Coping with Parametric Variation at Near-Threshold Voltages

Near-threshold voltage computing (NTC) promises improved energy efficiency but is more sensitive to parametric variation than conventional, super-threshold voltage computing (STC). In this article, the authors introduce an architectural model of parameter variation for NTC, show the shortcomings of adapting state-of-the-art STC techniques for variation mitigation to NTC, and discuss how to tailor variation mitigation to NTC.

Ulya R. Karpuzcu
University of Minnesota

Nam Sung Kim
University of Wisconsin-Madison

Josep Torrellas
University of Illinois at Urbana-Champaign

Contemporary scaling suffers from a growing power density with each technology generation. Although we can cram more devices onto a chip, cooling limitations prevent the power budget from expanding. As a result, the gap between what we can integrate on a chip and what we can operate simultaneously widens at every generation.[1]

A promising way to engage more cores in the computation is to operate at a lower supply voltage (V_{DD}). Lowering V_{DD} to slightly above the threshold voltage (V_{th}) reduces the energy per operation substantially.[2-4] This unconventional operation regime, called near-threshold voltage (NTV) computing (NTC) lets many more cores operate under a given power envelope.

Operating at near-threshold V_{DD} results in frequency (f) degradation. In many throughput-oriented applications, we can retain high performance by increasing the core count engaged in the computation. Because power consumption decreases more from operating at low V_{DD} than it increases from operating more cores, the result is a net power savings.[2]

Unfortunately, at NTV, devices have a higher susceptibility to parametric variation—that is, the device parameters' deviation from their nominal specifications. Variation produces not only slower and leakier cores, but also substantial differences in core speed and power across the chip. Variation particularly impacts on-chip memory because memory devices are sized more aggressively to satisfy high-density requirements. Moreover, memory structures are more sensitive to device mismatch.

Relying on the worst-case operating margins isn't practical at NTV, where the nominal f is already low. A further difficulty stems from the diminishing efficacy and increasing cost of state-of-the-art variation mitigation techniques for conventional, super-threshold voltage (STV) operation when adopted at NTV. These techniques rely heavily on V_{DD} tuning in independent V_{DD} domains on chip. Fine-grained V_{DD} tuning would appear to suit NTC well, because fine-grained V_{DD} domains can track the intensified within-die variation, and at NTV, small changes in V_{DD} have a larger impact on performance and power.

Published by the IEEE Computer Society

Unfortunately, state-of-the-art on-chip V_{DD} regulation is energy inefficient,[5] barely tolerable in the energy-conscious NTC environment.

Without addressing variation, we can't unlock NTC's potential. In this article, we confront variation by introducing an architectural model of parametric variation at NTV. We use the model to show the shortcomings of adopting state-of-the-art STC techniques for variation mitigation at NTV. We also examine how to tailor variation mitigation to NTC, using a single-V_{DD}-domain many-core organization called EnergySmart.

Background

In this section, we cover the basics of NTV operation and the impact of parametric variation at NTV.

NTV operation basics

For the current technology, the NTC V_{DD} is about 0.5 V, while the STC V_{DD} is about 1 V. Figure 1 depicts the power, *f*, and energy per operation as a function of V_{DD}.[3,6] At NTC, the energy per operation improves by 2 to 5× over STC, at the expense of a 5 to 10× *f* degradation. As a result, power reduces by 10 to 50×, which enables more cores to fit into a given power budget.

The minimum power and energy-per-operation points fall into the subthreshold regime ($V_{DD} < V_{th}$), where *f* degrades significantly. The STV regime accommodates the maximum *f*, at the cost of notably higher power and energy per operation. The NTV regime is a sweet spot, with power savings closer to subthreshold but with *f* closer to STV. Away from NTV, higher V_{DD} leads to substantially higher power, and lower V_{DD} to substantially lower *f*.

Impact of parametric variation at NTV

Parametric variation is the discrepancy between the device parameters' design specification and the actual values. It is caused by manufacturing imperfections, and gets worse as feature sizes shrink. Within-die (WID) variation has a systematic and a random component. The former is typically caused by lithographic irregularities, while the latter is caused by varying dopant concentrations.

