

REVIVAL: A VARIATION-TOLERANT ARCHITECTURE USING VOLTAGE INTERPOLATION AND VARIABLE LATENCY

PROCESS VARIATIONS WILL SIGNIFICANTLY DEGRADE THE PERFORMANCE BENEFITS OF FUTURE MICROPROCESSORS AS THEY MOVE TOWARD NANOSCALE TECHNOLOGY. DEVICE PARAMETER FLUCTUATIONS CAN INTRODUCE LARGE VARIATIONS IN PEAK OPERATION AMONG CHIPS, CORES ON A SINGLE CHIP, AND MICROARCHITECTURAL BLOCKS WITHIN ONE CORE. THE REVIVAL TECHNIQUE COMBINES THE POST-FABRICATION TUNING TECHNIQUES VOLTAGE INTERPOLATION (VI) AND VARIABLE LATENCY (VL) TO REDUCE SUCH FREQUENCY VARIATIONS.

Xiaoyao Liang
Gu-Yeon Wei
David Brooks
Harvard University

• • • • • Advances in CMOS device technology have been a driving force in the computing industry by providing ever-smaller transistors that lead to tremendous system integration benefits. Processor designers have capitalized on these improvements by providing more capable and higher-performance computing architectures. However, in the most advanced fabrication nodes (65 nanometer and beyond), difficulties in the manufacturing process manifest as potentially large variations in transistors' performance and power dissipation. These variations threaten to stall performance improvements sought through further reductions in minimum feature sizes.

Process variations occur at multiple spatial scales. Variations at the wafer level lead

to performance differences between individual microprocessor dies, but increasingly, process variations are becoming more fine-grained. Uneven mask exposure due to lithography limitations leads to systematic variations in transistor gate length at the microarchitectural block level. Random dopants fluctuations can change the threshold voltage of individual devices. Unlike die-to-die (D2D) variations, speed-binning techniques can't easily solve within-die (WID) and random variations because a handful of slow transistors can potentially lead to slow speed paths that affect the processor's overall clock frequency. For traditional synchronous designs, this can lead to a significant reduction in system performance for most fabricated chips.¹

In this article, we study the impact of applying two fine-grained post-fabrication tuning techniques, voltage interpolation (VI) and variable latency (VL), to individual microarchitectural units. Both techniques let us flexibly adapt blocks to different degrees of process variation. VI in effect provides a unique voltage for individual blocks by spatially dithering the supply voltage of logic within the blocks to operate off one of two distinct levels: a higher voltage (V_{DDH}) and a lower voltage (V_{DDL}).

The combination of VI with VL pipelines, which we call the *Revival technique*, offers significant advantages over using either in isolation.^{2–4} Our results show that this approach gives us a wide frequency-tuning range to deal with delay fluctuations arising from process variations with minimal power overhead. Even when compared to a comparatively expensive solution with per-core voltage control, our proposed technique can improve the energy-delay-squared metric (ED^2), commonly used to compare the energy and performance efficiency of different designs, by 21 percent.

Combating process variations

Process variation can greatly degrade the fabricated chips' maximum operating frequency. Previous research showed that variations can eat up to 30 percent of frequency gains sought by moving processor designs to the next technology node.⁴ With increasing WID variations, different microarchitectural units spread across the chip will exhibit different delay characteristics. The increasing amount of WID and random process variation calls for the development of novel post-fabrication tuning techniques that we can apply at the microarchitectural unit or block level.⁵

We explore two such techniques. The first lets us add extra latency to pipelines that might exhibit longer-than-expected delays to mitigate frequency degradation due to process variation. Although effective, the extra latencies can overly degrade system-level performance (such as instructions per cycle [IPC]) for latency-critical blocks with tight loops. To alleviate this, VI allows circuit blocks to statically choose between two supply voltages after fabrication. Using this approach, we

can view different microarchitectural units within a processor as operating at individually tuned effective voltage levels while maintaining a single, consistent operating frequency. VI not only combats the harmful effects of WID variations, but it also offers power savings when combined with VL.

