Low-Power Near-Threshold Design

Techniques to Improve Energy Efficiency

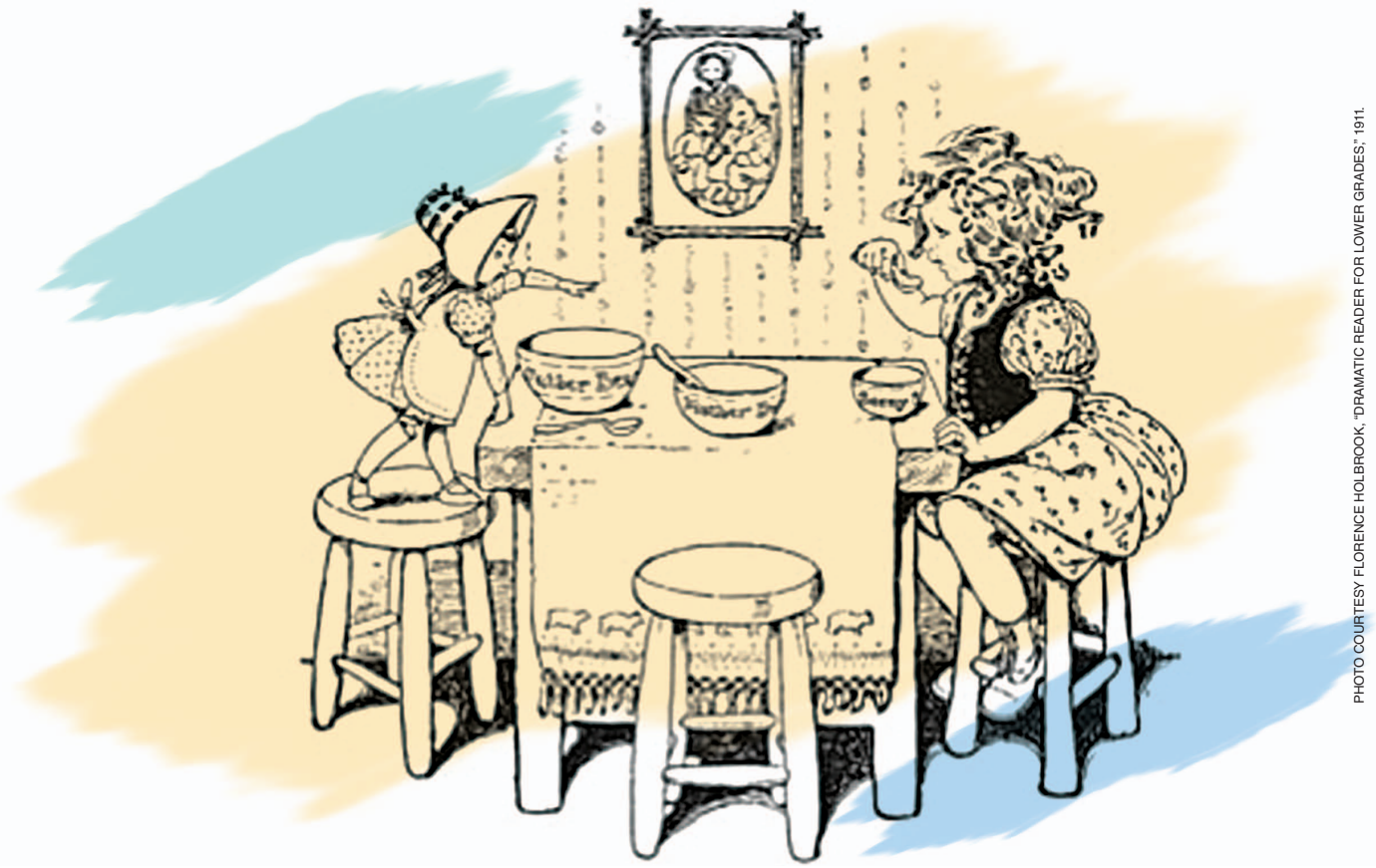

PHOTO COURTESY FLORENCE HOLBROOK, "DRAMATIC READER FOR LOWER GRADES," 1911.

Goldilocks says, "Ultra-low is too slow. Nominal is too hot. Near-threshold is just right!"

Nathaniel Pinckney,
David Blaauw, and Dennis Sylvester

Digital Object Identifier 10.1109/MSSC.2015.2418151
Date of publication: 25 June 2015

Energy-efficient near-threshold design has been proposed to increase energy efficiency across a wide range of applications. This article first provides a background motivating near-threshold and how it differs from super-threshold and subthreshold operation. Next, state-of-the-art near-threshold techniques are summarized that help overcome barriers to near-threshold adoption, namely high variation at low voltage. Last, example industrial and academic wide-voltage scaling systems are discussed.

INTRODUCTION

Background

A major problem facing the semiconductor industry is increased power and energy densities, caused by the inability to scale power supply voltage. As transistor density continues to increase, a corresponding reduction in the power dissipation per transistor must be realized. Without this corresponding reduction large portions of a processor will be inactive at any given time, leading to the term *dark silicon* [H. Esmaeilzadeh et al., *ISCA*, 2011]. Instead of moderate improvements in power, area, and performance generation-to-generation as was the norm with Dennard Scaling [R. Dennard et al., *JSSC*, 1992], today's process advancements now happen in bursts and generational gain is no longer a guarantee. Therefore, it is now imperative to exploit cutting-edge circuit and architectural techniques for improving microprocessor energy efficiency.

Supply voltage is a fundamental design parameter available to circuit and system designers for improving energy efficiency because of the strong quadric relationship of dynamic energy on voltage. Typical processors are designed to operate at the nominal, *super-threshold*, voltage for a given process technology, dictated by reliability and power. Traditional dynamic voltage and frequency scaling (DVFS) techniques are used to adjust a chip's performance and power consumption during runtime, slowing the clock and lowering the supply during periods of low utilization. Processor cores are conventionally designed to operate at the highest voltage during normal workloads, subject to thermal and power constraints.

A major problem facing the semiconductor industry is increased power and energy densities, caused by the inability to scale voltage.