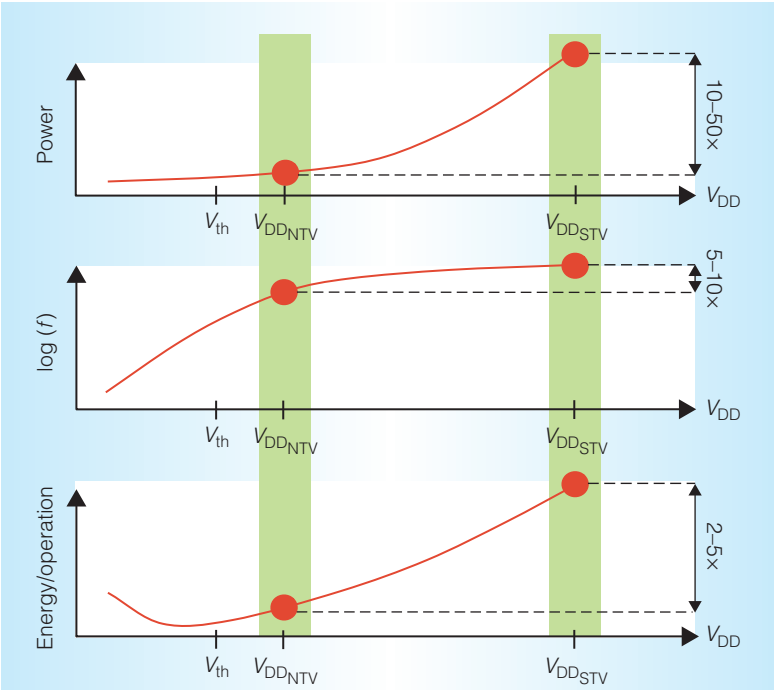

Figure 1. Power, frequency (*f*), and energy per operation as a function of supply voltage (V_{DD}).[3,6] At near-threshold voltage computing (NTC), the energy per operation improves by about 2 to 5× over super-threshold voltage computing (STC), at the expense of a 5 to 10× *f* degradation. (V_{DD}: supply voltage; V_{th}: threshold voltage.)

A processor or memory block's *f* of operation and power consumption depends on two key parameters vulnerable to variation: V_{th} and effective channel length (L_{eff}).

In a many-core setting, WID variation in V_{th} and L_{eff} widens the spread of the cores' (and memory blocks') *f* distributions. This results in a lower operating *f*, because the distribution tail determines the *f*. Moreover, WID variation in V_{th} increases the static power consumption because devices with lower V_{th} consume more than devices with higher V_{th} save. The higher the variation in V_{th} and L_{eff}, the higher the variation in *f* and power across cores and memories.

NTV operation intensifies the susceptibility to parametric variation. As V_{DD} gets close to V_{th}, the transistor's switching speed becomes more sensitive to variation. As a result, the timing guardband required to tolerate a fixed amount of V_{th} variation grows as V_{DD} decreases. This is shown in Figure 2, from Chang et al.[2]

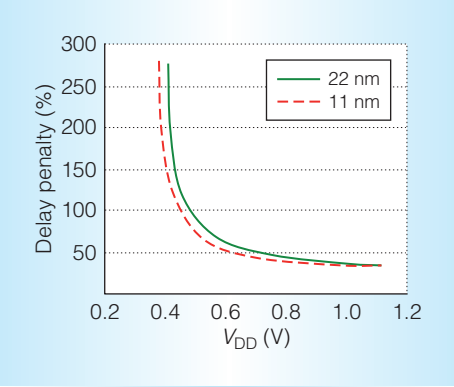

Figure 2. Evolution of variation-induced timing guardband with V_{DD}.[2] NTC intensifies the susceptibility to parametric variation.

The overall effect is higher f variation at NTV and, thus, higher dynamic power variation.

As V_{DD} gets closer to V_{th}, both dynamic and static power decrease. However, they do so at a different pace: static power reduces less. Thus, the share of static power increases. Because static power depends exponentially on V_{th}, V_{th} variation also causes substantial power variation at NTV.

Characterizing parametric variation at NTV

Quantitative characterization of the impact of variation is critical for the design of NTV-specific variation mitigation techniques. Consequently, we developed Varius-NTV,[7] an architectural variation model for NTC. The model is based on the Varius model for STC.[8] Varius-NTV addresses four main limitations of Varius when adopted at NTV.

- The Varius performance model relies on the alpha-power law, which isn't accurate in the NTV regime.
- The Varius memory model uses a six-transistor (6T) static RAM (SRAM) cell that can't reliably operate at NTV.
- The Varius memory model only considers read timing errors, while other memory error modes dominate at NTV.
- The Varius memory model neglects the leakage current's impact in the memory timing analysis, which becomes substantial at NTV.

To obtain the profile of V_{th} and L_{eff} values on a variation-afflicted chip, Varius-NTV proceeds as follows. To generate these parameters' systematic component, Varius-NTV superimposes a grid on the chip floorplan and samples a Gaussian distribution for each grid point. Because the systematic component exhibits spatial correlation, these sampled values follow a spatial correlation function. Varius-NTV uses a spherical function, wherein the correlation between the values of two points depends only on the distance between them.[7] Because the random component doesn't have spatial correlation, Varius-NTV captures random variation analytically. We then combine the systematic and random components.

