Fine-Grained Voltage Boosting for Improving Yield in Near-Threshold Many-Core Processors

Conference Paper · May 2015

DOI: 10.1145/2742060.2742105

3 authors:

Joonho Kong
Kyungpook National University
52 PUBLICATIONS **918** CITATIONS
SEE PROFILE

Arslan Munir
Kansas State University
140 PUBLICATIONS **2,539** CITATIONS
SEE PROFILE

Farinaz Koushanfar
University of California, San Diego
345 PUBLICATIONS **21,512** CITATIONS
SEE PROFILE

Fine-Grained Voltage Boosting for Improving Yield in Near-Threshold Many-Core Processors

Joonho Kong
School of EE
Kyungpook National University
joonho.kong@knu.ac.kr

Arslan Munir
Dept. of CSE
University of Nevada, Reno
arslan@unr.edu

Farinaz Koushanfar
Dept. of ECE
Rice University
farinaz@rice.edu

ABSTRACT

Process variation is a major impediment in optimizing yield, energy, and performance in near-threshold many-core processors. In this paper, we present a comprehensive analysis on yield losses in near-threshold many-core processors. Based on our analysis, we propose energy-efficient yield improvement techniques for near-threshold many-core processors: SRAM cell arrays and Wordline driver voltage Boosting (SWBoost) and Cache voltage Boosting (CBoost). Results reveal that SWBoost and CBoost improve a chip yield by up to 66% and 83%, respectively. Furthermore, runtime energy overheads of SWBoost and CBoost are only 0.46% and 0.54%, respectively, which are much lower than conventional voltage boosting techniques.

Categories and Subject Descriptors

C.1.2 [**Computer Systems Organization**]: Multiple Data Stream Architectures (Multiprocessors)—*Multiple-instruction-stream, multiple-data-stream processors (MIMD)*

General Terms

Design, Performance, Reliability

Keywords

Near-threshold computing; process variations; yield; voltage boosting

1. INTRODUCTION

An effective approach to sustain Moore's law and alleviate the dark silicon problem [5] in advanced process technologies is to lower the processor's supply voltage to near the transistor's threshold voltage (V_{th}): a computing paradigm known as near-threshold computing (NTC). Decreasing supply voltage from the nominal operating point (nominal V_{dd} operation is also known as super-threshold computing (STC)) decreases operating frequency and hence performance linearly, leakage power exponentially, and active energy per operation quadratically.

Although NTC enables sustaining Moore's law; process variations, which are caused by manufacturing imperfections and are also an issue in STC, exacerbate in the NTC regime. Each submicron technology generation becomes increasingly susceptible to process variations that manifest across the chip as fluctuations in transistor parameters (mainly, threshold voltage V_{th} and effective gate length L_{eff}) around the nominal values. The parametric variation at NTV causes substantial delay and power variation in circuits of identical processor cores, which limits the maximum operating frequency of the entire many-core processor [6]. Furthermore, this variation in chips' delay and power consumption beyond design margins severely hurts processors' yield. Improving processors' yield is imperative as it can significantly impact the revenue of a semiconductor industry.

In this paper, we conduct a comprehensive yield analysis in the near-threshold regime and propose efficient yield improvement techniques that selectively boost V_{dd} in a fine-grained manner for many-core processors. Compared to conventional coarse-grained voltage boosting techniques [12][13][15][16], our fine-grained voltage boosting techniques considerably improve processors' yield. Moreover, energy efficiency of our proposed techniques reduces leakage-induced yield losses. Our main contributions are summarized as follows:

Table 1: Failures classification in many-core processors.

Metric	Element	Category
Timing	Logic	Core logic timing failure
	SRAM & logic	L1-I timing failure
	SRAM & logic	L1-D timing failure
	SRAM & logic	L2 timing failure
Stability	SRAM	L1-I stability failure
	SRAM	L1-D stability failure
	SRAM	L2 stability failure
Power	Chip	Excessive leakage failure

- We conduct the first comprehensive component-level analysis of yield losses in a near-threshold tiled many-core processor;
- We identify that SRAM-based structures are most vulnerable to yield losses at NTV and justify the necessity of fine-grained voltage boosting mechanisms;
- We propose fine-grained and selective voltage boosting techniques for tiled many-core processors: SRAM cell arrays and Wordline driver voltage Boosting (SWBoost) and Cache voltage Boosting (CBoost);
- We consider leakage-induced yield losses and reveal ineffectiveness of the previous course-grained voltage boosting techniques. We also quantify yield improvement by our proposed fine-grained voltage boosting techniques over the existing techniques.

