Efficient Parameter Variation Sampling for Architecture Simulations

Feng Lu Russ Joseph Goce Trajcevski Song Liu
Department of Electrical Engineering and Computer Science
Northwestern University, Evanston, Illinois 60208-0834 USA
Email:{feng.lu, rjoseph, goce, song.liu}@eecs.northwestern.edu

***Abstract*—This paper addresses the problem of efficient and effective parameter variation modeling and sampling in computer architecture simulations. While there has been substantial progress in accelerating simulation time for circuit designs subject to manufacturing variations, these approaches are not generally suitable for architectural studies. Toward this we investigated two complementary avenues: (1) adapting *low-discrepancy* sampling methods for use in Monte Carlo architectural simulations. We apply techniques previously developed for gate-level circuit models to higher level component models and in so doing drastically reduce the number of samples needed for detailed simulation; (2) applying *multi-resolution analysis* to appropriately decompose geometric regions of a chip, and achieve more effective description of parameter variations without increasing computational complexity. Our experimental results demonstrate that the combined techniques can reduce the number of Monte Carlo trials by a factor of 3.3, maintaining the same accuracy while significantly reducing the overall simulation run-time.**

I. Introduction

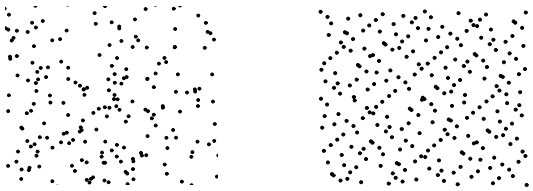
In both architecture and design automation communities, large scale Monte Carlo simulations are widely used to investigate the probabilistic impacts of manufacturing variation [1]. These variations follow complex, random behavior and influence the behavior of circuits and architectures in profound manner, limiting the applicability of analytical models and steering researchers toward Monte Carlo simulation. Typically, a single Monte Carlo experiment consists of generating hundreds or thousands of random parameter variation scenarios and simulating either a circuit or processor design under each of those scenarios. However, the total simulation cost for many parameter variation studies can be enormous. In the realm of architecture, each Monte Carlo simulation would require running a detailed architecture simulator for anywhere from one hundred million to one billion instructions – a task which may take hours. Given that most recent studies in the architecture community may incorporate ten or more individual benchmark programs [2]–[4], the full set of Monte Carlo simulations may require thousands of compute-hours.

If left unaddressed, the burdensome architectural simulation time associated with parameter variation studies may have adverse impacts. At the risk of reducing simulation accuracy, researchers may choose to use fewer Monte Carlo samples, simulate a smaller window of program execution (e.g. 10 million instead of 100 million instructions), select a faster but cruder and less detailed simulator model, or subset the benchmark suite. As previous work has shown (e.g., [5], [6]), corner-cutting in the name of reducing simulation time can have disastrous effects on accuracy of architecture studies and in some extreme cases may draw researchers to incorrect conclusions.

978-3-9810801-7-9/DATE11/©2011 EDAA

While there have been successful attempts to reduce Monte Carlo simulation time in the circuit domain, these approaches cannot directly be applied to architecture [7]. The circuit approaches attempt to reduce the simulated samples while retaining the same statistical properties. In particular, circuit-level studies assume knowledge of circuit structure and model variation at the gate level while architecture studies are at a much higher semantic level and investigate designs with billions of gates. We address this problem by bridging this semantic gap and making the approach scalable to architecture.

At the heart of the motivation for this work is the observation that significant gains in the efficiency of variation-aware architecture simulation can be achieved if better sampling methodology for parameter variation is accommodated. Specifically, we postulate that we can reduce the number of samples needed to achieve statistically sound results if we use sequences that are guaranteed to give faster convergence than Monte Carlo. To do this we must bridge a gap in understanding between circuit and architecture. We adapt several existing circuit-level techniques to make them suitable for this domain and introduce several novel approaches that further improve simulation efficacy. The main contributions of this paper can be summarized as follows:
(1) *Adapting Low-Discrepancy Sampling Methods to Architectural Simulation:* Low-Discrepancy (LD) sampling techniques generate quasi-random samples defined to have lower integration errors than true Monte Carlo sequences [8]. By implementing low-discrepancy techniques into variation map generation, the sample space of parameter variation can be covered by fewer samples relative to Monte Carlo sampling approaches. This efficient sampling methodology leads to large reduction in architectural simulation time.
(2) *Introducing Multi-Resolution Grid Maps:* To better represent sensitive geographic regions of the chip, we divide it into a non-uniform grid. For processor components that are more sensitive to the parameter variation, we assign a finer grid resolution, and apply coarser granularity to those components which are less sensitive. In total, we improve the effectiveness and efficiency of the parameter variation representation, while maintaining the same overall complexity of representation.
(3) *Comprehensive Experimental Evaluation:* We implement and evaluate the proposed methodologies. Our results demonstrate



