Mitigating the Effects of Process Variation in Ultra-low Voltage Chip Multiprocessors using Dual Supply Voltages and Half-Speed Units

Timothy N. Miller, Renji Thomas, Radu Teodorescu
Department of Computer Science and Engineering, The Ohio State University
{millerti, thomasr, teodores}@cse.ohio-state.edu

Abstract—Energy efficiency is a primary concern for microprocessor designers. One very effective approach to improving processor energy efficiency is to lower its supply voltage to very near to the transistor threshold voltage. This reduces power consumption dramatically, improving energy efficiency by an order of magnitude. Low voltage operation, however, increases the effects of parameter variation resulting in significant frequency heterogeneity between (and within) otherwise identical cores. This heterogeneity severely limits the maximum frequency of the entire CMP. We present a combination of techniques aimed at reducing the effects of variation on the performance and energy efficiency of near-threshold, many-core CMPs. Dual Voltage Rail (DVR), mitigates core-to-core variation with a dual-rail power delivery system that allows post-manufacturing assignment of different supply voltages to individual cores. This speeds up slow cores by assigning them to a higher voltage and saves power on fast cores by assigning them to a lower voltage. Half-Speed Unit (HSU) mitigates within-core variation by halving the frequency of select functional blocks with the goal of boosting the frequency of individual cores, thus raising the frequency ceiling for the entire CMP. Together, these variation-reduction techniques result in almost 50% improvement in CMP performance for the same power consumption over a mix of workloads.

Index Terms—Energy efficiency, near-threshold voltage, process variation, chip multiprocessors.

1 Introduction

Power consumption is one of the most significant roadblocks to continued growth of microprocessor performance. Physical limits on power delivery and heat removal are already constraining designs today [9] and will continue to severely restrict performance in the future [15]. If current integration trends continue, chips could see a 10-fold increase in power density by the time 11nm technology is in production [3]. The only way to ensure continued scaling and performance growth is to develop solutions that dramatically increase computational energy efficiency.

A very effective approach to improving the energy efficiency of a microprocessor is to lower its supply voltage (V_{dd}) to very close to the transistor's threshold voltage (V_{th}), into the so-called near-threshold (NT) region [1], [8], [10]. This is significantly lower than what is used in standard dynamic voltage and frequency scaling (DVFS), resulting in aggressive reductions in power consumption (up to 100×) with about a 10× loss in maximum frequency. Even with the lower frequency, chips running in near-threshold have the potential to achieve significant improvements in energy efficiency. In a power-constrained chip multiprocessor (CMP), near-threshold operation will allow many more cores to be powered on than in a CMP at nominal V_{dd} increasing aggregate CMP throughput.

Unfortunately, near-threshold CMPs are very sensitive to *process variation*. Variation is caused by manufacturing challenges with very small feature technologies. One parameter most severely affected by variation is the transistor threshold voltage (V_{th}). Variation in V_{th} causes heterogeneity in transistor delay and power consumption within processor dies leading to sub-optimal performance. Near-threshold operation greatly exacerbates these effects because supply voltage is much closer to the threshold voltage, making the impact of V_{th} variation much more pronounced. In this paper, we show that at NT, frequency variation can easily be an order of magnitude higher than at nominal V_{dd}.

This paper introduces two simple, low-overhead, but highly effective techniques for mitigating frequency variation in near-threshold CMPs. The first technique, Dual Voltage Rails (DVR), consists of a power supply system that provides the CMP with two power supply rails. Each power rail supplies a different externally controlled voltage. Each core in the CMP can be assigned to either of the two power supplies using a simple power gating circuit [4]. We show that DVR can reduce frequency variation from 30.6% standard deviation from the mean (σ/μ) down to 23.1%, improving CMP frequency by 30%.