Variable latency

Several architecture groups have recently studied VL's architectural impact.^{2–4} The basic idea is to make the latency of specific microarchitectural units adjustable while keeping the global frequency unchanged. If some units exhibit speed loss due to variation, VL can extend the latency of those units. The benefit of variable latency over other schemes (such as globally asynchronous locally synchronous [GALS]) is that the system is still fully synchronized and the entire machine operates at a single frequency. Previous studies have shown that one cycle of extra latency is sufficient for most units to cover the expected frequency spread resulting from severe variations.² For less IPC-critical units, an extra latency cycle does not degrade performance by much and, thus, will salvage chips from suffering large frequency loss.

The weakness of VL is its potentially large performance impact on IPC-critical units. In the past, researchers have either only shown VL's effectiveness on limited sets of microarchitectural units^{2,3} or didn't consider its effects on tight single-cycle loops.⁴ We provide a detailed study and comparison of how well VL can combat variations across units within a CPU core with both tight and long loops.

Figure 1a illustrates the basic concept of VL for a piece of a long pipeline consisting of multiple stages, such as a floating-point unit (FPU). This pipeline assumes a latch-based synchronous design operating off of two complementary clock phases. One benefit of latch-based designs over a flip-flop-based design is they let us borrow time across logic stages owing to the soft barriers imposed by the latches, unlike the hard barriers associated with flip-flops (FF). Ideally, approximately half a clock cycle of time slack can be borrowed across pipeline stages. This time borrowing inherently hides delay imbalances between stages (within limits), which

also helps to mitigate the effects of process variations.

The upper block diagram in Figure 1a shows a pipeline configured to operate at the default latency with a flow-through latch between stages 1b and 2a. Clocking that latch, as the lower block diagram shows, adds an extra half cycle of latency to the pipeline and provides extra time borrowing to absorb longer-than-expected delays due to process variation in the preceding logic stages. Switching between the modes with and without the extra latency allows for post-fabrication tuning as needed by each unit.

In addition to logic-dominated structures, we can apply a modified version of the VL technique to memory-dominated structures such as register files and issue queue (issueQ). Because a memory array typically requires precharging before it can be accessed, it's difficult to borrow into the memory-access time. Hence, we simply insert extra latches at the boundary between the memory array and other logic structures. Figure 1b shows how we can pipeline a register file into two stages. If necessary, we can add an extra clocking stage between the decoder and the memory array. Similarly, we can pipeline the issueQ into two stages by separating the wake-up content-addressable memory (CAM) and selection logic.

Voltage interpolation

Because VL alone can't solve the variation problem (which we explain in more detail later), we propose VI as a necessary second tuning technique. Instead of providing one supply voltage to all the circuit blocks, we provide V_{DDH} and V_{DDL} . To best utilize these two voltages, we divide the microarchitectural blocks into multiple domains (see Figure 2). For example, assume the combinational logic within a unit is divided into three voltage domains, where each domain can select between V_{DDH} (1.2 V) and V_{DDL} (1.0 V) individually via the PMOS power switches. We assume these power-switching devices introduce little additional overhead because V_{DD} gating is often already used to cut leakage power for idle blocks. This ability to choose between two voltages lets us tune the unit's delay corresponding to the voltage