Within the past decade there has been increased interest in ultra-low voltage (ULV) designs, primarily limited to special applications, such as low-power autonomous sensor nodes [W. Lim et al., *ISSCC*, 2015] or specialized hardware accelerators [A. Wang et al., *ISSCC*, 2004; D. Jeon, *JSSC*, 2012]. ULV applications generally have low performance requirements or are algorithms that tolerate slow clock frequencies by easily scaling across additional hardware. ULV designs usually operate at the minimum energy point, below which leakage power dominates and sometimes even lower to further save power at the cost of energy.

Between the two extremes of nominal voltage and subthreshold operation is near-threshold computing (NTC; the term near-threshold design was first used in [B. Zhai et al., *ISLPED*, 2007]), which trades some performance loss for moderate energy gain, as shown in Figure 1. Performance loss from slowed clock frequency can be regained through parallelization by adding cores to maintain reasonable latency and throughput for many

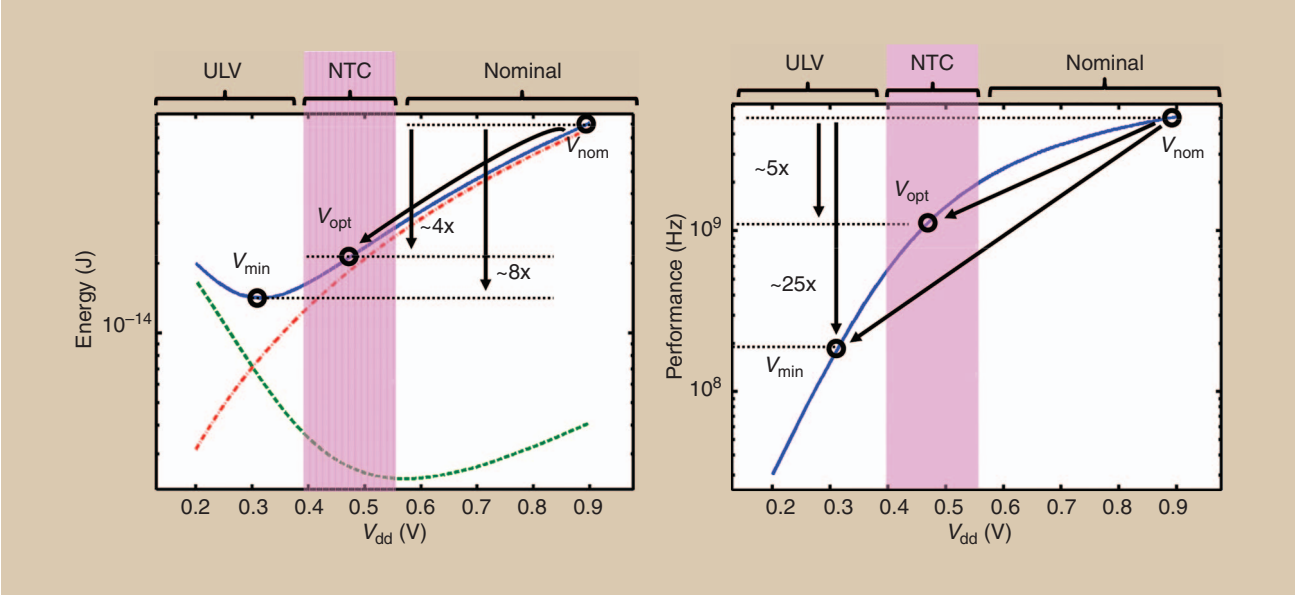

FIGURE 1: Energy versus supply voltage for three ranges of operation (nominal, ultra-low voltage, and near-threshold). NTC balances performance loss and energy gain.

applications. Unlike ULV designs, NTC is intended for general-purpose computing applications that have moderate performance requirements and do not parallelize perfectly.

Near-Threshold

The idea of sacrificing performance for improved power or area is not new; instead, it is fundamental to circuit design. An early work argued [A. Chandrakasan et al., JSSC, 1992] that architectural parallelism, through pipelining and replicating function blocks such as ALUs allows for reasonable performance while leveraging voltage scaling to minimize power (though not necessarily minimum energy). By using an analytical model of power and performance tradeoffs, architectural parallelism was shown to be more effective than technological techniques, such as transistor sizing, for achieving good speed after voltage scaling.

Recent microprocessor architecture papers [B. Zhai et al., *ISLPED*, 2007; R.G. Dreslinski et al., *Proc. IEEE*, 2010] have advocated for parallelizing a workload across many low-voltage, energy-efficient cores, distributing the task to balance the slow clock frequency from low voltage operation. Despite using more cores to run the task than at nominal voltage, energy savings can be achieved because dynamic energy has a quadratic dependence on voltage, $E_{\text{dynamic}} \propto Vdd^2$, yet the number of cores needed is initially linear with Vdd. This can be seen from the task runtime's dependence on supply voltage, $T_{\text{task}} \propto Vdd/(Vdd - V_t)^2$, and simplifies to $T_{\text{task}} \propto 1/Vdd$ if supply voltage is much higher than threshold voltage. Therefore, if a task can be parallelized across cores with little overhead, only a linear number of cores must be added to match task completion time, whereas quadratic energy savings are achieved. Of course, at supply voltages close to threshold these assumptions break down.

Even for performance insensitive applications, achievable energy gains are limited by static energy (from leakage currents), which becomes dominant at very low voltages. Since task completion time increases exponentially close to threshold, if performance is constrained then the number of cores needed for parallelization rapidly increases at low voltage. In [Pinckney et al., *DAC*, 2012] we provided a systemic definition of near-threshold to better understand how close to threshold is practical for many workloads and then using this definition to examine trends across technology nodes. In this methodology, energy is minimized subject to a performance constraint; specifically, that latency is fixed to that of a single core running a workload at high voltage.