2. YIELD ANALYSIS FOR NTC

Operation in near-threshold regime significantly impacts yield of manufactured chips due to enhanced effects of process variation. This section classifies and analyzes yield losses for different process variation severities. We classify failures that cause yield losses into eight different categories as summarized in Table 1.

2.1 Reference NTV many-core architecture

Our reference architecture is a tiled many-core processor consisting of 64 tiles similar to Tilera's TILEPro64 processor [14]. The processor features an 8×8 grid of 64 tiles (processor cores) implemented in 11nm process technology. The nominal V_{dd} and V_{th} are $0.55V$ and $0.33V$ [7], respectively. The architectural parameters for cache memories follow Tilera's TILEPro64 specifications as close as possible [14]. Our NTV many-core processor's cache memories are composed of 8T SRAM cells which are more robust to process variation than 6T SRAM cells [6].

2.2 Analysis of Yield Losses

Effect of Target Clock Frequency & Process Variation: Chip yield depends on target clock frequency and process variation severity. Process variation severity is expressed as

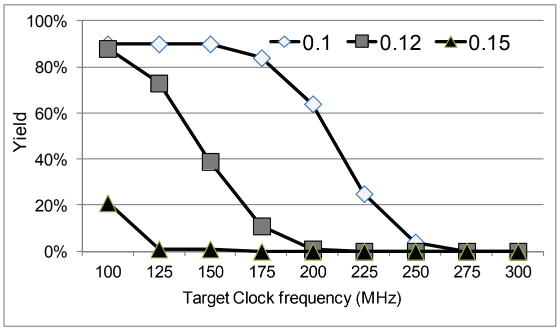

Figure 1: Yield versus target clock frequency for different process variation severities.

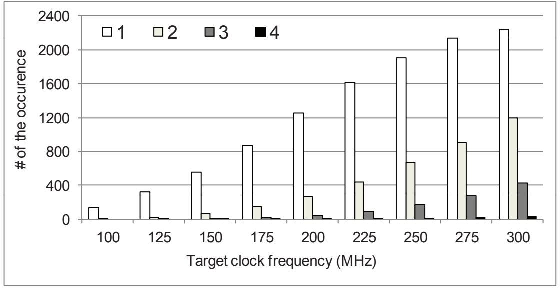

Figure 2: Number of tiles with faulty components out of 100 chips when $(\sigma/\mu)_{V_{th}} = 0.15$.

σ/μ where σ denotes standard deviation of process variation around mean μ. Fig. 1 depicts yield results for various target clock frequencies and three V_{th} process variation severities: $(\sigma/\mu)_{V_{th}} = 0.1$, 0.12, and 0.15 [7]. The L_{eff} variations are exactly half of V_{th} variations for each variation severity and are not denoted for conciseness. We set the leakage yield cutoff to be 10% of the baseline leakage-induced yield loss (approximately matched to the cutoff presented in [8]). We assume the nominal clock frequency (i.e., without process variation) of our target processor to be 1GHz [7].

Results reveal that as the target clock frequencies become higher, yield is significantly reduced. For instance, yield is 0% irrespective of process variation severity when the target clock frequency is higher than 250 MHz. Similarly, obtaining a target clock frequency of 225 MHz when $(\sigma/\mu)_{V_{th}} = 0.1$ is challenging because of diminishing yield. For example, yield is 25% for the target clock frequency of 225 MHz as compared to the yield of 64% for the target clock frequency of 200 MHz. Results indicate that yield decreases as process variation severity increases. For example, yield is only 21% for the target clock frequency of 100 MHz when $(\sigma/\mu)_{V_{th}} = 0.15$. These results verify that yield significantly limits the manufactured chips' performance.