(a) Standard Monte Carlo (b) Quasi-Monte Carlo

Fig. 1. Comparing 2D sequences generated with standard Monte Carlo and Low-Discrepancy techniques. The two examples have an equal number of points.

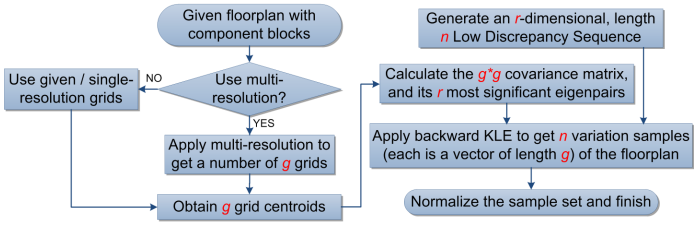


Fig. 2. Process flow for generating n LD variation maps

that for the selected microarchitecture level timing error and leakage power estimation, the low-discrepancy sampling and multi-resolution grid model give at least 3.3× faster convergence than Monte Carlo sampling.

The rest of this paper is structured as follows. In Section II we recollect some necessary background for understanding the proposed techniques, which we elaborate upon in Section III. Section IV details the quantitative benefits of our work by presenting the results of our experimental evaluations. Section V concludes the article and outlines directions for future work.

II. Background

Predicting the impact of manufacturing variation on circuit and architecture designs has become a challenging and increasingly important task for several reasons [1], [9]. The fabrication process introduces prominent variations to the threshold voltage, V_{th}, and the effective gate length, L_{eff} of transistors [1], [10], [11]. These parameter variations include both true random components which are independent and systematic components that are a function of the chip geometry and exhibit complex correlation patterns [12]. Modeling and simulation approaches must correctly account for the way that the parameter variation impacts circuit delay while capturing the spatial correlations.

Due to the probabilistic nature of manufacturing variations, and the complex interactions between transistor parameters, stochastic methods including Monte Carlo(MC) experiments based on repeated trials have become powerful tools for studying the consequences of parameter variation and developing architectural and circuit innovations to counter them [3], [13]. At a high-level, the approach consists of generating two-dimensional fields which represent random parameter variation which obey the before mentioned statistical properties and then running detailed simulation for each one of these scenarios. For gate-level Statistical Static Timing Analysis (SSTA), a natural way to model spatially correlated parameter variation is with a correlation matrix which captures the statistical relationship between every pair of transistors in the circuit [7]. Many random parameter fields can then be generated using this correlation matrix as a starting point. In the case of SSTA, SPICE simulations are run with each field sample. Since the total number of samples needed to guarantee convergence can be quite large, the number of MC trials becomes the biggest factor in simulation run time.

At the circuit-level, some innovative sampling techniques have been able to drastically reduce this factor and improve simulation runtime. Singhee *et al.* [7], [14] recognized that with conventional random field generation, Monte Carlo techniques require many samples to guarantee convergence because its accuracy obeys a $O(n^{-0.5})$ proportionality with sample set size n [15]. They further noticed that comparing to true random sequences, some classes of Quasi-Monte Carlo sequences with the same number of samples have better coverage for the sample space, hence give faster convergence. In particular, *Low-Discrepancy* (LD) methods are known to generate high quality deterministic patterns that are guaranteed to give approximately $O(n^{-1})$ [8] convergence, a significant improvement over random sequences. Figure 1 illustrates the difference between the coverage natures of conventional Monte Carlo points and low-discrepancy points in a two-dimensional space. The conventional Monte Carlo samples show both clusters and sparse regions while the low-discrepancy samples give much better coverage of the space. One can imagine extending this concept to higher dimensional spaces where each dimension might represent a physical factor (e.g. V_{th} for each transistor in a circuit).