We apply the resulting V_{th} and L_{eff} profiles to the Varius-NTV performance model derived from Markovic et al.[4] The result is the f supported on a per-core and memory-block basis. From the V_{th} and L_{eff} profiles, we also extract the minimum operating V_{DD} for memories. The V_{th} and L_{eff} profiles, combined with the V_{DD} and f distributions, generate the values of the static and dynamic power consumption in cores and memory blocks.

Varius-NTV uses an eight-transistor (8T) SRAM cell.[9] This cell can operate robustly at NTV because we can independently optimize the read and write paths with a marginal cell-area increase. In the classical 6T cell, read optimizations conflict with write optimizations.[10]

Varius-NTV also provides the variation-induced error rates as a function of the operating V_{DD} and f. These errors are timing errors for logic, and timing, stability, and hold errors for memories. A logic timing error occurs if variation slows down logic so that it fails to meet timing at the designated f. For memories, Varius-NTV covers various types of errors:

- read (and write) timing errors, which occur if a read (or write) isn't performed within the designated duration owing to variation-induced slowdown;
- a write stability error, which occurs if the cell content can't be modified

Using Varius-NTV to Provide Timing Analysis

Figure A shows how to use Varius-NTV to provide timing information for a chip. Varius-NTV takes the chip floorplan and a distribution of the variation-free path delays for each pipeline stage. On the bottom left of the figure, we see the path delay distribution for one such stage. The stage can cycle at f_{NOM}. Varius-NTV has models to compute the parametric variation's effect on logic structures (Figure A, upper middle box) and memory structures (Figure A, lower middle box). The top right of the figure shows the impact of variation on the stage's path delay distribution. Some paths become faster, while others become slower. The stage's new frequency is now lower (f_{VAR}). Varius-NTV can also generate the timing error rate as a function of V_{DD} for a fixed operating frequency f (Figure A, bottom right). A higher V_{DD} results in faster paths, reducing the likelihood of timing errors.

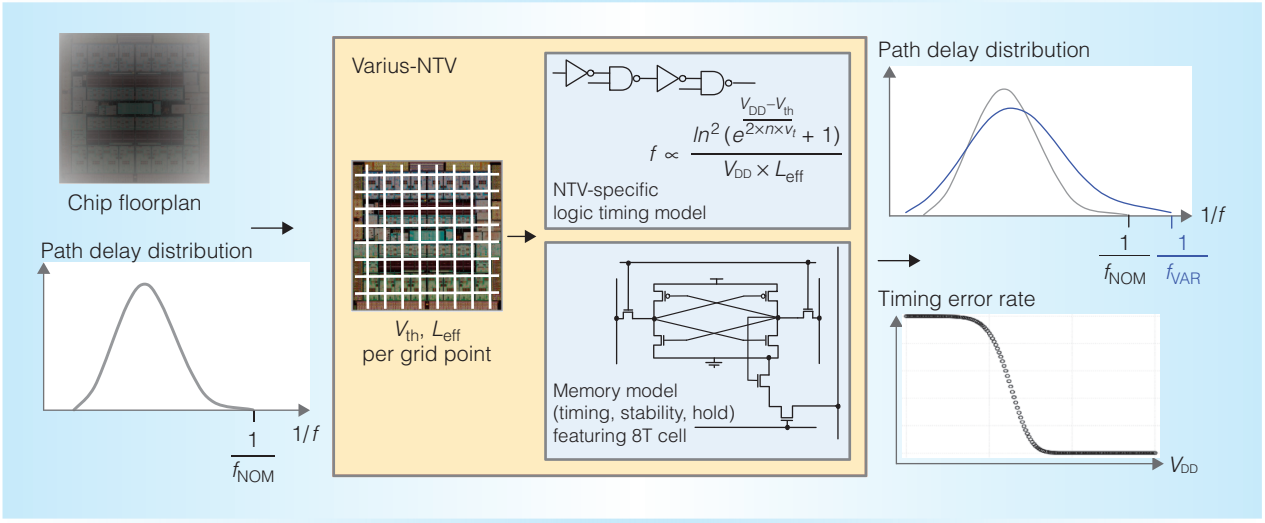

Figure A. Using Varius-NTV to provide timing information for a chip. Varius-NTV has models to compute the parametric variation's effect on logic and memory structures.

even if the write duration were to be extended to infinity; and

- a hold error, which results in the loss of the cell content due to excessive leakage under variation while the cell isn't being accessed.

The 8T cell eliminates bit-flips during reads by construction.