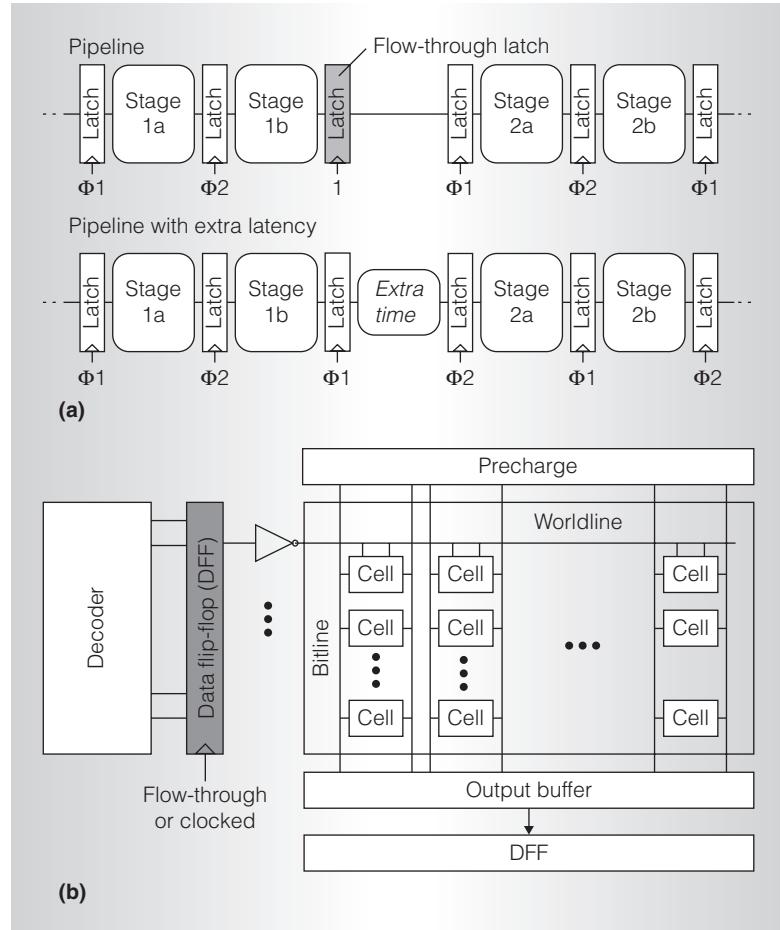


Figure 1. Block diagrams of variable latency applied to logic-dominated (a) and memory-dominated structures (b).

range between V_{DDH} and V_{DDL} (ΔV). If all the domains utilize V_{DDH} , the unit has a maximum effective voltage of 1.2 V and the minimum delay possible. Conversely, if all stages operate off of V_{DDL} , the unit will have the minimum effective voltage of 1.0 V and the longest delay. Configurations between these two extreme scenarios lead to a spread of delay possibilities—that is, the unit operates off an effective voltage somewhere between V_{DDH} and V_{DDL} . For example, if two domains are connected to V_{DDH} and one domain to V_{DDL} , this results in the high “effective” voltage we show in the left-hand side of Figure 2. If one domain is connected to V_{DDH} and two domains to V_{DDL} , the effective voltage is relatively lower, as we show in the right-hand side of Figure 2.