The second technique, Half-Speed Unit (HSU), targets within-core variation responsible for increasing critical path delay and lowering the maximum core frequency. With HSU, functional units have two possible speeds: full speed (running at the core's frequency) and half speed (running at half the core's frequency). Slower units run at half speed, allowing the frequency of that core (and the entire CMP) to be increased substantially. Our evaluation shows DVR alone improves the performance of a variation-unaware 64-core NT CMP by 30% and HSU alone by 33%. When combined, DVR and HSU together achieve a 48% average performance improvement.

Overall, this paper makes the following contributions:

- Analyzes the impact of process variation on large CMPs running at near-threshold voltages.
- Presents DVR, a simple and powerful solution for reducing core-to-core frequency variation in NT CMPs.
- Presents HSU, a low-overhead, low-complexity solution for mitigating within-core variation in NT CMPs.

More details and an extended evaluation can be found in the workshop version of this paper [11].

2 Related Work

Previous work has proposed techniques for reducing variation in processors operating at nominal voltages including body biasing [14], variable pipeline stage latency [6] and the GALS architecture [7]. Most previous solutions fine-tune the delay of pipeline stages to reduce delay variation and improve frequency. These designs incur significant overheads compared to our work: multiple independent bias voltages (and wells) for body biasing, complex calibration for variable pipeline latency designs.

Previous work has examined dual and multi-V_{dd} designs with the goal of improving energy efficiency. Most previous work has focused on tuning the delay vs. power-consumption of paths at fine granularity within the processor. For instance, in [5], circuit blocks along critical paths are assigned to the higher power supply, while blocks along non-critical paths are assigned to a lower power supply. A dual-V_{dd} design was proposed by R. Dreslinski in his Ph.D. dissertation [2] to speed up transactional bottlenecks.

3 Architecture Design

3.1 Dual Voltage Rails (DVR)

Within-die variation causes power consumption and maximum operating frequency to vary widely from core to core. This can severely limit CMP frequency because the system clock is limited by the slowest core. At the same time, any core that can run faster than the system clock is wasting energy. This is because these cores could run at a lower voltage for the same speed and therefore save power.

DVR addresses these inefficiencies by providing two power supply rails in the CMP. Each power rail supplies a different voltage, both near-threshold, with one slightly higher than the other. Cores can be assigned, post-manufacturing, to either of the two supply voltages as follows: "fast" cores are assigned to run on the lower V_{dd}, reducing their power consumption, while "slow" cores run on the higher V_{dd}, improving their frequency. This reduces within-die frequency variation and therefore reduces wasted energy. At near-threshold even small changes in V_{dd} significantly effect frequency. Thus, even a small difference (100mV) between the two rails dramatically reduces frequency variation.

DVR is low overhead and relatively easy to implement. Each core is provided with two power gates [4], allowing it to be assigned to either power rail. Within each core, only a single power distribution network is needed, resulting in a much lower overhead compared to solutions that employ dual voltages at much finer granularity [5], [6].

3.1.1 Post-manufacturing Calibration

For DVR to be effective at reducing within-die variation, a post-manufacturing calibration process is needed. Calibration can be performed during burn-in while the chip is also tested for defects. Calibration involves two stages. In the first stage, a set of built-in self-tests (BIST) will be used to characterize the variation profile of the die. The variation profile provides a mechanism for estimating the maximum frequency each core can achieve as a function of V_{dd} and its internal V_{th} distribution. The second calibration step uses the variation profile of each chip to perform an off-line (and off-chip) optimization to choose the V_{dd} levels for the two DVR rails and which cores should be assigned to each rail. Various optimization criteria may be used for this step. For instance, one straightforward optimization is to maximize CMP frequency under iso-power constraints.

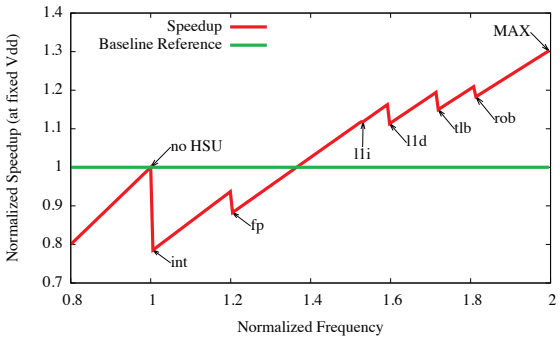

Fig. 1: Frequency vs. average speedup for a core with HSU running SPEC2000 benchmarks. Performance drops when a unit's frequency is dropped to half-speed, but core frequency can then be increased, offsetting the the loss.