However, low-discrepancy sequences alone cannot replace Monte Carlo sequences in generating parameter variation samples for even small circuits. For a design with n gates, one would need to generate a low-discrepancy sequence with dimensionality of n. Current best low-discrepancy sequence generators offer practical advantage over standard Monte Carlo sequences only in the early r dimensions ($r \leq 12$ [16]). Consequently, for efficient parameter variation modeling of circuits, we apply the Karhunen-Loeve Expansion (KLE) [17], a model simplification technique similar to Principle Component Analysis (PCA) [7]. Recall that a correlation matrix can be used to represent gate-level variations across the chip. This serves as a very precise description for high-dimension model of the chip. The first r ($r \sim 25$) components of KLE, composed of the r-dimenional random (or quasi-random) sequence and the r most significant eigenpairs of the correlation matrix, is an accurate estimate of KLE [7]. This effectively reduces a large number of correlated variables – in this case transistor parameters which are geometrically correlated – into a much smaller number of values and hence lower dimensionality. With a drastically reduced representation of the parameter variation, low-discrepancy techniques can be safely applied to reduce the number of required samples.

III. Variation Modeling and Sampling

As described in the previous section, Quasi-Monte Carlo sampling methods have been applied to accelerate gate-level SSTA simulations under parameter variation [7], [14] in the circuit domain. Designs with $\Omega(10^4)$ gates are evaluated for these studies, where spatial correlations of these gates can be captured in a correlation matrix of tractable size. However, these techniques do not directly scale to architectural simulations for a few reasons. First, gate-level descriptions of most modern processors are unavailable for academic researches where they

are obviously available for circuit-level designs. Second, even if a complete processor could be modeled at the gate-level, the netlist of the design, which may contain hundreds of millions to billions of gates, exceeds the capacity of existing gate-level algorithms which have $O(n^2)$ spatial complexity for n gates. Finally, most computer architects work on a higher and more abstract level, and architectural simulations aim for more complex and comprehensive evaluations for the system. For example, recent work at the architecture level has examined whole-chip leakage power and timing error rates as functions of parameter variations [1]. These studies must include program state and microarchitecture-level models that are fundamentally different from transistor-level simulations in SSTA.

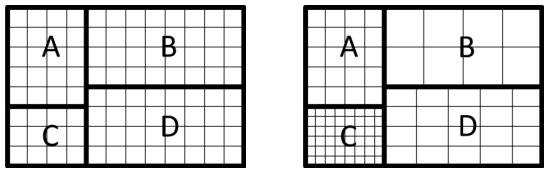
To address these challenges, our proposed techniques have special considerations for architectural variation simulations. First, instead of gate-level, we model parameter variations at block/grid level. Grid size here poses a tradeoff between the computational complexity and the modeling accuracy. Second, within the processor each structural block will have its own susceptibility to and distinct behavior under parameter variation. We leverage the fact that some components may have a greater overall impact on the system than others and introduce *multi-resolution modeling* of parameter variation. Figure III gives a process flow for generating n LD samples. In the following section, we first demonstrate how to model a block/grid variation map with Quasi-Monte Carlo methods. Then, we discuss algorithms that generate grid structures with the best accuracy-complexity tradeoffs.

A. Compact Systematic Variation Representations

Our parameter variation modeling approach assumes a high-level physical model for microarchitectural components nominally described via a floorplan. Depending on the application, one may choose to model structures within a single processor pipeline, or cores and caches in a many-core chip. Given this floorplan, we represent the physical variation of parameters such as L_{eff} and V_{th} for diverse usages and abstractions. Either block-based variation model is applied, where we assume the parameter within each component is a constant and use its centroid for correlation calculation, or we further decompose the blocks into regular grid regions and generate variation samples with finer granularity. Note that, although block level models may lose some accuracy comparing to grid level models, they are still acceptable models for certain architectural study [1].