Percentage of Faulty Tiles: Fig. 2 depicts the number of tiles containing faulty microarchitectural components (among L1D$, L1I$, L2$, and core logic) when $(\sigma/\mu)_{V_{th}} = 0.15$. We count the number of tiles containing timing or stability failure in microarchitectural components for a sample of 100 chips where the maximum number of failure occurrences is 64 tiles × 100 chips = 6400. Results reveal that there are more tiles containing one failing component as compared to the tiles with multiple failing components. For example, the number of tiles with one failing component is 34.9% when the target clock frequency is 300 MHz. These results indicate that component-level fine-grained voltage boosting techniques would be more beneficial in terms of yield and energy optimization as compared to tile-level coarse-grained voltage boosting techniques. Consequently, the finer-grained techniques would also be beneficial for a leakage-induced yield loss reduction due to their lower leakage power consumption.

Composition of Yield Losses: Our yield loss analysis in this subsection focuses on component-level failures (i.e., among L1-I, L1-D, L2, and processor core) as classified in Table 1 excluding leakage power failure. We count the number of tiles containing timing or stability failures in microarchitectural components for a sample of 100 chips where the maximum number of failure occurrences is 64 tiles × 100 chips = 6400. Fig. 3 depicts composition of yield losses for $(\sigma/\mu)_{V_{th}} = 0.15$.

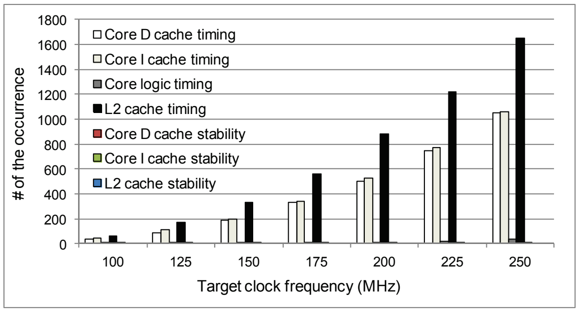

Figure 3: Composition of yield losses.

Results reveal that the timing failures in 8T SRAM-based components (i.e., caches) are dominating factors in yield losses. The core logic timing failures are negligible and only appear when the target frequency ≥ 225 MHz. The stability failures are also negligible in our many-core processor operating at 0.55V. Results further indicate that L2 caches are most susceptible to failures due to process variation as compared to other 8T SRAM-based components. The large size of L2 caches as compared to L1 caches result in greater timing failure rate in L2 caches since large cache size implies large number of 8T SRAM cells and a large number of parallel independent delay paths.

3. VOLTAGE BOOSTING TECHNIQUES FOR NTV MANY-CORE PROCESSOR

In this section, we propose fine-grained microarchitectural component-level voltage boosting techniques: SWBoost and CBoost. Our proposed techniques are based on our comprehensive yield analysis, which reveals that SRAM-based components (L1-I, L1-D, and L2 caches) are more susceptible to the effects of process variations than the processor core logic (Section 2.2) in the NTC regime. This section also discusses coarse-grained voltage-boosting techniques proposed in prior work [12][13][16]: tile-level boost (TBoost) and voltage margining (VM). We also provide implementation guidelines of our proposed techniques in this section.

3.1 Proposed Voltage Boosting Techniques

SWBoost:SWBoost supplies the boosted V_{dd} only to the wordline drivers and SRAM cell arrays. The range of boosted V_{dd} is 0.57V–0.65V. SWBoost can provide yield improvements as most failures in cache memories occur due to timing and stability failure in 8T SRAM cells. SWBoost can provide energy savings as compared to boosting the whole tile or whole cache memories. Fig. 4 depicts our proposed boosting techniques. In the case of SWBoost, we only boost wordline and SRAM cell supply voltage. SWBoost is based on dual-voltage rail (DVR); however, SWBoost operates in a finer-grained manner as compared to the conventional DVR [12][13][15][17]. SWBoost selectively applies the boosted V_{dd} to faulty cache components to improve yield and energy efficiency. Two power gating P-type metal-oxide-semiconductor (PMOS) transistors are required for a component, which are employed to select either nominal V_{dd} or boosted V_{dd} for each component. Since there are three cache memory components (L1-I, L1-D, and L2) in each tile, a total of six PMOS transistors are needed for each tile. To support SWBoost for all of the 64 tiles in the processor, 384 power gating PMOS transistors are required.