We can use Varius-NTV in numerous ways. For example, we can use it to generate a safe f and V_{DD} operating point for a chip or to generate on-chip distributions of f, minimum V_{DD}, or power dissipation. It can also provide a given design's timing, stability, or hold error rates. Finally, we can use it to explore the design space by having a chip conform to a target f and power budget or a target timing, stability, or hold error rate in the presence of variations. For the case of timing analysis, see the "Using Varius-NTV to Provide Timing Analysis" sidebar.

Coping with parametric variation at NTV

Addressing NTV's higher susceptibility to parametric variation is critical. Currently, designers mostly handle variation at STV using adaptive body biasing (ABB)[11] and multiple on-chip V_{DD} domains.[12] Unfortunately, these techniques exhibit diminished efficacy and higher cost at NTV.

Specifically, ABB's usefulness will likely abate in the future given the emerging device technologies. On the other hand, multiple on-chip V_{DD} domains with independent V_{DD} scaling would appear to be useful at NTV. Indeed, because WID variation becomes more significant at NTV, chip neighborhoods could benefit more from the decoupling of V_{DD} values. Moreover, at NTV, small changes in V_{DD} have a relatively larger impact on performance and energy.

However, several limitations make the use of multiple V_{DD} domains at NTV less attractive.[13] First, the power efficiency of on-chip

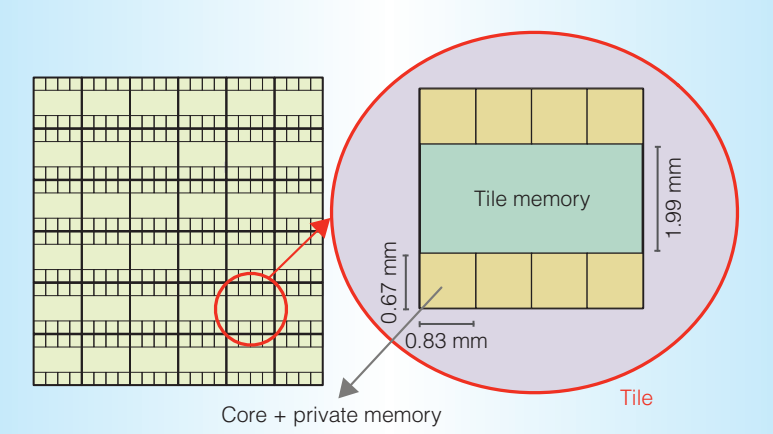

Figure 3. A hypothetical 288-core near-threshold voltage (NTV) chip at 11 nm. The chip size is about 20 mm × 20 mm. Each tile in the chip has eight cores with their own memory block, and a tile memory block.

V_{DD} switching regulators is limited—it's often in the 75 to 90 percent range for realistic operating conditions. Having V_{DD} regulators on-chip is the only way to support many domains with modest cost. Second, small V_{DD} domains are susceptible to deeper V_{DD} droops, owing to the lower averaging effects in the current that the domain draws. To guard against droop-induced errors, we need larger V_{DD} guardbands in these small V_{DD} domains. Finally, an NTV chip will be physically large, possibly containing hundreds of cores. To minimize the cost, each domain will likely include a sizable number of cores, with nontrivial within-domain variation. This will lead to a V_{DD} setting that's suboptimal for individual cores. On top of all this, on-chip switching regulators also consume substantial area and introduce design complexity.

Such limitations suggest a different type of architecture for future NTV many-core chips. Thus, we propose EnergySmart, an architecture that eschews multiple on-chip V_{DD} domains for energy efficiency.[13] EnergySmart keeps a single V_{DD} in the whole chip. It supports dynamic voltage and frequency scaling (DVFS), but applies it globally across the chip. To inexpensively handle variations, EnergySmart supports only f domains. It is organized in clusters of cores, where each cluster is potentially an f domain. With many f domains, the chip still has many degrees of freedom to tackle process variation and is simple.

To attain energy-efficient performance, EnergySmart has an effective core-to-job assignment algorithm. Thanks to the chip lacking V_{DD} domains, the assignment algorithm is simple: it must select only the f of the chosen clusters at the global V_{DD}, rather than the V_{DD} and f of all the clusters. The result is an effective algorithm.[13] In addition, we also show that the lower speed of V_{DD} changes without on-chip V_{DD} regulators doesn't hamper effective DVFS.[13]

Evaluation

To quantify the impact of variation at NTV, we simulate a hypothetical chip with 288 cores organized in 36 tiles at 11 nm (Figure 3). The chip size is about 20 mm × 20 mm. Each tile has eight cores with their own memory block, and a tile memory block. Each core is a single-issue engine where memory accesses can be overlapped with each other and with computation. The per-core memory is used as a private level-1 (L1) cache, whereas the tile memory is used as a shared level-2 (L2) cache. The chip uses a fully mapped directory-based MESI (modified, exclusive, shared, invalid) coherence protocol where each pointer corresponds to one tile. The interconnection network is a 2D torus across tiles and a bus inside each tile.