B. Implementing Quasi-Random Samples

Figure III shows how to generate our Quasi-Monte Carlo samples. First, given the block floorplan and grid resolution, the correlation matrix is calculated for KLE decomposition. To maintain consistency with [1], here the matrix concerns purely the covariance factor between grid regions. This differs from the circuit-level approach in [7] where the correlation factors are normalized by grid area. Second, there are many possible methods for constructing LD sequences. We select Niederreiter's sequence, which has been proved to have less integration error [18] than Sobol's sequence which was used by [7], [14]. This LD sequence is then combined with KLE to generate an original set of systematic parameter variation samples. Finally, before



(a) Single-Resolution grids. (b) Multi-Resolution grids.

Fig. 3. The illustration of SR and MR grids distributed over a 4-block floorplan. Both figures are with the same number of grids.

"publishing" the sample set, we adjust the set to improve the sample space coverage. Systematic variation is supposed to have a statistical mean μ_{var} of zero and a specified standard deviation σ_{var} (according to this parameter's given μ and σ/μ [1]). For each block/grid region i, we apply linear normalization to its variation values across all samples, so that $\mu_{var,i} = 0$ and $\sigma_{var,i} = \sigma_{var}$. After doing this, the KLE-based LD variation sample set are well positioned within the targeted statistical range.

C. Enhancing Localization with Multi-Resolution Analysis

We make another observation relevant to microarchitectural parameter variation studies, namely that some components of the processor are known to be more sensitive to variation than others. In this paper, we apply this to evaluate two important architecturally relevant component properties that are strong functions of parameter variation: timing error rate P_e and leakage power P_{leak}. In an era where architects are considering timing speculation as a way to improve performance and efficiency, timing error rates are important properties of a design [2], [3], [13]. In deep submicron technology, leakage power comprises a significant portion of total chip power and therefore serves as an essential design characteristic.

We first consider P_e, modeling an n-stage pipeline as a series failure system. The total P_e can be represented as a weighted summation of the error rate of each pipeline stage i:

$$P_e = \sum_{i=1}^{n} (\alpha_i \times P_{e_i}) \tag{1}$$

α_i is the activity factor of block i. Intuitively, pipeline stages which have either high activity factors or error rates P_{e_i} are more likely to produce timing errors and will have a greater impact on total error rate. Activity factors are a strong function of program characteristics (e.g. floating-point applications with have high activity factors for their FP execution units while integer programs will not) and in many cases activity magnitudes can be predicted before simulation. We now consider P_{leak}. Chip-wide leakage power can be seen as the integration of the leakage power of each component i:

$$P_{leak} = \sum_{i=1}^{n} P_{leak_i} \tag{2}$$

Leakage for a component depends on both the temperature of that block and its area. Since area is known a priori and temperature is dependent on activity, we can reasonably ascertain which component blocks are likely to be dominant. As two of the more important characteristics of a processor under parameter variation, both of timing error rate and leakage power are in the form of $f = \sum_{i=1}^{n} f_i$. Let f_0 and f_{i_0} denote the true values of f and f_i, to optimize the estimation of f, we need to minimize the estimate error ε:

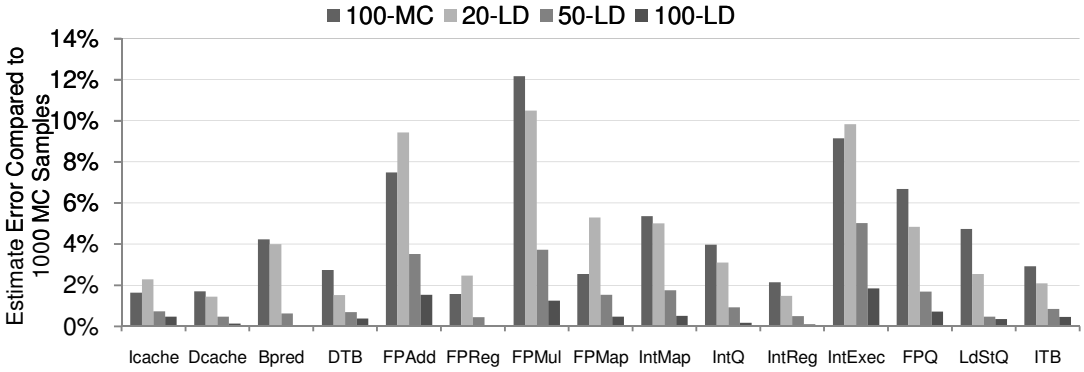


Fig. 4. The estimate error of P_e relative to 10,000 MC samples: for 15 cpu blocks, 100 MC samples, 20 LD samples, 50 LD samples and 100 LD samples

$$\varepsilon = |f - f_0|/f_0 = |\sum_{i=1}^{n}(f_i - f_{i_0})/f_0| = |\sum_{i=1}^{n}(\varepsilon_i \times f_{i_0}/f_0)| \quad (3)$$