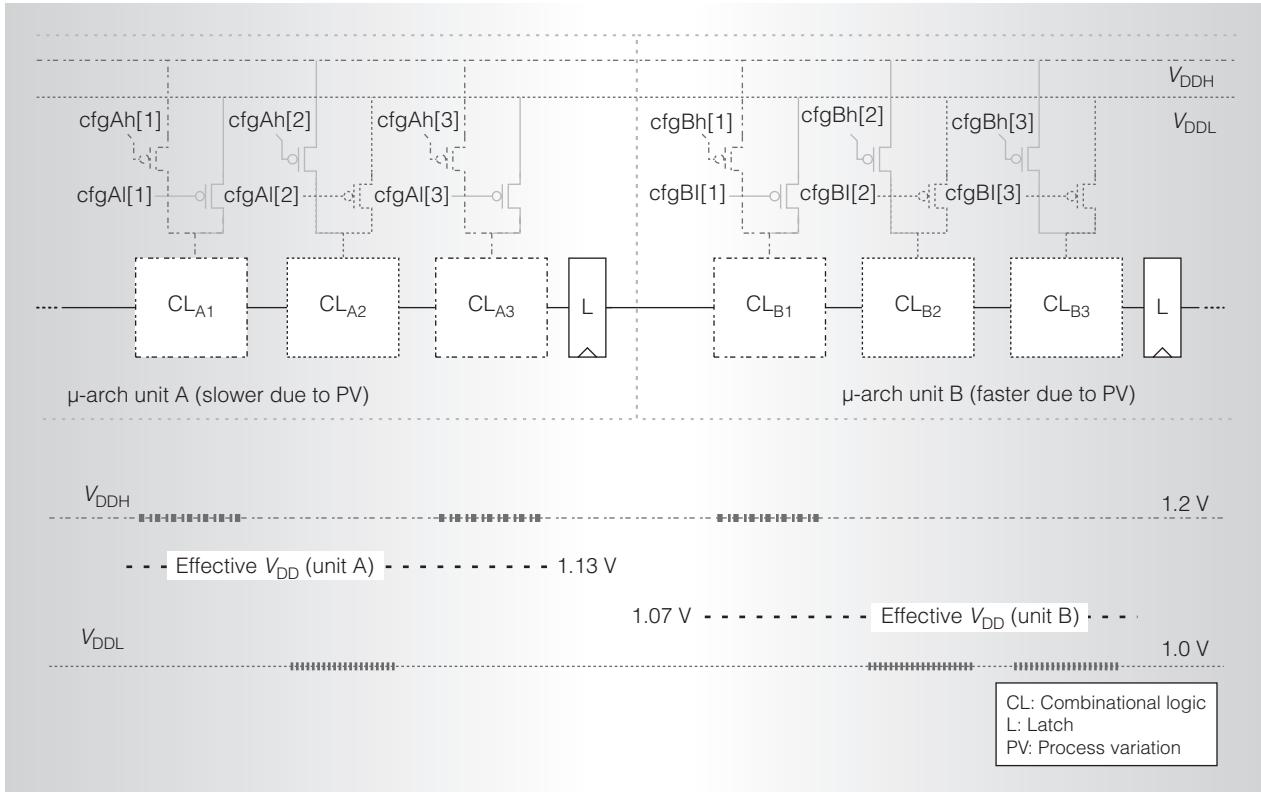


Figure 2. Schematic of voltage interpolation and illustration of effective voltage.

Hence, this spatial dithering of the voltage enables our notion of VI. Given the soft barriers (due to time borrowing) with latch-based clocking, both units operate at an intermediate, effective voltage level when we view them in combination. Process variation will invariably lead to a spread of delays across the different stages even if they're perfectly balanced at design time under nominal conditions. VI combats this by configuring microarchitectural units to operate at a specific target frequency, speeding up slower paths and slowing down faster ones.

The main benefit of VI is that we can arbitrarily select the different effective voltages each unit needs to run at a single nominal frequency. Per-block voltage tuning might also be possible by individually supplying power to each block, but the hardware overhead is prohibitively high. One concern that arises when we consider using two supply voltages is whether we need level shifters at voltage boundaries to break the static current path in a gate operating off V_{DDH} driven by a block operating off V_{DDL} .

Fortunately, if the difference between V_{DDH} and V_{DDL} required to cover delay variations resulting from process variation is small, the transistors' nonzero threshold voltage obviates level shifters.

For pipelined microarchitectural units (such as FPUs), stage boundaries can be natural cut points for the voltage domains that individually select between V_{DDH} and V_{DDL} . Because a latch-based design lets us use time borrowing between pipeline stages, we can use VI to tune the entire pipeline's speed with respect to the target system's operating frequency, even in the presence of process variation.

Validating the techniques

To validate the proposed schemes, we applied both VI and VL to a single-precision FPU (compatible with IEEE Std 754), designed using a standard CAD flow in a 130-nm CMOS logic process with eight metal layers. We pipelined the FPU into six stages for the nominal case and were able to insert extra latency, which resulted

in a second seven-stage configuration. Each pipeline stage can choose between two supply voltages (V_{DDH} and V_{DDL}) through configuration registers, enabling VI. We taped out the design and measured 15 chips for functionality, performance, and power. Figure 3 presents a die photo of the test chip with an overlay identifying the main blocks. (Additional details and measurements of the test chip are available elsewhere.⁶)

