E. Castillo, *Extreme Value and Related Models With Applications in Engineering and Science*. New York: Wiley, 2004.

[12] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*. New York: Springer, 2005.

[13] N. Cressie, *Statistics for Spatial Data*. New York: Wiley, 1993.

[14] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," in *Proc. Int. Symp. Physical Design*, Apr. 2006.

[15] P. Ribeiro and P. Diggle, The geoR Package [Online]. Available: http://www.est.ufpr.br/geoR

[16] The R Project for Statistical Computing [Online]. Available: http://www.r-project.org

[17] International Technology Roadmap for Semiconductors 1999.

[18] A. Kahng, "The road ahead: Variability," *IEEE Design Test Computers*, vol. 19, no. 3, pp. 116–120, May–Jun. 2002.

[19] A. Kahng, "How much variability can designers tolerate?," *IEEE Design Test Computers*, vol. 20, no. 6, pp. 96–97, Nov.–Dec. 2003.

[20] T. Karnik, S. Borkar, and V. De, "Probabilistic and variation-tolerant design: Key to continued Moore's law," in *Proc. Workshop Timing Issues in Specification Synthesis Digital Systems*, Feb. 2004.

[21] A. Keshavarzi *et al.*, "Measurements and modeling of intrinsic fluctuations in MOSFET threshold voltage," in *Proc. Int. Symp. Low Power Electronics Design*, Aug. 2005.

[22] S. Mukherjee, C. Weaver, J. Emer, S. Reinhardt, and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," in *Proc. Int. Symp. Microarchitecture*, Dec. 2003.

[23] E. Humenay, D. Tarjan, and K. Skadron, "Impact of parameter variations on multicore chips," in *Proc. Workshop Architectural Support Gigascale Integration*, Jun. 2006.

[24] Z. Huang and M. D. Ercegovac, "Effect of wire delay on the design of prefix adders in deep-submicron technology," in *Proc. Asilomar Conf. Signals Systems*, Oct. 2000.

[25] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Zeisler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Proc. Int. Symp. Microarchitecture*, Dec. 2003.

[26] S. Mukhopadhyay, H. Mahmoodi, and K. Roy, "Modeling of failure probability and statistical design of SRAM array for yield enhancement in nanoscaled CMOS," *IEEE Trans. Computer-Aided Design Integrated Circuits*, vol. 24, no. 12, pp. 1859–1880, Dec. 2005.

[27] S. Das, S. Pant, D. Roberts, S. Lee, D. Blaauw, T. Austin, T. Mudge, and K. Flautner, "A self-tuning DVS processor using delay-error detection and correction," in *Proc. IEEE Symp. VLSI Circuits*, Jun. 2005.

[28] E. Karl, D. Sylvester, and D. Blaauw, "Timing error correction techniques for voltage-scalable on-chip memories," in *Proc. Int. Symp. Circuits Systems*, May 2005.

[29] A. Agarwal *et al.*, "Path-based statistical timing analysis considering inter- and intra-die correlations," in *Proc. Int. Workshop Timing Issues Specification Synthesis Digital Systems*, Jun. 2002.

[30] X. Liang and D. Brooks, "Mitigating the impact of process variations on CPU register file and execution units," in *Proc. Int. Symp. Microarchitecture*, Dec. 2006.

[31] A. Mallik and G. Memik, "A case for clumsy packet processors," in *Proc. Int. Symp. Microarchitecture*, Dec. 2004.

[32] G. Memik, M. H. Chowdhury, A. Mallik, and Y. I. Ismail, "Engineering over-clocking: Reliability-performance trade-offs for high-performance register files," in *Proc. Int. Conf. Dependable Systems Networks*, Jun. 2005.

Smruti R. Sarangi received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kharagpur, and the M.S. and Ph.D. degrees in computer science from the University of Illinois, Urbana-Champaign, in 2007.

He is now at Synopsys Research, Bangalore, India, where his research interests include processor reliability, schemes to mitigate the effects of process variation, and power management schemes.

Brian Greskamp (S'07) received the B.S. degree in computer engineering from Clemson University, Clemson, SC, in 2003. He is working toward the Ph.D. degree in computer science at the University of Illinois, Urbana-Champaign.

His research focuses on microarchitectural techniques for improving processor performance in the presence of parameter variation.

Radu Teodorescu (S'06) received the B.S. degree in computer science from the Technical University of Cluj-Napoca, Romania, and the M.S. degree in computer science from University of Illinois, Urbana-Champaign. He is currently working toward the Ph.D. degree in computer science from the University of Illinois.

His research interests include processor design for reliability and variation tolerance.

Jun Nakano (M'07) received the M.S. degree physics from the University of Tokyo, Japan, and Ph.D. degree in computer science from the University of Illinois, Urbana-Champaign, in 2006.

He is now an Advisory IT Specialist at IBM, Japan. His research interests include reliability in computer architecture and variability in semiconductor manufacturing.

Dr. Nakano is a member of the ACM.

Abhishek Tiwari (S'07) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology, Kanpur, and the M.S. degree in computer science from the University of Illinois, Urbana-Champaign. He is working toward the Ph.D. degree in computer science at the University of Illinois.

His research focuses on variability and its impact on the performance, power and reliability of processors.

Josep Torrellas (S'87–M'90–SM'03–F'04) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He is a Professor of computer science and the Willett Faculty Scholar at the University of Illinois, Urbana-Champaign. His research interests include multiprocessor computer architecture, thread-level speculation, low-power design, and hardware and software reliability.

Dr. Torrellas serves as the Chair of the IEEE Technical Committee on Computer Architecture (TCCA). He is a member of the ACM.