As core voltage is reduced and its clock frequency decreases, a workload is parallelized across cores until the target latency is achieved. This iso-latency analysis is workload-dependent, and parallelization overheads, arising from algorithmic and architectural sources, are assessed through system-level simulations of the SPLASH-2 benchmark suite [S.C. Woo et al., *ISCA*, 1995]. Additionally, circuit energy and performance scaling are simulated with SPICE models of six industrial processes from 180 nm to 32 nm. A key finding is that, across the scientific benchmarks studied, the near-threshold region tracked roughly 200–400 mV above V_t, as shown in Figure 2. Across the SPLASH-2 benchmarks in 32 nm, parallelism across 12 cores is needed on average. Additionally, NT energy gain was decreasing from 8× in 180 nm to

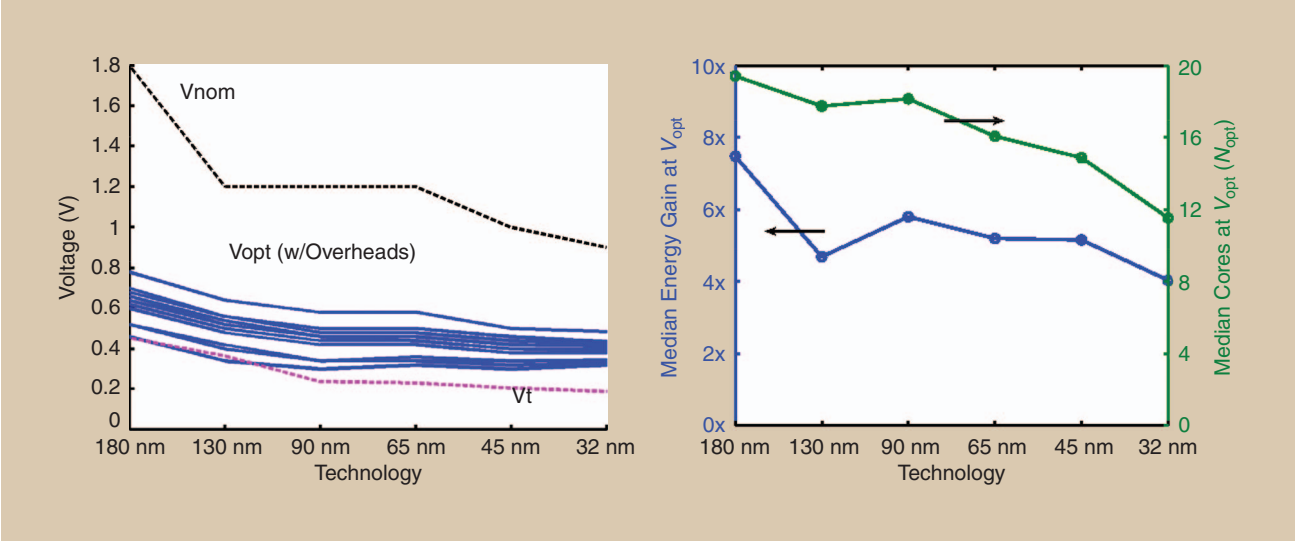

FIGURE 2: Left, minimum energy points, blue lines, when considering algorithmic and architectural parallelism overheads for SPLASH-2 benchmarks across six technology nodes. This near-threshold region tracks approximately 200 mV–400 mV above threshold voltage, purple line. Nominal voltage is shown with the black line. Right, median energy gain from running at near-threshold instead of nominal, and number of cores parallelized across SPLASH-2 benchmarks. Near-threshold had ~8× energy gain in 180 nm and has reduced to 4× in 32 nm.

4× in 32 nm, and therefore is becoming less effective as planar technologies reach end of life.

Near-Threshold Techniques

Near-threshold computing presents a number of challenges that differ from super-threshold or ultra-low voltage operation and that must be overcome before energy improvements can be realized. Variability in path delays and susceptibility to noise is much worse in near-threshold compared to super-threshold operation. Additionally, achieving good performance across a wide voltage range is critical to enable worthwhile dynamic adjustment of a core's efficiency against varying workloads.

Heterogeneous Architectures and Supply Boosting

Parallelism of near-threshold many-core systems need not be across identical cores running on a single, shared voltage supply. Instead, the availability of different voltages, architectures, and accelerators aid improving efficiency for a variety of workloads. An example of a heterogeneous system with multiple independent core architectures is ARM's big.LITTLE processor arrangement [P. Greenhalgh, ARM white paper, 2011], in which large high-performance cores are used when speed is critical; otherwise small low-power cores are used for high efficiency.

If supply voltage can be controlled independent for each core in a many-core processor, the system is in effect heterogeneous except through different voltages instead of different microarchitectures. Nominal voltage cores are used for high-performance workloads while near-threshold cores for maximum efficiency. Dynamic voltage adjustment allows for the system to adapt to workload during runtime, further increasing efficiency. We envision NTC processors featuring an array of cores operating at low-voltage for most workloads, and with the ability to raise the operating voltage of cores when fast single-threaded performance is required.

Dynamic voltage scaling techniques, such as with off-chip chip regulators, have been used in commercial chips for years. Recently there has been a push toward moving regulators on-chip, enabling voltage control of individual core, with examples from Intel, IBM, and Berkeley [E. Burton, *APEC* 2014; Z. Toprak-Deniz, *ISSCC* 2014; Jevti, *VLSI* 2014]. For instance, Intel's Haswell Xeon microprocessor features fully-integrated voltage regulators (FIVRs) and uses air core inductors in the processor's package to individually regulate each of the processor core supplies [B. Bowhill et al., *ISSCC* 2015].

An alternative to using on-chip regulators for per-core adjustment is proposed in [Pinckney et al., *VLSI*, 2013]. The technique, called *Shortstop*, uses three external supplies to quickly raise the voltage of a core, within tens of nanoseconds, without inducing supply droop. This is achieved by leveraging parasitic inductance of a package similar to a boost converter arrangement. A transient supply rail is temporarily shorted to ground to energize its associated parasitic inductance before being connected to a core. The energized inductor is then able to quickly transfer energy to the core's virtual supply rail, charging the core's intrinsic capacitance. An added on-chip capacitor provides an initial boost to the core, and also to capture remaining energy in the inductor after rising of the core's supply rail is complete. The 28-nm wire-bonded demonstration chip was able to raise the voltage of a core 1.7× than PMOS headers and with 3.5–6× less droop. An update to previously mentioned near-threshold study [Pinckney et al., *DAC*, 2012] shows that ability to boost a core for high performance leads to modest energy gains, especially for those applications that are only mildly parallelizable, with a 15–60% Amdahl serial coefficient [Pinckney et al., *IEEE Micro*, 2013].