To determine whether to supply nominal or boosted V_{dd} to a cache component, SWBoost uses three fault indicator

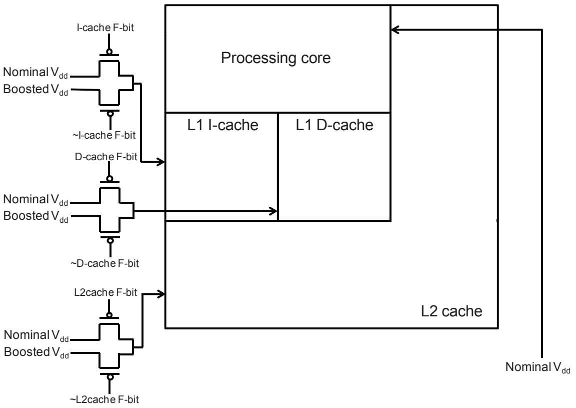

Figure 4: Our proposed voltage boosting technique.

(F) bits (stored in a non-volatile memory) for each tile: *D-cache F-bit*, *I-cache F-bit*, and *L2cache F-bit*. *D-cache F-bit*, *I-cache F-bit*, and *L2cache F-bit* determine if a timing- or stability-related failure exists in L1-D, L1-I, and L2 cache, respectively. A total of 192 bit non-volatile storage is required for storing these F-bits for our many-core processor (3 × 64 = 192). The contents of these F-bits are determined during the chip testing phase. Since current processors already possess structures for testability, our proposed techniques do not need additional components for testing.

CBoost:CBoost has the same V_{dd} boosting range and granularity as that of SWBoost, however, the boosted V_{dd} is supplied to the entire cache memories including cache peripheral circuits (e.g., address decoders and multiplexers). CBoost can provide higher yield than SWBoost as it can alleviate more timing-related failures in cache memories (cache peripheral circuits with the boosted V_{dd} will also be faster than non-boosted V_{dd}), however, CBoost consumes more energy than SWBoost. As in the case of SWBoost, CBoost also needs 384 power gating PMOS transistors and 192 F-bits.

3.2 Existing Voltage Boosting Techniques

TBoost:TBoost is a coarse-grained voltage boosting technique that supplies boosted voltage at the tile-level in case of failure in any of the tile's component (processor core, L1-I, L1-D, and L2 cache). In our TBoost implementation, we use the same boosting granularity as presented in [12][13], although prior work does not boost L2 caches.

Voltage Margining (VM):VM [16] is a coarse-grained voltage boosting technique that increases the chip-wide V_{dd} to save the faulty chips (or tiles) and improve performance. VM consumes more energy than SWBoost, CBoost, and TBoost because of being coarser-grained than other techniques.

3.3 Implementation Issues

Our proposed fine-grained voltage boosting designs (SWBoost and CBoost) require power gating PMOS transistors and non-volatile F-bits, and hence requires some area overhead. A single power gating PMOS transistor requires huge area compared to regular transistors (∼ 6K transistors) [12], however, the overall area overhead is negligible compared to the total processor area as the state-of-the-art processors integrate more than a billion transistors. A conservative estimate of area overhead of all the power gating PMOS transistors in our many-core processor is 0.23%. The area overhead of non-volatile storage for 192 F-bits is negligible.

Although our designs use two different voltage levels (nominal and boosted), our designs do not need voltage level converters because of small difference between these voltage levels [11]. In our designs, devices with non-zero threshold voltages can be used in place of voltage level converters.

4. EVALUATION

This section presents evaluation results focusing on yield improvement and energy overhead of our proposed fine-grained voltage boosting techniques (SWBoost and CBoost) and coarse-grained voltage boosting techniques proposed in prior work: TBoost and VM. We also present yield results for the spare tiles (ST) technique for yield improvement: ST$_x$ represents the case when 'x' number of spare tiles are employed for mitigating the effects of process variation. We compare our voltage boosting techniques with a *baseline* case that denotes no voltage boosting or no process variation-aware technique. We investigate the yield results for three different process variation severities: $(\sigma/\mu)_{V_{th}}$ = 0.1, 0.12, and 0.15.