Equation 3 implies that for blocks with larger f_{i_0}, the estimate error ε_i needs to be smaller to minimize the total error. Hence in this work, we introduce *Multi-Resolution* (MR) variation sampling, in which the on-chip parameter map is composed of blocks with varying grid density. The total number of grids points G are distributed to each block i following the rule

$$G_i = (f_{i_0}/f_0) \times G, \quad (4)$$

which intuitively means that the grid density within one block is proportional to the "function" density within it, which we can obtain from nominal emperical results. Figure 3 illustrates this idea with both Single-Resolution (SR) and MR grids, where block C has the greatest density of the targeting function and block B has the least. As experimental results show, for identically sized parameter maps, MR samples converge faster than SR samples.

We conclude this section with a note that, combining the LD and MR techniques, generating 1,000 samples typically takes several seconds to a few minutes on a standard Linux desktop system. The sample generation time is therefore negligible when compared to the detailed simulation time which follows.

IV. Evaluation

Our Quasi-Monte Carlo and Multi-Resolution variation models are suitable for examining the impact of parameter variation on many aspects of a microarchitecture. In this section, we evaluate our variation model and sampling methodology by applying it to two aspects of high-performance processor design which are extremely sensitive to parameter variation: (1) timing errors associated with timing speculative architectures [2] and (2) chip leakage power. Our first application examines trade-offs in observed timing errors versus clock frequency and compares convergence rates of timing error rates under low-discrepancy sequences versus standard Monte Carlo samples. In the second application, we examine the on-die leakage power variations with both SR-LD sampling and MR-LD sampling comparing to MC. Both applications are compared against VARIUS [1] Monte Carlo samples as a baseline case, which has been widely adaped for architectural parameter variation sampling [2], [4].

For the timing error estimation, we use the VARIUS timing model. It adopts the Alpha-Power Law [19] to relate threshold voltage V_{th} and effective gate length L_{eff} to gate delay:

$$T_g \propto \frac{L_{eff}V}{\mu(V - V_{th})^{\alpha}} \quad (5)$$

where V is the supply variation, the μ is carrier mobility and α is an empirically derived constant. The gate delay is then used to estimate the timing error rate for logic and memory structures under process, voltage, and temperature variations. For leakage, we apply the HotLeakage [20] model which suggests that

$$I_{leak} \propto (\frac{kT}{q})^2 e^{q\frac{-V_{th}-V_{off}}{\eta kT}}, \quad (6)$$

and the leakage power is proportional to I_{leak}. A factor of the total leakage power across the chip can be obtained by an integration of Equation 6, where k is the Boltzmann Constant, q is the electron charge, and V_{off} and η are empirically determined parameters. We adopt these parameters from [1], [21] and [22] and scale them to 32nm technology.

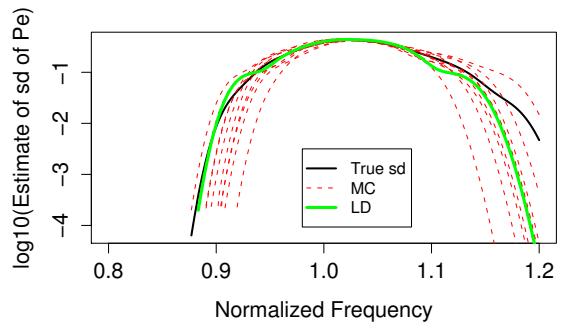
We model a single core design featuring an Alpha 21264 processor scaled to a 32nm technology and use a floorplan detailing the microarchitectural structures of this design. In our experiments, we model random and systematic variation. A spherical correlation model [1] is used for all the variation samples. We assume V_{th} and L_{eff} are highly correlated [1], and use identical systematic variation samples for the two parameters. Our models apply σ/μ of parameter variation, nominal supply and threshold voltage, and the decomposition of systematic and random components which follow that of [1]. These parameters are suitable for modeling high-performance designs in a 32nm technology.