Path Variability

The increased sensitivity to power supply and process variability can be shown through basic modeling of the transistor. The "on" current of a transistor in saturation can be approximated using the alpha-power law [T. Sakurai et al., *JSSC*, 1990]

$$\mathrm{I_{on}} \propto \frac{W}{L}(V_{\mathrm{dd}} - V_t)^{\alpha}.$$

As Vdd is lowered, the current becomes more sensitive to changes in the transistor's threshold voltage, $\triangle V_t$

$$\mathrm{I_{on}} = I_{\mathrm{on-nominal}}\left(1 - \frac{\alpha}{Vdd - V_t} \triangle V_t\right).$$

Pelgrom's law, $\sigma_{V_t} = (A_{V_t}/\sqrt{L * W})$, tells that the variation in V_t is inversely related to the square of the gate area. Therefore, upsizing transistors or increasing gate length will improve V_t. Also, longer path lengths exhibit less V_t variation from mismatch sources, such as random dopant fluctuations (RDF), as uncorrelated noise averages out. Thus, mismatch variation is especially bad for short paths.

A common technique to improve voltage scalability for standard cell-based design flows is to remove gates exhibiting poor delay variation at low voltages, such as tall transistor stacks and wide transmission gate logic [S. Jain et al., *ISSCC*, 2012]. In a 32-nm process, four-stack gates were found to have 108% increase in delay variability at 300 mV as compared to three-stack, Figure 3. Similarly, four-wide transmission gate multiplexers also had more than double delay variability as three-wide. Limiting transistor selection to low-threshold devices improves headroom, $V_{\mathrm{dd}} - V_t$, reducing delay variability. Last, excluding gates with small sized devices, or upsizing those devices, improves variability because of Pelgrom's law mentioned above.

Sequential Element Data Retention and Hold Times

Latch keepers are sized to be weak; therefore, they generally have a

small gate area and subsequently high variability. This is exacerbated as voltage is reduced, causing static noise margin collapses and data retention failures [C.H. Chen et al., *ISLPED*, 2013]. One way to improve the minimum retention voltage V_{min} is to upsize keepers, thereby reducing sensitivity to process variation at the cost of increased area, power, and delay. In a 32-nm design, upsizing interrupted keepers decreased V_{min} by 100 mV [Jain et al., ISSCC, 2012].

Transmission gate latches and flip-flops (TGFFs) exhibit dramatically increased variation in hold time at low voltages, primarily due to misalignment of transmission gate and feedback tristate clock signals [C.H. Chen et al., *ISLPED*, 2013]. The complemented clock signal is usually driven by a single small inverter that is very susceptible to mismatch variation due to its size. Hold time violations occur with short paths by definition, which have few gates. Uncorrelated variation averages out over long logic chains but remains significant if the number of gates is small, as with short logic paths and the single clock inverter in a TGFF. This problem is exacerbated at near-threshold since circuit delay is more sensitive to threshold voltage variation.

Prototype near-threshold systems have been published from both academia and industry.

Increased hold time and short variation can be fixed through adding additional hold time buffers. However, this can significantly increase area and power of a design depending on switching activities of the paths affected. Furthermore, a larger area can negatively impact performance due to longer wire loads between cells. More subtly, a logic path may branch and simultaneously share a short path and critical path. Therefore, by fixing hold time violations critical paths may also be affected and impact performance of the design.

Single-phase flip-flops can help alleviate hold time variation since the clock's complement is not used. Instead only a single phase is used throughout the cell, removing the problem of skew between true and complement signals. An example single phase sequential element is the Static Single-Phase Contention-Free Flip-Flop (S^2CFF) [Y. Kim et al., *ISSCC*, 2014] that exhibits 3.4× lower 3-sigma hold time variation at low voltage, Figure 4.

Clock Skew and Timing

Clock distribution, such as clock trees and meshes, exhibits increased clock skew in near-threshold because of delay variation of clock buffers. Modifying the clock network topology can improve skew, for example by exploiting the fact that circuit delay becomes dominant at low voltages compared to wire RC delay. This

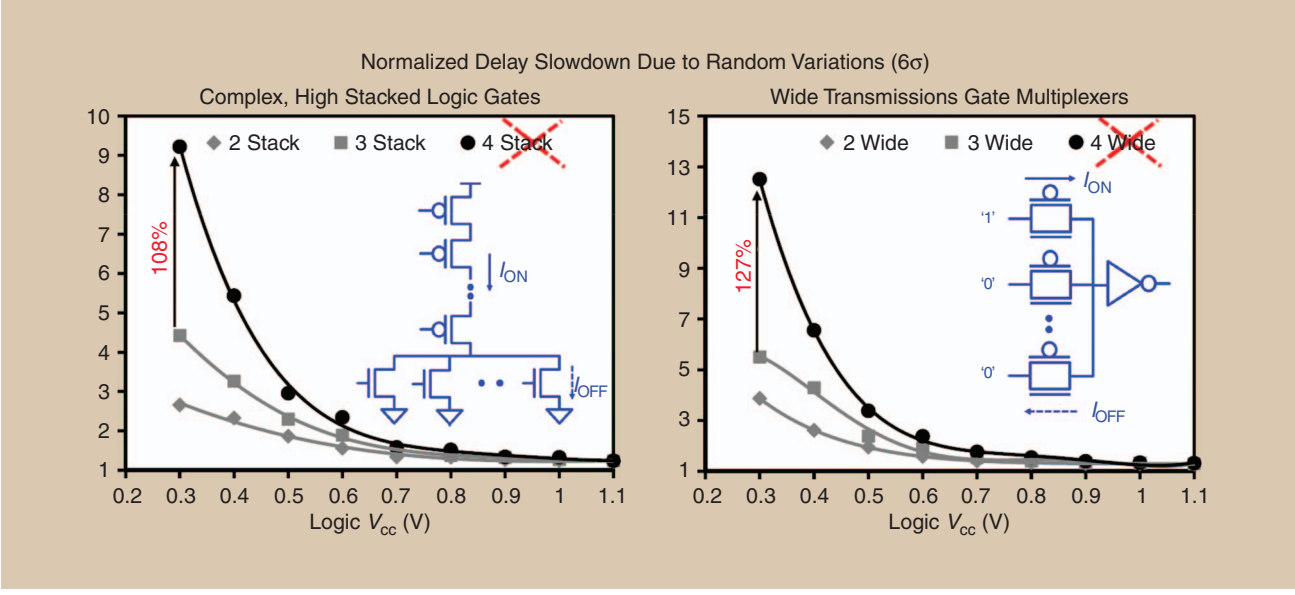

FIGURE 3: High stacked logic gates and wide transmission or pass logic gates exhibit high variation at low voltage. In this example, four-stack gates and four-wide TG multiplexers are pruned since they both have > 100% more variation then three-stack or three-side when Vdd = 300 mV.