3.2 Half-Speed Unit (HSU)

Within-core variation is another important hinderance to the efficiency of NT CMPs. At very low V_{dd}, delay variation between functional units can be substantial. This results in lower core frequencies, because the frequency of a core is dictated by the critical path delay of the slowest functional unit. To improve individual core frequency in the presence of a few slow units, Half-Speed Unit (HSU) allows slow units to operate at half the main clock frequency. This moves the slow units out of the critical path, allowing core frequency to be raised substantially. Applying HSU to the slowest cores in the CMP allows the CMP clock frequency to be raised, improving the aggregate CMP performance. Even if the performance of some cores is reduced by HSU, the loss is more than offset by the higher CMP frequency.

Figure 1 shows the effect of HSU on the mean performance of a core, randomly chosen from our variation model, running the SPEC benchmarks. At baseline frequency, all functional units are running at full speed. As frequency increases, the first unit that becomes critical is, in this case, the integer ALU cluster ("int"). It is set to half speed, and performance initially drops by 20%. Frequency however can be raised by about 15%, making up for some of the performance loss, before the next slower unit must have HSU applied. After applying HSU to the "fp" cluster the frequency can continue to rise, bringing performance above the initial baseline.

The previously proposed GALS [7] architecture runs the main functional units on completely independent clocks to mitigate variation. GALS requires asynchronous queues to control dataflow between clock domains, and these can add significant latency. The HSU design is much simpler because it does not require inter-stage communication queues beyond those present in an out-of-order processor. Slow functional units will simply have double the latency of the same unit running at full speed.

Our HSU implementation divides a processor into functional blocks (groups of functional units) so as to minimize the architectural challenges associated with having one component communicating with another that is operating at half speed. Figure 2 shows the HSU granularity in our design. The following functional blocks can be independently switched to half-speed if needed: *inor*, the entire in-order section (fetch, decode, etc.); *l1i* and *l1d*, the L1 caches; *tlb*, the translation lookaside buffer; *ls*, loads, stores, the load-store queue, and address calculations; *int*, all integer ALU units; *fp*, all floating-point ALU units; and *rob*, the unified reorder buffer.

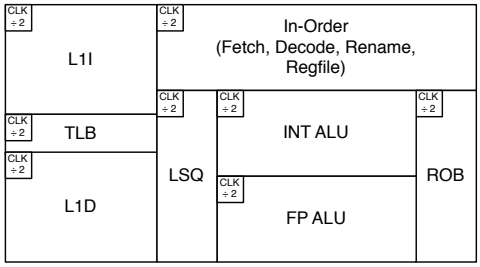

Fig. 2: Overview of Half-Speed Unit, with clock dividers for each functional unit block. Units can run on the system clock or enable the divider to run at half-speed.

CMP architecture	
Cores	64, out-of-order
Fetch/issue/commit width	2/2/2
Register file size	40 entry
L1 data cache	2-way 16K, 1-cycle access
L1 instruction cache	1-way 16K, 1-cycle access
Shared L2	8-way 16 MB, 10 cycle access
Technology	32nm
Nominal V_{dd}	900mV
Near threshold V_{dd}	300mV – 500mV
Nominal Frequency	2GHz @ 900mV
Near threshold Frequency	400Mhz @ 400mV
Variation parameters	
V_{th} mean (μ),	210mV
V_{th} std. dev./mean (σ/μ)	3% – 12%
ϕ (correlation distances)	0.1, 0.4, 1.0 of die width

TABLE 1: Summary of the experimental parameters.