The nominal values of V_{DD} and f are 0.55 V and 1.0 GHz, which approximately correspond to 0.77 V and 3.0 GHz for STC. The Varius-NTV parameter values are $(\sigma/\mu)_{Vth} = 15$ percent, $(\sigma/\mu)_{Leff} = 7.5$ percent, and $\phi = 0.1$. We derived the technology parameters from the *International Technology Roadmap for Semiconductors* (www.itrs.net) and fine-tuned them considering industry projections for 11 nm. The power budget is fixed at 100 W.

To evaluate performance and power, we interfaced Pin[14] over a user-level Pthreads library to the SESC simulator (http://sesc.sourceforge.net). The power analysis used McPAT[15] scaled to 11 nm. We ran multiprogrammed workloads that contained the following Parsec (Princeton Application Repository for Shared-Memory Computers) applications:[16] blackscholes, ferret, fluidanimate,

raytrace, swaptions, canneal, dedup, and streamcluster. Each application ran with 8 threads in parallel and was mapped to a single tile. The complete region of interest was executed for the simsmall input data set.

Operating point under variation

To determine the operating V_{DD} and f in the presence of variation, we began by finding the minimum sustainable V_{DD} per memory block (V_{DD_MIN}). To this end, Varius-NTV first estimated the memory block's V_{DD_HOLD}, which is the minimum V_{DD} that still avoids hold errors. Varius-NTV then performed write stability analysis for the memory block, to guarantee that operation at V_{DD_HOLD} didn't cause write stability errors. After this step, Varius-NTV could select a higher V_{DD} if the write stability error rate at V_{DD_HOLD} exceeded a tolerable threshold. If the resulting V_{DD_MIN} is higher than V_{th} (a very likely scenario), then it represents the minimum possible V_{DD} at NTV for the memory block. We performed this process in all the memory blocks.

The V_{DD_MIN} for the logic blocks in a tile is the minimum V_{DD} for which there are no timing errors in the logic when operating at a target f. In our analysis, however, we imposed the highest V_{DD_MIN} of the memory blocks in the tile on the logic blocks of the tile, and set the operating f of the cores in the tile accordingly. Hence, given the V_{DD_MIN} of the memory blocks in the tile, Varius-NTV performed timing analysis to calculate the fs of memory and logic blocks in the tile. The tile's slowest path determined the highest f that all the blocks in the tile can support at the fixed V_{DD_MIN}.

Impact of variation at STV versus NTV

To characterize variation in the chip of Figure 3, we considered three types of on-chip blocks separately: logic (core pipelines), small memories (per-core local memories), and large memories (tile memories). The highest V_{DD_MIN} across all of the blocks in the chip sets the chip's operating voltage (V_{DD_OP}). For this experiment only, each block cycles at the maximum f that it can support at V_{DD_OP}. We sampled 100 chips generated by Varius-NTV.

To quantify intra-tile variations, we computed, for each tile, the f ratio of the fastest pipeline to the slowest one in the tile. We recorded the mean of this f ratio over all of a chip's tiles and called it *IntraPipe*. We repeated the same process for the local memories in the tile and called it *IntraMem*. For inter-tile variations, we measured the f ratio of the fastest tile memory to the slowest one and called it *InterMem*. We then considered the maximum f that each tile could support at V_{DD_OP} (which is the lowest f of its pipelines, local memories, and tile memory), and computed the ratio of the f of the fastest and slowest tiles. We call it *InterPipe+Mem*. For all these measures, we report the mean over 100 chips, along with 95-percent confidence intervals.

Figure 4a compares these f ratios for NTV and STV. The f ratios are significantly higher at NTV than at STV for the same parametric variation profile. For example, *InterPipe+Mem* is about 4 at NTV but only about 2.5 at STV. This is because a low V_{DD} amplifies the effect of parametric variation on delay. A similar trend applies for the variation in power. Overall, the chip becomes more heterogeneous at NTV.

These experiments used a safe V_{DD_OP} for the whole chip, corresponding to the maximum V_{DD_MIN} across all blocks. However, many blocks could support lower V_{DD_MIN} values. Figure 4b shows the distribution of the per-tile V_{DD_MIN} in a representative chip. The per-tile V_{DD_MIN} is the maximum V_{DD_MIN} across all the blocks within the tile. The data is shown as a histogram. Per-tile V_{DD_MIN} values vary between 0.49 and 0.57 V.

For the memories, the minimum V_{DD} needed to avoid write-stability errors is typically slightly higher than the minimum V_{DD} needed to avoid hold errors. Thus, the former usually determines the V_{DD_MIN} of the memory blocks in our experiments.