Figure 4 summarizes the measured results of frequency tuning versus power consumption for the FPU test chip. For comparison purposes, the dashed line plots the traditional trade-off between frequency (presented as clock period) and power when the global voltage (V_{DD} Global) is swept from 0.95 to 1.4 V while operating in the six-stage configuration. The 64 circles correspond to all (2^6) possible switchable voltage configurations for six-stage operation given two power supplies: $V_{DDH} = 1.2$ V and $V_{DDL} = 0.9$ V. The results verify that a broad tuning range of frequencies are possible with only two supplies at fixed voltages, extending above and below a nominal frequency of 264 MHz (or 3.8 ns clock period) given a nominal voltage (1.1 V). Most of the circles hover above the dashed line, indicating a small power penalty associated with VI. On the other hand, some circles fall below the line. Due to some inherent imbalances between stages in the design, some voltage configurations are better than others. Hence, it's necessary to carefully select between the 64 configurations to find the lowest power solution. Although these results are for a single FPU chip, we can deduce that the tuning range provided by VI can cover the 30 percent delay variations we expect from process variation.

The FPU test chip can also operate in a seven-stage configuration by clocking additional latches in the pipeline such that data exits the FPU pipeline after seven clock cycles. Figure 4 also includes measured power versus frequency for 64 voltage configurations (triangles) while operating in seven-stage mode with the same V_{DDH} and V_{DDL} . The additional time borrowing provided by the extra cycle latency allows the FPU to operate at higher clock frequencies. The overall frequency tuning range with both VI and VL grows to more than 40 percent.

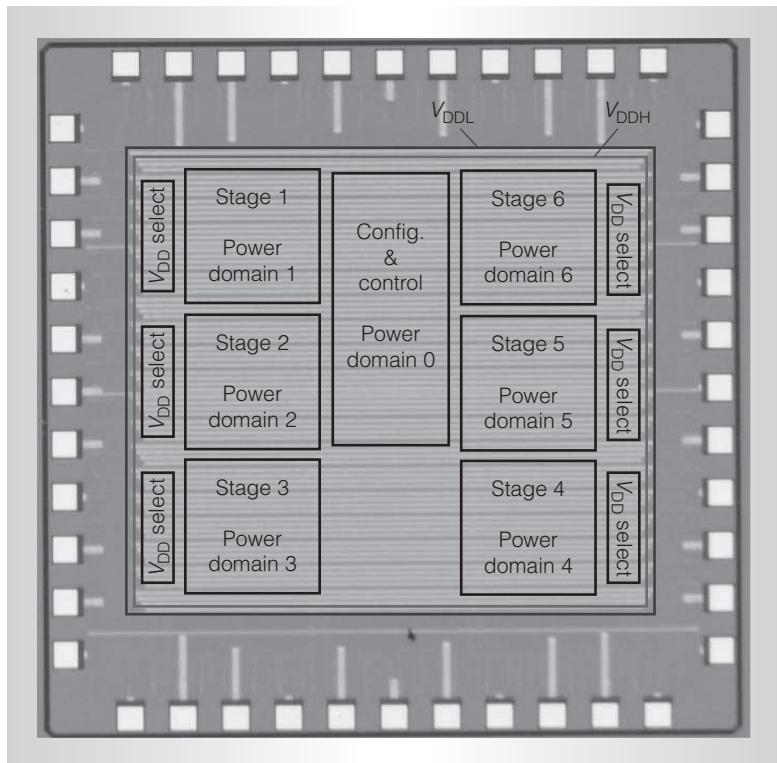


Figure 3. Die photo for the test chips.