4 Evaluation Methodology

We model a 32nm 64-core CMP. Each core is dual-issue out-of-order (see Table 1). We modified SESC [12] to simulate the CMP and ran the SPEC CPU2000 benchmarks, SPECint (*crafty, mcf, parser, gzip, bzip2, vortex,* and *twolf*) and SPECfp (*wupwise, swim, mgrid, applu, apsi, equake,* and *art*).

To simulate the impact of HSU on performance, we ran all benchmarks for each possible HSU profile. Since there are eight different blocks that can be run at half speed, this required 256 (e.g. mcf_0 to mcf_{255}) simulations for each benchmark.

We model variation in threshold voltage (V_{th}) using VARIUS [13]. Each chip is modeled as a grid of points and each point is given one value of V_{th} assumed to have a normal distribution with mean μ and standard deviation σ. Variation is also characterized by a spatial correlation, so that adjacent areas on a chip have roughly the same V_{th}. Spatial correlation is characterized by a correlation distance ϕ, at which there is no significant correlation between two grid points. ϕ is expressed as a fraction of the chip width. Table 1 shows some of the process parameters used. Each individual experiment uses a batch of 600 chips that have a different variation map generated with the same mean μ, same standard deviation σ, and three different correlation distances ϕ (Table 1).

We use the power and frequency models for near-threshold operation of Markovič et al. [8].

5 Evaluation

We evaluate the performance improvement and energy savings achieved by a CMP with DVR and HSU applied both independently and in conjunction. We begin by evaluating the impact of process variation on the frequency of NT CMPs.

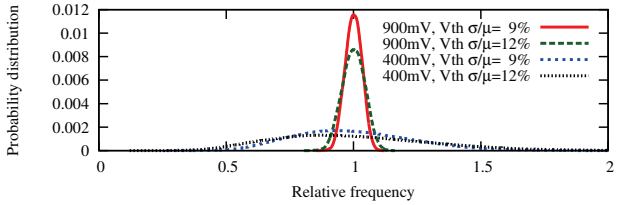

Fig. 3: Core-to-core frequency variation at nominal and near-threshold V_{dd}, relative to die mean.

5.1 Frequency Variation at Near-Threshold

Process variation has a much greater effect on core frequency at near-threshold than at nominal V_{dd}. Figure 3 illustrates core-to-core variation in frequency as a probability distribution of core frequency divided by die mean (average over all cores in the same die). Distributions are shown for 9% and 12% within-die V_{th} variation (σ/μ). At nominal V_{dd} the distribution is tight, with only 4.4% frequency σ/μ. At NT, cores vary from less than half to more than 1.5× mean, for a very large 30.6% σ/μ variation. Table 2 shows the impact of different V_{th} variation levels on the σ/μ of frequency variation at nominal and NT voltages.

V_{th} σ/μ	Freq. σ/μ at 900mV	Freq. σ/μ at 400mV
3%	1.0%	7.5%
6%	2.1%	15.1%
9%	3.2%	22.8%
12%	4.4%	30.6%

TABLE 2: Frequency variation as a function of V_{th} variation and V_{dd}.

The high variation has a dramatic impact on CMP frequency. Without variation, a 32nm CMP would be expected to run at about 400MHz at V_{dd} = 400mV. With a 12% V_{th} variation our model indicates an average frequency across all dies of 149MHz, with a minimum of 75MHz and a maximum of 230MHz, for the same V_{dd}. Clearly, variation has a very detrimental effect on the frequency of NT CMPs.

5.2 Variation Reduction with DVR and HSU

5.2.1 Performance Improvements from DVR

The goal of DVR is to raise the minimum frequency of the CMP while keeping power consumption constant. This is achieved by assigning slow cores to the high voltage rail and fast cores to the low voltage rail, reducing frequency variation. Figure 4 shows the effect of DVR on core frequency distribution, compared to a single voltage rail (SVR), for the same power. DVR significantly tightens the frequency variation, with core frequency σ/μ reduced from 30.6% to 23.1%. Frequency distribution is a non-linear function of the V_{th} distribution, so the distribution tightens more than the minimum increases, resulting in a lower average frequency for DVR compared to SVR. The minimum frequency however increases by about 30% on average, as shown in Figure 5.