A. Low-Discrepancy Variation Samples

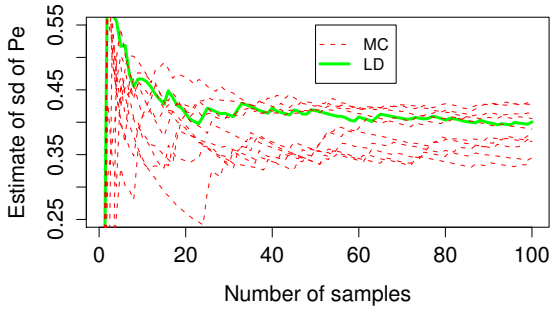
To evaluate the effect of low-discrepancy sampling, we apply block-based LD variation samples to the VARIUS [1] timing error model, and estimate the distribution of the resulting timing error rates P_e for all the pipeline stages of a processor floorplan under a sequence of clock frequencies.

For comparison, the process is repeated with several sets of VARIUS Monte Carlo samples. The results of a large Monte Carlo set with 10,000 samples are used as a gold standard. This is a sample set size sufficiently large that sample mean and variance are very close to true distribution mean and variance. Note, that these sample sizes are prohibitively large for most simulation studies – they represent a best case result.

Although [4] suggests that 100 Monte Carlo samples show enough convergence when applying to VARIUS timing error



(a) The estimate of *sd* over a normalized frequency set



(b) The estimate of *sd* converges with increasing run size

Fig. 5. The estimate of P_e's standard deviation and its convergence for Icache: Comparing 1 LD to 10 MC runs with (a) Fixed run size of 100 samples. (b) Fixed clock frequency at 1.0.

model, our experiments show that on average any group of 100 MC samples still have considerable error when compared to the gold standard. On the contrary, Low-discrepancy samples produce high fidelity results. Figure 4 presents the error of 100 MC, 20 LD, 50 LD and 100 LD samples relative to 10,000 MC samples when estimating the mean of P_e of each processor pipeline component. For 10 out of 15 components, 20 LD samples have better accuracy than 100 MC samples, and 50 LD samples outperforms 100 MC on all components. One can view this in an alternative way. With the same number of samples, 100 LD gives an accuracy at least 75% better than 100 MC. This experiment proves that LD samples converge much faster than MC, which translates to either significant reduction of samples needed or better accuracy with the same number of samples. Since generating LD samples is a deterministic process, the shown results are repeatable and consistent. MC trials in contrast produce dramatic fluctuations for different runs and hence do not guarantee fast convergence.

The LD estimate of the standard deviation (*sd*) also shows faster convergence. Due to space limit, we only show in Figure 5 the estimate and convergence of standard deviation for the Icache block. The two sets of curves intuitively show the difference between the natures of LD and MC sampling, and are consistent with our expectations. In summary, low-discrepancy techniques allow much faster convergence, resulting in large reductions in sample set size.

B. Low Discrepancy, Multi-Resolution Variation Samples

Now we evaluate the sampling of low discrepancy and multi-resolution grids, and we do this by estimating the deviations in chip leakage power. Multi-resolution analysis allows us to

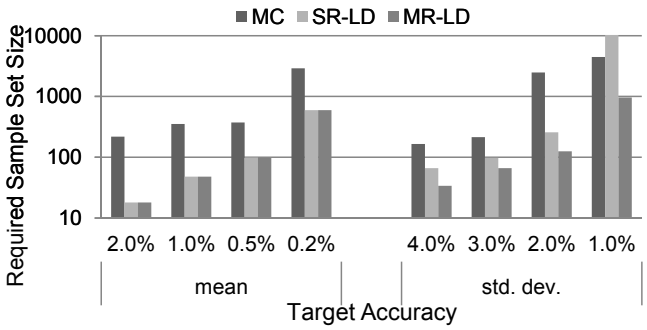


Fig. 6. The number of samples needed for targeting accuracy when estimating chip leakage power: MC, SR-LD and MR-LD.