Clock distribution exhibits increased clock skew in near-threshold because of delay variation of clock buffers.

is because circuit delay increases rapidly at low Vdd while wire RC remains fixed. Therefore, it is possible to optimize a clock network for low-voltage operation by reducing or eliminate clock buffers. A low-skew low-voltage network will favor few levels of clock hierarchy and large buffers, as opposed to many levels of hierarchy and small buffers in traditional designs [M. Seok et al., *JETCAS*, 2011]. Additionally, variation-tolerant techniques, such as *soft clocking* with latches or specially designed flip-flops, can be used to reduce sensitivity to clock skew along logic paths [M. Wieckowski et al., *CICC*, 2008]. A 16-bit multiplier using time borrowing with two-phase latch-based clocking shows 1.6–3.6× speedup and 18–30% lower energy as compared to a conventional flip-flop design [M. Seok et al., *DAC*, 2011].

Logic delay also exhibits more variation at near-threshold and margins for this can impact performance and power budgets. Conventionally chips are designed for worst-case margins, including process, voltage, temperature, and noise sources. However, these sources rarely occur simultaneously, and therefore chips are over-margined from the typical case, wasting energy, and lowering performance. Margin reduction techniques, such as critical-path and in-situ monitoring, help to regain lost performance and power. Replica critical-path monitoring features multiple replica circuits to track delay behavior of critical paths in the design. These can be helpful for slow changing or fixed variations, such as global process skew and temperature. In situ error detection and correction techniques [D. Ernst et al., *IEEE Micro*, 2003; M. Fojtik et al., *ISSCC*, 2012; S. Kim et al., *ISSCC*, 2013] take replica monitoring a step further and detect errors on critical paths themselves, therefore removing margin for power supply noise and within-die variation, though these have yet to be proven on large commercial designs. A recent in-situ monitor focused on voltage scalability by avoiding clock-to-q mismatch between data and shadow latches used for error detection, and was found to operate down to 300 mV in 65 nm [S. Kim et al., *VLSI-C*, 2014]. Combined with soft clocking through time borrowing, the proposed design demonstrates a 59% decrease in energy and 4.9× higher throughput at low voltage, as shown in Figure 5.

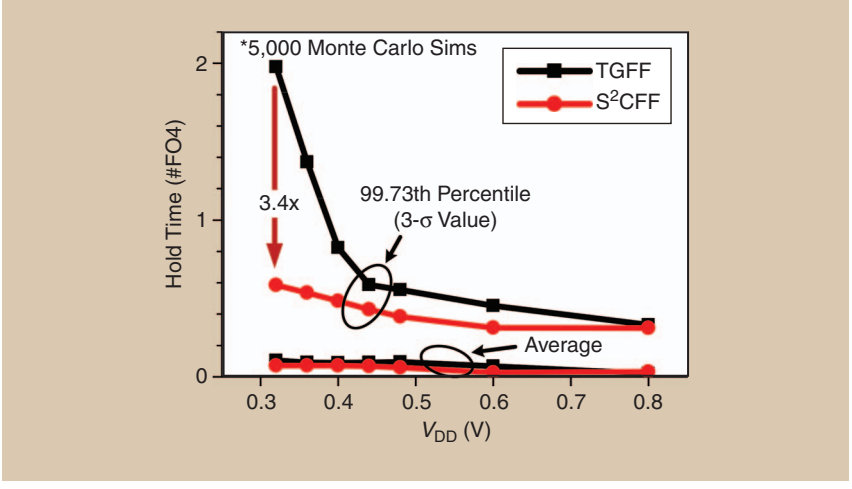

FIGURE 4: Hold time mean and 3-sigma for conventional transmission gate flip-flop (TGFF) versus improved flip-flop called S²CFF. Improved flip-flop has 3.4× lower 3-sigma variation in hold time then TGFF at 0.3 V. Therefore, hold time variability is much better for low-voltage operation, easing min-time buffering and margining.

SRAM and Caches

SRAMs pose a unique and formidable issue in near-threshold designs because they are very sensitive to mismatch and have high density. Within-die variation that may barely impact logic could still have a large effect on SRAM arrays because of the large number of bit cells. Operating below nominal voltage reduces SRAM read, write, and retention margins and eventually, if voltage continues to be reduced, margins may be completely eliminated leading to functional failures [G. Chen et al., *TVLSI*, 2010]. Upsizing SRAM bit-cell transistors is costly since area proportionally increases, reducing array density. Mismatch variation, such as random dopant fluctuations (RDFs), cause large differences between individual bit cells, so that any mitigation through tuning or calibration is difficult. Lastly, by design bit lines experience small-swings. Sense amps are used to detect small differential signals. Therefore, reducing operating voltage of an array does not imply bit-line power savings [G. Chen et al., *ICCAD*, 2007]. Additionally, memory arrays tend to be leakage-dominated, so any dynamic energy savings may be lost because of increased static energy from slower operation.