Eschewing multiple V_{DD} domains at NTV

To quantify the limitations that make the use of multiple V_{DD} domains less attractive at NTV, we ran our applications on four different variations of the chip in Figure 3. Specifically, the *Perf* chip environment is a

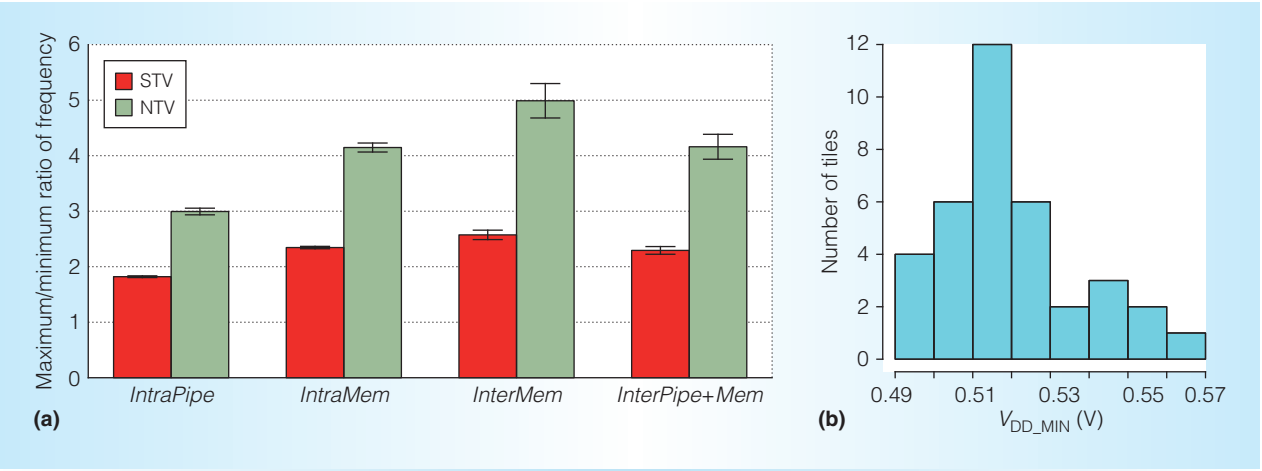

Figure 4. Impact of variation on a many-core chip's operating point. Variation in *f* at NTV and STV (a). The *f* ratios are higher at NTV than at STV. Distribution of V_{DD_MIN} over all the tiles of a representative NTV chip (b). Per-tile V_{DD_MIN} values vary between 0.49 and 0.57 V.

perfect chip configuration with a V_{DD} and an *f* domain per tile which doesn't suffer from any V_{DD} regulator losses. *Eff* is Perf with the power losses due to the on-chip V_{DD} regulators. *EffCoa* is Eff with coarser V_{DD} domains—namely, four tiles per V_{DD} domain. *EffCoaDyn* is EffCoa plus a 5-percent additional V_{DD} guardband. This tolerates potentially deeper dynamic V_{DD} droops resulting from the lower averaging effects in the current drawn by the small domains. Finally, EnergySmart relies on a single V_{DD} domain and per-tile *f* domains.

Figure 5 compares the execution's energy efficiency on the different environments in millions of instructions per second (MIPS) per watt, which is an inverse measure of the energy per operation. The figure shows lines for different power inefficiencies of the V_{DD} regulators (5, 10, 15, 20, and 25 percent). For each environment, we report the maximum achievable MIPS/W, assuming oracular application scheduling. This approach favors the multi-V_{DD}-domain environments, which require more complex core assignment algorithms owing to the larger number of degrees of freedom—as induced by the multiple V_{DD} values. The numbers in the figure are normalized to Perf's MIPS/W.

EnergySmart attains only about 80 percent of Perf's MIPS/W. This is because it doesn't exploit Perf's multiple V_{DD} domains and, therefore, operates less energy-efficiently. In the environments with multiple V_{DD} domains, the MIPS/W keeps decreasing as more practical overheads are considered (moving from Perf to Eff, EffCoa, and EffCoaDyn). For the entire range of V_{DD} regulator power losses considered, EnergySmart's MIPS/W remains higher than that of the realistic environment EffCoaDyn.

For a multi-V_{DD}-domain chip to clearly beat EnergySmart, it must require no increase in V_{DD} guardband (EffCoa) and use regulators of at most 5 percent power loss. Alternatively, it must support per-tile V_{DD} domains without any increase in V_{DD} guardband (Eff) and rely on regulators of at most 10 percent power loss. All of these chips would be significantly more complicated and use more area than EnergySmart.

To unlock NTC's energy-efficiency potential, we must understand parametric variation through experimentation and modeling, and develop effective variation-mitigation techniques. Otherwise, the promised energy savings of low-V_{DD} operation will disappear through guardbands and conservative designs.