We also compare the DVR improvement with the ideal case of having each core at its own optimal V_{dd} (64Vdd in Figure 5). DVR, with only two voltage rails, improves efficiency (+30%) by more than half as much as having independent voltage rails for each core (+57%). Note that the ideal case is not practical to implement because of the large number of power lines and voltage regulators required.

DVR yields significant performance improvements even though the voltage difference between two power rails is not very large. The average difference between $V_{\text{DVR}-\text{low}}$ and $V_{\text{DVR}-\text{high}}$, across all chips we simulate is 66mV. The maximum difference is 120mV, and the minimum is 30mV. The average $V_{\text{DVR}-\text{low}} = 364\text{mV}$ and $V_{\text{DVR}-\text{high}} = 429\text{mV}$.

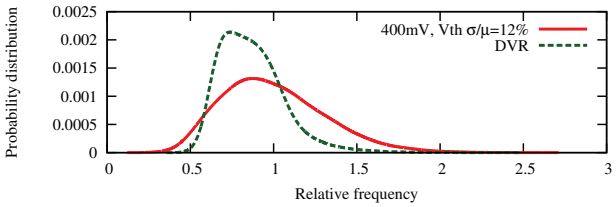

Fig. 4: Core-to-core frequency variation for DVR versus SVR. Data points are normalized to SVR die mean.

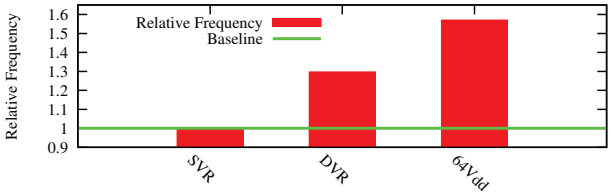

Fig. 5: Average frequency increase from DVR relative to the SVR baseline. For reference, we show the theoretical best case where every core has its own ideal voltage supply ($64V_{dd}$).

5.2.2 Performance Improvements from HSU

HSU helps improve chip performance by mitigating within-core variation. We show two options for applying HSU. The first (HSU_{isoP}) is iso-power. Both the supply voltage and the HSU profile are optimized to improve CMP performance while keeping power consumption the same as baseline. This may reduce the performance of some cores to below baseline.

The second (HSU_{isoV}) keeps V_{dd} unchanged at 400mV and raises frequency as much as possible to achieve the greatest performance, without limiting power. This has the advantage of ensuring that no core's performance is lower than baseline.

Figure 6 shows the performance improvement from HSU, averaged across all chips in our experiments, broken down by benchmark. Each benchmark is run on all cores of each chip, in isolation. HSU_{isoP} achieves an average speedup of 32% over the baseline, for the same power consumption. HSU_{isoV} does even better, with a speedup of 58% over the baseline, at the same V_{dd}, but with a higher power consumption.

5.2.3 Performance Improvements from DVR and HSU

DVR and HSU can be combined to further improve performance in the presence of variation. DVR and HSU address different variation issues and therefore synergize very well.

Figure 7 shows the per-benchmark effects of DVR, HSU, and their combination. On average DVR alone improves performance by 29%. When combined with HSU_{isoP} and HSU_{isoV} the performance improvement jumps to 48% and 49% respectively. This shows that DVR and HSU combine very well to achieve an almost 50% performance improvement over the baseline NT CMP.

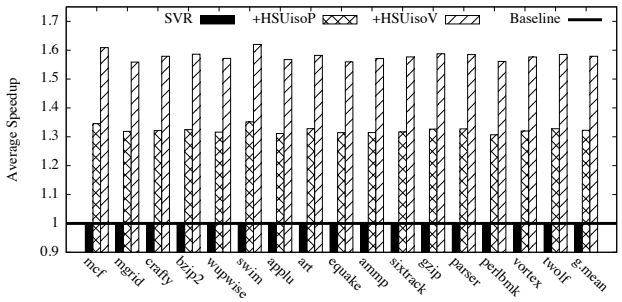

Fig. 6: Per-benchmark speedup relative to SVR.