Alternatives to 6T SRAM for improving low-voltage robustness have been proposed. A single-ended 6T cell uses upsized gates to reduce variation, and gating of virtual power and ground rails of the bit-cell feedback inverters, to improve write margins [B. Zhai et al., *JSSC*, 2008]. The proposed SRAM

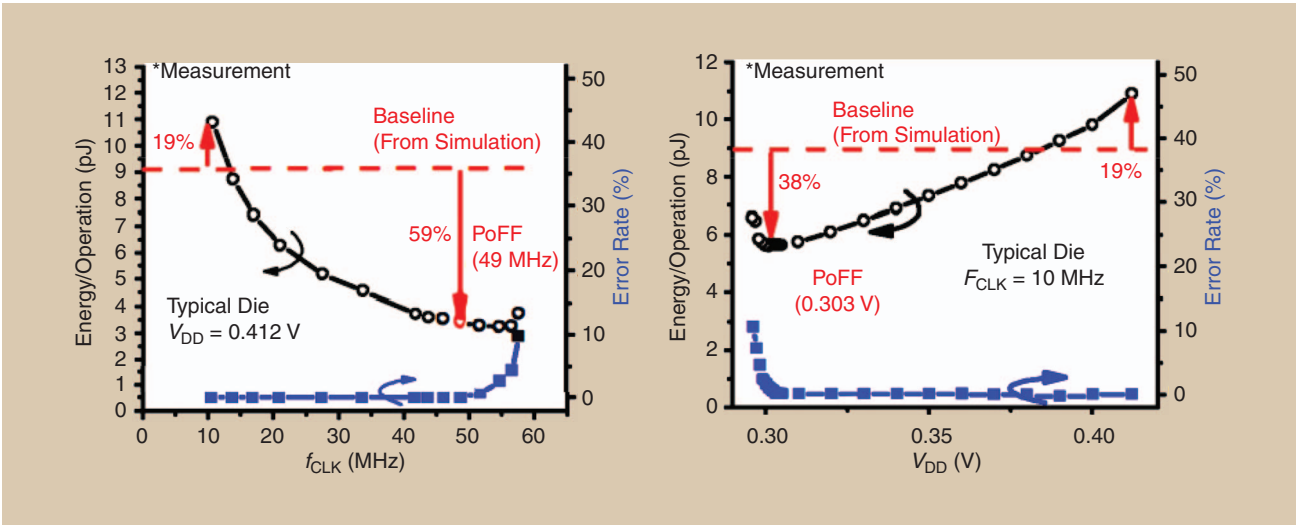

FIGURE 5: Energy efficiency increases by allowing higher frequency for a fixed voltage, left, or low voltage for a fixed frequency. Energy per operation decreases until error rate rises significantly passed point of first failure (PoFF).

was able to operate in subthreshold, down to 193 mV in a 130-nm process with a 1.2-V nominal operating voltage. Other designs have sought to decouple bit-cell reads from writes, such as 8T SRAM [L. Chang et al., *VLSIT*, 2005]. These allow for easier sizing optimizations since reads and writes are decoupled but could suffer from read disturbs of half-selected cells, and may suffer from slow or unreliable reads because single-ended reads are used. Even larger bit cells have been proposed, such as 9T or 10T designs, which remedy these problems at the cost of decreased array density [I.J. Chang et al., *JSSC*, 2009].

Level Shifters

The increase in the number of cores on a chip multiprocessor leads to an increase in the number of power domains, each requiring level shifters on domain boundaries. Differential cascode voltage switch (DCVS) level shifters rely on low-voltage driven pull-down NMOSs overcoming weak PMOS headers to obtain full high-voltage swing [Y. Kim et al., *VLSI Circuits*, 2011]. As the low supply voltage is further decreased, the circuit becomes increasingly susceptible to PVT variation and may lead to functional failures of the level shifter. Even in the absence of functional failures, circuit delay increases rapidly for DCVS level shifters topologies leading to timing failures.

Improved level shifter topologies help mitigate these issues, especially for ultra-low voltage operation, but are still applicable to near-threshold as well. An example improved ULV topology is the ultra-low voltage split-output (ULVS) level shifter that reduces contention by inserting a PMOS between the contending devices [S. Hsu et al., *JSSC*, 2013]. Shown in Figure 6, the ULVS topology lowers V_{min} by 125 mV when demonstrated in 22 nm.

Variability remains one of the biggest challenges for low-voltage operation.

Example Implementations

Prototype near-threshold systems have been published from both academia and industry. The

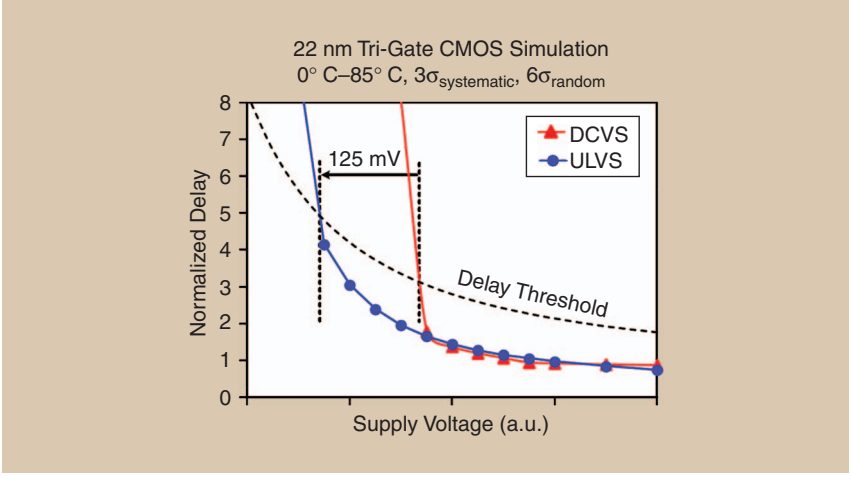

FIGURE 6: Traditional differential cascode voltage switch (DCVS) level shifters are designed with contention which can have fail at low voltages. By improving level shifter topology and reducing device contention, supply voltage can be reduced without failures.

Near-threshold holds promise for improving the energy efficiency of modern processors as process scaling continues to slow.

early low-power designs focused on pushing minimum voltage as low as possible, while ensuring no functional errors [A. Wang, *ISSCC*, 2004]. Recent interest has shifted to wide-range voltage scaling designs, which strive to maintain good performance and energy across a wide voltage range, from near-threshold to super-threshold.