We expect that new transistor organizations or other devices will appear that might be affected differently by variation. However, it's unlikely that parametric

variation will cease to be an important problem, given the unrelenting pursuit of aggressive designs. An example is the FinFET, which is also affected by variation.[17] Thus, we will have to continuously refine and extend our variation models.

Power losses in V_{DD} regulation are a major obstacle as we seek energy-efficient computing. We need V_{DD} regulation that has minimal power, area, and complexity costs. One possible approach is to design V_{DD} regulators hierarchically.[5] The first level comprises one or a handful of switching V_{DD} regulators (SVR) placed on a stacked die, with devices optimized for the SVR inductances. The second level comprises many on-chip low-drop-out (LDO) V_{DD} regulators. Each LDO is connected to one of the first-level SVRs and provides the V_{DD} for, say, a single tile. LDOs have a high efficiency if the ratio of their input (V_I) to output (V_O) V_{DD} is close to 1. Thanks to systematic variation, the LDOs in a region of the chip will need to provide very similar V_O. Because they take their V_I from the same first-level SVR and their V_O is similar to each other, we can design them so that their efficiency is close to 95 percent. Their area is negligible because their hardware reuses a power-gating circuit. Such a circuit is likely to be already present on chip to power-gate the tile.

For best operation at NTV, we must also optimize the device parameters (such as L_{eff}, channel width, V_{th}, and oxide thickness t_{ox}) for operation at NTV. They should be designed to yield the best performance and power tradeoff at NTV—rather than at STV, as current technologies do. This is because a technology is typically optimal for only a relatively narrow range of V_{DD}.

In the end, robust approaches to mitigate and tolerate parametric variation for 11-nm technologies and beyond will have to come from solutions that span multiple computing-stack layers. If the focus is only a single layer—say, circuits approaches—the design will necessarily be conservative, because it will have to assume worst-case conditions for the other layers. We are currently pursuing such a multilayer approach as part of DARPA's Perfect (Power Efficiency Revolution for Embedded Computing Technologies) program, together with another investigator.[18] MICRO

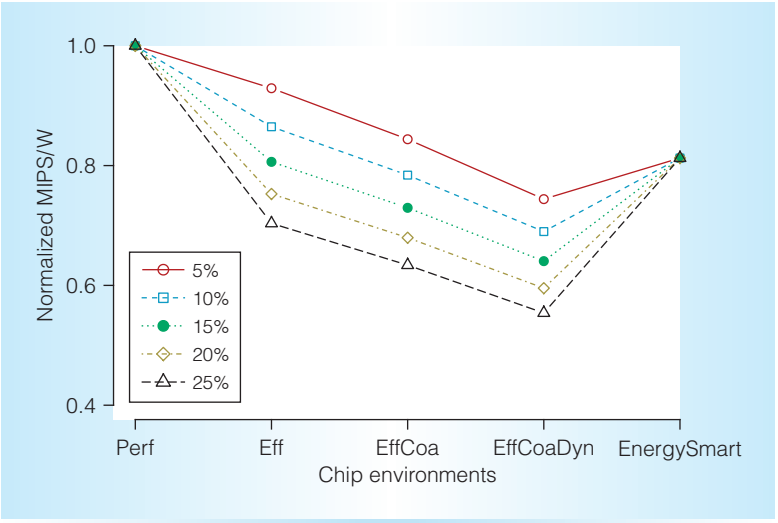

Figure 5. Normalized energy efficiency across different chip environments for a range of V_{DD} regulator inefficiencies. Lines for different power inefficiencies of the V_{DD} regulators (5, 10, 15, 20, and 25 percent) are shown.

Acknowledgments

This work was supported in part by generous grants from Intel, Advanced Micro Devices, IBM, the National Science Foundation (CCF-095360 and CCF-1012759), DOE ASCR under award number DE-FC02-10ER2599, KMEST (2012M3A6A6054194), and DARPA under UHPC contract number HR0011-10-3-0007 and under cooperative agreement HR0011-12-2-0019. Nam Sung Kim has a financial interest in AMD.