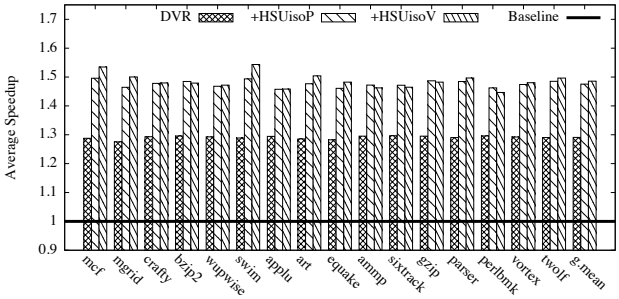

Fig. 7: Per-benchmark speedup relative to SVR.

5.2.4 Energy Savings

Since DVR and HSU reduce runtime for the same power, energy is reduced. DVR reduces CMP energy by about 23% of baseline, HSU by around 25%, and together around 32%.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grant CCF-1117799 and an allocation of computing time from the Ohio Supercomputer Center. The authors would like to thank the anonymous reviewers for their suggestions and feedback.

REFERENCES

[1] R. Dreslinski, M. Wieckowski, D. Blaauw, D. Sylvester, and T. Mudge, "Near-threshold computing: Reclaiming Moore's law through energy efficient integrated circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253 –266, February 2010.

[2] R. Dreslinski, "Near threshold computing: From single core to many-core energy efficient architectures," Ph.D. dissertation, The University of Michigan, 2011.

[3] "International Technology Roadmap for Semiconductors (2009)."

[4] H. Jiang and M. Marek-Sadowska, "Power gating scheduling for power/ground noise reduction," in *Design Automation Conference*, 2008, pp. 980–985.

[5] S. Kulkarni, A. Srivastava, and D. Sylvester, "A new algorithm for improved VDD assignment in low power dual VDD systems," in *International Symposium on Low Power Electronics and Design*, May 2004, pp. 200–205.

[6] X. Liang, G.-Y. Wei, and D. Brooks, "Revival: A variation-tolerant architecture using voltage interpolation and variable latency," *IEEE Micro*, vol. 29, no. 1, pp. 127–138, 2009.

[7] D. Marculescu and E. Talpes, "Variability and energy awareness: A microarchitecture-level perspective," in *Design Automation Conference*, June 2005, pp. 11–16.

[8] D. Markovic, C. Wang, L. Alarcon, T.-T. Liu, and J. Rabaey, "Ultralow-power design in near-threshold region," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 237–252, February 2010.

[9] R. McGowen, C. Poirier, C. Bostak, J. Ignowski, M. Millican, W. Parks, and S. Naffziger, "Power and temperature control on a 90-nm Itanium family processor," *Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 229–237, January 2006.

[10] T. Miller, J. Dinan, R. Thomas, B. Adcock, and R. Teodorescu, "Parichute: Generalized turbocode-based error correction for near-threshold caches," in *International Symposium on Microarchitecture*, 2010, pp. 351–362.

[11] T. 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 stages," in *Workshop on Energy-Efficient Design, in conjunction with ISCA*, 2011.

[12] J. Renau, B. Fraguela, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, K. Strauss, S. Sarangi, P. Sack, and P. Montesinos, "SESC Simulator," January 2005, http://sesc.sourceforge.net.

[13] S. R. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "VARIUS: A model of parameter variation and resulting timing errors for microarchitects," *IEEE Transactions on Semiconductor Manufacturing*, vol. 21, no. 1, pp. 3–13, February 2008.

[14] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas, "Mitigating parameter variation with dynamic fine-grain body biasing," in *International Symposium on Microarchitecture*, December 2007, pp. 27–39.

[15] J. Torrellas, "Architectures for extreme-scale computing," *IEEE Computer*, vol. 42, pp. 28–35, November 2009.