4.1 Evaluation setup

For yield estimation, we use VARIUS-NTV [6] process variation model, which is specialized for NTC. We specify V_{th} and L_{eff} variation severities whereas other parameters are set to their default values in VARIUS-NTV. For workload-dependent energy consumption of our many-core processor, we use Snipersim [1] to extract the access counts of each functional unit, which are then given as an input to McPAT [10] scaled for 11nm technology node. We use 15 multi-threaded benchmarks from SPLASH-2 (barnes, cholesky, lu, ocean, radiosity, radix, and raytrace) and PARSEC (blackscholes, bodytrack, dedup, fluidanimate, freqmine, raytrace, streamcluster, and swaptions) for our evaluations.

4.2 Yield

Table 2 summarizes yield results for various boosting techniques with boosted V_{dd} of 0.57V, 0.61V, and 0.65V and spare tiles for various target clock frequencies (TCF) and process variation severities. Results indicate that ST techniques impart best yield when $(\sigma/\mu)_{V_{th}} = 0.1$ and target clock frequency is 200 MHz. Our CBoost technique delivers best yield for relatively high target clock frequencies.

For instance, CBoost enables 54% yield, which is highest across other considered techniques, when target clock frequency is 300 MHz and boosted V_{dd} is 0.65V. CBoost enables 15% yield improvement over TBoost when target clock frequency is 300 MHz and boosted V_{dd} is 0.65V. Better yield of CBoost than TBoost is due to the coarser-grained boosting (tile-wide) of TBoost that results in more power/energy consumption. This additional energy consumption in TBoost causes additional leakage-induced yield losses, which limits the yield attainable from TBoost. SWBoost can obtain comparable yield to CBoost and TBoost when the target frequency is low ($\leq$ 200 MHz), however, the yield attainable from SWBoost decreases sharply as the target frequency increases. VM proffers the worst yield among all the considered techniques due to low leakage power efficiency that results in high leakage-induced yield losses.

Results indicate that attainable yield from all voltage boosting techniques deteriorates for $(\sigma/\mu)_{V_{th}}$ = 0.12 and $(\sigma/\mu)_{V_{th}}$ = 0.15 due to increased process variations. However, overall trend of yield results across various techniques is similar to the case of $(\sigma/\mu)_{V_{th}} = 0.1$.

4.3 Energy

Fig. 5 shows the geometric mean of normalized energy results for various multi-threaded workloads for $(\sigma/\mu)_{V_{th}}$ = 0.15. The 'TBoost_exL2' in Fig. 5 denotes the TBoost technique in which the L2 cache is excluded from the tile-level V_{dd} boosting similar to the technique introduced in [12]. Results indicate that SWBoost is most energy-efficient because of fine-grained boosting of only the SRAM arrays and wordline drivers of faulty cache memory components. SWBoost and CBoost show a runtime energy overhead of 0.46% and 0.54% on average, respectively, as compared to the baseline.

Table 2: Yield results of boosting techniques and spare tiles (TCF denotes target clock frequency and PV process variation).

PV	**Boosted V_{dd}**	**N/A**	**0.57V**				**0.61V**				**0.65V**				**N/A**		
$(\frac{\sigma}{\mu})_{V_{th}}$	**TCF**	**Base**	**SWBoost**	**CBoost**	**TBoost**	**VM**	**SWBoost**	**CBoost**	**TBoost**	**VM**	**SWBoost**	**CBoost**	**TBoost**	**VM**	**ST$_1$**	**ST$_2$**	**ST$_3$**
0.1	200MHz	64%	77%	79%	79%	49%	86%	89%	89%	0%	89%	89%	89%	0%	79%	93%	98%
	250MHz	4%	18%	19%	19%	10%	45%	68%	67%	0%	70%	87%	86%	0%	20%	38%	52%
	300MHz	0%	0%	0%	0%	0%	4%	11%	9%	0%	12%	54%	47%	0%	0%	0%	2%
0.12	200MHz	1%	5%	7%	7%	5%	23%	38%	34%	0%	43%	72%	64%	0%	3%	14%	34%
	250MHz	0%	0%	0%	0%	0%	1%	1%	1%	0%	2%	20%	11%	0%	0%	0%	0%
	300MHz	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	1%	0%	0%	0%	0%	0%
0.15	100MHz	21%	49%	45%	44%	27%	64%	76%	76%	9%	74%	86%	86%	0%	53%	76%	85%
	150MHz	1%	1%	1%	1%	0%	6%	14%	13%	2%	19%	52%	45%	0%	1%	5%	12%
	200MHz	0%	0%	0%	0%	0%	0%	1%	1%	0%	0%	2%	1%	0%	0%	0%	0%