References

1. M.B. Taylor, ''Is Dark Silicon Useful? Harnessing the Four Horsemen of the Coming Dark Silicon Apocalypse,'' *Proc. 49th ACM/EDAC/IEEE Design Automation Conf.,* IEEE CS, 2012, pp. 1131-1136.
2. L. Chang et al., ''Practical Strategies for Power-Efficient Computing Technologies,'' *Proc. IEEE,* Feb. 2010, pp. 215-236.
3. R.G. Dreslinski et al., ''Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits,'' *Proc. IEEE,* Feb. 2010, pp. 253-266.
4. D. Markovic et al., ''Ultralow-Power Design in Near-Threshold Region,'' *Proc. IEEE,* Feb. 2010, pp. 237-252.
5. A.A. Sinkar et al., ''Low-Cost Per-Core Voltage Domain Support for Power-Constrained

High-Performance Processors,'' *IEEE Trans. VLSI Systems,* May 2013.
6. S. Jain et al., ''A 280mV-to-1.2V Wide-Operating-Range IA-32 Processor in 32 nm CMOS,'' *Proc. IEEE Int'l Solid-State Circuits Conf.,* IEEE CS, 2012, pp. 66-68.
7. U.R. Karpuzcu et al., ''VARIUS-NTV: A Microarchitectural Model to Capture the Increased Sensitivity of Manycores to Process Variations at Near-Threshold Voltages,'' *Proc. 42nd Ann. IEEE/IFIP Int'l Conf. Dependable Systems and Networks,* IEEE CS, 2012, pp. 1-11.
8. S. Sarangi et al., ''VARIUS: A Model of Process Variation and Resulting Timing Errors for Microarchitects,'' *IEEE Trans. Semiconductor Manufacturing,* Feb. 2008, pp. 3-13.
9. L. Chang et al., ''An 8T-SRAM for Variability Tolerance and Low-Voltage Operation in High-Performance Caches,'' *IEEE J. Solid-State Circuits,* Apr. 2008, pp. 956-963.
10. G. Chen et al., ''Yield-Driven Near-Threshold SRAM Design,'' *Proc. IEEE/ACM Int'l Conf. Computer-Aided Design,* IEEE, 2007, pp. 660-666.
11. J.W. Tschanz et al., ''Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage,'' *IEEE J. Solid-State Circuits,* Nov. 2002, pp. 1396-1402.
12. R. Teodorescu and J. Torrellas, ''Variation-Aware Application Scheduling and Power Management for Chip Multiprocessors,'' *Proc. 35th Ann. Int'l Symp. Computer Architecture,* IEEE, 2008, pp. 363-374.
13. U.R. Karpuzcu et al., ''EnergySmart: Toward Energy-Efficient Manycores for Near-Threshold Computing,'' *Proc. IEEE 19th Int'l Symp. High Performance Computer Architecture,* IEEE CS, 2013, pp. 542-553.
14. C.-K. Luk et al., ''Pin: Building Customized Program Analysis Tools with Dynamic Instrumentation,'' *Proc. ACM SIGPLAN Conf. Programming Language Design and Implementation,* ACM, 2005, pp. 190-200.
15. S. Li et al., ''McPAT: An Integrated Power, Area, and Timing Modeling Framework for Multicore and Many-core Architectures,'' *Proc. 42nd Ann. IEEE/ACM Int'l Symp. Microarchitecture,* IEEE CS, 2009, pp. 469-480.
16. C. Bienia et al., *The PARSEC Benchmark Suite: Characterization and Architectural Implications,* tech. report TR-811-08, Computer Science Dept., Princeton Univ., 2008.
17. X. Sun et al., ''Variation Study of the Planar Ground-Plane Bulk MOSFET, SOI FinFET, and Trigate Bulk MOSFET Designs,'' *IEEE Trans. Electron Devices,* Oct. 2011, pp. 3294-3299.
18. J. Torrellas, N.S. Kim, and R. Teodorescu, *Parameter Variation at Near Threshold Voltage: The Power Efficiency versus Resilience Tradeoff,* tech. report, Dept. Computer Science, Univ. Illinois at Urbana-Champaign, 2012; http://iacoma.cs.uiuc.edu/perfect/technical_report.pdf.

Ulya R. Karpuzcu is an assistant professor in the Department of Electrical and Computer Engineering at the University of Minnesota. Her research focuses on the impact of process technology on computer architecture. Karpuzcu has a PhD in electrical and computer engineering from the University of Illinois at Urbana-Champaign. She's a member of IEEE.

Nam Sung Kim is an associate professor in the Department of Electrical and Computer Engineering at the University of Wisconsin-Madison. His research interests include low-power and high-performance circuits, circuit-microarchitecture codesigns, CAD algorithms, and bio-inspired computing systems. Kim has a PhD in computer science and engineering from the University of Michigan, Ann Arbor. He's a senior member of IEEE.

Josep Torrellas is a professor in the Department of Computer Science at the University of Illinois at Urbana-Champaign. His research interests include multiprocessor computer architecture, thread-level speculation, low-power design, and hardware and software reliability. Torrellas has a PhD in electrical engineering from Stanford University. He's a fellow of IEEE and the ACM.

Direct questions and comments about this article to Ulya R. Karpuzcu, 4-174 Keller Hall, 200 Union Street S.E., Minneapolis, MN 55455; ukarpuzc@umn.edu.