An academic example of a many-core, 3D-stacked processor and DRAM concept is the Centip3De project from the University of Michigan [D. Fick et al., *ISSCC*, 2012]. The 90-nm system is arranged in clusters of four cores, where one to four cores can be active depending if the workload demands are for single-threaded performance or high energy efficiency. In single-core mode, the ARM Cortex-M3 processors run at 1.15 V and 80 MHz with an efficiency of 860 DMIPS/W. In four-core mode, the processors run at 0.65 V and 10 MHz with a peak efficiency of 3930 DMIPS/W, over 4.5 times as efficient compared to single core mode. The L1 caches run at integer multiples of the core frequencies to time-multiplex requests reducing collisions in cache access.

A wide-range voltage scaling Pentium-class, IA-32 core from Intel demonstrates respectable performance from 280 mV to 1.2 V [S. Jain et al., *ISSCC*, 2012]. Characterization of the standard cell libraries was done at both high and low voltages to optimize and identify wide-range scaling performance. Standard cells with poor delay scaling in the presence of variation were pruned from the library. The minimum energy for the 32-nm test chip was observed at 450 mV for the core supply, reducing energy consumption by 4.7× over nominal 1.2-V operation, shown in Figure 7. The memory is on a separate power domain than logic, so as to not violate the 0.55-V memory retention voltage. Synthesis targets were very important as well, with the low-voltage targeted design achieving approximately 67% clock frequency than a high-voltage targeted design scaled down. The design also features programmable delay buffers to tune for skew between processor blocks.

Near-threshold many-core architectures will require efficient interconnect fabrics that scale well to different voltages. Intel proposed a wide-range, 340 mV-to-0.9 V, FinFET network-on-chip demonstrated in 22 nm FinFET [G. Chen et al., *ISSCC*, 2014]. The chip is partitioned into a 16 × 16 mesh with 256 separate power and clock domains. The energy efficiency of the network, highest bandwidth per watt, was achieved at 400 mV and 18.3 Tb/s/W as compared to 7.0 Tb/s/W at 0.9-V nominal supply. Specialized wide-range near-threshold processor blocks, such as the SIMD permutation engine, have also been explored by Intel [S. Hsu et al., *ISSCC*, 2012].

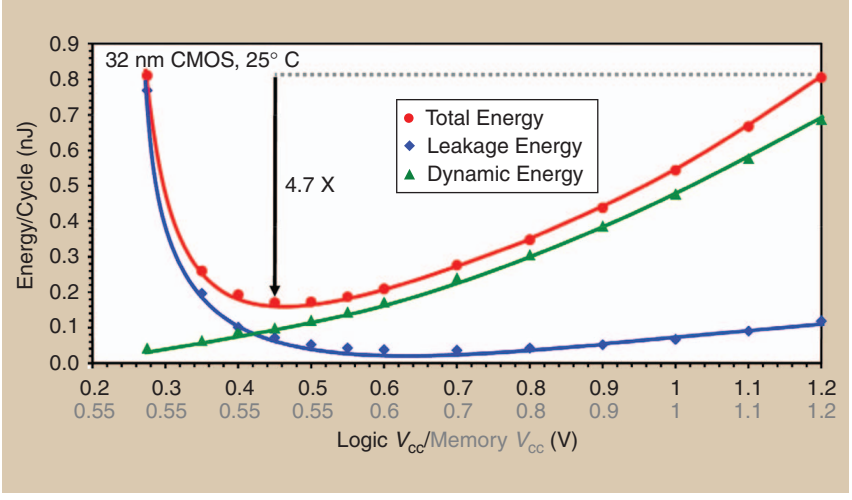

FIGURE 7: A wide-range Intel IA-32 processor exhibits minimizes energy at 450 mV, 4.7× less than nominal 1.2-V operation. Below 450 mV, leakage dominates total energy.

Conclusions

Near-threshold holds promise for improving the energy efficiency of modern processors as process scaling continues to slow. Increased transistor packing densities will enable advanced voltage scaling techniques. Initial near-threshold designs have shown promise, yet many obstacles remain before NTC can be widely adopted. Variability remains one of the biggest challenges for low-voltage operation but variation tolerant techniques, such as soft clocking and in-situ monitoring, can help mitigate these issues. Improved topologies for blocks that traditionally scale very poorly because of sensitivity to mismatch, such as SRAMs, demonstrate that low-voltage operation is possible.

References

1) *Background + Near-Threshold Computing*

[1] R. Dennard et al., "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE J. Solid-State Circuits*, 1974.

[2] A. Chandrakasan et al., "Low-power CMOS digital design," *IEEE J. Solid-State Circuits*, 1992.

[3] R. G. Dreslinski et al., "Near-threshold computing: reclaiming Moore's law through energy efficient integrated circuits," *Proc. IEEE*, Feb. 2010.

[4] H. Esmaeilzadeh et al., "Dark silicon and the end of multicore scaling," in *Proc. ISCA*, 2011.