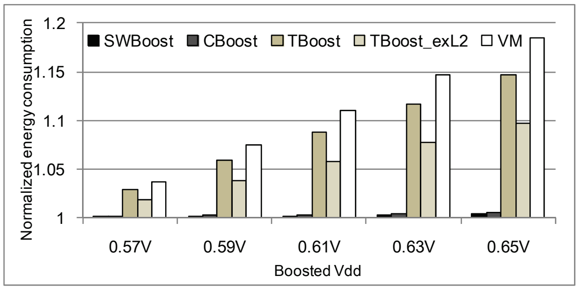

Figure 5: Geometric mean of energy results normalized to the baseline for $(\sigma/\mu)_{V_{th}} = 0.15$.

5. RELATED WORK

Process variation is a major issue for optimizing yield, performance, and energy in the NTV regime. Several earlier studies investigated cache and multi-core designs for NTC. Chen et al. [2] compared conventional 6T SRAM, single-ended 6T SRAM, and 8T SRAM designs for NTC. Dreslinski et al. [4] proposed an energy-efficient L1 cache architecture for NTC. Dreslinski et al. [3] proposed a multi-voltage, clustered multi-core design that supports different supply voltages and threshold voltages in the caches for each cluster (a group of processor cores was termed as cluster). Our work differs from this work as our proposed designs supply higher V_{dd} to only faulty cache memories for yield improvement. Moreover, our designs do not need V_{th} tuning nor require voltage-level converters, which makes our designs much simpler than that presented in [3].

Miller et al. [13] proposed dual-voltage rails and half speed units for mitigating process variation in the NTV regime. The authors proposed core-level voltage boosting by supplying dual V_{dd}, however, this course-grained voltage boosting results in energy and yield inefficiency. Our proposed designs only boost faulty cache memories that present a major bottleneck in yield improvement in NTC. Seo et al. [16] proposed various techniques to mitigate the effects of process variation in near-threshold single instruction multiple data (SIMD) architectures including VM, spare components (cores/tiles), and frequency margining. Results reveal that our proposed fine-grained voltage boosting techniques depict better yield as compared to VM and spare tiles. In [9], a variation-aware SIMD architecture was proposed, however, the proposed design focused only on the processing unit and not on memory components. Karpuzcu et al. [7] proposed a process variation-aware thread scheduling and frequency assignment technique that exploits a heterogeneity of clusters in an NTV many-core processor. Some prior work demonstrated silicon implementations for NTV processors [15] which use different voltages for SRAM arrays for better stability; however, our designs selectively supply the boosted V_{dd} only to the faulty cache memories.

6. CONCLUSIONS

In this paper, we present a comprehensive analysis of yield losses in tile-based many-core processors. Results indicate that SRAM-based components are vulnerable to yield losses in the near-threshold regime. We observe that fine-grained selective voltage boosting techniques not only reduce both timing- and leakage-induced yield losses but also improve runtime energy efficiency. Based on our yield analysis, we propose architectural component-level V_{dd} boosting techniques: SWBoost and CBoost. Results reveal that SWBoost and CBoost improve a chip yield by up to 66% and 83%, respectively. Results also verify that energy overhead of our proposed techniques is significantly less than the conventional techniques. SWBoost is most energy-efficient among all the evaluated techniques with a maximum energy overhead of only 0.46% as compared to the baseline. In our future work, we plan to incorporate a process variation model for interconnect and optimize our design to take into account both interconnects and tiles under process variations in the NTV regime.