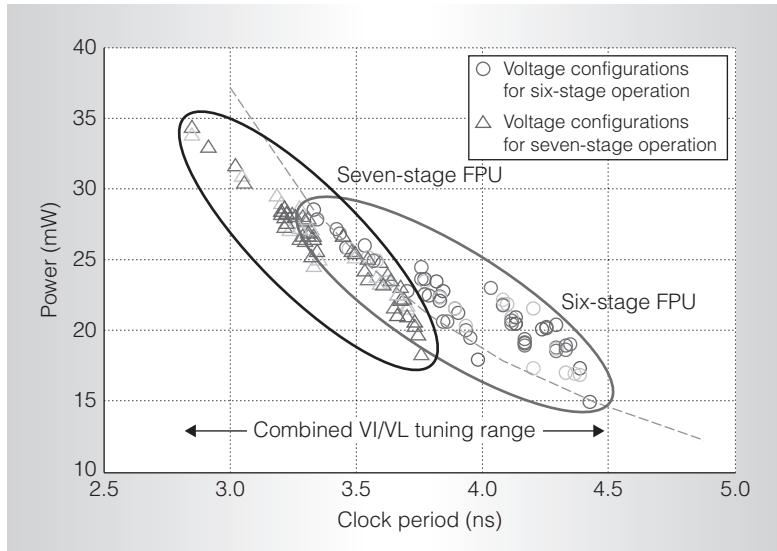


Figure 4. Power versus delay relationship for voltage interpolation (VI) and variable latency (VL) and the frequency tuning range. The dashed line plots the traditional trade-off between frequency and power. The lower gray oval gives results for the six-stage configuration, and the upper black oval gives those for the seven-stage configuration.

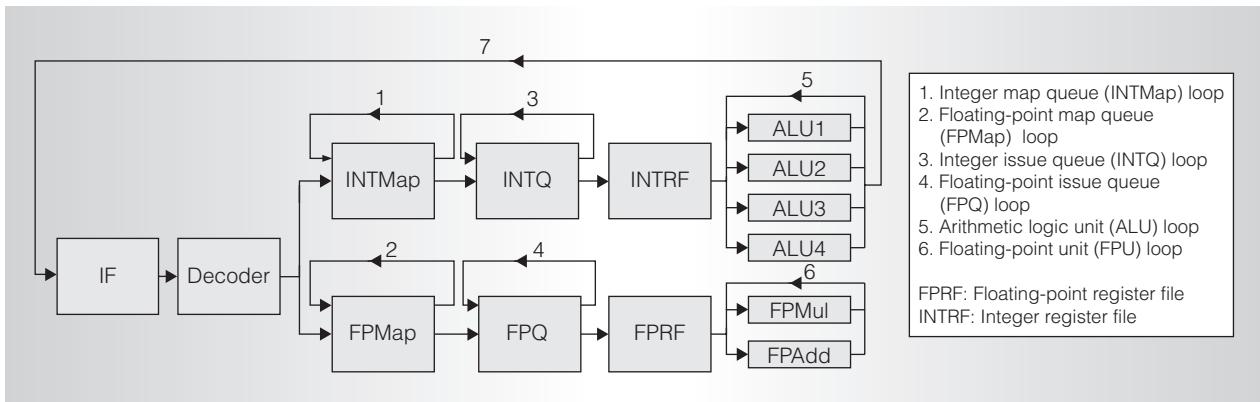


Figure 5. Several types of architectural loops in a multicore processor architecture.

In addition to higher frequency operation, the power consumption is lower for comparable six-stage speeds because more of the stages can operate off V_{DDL} . The small increase in clock power to switch more latches is offset by the decrease in dynamic power consumed by the logic.

If an FPU runs slowly due to process variation, there are two ways to achieve the nominal operating frequency. One choice is to maintain a six-stage pipeline and connect more stages to V_{DDH} so that the effective voltage increases. Another choice is to extend to a seven-stage pipeline to provide additional time for computation while reducing logic power by switching more stages to V_{DDL} . If power consumption is the only metric of interest, the seven-stage configuration is always better than the six. However, there are performance penalties for architectural units with a longer latency, especially for some units such as the issueQ and arithmetic logic unit (ALU). Therefore, power alone isn't sufficient to judge VL's effectiveness.

Architecture analysis

To fully understand this problem, we performed a detailed architecture-level analysis and trade-off study that explored the impact of applying these two techniques across a multicore processor architecture. We describe the impact in the context of well-known architectural loops in an out-of-order microprocessor.⁷ Figure 5 shows that there are several types of loops in a processor composed of different microarchitectural units. We

assumed each has a one-cycle default latency, except for the FPU, which has a six-cycle latency. We considered most of the key architectural units with the exception of large arrays such as caches. Process variations in array structures will impact both cell stability and performance,⁸ so techniques that specifically target large cache structures are most appropriate.^{3,9,10} This work focuses on the core pipeline logic.