[5] N. Kurd et al., "Next generation Intel core micro-architecture (Nehalem) clocking," *IEEE J. Solid-State Circuits*, 2009.
[6] N. Pinckney et al., "Assessing the performance limits of parallelized near-threshold computing," in *Proc. DAC*, 2012.
[7] S. C. Woo et al., "The SPLASH-2 programs: characterization and methodological considerations," in *Proc. ISCA*, 1995.
[8] N. Pinckney et al., "Limits of parallelism and boosting in dim silicon," *IEEE Micro*, 2013.
[9] M. B. Taylor, "Is dark silicon useful?: harnessing the four horsemen of the coming dark silicon apocalypse," in *Proc. DAC*, 2012.
[10] B. Zhai et al., "Energy efficient near-threshold chip multi-processing," in *Proc. ISLPED*, 2007.
[11] W. Lim et al., "Batteryless sub-nW Cortex-M0+ processor with dynamic 9:00 AM leakage-suppression logic," in *Proc. ISSCC*, 2015.
[12] D. Jeon, "A super-pipelined energy efficient subthreshold 240 MS/s FFT core in 65 nm CMOS," *IEEE J. Solid-State Circuits*, , 2012.
[13] P. Greenhalgh, "Big.little processing with arm cortex-a15 & cortex-a7," ARM White Paper, 2011.
[14] T. Sakurai et al., "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, 1990.
2) *Per-Core Voltage Control and Regulation*
[15] E. A. Burton et al., "FIVR—fully integrated voltage regulators on 4th generation Intel Core SoCs," in *Proc. APEC*, 2014.
[16] R. Jevtić, "Per-core DVFS with switched-capacitor converters for energy efficiency in many core processors," in *Proc. VLSI Systems*, 2014.
[17] N. Pinckney et al., "Shortstop: An on-chip fast supply boosting technique," in *Proc. VLSI Circuits*, 2013.
[18] Z. Toprak-Deniz et al., "Distributed system of digitally controlled microregulators enabling per-core DVFS for the POWER8 microprocessor," in *Proc. ISSCC*, 2014.
[19] B. Bowhill et al., "The Xeon® Processor E5-2600 v3: A 22nm 18-core product family," in *Proc. ISSCC*, 2015.
3) *Clocking + Sequential Elements*
[20] C. H. Chen et al., "Minimum supply voltage for sequential logic circuits in a 22nm technology," in *Proc. ISLPED*, 2013.
[21] D. Ernst et al., "Razor: a low-power pipeline based on circuit-level timing speculation," *IEEE Micro*, 2003.
[22] M. Fojtik et al., "Bubble razor: an architecture-independent approach to timing-error detection and correction," in *Proc. ISSCC*, 2012.
[23] S. Kim et al., "Razor-lite: A side-channel error-detection register for timing-margin recovery in 45nm SOI CMOS," in *Proc. ISSCC*, 2013.
[24] S. Kim et al., "R-processor: 0.4V resilient processor with a voltage-scalable and low-overhead in-situ error detection and correction technique in 65nm CMOS," in *Proc. VLSIC*, 2014.
[25] Y. Kim et al., "A static contention-free single-phase-clocked 24T flip-flop in 45nm for low-power applications," in *Proc. ISSCC*, 2014.
[26] M. Seok et al., "Pipeline strategy for improving optimal energy efficiency in ultra-low voltage design," in *Proc. DAC*, 2011.
[27] M. Seok et al., "Robust clock network design methodology for ultra-low voltage operations," in *Proc. JETCAS*, 2011.
[28] M. Wieckowski et al., "Timing yield enhancement through soft edge flip-flop based design," in *Proc. CICC*, 2008.
4) *Level Conversion*
[29] Y. Kim et al., "LC2: Limited contention level converter for robust wide-range voltage conversion," in *Proc. VLSI Circuits*, 2011.
5) *SRAM + Cache Architecture*
[30] I. J. Chang et al., "A 32 kb 10T sub-threshold SRAM array with bit-interleaving and differential read scheme in 90 nm CMOS," *IEEE J. Solid-State Circuits*, , 2009.
[31] L. Chang et al., "Stable SRAM cell design for the 32 nm node and beyond," in *Proc. VLSIT*, 2005.
[32] G. Chen et al., "Yield-driven near-threshold SRAM design," in *Proc. TVLSI*, 2010.
[33] R. G. Dreslinski et al., "Reconfigurable energy efficient near threshold cache architectures," *Micro*, 2008.
[34] B. Zhai et al., "A variation-tolerant sub-200 mV 6-T subthreshold SRAM," *IEEE J. Solid-State Circuits*, 2008.
6) *Near-Threshold Implementations*
[35] A. Wang et al., "A 180mV FFT processor using subthreshold circuit technique," in *Proc. ISSCC*, 2004.
[36] G. Chen et al., "A 340mV-to-0.9V 20.2Tb/s source-synchronous hybrid packet/circuit-switched 16×16 network-on-chip in 22nm tri-gate CMOS," in *Proc. ISSCC*, 2014.
[37] D. Fick et al., "Centip3De: A 3930DMIPS/W configurable near-threshold 3D stacked system with 64 ARM Cortex-M3 cores," in *Proc. ISSCC*, 2012.
[38] S. Hsu et al., "A 280 mV-to-1.1 V 256b reconfigurable SIMD vector permutation engine with 2-dimensional shuffle in 22 nm tri-gate CMOS," *IEEE J. Solid-State Circuits*, 2013.
[39] S. Jain et al., "A 280mV-to-1.2V wide-operating-range IA-32 processor in 32nm CMOS," in *Proc. ISSCC*, 2012.
[40] D. Marković et al., "Ultralow-power design in near-threshold region," *Proc. IEEE*, Feb. 2010.

About the Authors

Nathaniel Pinckney is a Ph.D. student at the University of Michigan. He has authored or coauthored over 25 publications in the areas of low-power, near-threshold VLSI design and cryptographic accelerators. He received his bachelor of science degree from Harvey Mudd College in 2008 and his master of science from the University of Michigan in 2012. Prior to joining the University of Michigan, he worked in Sun Microsystems' VLSI research group. He is a Member of the IEEE.

David Blaauw received his B.S. in physics and computer science from Duke University in 1986 and his Ph.D. in computer science from the University of Illinois, Urbana, in 1991. After his studies, he worked for Motorola, Inc. in Austin, Texas, where he was the manager of the High Performance Design Technology Group. Since August 2001, he has been on the faculty at the University of Michigan where he is a professor. He has published over 450 papers and holds 40 patents. His work has focused on VLSI design with particular emphasis on ultra-low-power and high performance design. He was the technical program chair and general chair for the International Symposium on Low Power Electronic and Design. He was also the technical program cochair of the ACM/IEEE Design Automation Conference and a member of the ISSCC Technical Program Committee.

Dennis Sylvester received a Ph.D. from the University of California, Berkeley, and is professor of electrical engineering and Computer Science at the University of Michigan, Ann Arbor. He is the director of the Michigan Integrated Circuits Laboratory (MICL), a group of ten faculty and 70+ graduate students. He has held research staff positions in the Advanced Technology Group of Synopsys and Hewlett-Packard Laboratories and visiting professorships at the National University of Singapore and Nanyang Technological University. He has published over 400 articles along with one book and several book chapters. His research interests include the design of millimeter-scale computing systems and energy efficient near-threshold computing. He holds 22 U.S. patents and serves as a consultant and technical advisory board member for electronic design automation and semiconductor firms in these areas. He cofounded Ambiq Micro, a fabless semiconductor company developing ultra-low-power mixed-signal solutions for compact wireless devices. He is an IEEE Fellow.

SSC