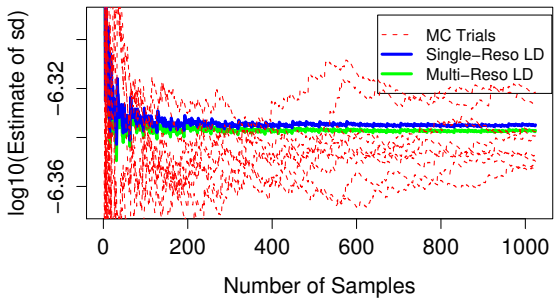
configure grid granularity within a component block according to its importance. For this study, we focus on leakage power and make grid densities proportional to power densities for all blocks, as explained in Section III. Power density of a block is determined by its temperature when with nominal V_{th0}, and we use the temperature distribution from [23] for the processor floorplan. After distributing the grid resolution we generate the MR-LD variation samples with KLE-based LD methods.

We generate a set of MC, SR-LD and MR-LD samples for comparison. For the three different modeling methodologies, all samples are with the same sized parameter map (25×25), and the resulting leakage estimates are compared to that of a gold standard, 10,000 MC samples of resolution 50×50.

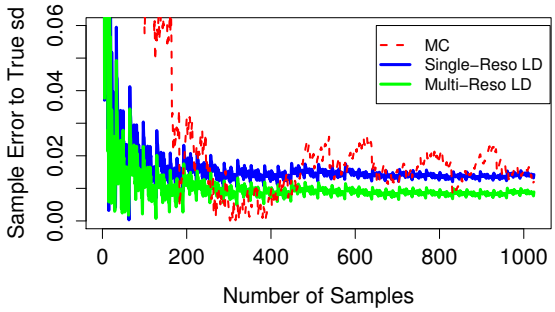
Figure 6 shows the number of samples needed to achieve the targeting accuracies. For the mean, the LD samples show at least $4\times$ faster convergence than MC. However, SR-LD and MR-LD do not have significant difference themselves, which could be possible because the estimate errors of the mean are already low. We note that, although not shown in the figure, the average error of MR-LD is 0.4% smaller than SR-LD. For the standard deviation, LDs still converge faster than MC, and at the same time, MR-LD outperforms SR-LD, with speedup of at least $3.3\times$ and $2.2\times$ respectively. Figure 7 presents more intuition for the estimate of *sd* as the number of samples grows (for clarity only until 1000 samples are shown), which leads to the observation that MR-LD converges to a better accuracy than SR-LD. Considering the fact that the difference between the computational efforts of implementing single-resolution and multi-resolution models is only distributing the grids with different density, the potential of the multi-resolution model is attractive, especially when accuracy is critical.

V. Conclusions and Future Work

In this work, we introduced a collection of techniques to help computer architects rethink the parameter variation model and improve sampling methodology when applying Monte Carlo simulations. Our key contributions were: (1) to develop spatial variation representations that could be applied to study architectural components while leveraging properties of the low-discrepancy and (2) to introduce multi-resolution models that adapt grid resolution to suit the relative importance of a component. We evaluated our techniques using a series of Monte Carlo experiments and found that in most cases our improved modeling and sampling methodology can dramatically reduce the number of samples needed to achieve convergence.



(a) The estimate of *sd*



(b) The sample error to the true *sd*

Fig. 7. The estimate and the sample error of the standard deviation of the leakage factor distribution with increasing sample set size: comparing MR-LD, SR-LD and standard MC.

As one of the most straightforward ways to decompose parameter variation, Karhunen-Loeve Expansion (KLE) is adapted for the spatially correlated parameter variation model. However, KLE is still Fourier-like, meaning that each orthogonal term in the decomposition captures the information across the whole spatial domain. Considering the target of the entire processor where the pipeline stages' characteristics differ from each other, there might be other ways to decompose the parameter variation while taking the differences between different stages into consideration. One possible way is *wavelet decomposition*, in which each term localizes one specific part of the domain, and hopefully this could lead to a better approach to represent the different variation scenarios in different pipeline stages.

We evaluated our Multi-Resolution approach by distributing the grid densities proportional to the target function densities. While sharing a similar motivation as the multi-level grid files from database research [24] used for selectivity estimation, in this work we have a slightly different context. Namely, the coarseness of the resolution is varied based on the sensitivity to variations. For future work, we would like to further investigate the problem of dynamic fine-tuning of the grid-map and sample generation, in reaction to some (observed) changes in the parameters variation and component activity factors that may affect the validity of the experiments. Towards that, we will try to apply some of the techniques for streaming data management [25] in our context.

Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful comments. This work is in part supported by NSF grants CAREER CCF-0644332, CNS-0720820 and CNS-0910952.

References

[1] R. Teodorescu, B. Greskamp, J. Nakano, S. Sarangi, A. Tiwari, and J. Torrellas, "Varius: A model of parameter variation and resulting timing errors for microarchitects," *IEEE Trans on Semiconductor Manufacturing*, 2008.

[2] S. Sarangi, B. Greskamp, A. Tiwari, and J. Torrellas, "EVAL: Utilizing processors with variation-induced timing errors," in *IEEE/ACM Int. Symp. on Microarchitecture*. IEEE Computer Society, 2008.

[3] B. Greskamp, L. Wan, U. R. Karpuzcu, J. J. Cook, J. Torrellas, D. Chen, and C. Zilles, "Blueshift: Designing processors for timing speculation from the ground up," in *Int. Symp. on High-Performance Computer Architecture*, 2009.

[4] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating parameter variation with dynamic fine-grain body biasing," in *Int. Symp. on Microarchitecture*, 2007.

[5] D. Citron, "MisSPECulation: partial and misleading use of SPEC CPU2000 in computer architecture conferences," in *Int. Symp. on Computer Architecture*. ACM, 2003.

[6] J. J. Yi, R. Sendag, D. J. Lilja, and D. M. Hawkins, "Speed and accuracy trade-offs in microarchitectural simulations," *IEEE Trans. on Computers*, 2007.

[7] A. Singhee, S. Singhal, and R. Rutenbar, "Practical, fast monte carlo statistical static timing analysis: Why and how," in *Int. Conf. on Computer-Aided Design*, 2008.

[8] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, 1992.

[9] C. S. Amin, N. Menezes, K. Killpack, F. Dartu, U. Choudhury, and N. H. andYehea I. Ismail, "Statistical static timing analysis: How simple can we get?" in *Design Automation Conference*, 2005.

[10] T. Karnik, S. Borkar, and V. De, "Probabilistic and variation-tolerant design: Key to continued moore's law," 2004.

[11] E. Humenay, D. Tarjan, and K. Skadron, "Impact of parameter variations on multi-core chips," in *Workshop on Architectural Support for Gigascale Integration*, 2006.

[12] 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 *Int. Symp. on Quality Electronic Design*, 2005.

[13] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner1, and T. Mudge, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Int. Symp. on Microarchitecture*, 2003.

[14] A. Singhee and R. Rutenbar, "From finance to flip flops: A study of fast quasi-monte carlo methods from computational finance applied to statistical circuit analysis," in *Int. Symp. on Quality Electronic Design*, 2007.

[15] J. Kiefer, "On large deviations of the empirical d. f. of vector chance variables and a law of the iterated logarithm," *Pacific Journal of Mathematics*, 1961.

[16] P. Bratley, B. Fox, and H. Niederreiter, "Implementation and tests of low-discrepancy sequences," *ACM Trans. on Modeling and Compurter Simulation*, 1992.

[17] M. Loeve, "Probability theory," 1977.

[18] T. Davies and R. Martin, "Low-discrepancy sequences for volume properties in solid modelling," in *CSG '98 Conference*, 1998.

[19] T. Sakurai and A. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverterdelay and other formulas," *IEEE Journal of Solid-State Circuits*, 1990.

[20] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan, "Hotleakage: A temperature-aware model of subthreshold and gate leakage for architects," University of Virginia, 2003.

[21] L. Zhang, L. S. Bai, R. P. Dick, L. Shang, and R. Joseph, "Process variation characterization of chip-level multiprocessors," in *Design Automation Conference*, 2009.

[22] W. Zhao and Y. Cao, "New generation of predictive technology model for sub-45nm design exploration," in *Int. Symp. on Quality Electronic Design*, 2006.

[23] Y. Han, I. Koren, and C. A. Moritz, "Temperature aware floorplanning," in *Workshop on Temperature-Aware Computer Systems*, 2005.

[24] K.-Y. Whang, S. Kim, and G. Wiederhold, "Dynamic maintenance of data distribution for selectivity estimation," *VLDB Journal*, 1994.

[25] I. Sharfman, A. Schuster, and D. Keren, "A geometric approach to monitoring threshold functions over distributed data streams," in *SIGMOD Conference*, 2006.