Acknowledgements. This work was supported in part by NSF career grants (No. 0644289 and 1116858), ARO YIP grant (R17450), ONR R17460, and the University of Nevada, Reno.

7. REFERENCES

[1] T. Carlson, W. Heirman, and L. Eeckhout. Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation. In *the 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 1–12, 2011.

[2] G. K. Chen, D. Blaauw, T. Mudge, D. Sylvester, and N. S. Kim. Yield-driven near-threshold SRAM design. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, pages 660–666, 2007.

[3] R. G. Dreslinkski, B. Zhai, T. Mudge, D. Blaauw, and D. Sylvester. An energy efficient parallel architecture using near threshold operation. In *Proceedings of the 16th International Conference on Parallel Architecture and Compilation Techniques*, pages 175–188, 2007.

[4] R. G. Dreslinski, G. K. Chen, T. Mudge, D. Blaauw, D. Sylvester, and K. Flautner. Reconfigurable energy efficient near threshold cache architectures. In *Proceedings of 41st IEEE/ACM International Symposium on Microarchitecture*, pages 459–470, 2008.

[5] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. In *Computer Architecture (ISCA), 2011 38th Annual International Symposium on*, pages 365–376, 2011.

[6] U. R. Karpuzcu, K. B. Kolluru, N. S. Kim, and J. Torrellas. VARIUS-NTV: A microarchitectural model to capture the increased sensitivity of manycores to process variations at near-threshold voltages. In *Proceedings of the 2012 42nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 1–11, 2012.

[7] U. R. Karpuzcu, A. Sinkar, N. S. Kim, and J. Torrellas. EnergySmart: Toward energy-efficient manycores for near-threshold computing. In *Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, pages 542–553, 2013.

[8] J. Kong, Y. Pan, S. Ozdemir, A. Mohan, G. Memik, and S. W. Chung. Fine-Grain Voltage Tuned Cache Architecture for Yield Management under Process Variations. *IEEE Transactions on VLSI Systems*, 20(8):1532–1536, 2012.

[9] E. Krimer, R. Pawlowski, M. Erez, and P. Chiang. Synctium: A near-threshold stream processor for energy-constrained parallel applications. *IEEE Computer Architecture Letter*, 9(1):21–24, 2010.

[10] S. Li, J.-H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, pages 469–480, 2009.

[11] X. Liang, G.-Y. Wei, and D. Brooks. ReVIVaL: A variation-tolerant architecture using voltage interpolation and variable latency. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, pages 191–202, 2008.

[12] T. N. Miller, X. Pan, R. Thomas, N. Sedaghati, and R. Teodorescu. Booster: Reactive core acceleration for mitigating the effects of process variation and application imbalance in low-voltage chips. In *Proceedings of the 2012 IEEE 18th International Symposium on High Performance Computer Architecture (HPCA)*, pages 27–38, 2012.

[13] T. N. Miller, R. Thomas, and R. Teodorescu. Mitigating the effects of process variation in ultra-low voltage chip multiprocessors using dual supply voltages and half-speed units. *Computer Architecture Letters*, 11(2):45–48, 2012.

[14] A. Munir, F. Koushanfar, A. Gordon-Ross, and S. Ranka. High-performance optimizations on tiled many-core embedded systems: A matrix multiplication case study. *The Journal of Supercomputing*, 66(1):431–487, 2013.

[15] G. Ruhl, S. Dighe, S. Jain, S. Khare, and S. Vangal. IA-32 processor with a wide-voltage-operating range in 32-nm CMOS. *IEEE Micro*, 33(2):28–36, 2013.

[16] S. Seo, R. G. Dreslinski, M. Woh, Y. Park, C. Charkrabari, S. Mahlke, D. Blaauw, and T. Mudge. Process variation in near-threshold wide SIMD architectures. In *Proceedings of the 49th Annual Design Automation Conference*, pages 980–987, 2012.

[17] B. Stolt, Y. Mittlefehldt, S. Dubey, G. Mittal, M. Lee, J. Friedrich, and E. Fluhr. Design and implementation of the POWER6 microprocessor. *IEEE Journal of Solid-State Circuits*, 43(1):21–28, 2008.