VL and time borrowing

The performance of different architectural loops has varying sensitivity to latency. In other words, depending on the loops' span and latency, each loop can affect the processor's overall performance differently. For example, latency of the issueQ and ALU loops determines the dispatch of dependent instructions, which has a strong correlation to overall instruction throughput. Increasing the latency of tight loops, such as the issueQ and ALU, prevents back-to-back issue of instruction and significantly impacts system throughput. The branch-resolution loop is important for pipeline flush operations, dictating branch mispredict penalties and the amount of mis-speculative instructions. Although this loop can have a large impact on IPC if it's overextended, extending the loop by one or two cycles will likely have negligible impact on performance for most applications.

In addition to the IPC impact of increasing loop latency, we must also consider time borrowing's impact across units. Time borrowing essentially lets slow blocks take up

the timing slack provided by fast blocks in subsequent pipeline stages. When combined with VL operation, time borrowing can effectively mitigate the impact of variations. However, we must carefully consider which loops can use time borrowing and which loops are sensitive to increased latency. Single-cycle loops (such as loops 1 through 5 in Figure 5) can't time borrow because they can't borrow time from themselves. In contrast, longer loops can borrow time between multiple stages to meet timing—for example, in a branch-resolution loop. The FPU loop can leverage time borrowing effectively to balance delay fluctuations between individual stages as long as the entire FPU can meet a six-cycle delay. All the stages in the branch-prediction loop can borrow time from one another as long as the entire loop meets a six-cycle delay requirement and all the tight loops residing inside it can meet their own timing. In general, long loops (such as loops 6 and 7 in Figure 5) are least sensitive to variable latency and can make the most use of time borrowing.

To further illustrate the performance impact of time borrowing and variable latency, Figure 6 plots overall system performance in billions of instructions per second (BIPS) when different techniques are applied to three representative loops: the integer issueQ (INTQ), FPU, and branch resolution. We normalized the results with respect to an ideal machine without any variation. We then considered 50 individual chips that suffer from process variation, where each chip can have its own operating frequency. In this case, the slowest stage or unit within each loop determines that loop's frequency. We then applied time borrowing and VL to the same chips, attempting to run the loops at or close to the nominal frequency in an ideal machine. As expected, time borrowing offers no benefit to the INTQ, but it helps the FPU and branch-resolution loops because it allows some averaging of delays across stages and units. VL negatively impacted performance on the INTQ loop because the frequency increase is more than offset by IPC losses. However, VL works reasonably well for loops with longer pipelines.

This analysis shows that although VL can be effective for certain loops, we must use it

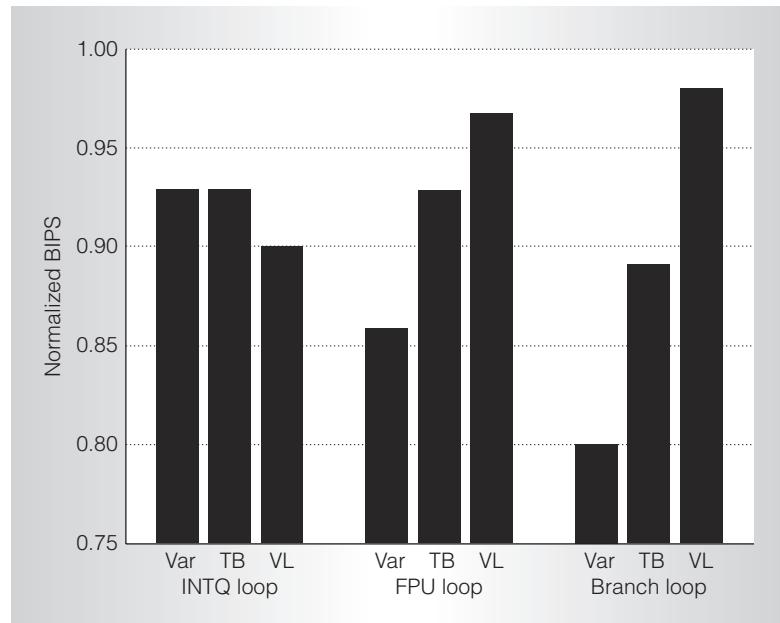


Figure 6. Performance sensitivity of three loops. The Var bar represents the average performance loss due to process variations (for that particular loop) across all chips without time borrowing (TB) or variable latency (VL).

judiciously. In contrast, we can apply VI across microarchitectural units, but we must carefully consider power overhead.

VI alone

In a typical microprocessor with only one global power supply, the global voltage, V_{DD} Global, is set by the entire chip's worst-case critical path and the desired power-performance target. The key idea behind VI is that we can selectively apply an effective voltage, somewhere between V_{DDH} and V_{DDL} , to individual blocks within the CPU to individually meet their timing needs. To illustrate VI's frequency-tuning capabilities in a multicore processor, we considered a 16-core chip with process variations and assumed a nominal voltage of 1.0 V for this chip. Figure 7 plots its simulation results in a power-frequency space for different voltage and frequency tuning scenarios.

We normalized all the results to the power and performance of an ideal chip without variations operating at a nominal frequency and voltage. The chip's global-frequency configuration (see the solid inverted triangle) corresponds to a traditional scenario

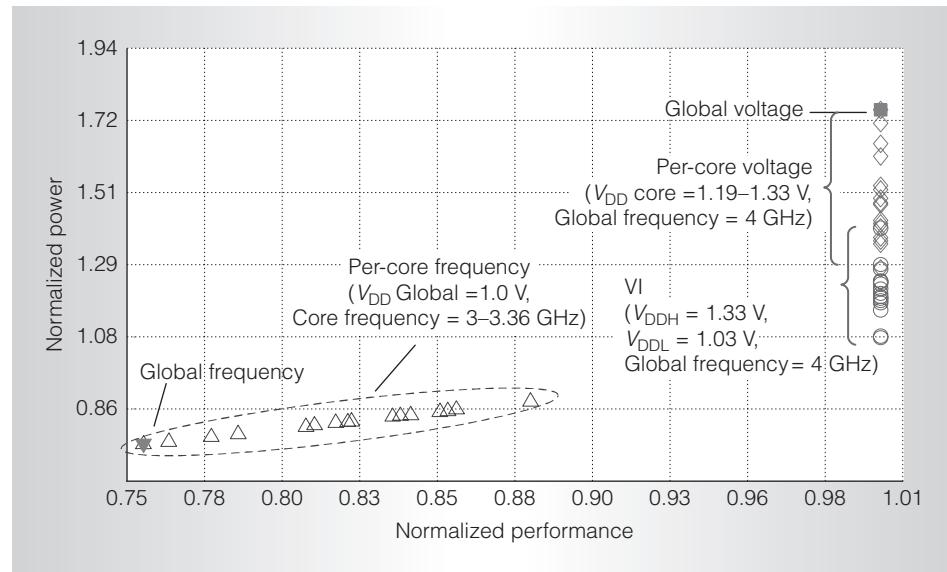


Figure 7. Normalized power versus normalized performance of different techniques applied to a 16-core processor with process variations. The solid, inverted triangle shows the performance and power of the chip using a global-frequency configuration: 1 V and 3 GHz. The solid box shows the global voltage point: 1.33 V and 4 GHz.

in which the global clock frequency is lower than the nominal frequency to accommodate the slowest core in the chip given the 1.0 V supply voltage. Figure 7 also plots the performance of each of the chip's cores with per-core frequencies, showing the frequency and power of each individual core running at its maximum speed with the 1.0 V supply. This configuration can loosely be thought of as a GALS approach applied at the core level. The remaining configurations are shown with respect to settings where voltages are tuned to enable all cores to operate at the nominal frequency. The top data point (see the solid black box) shows that in order to achieve this target performance with one global voltage applied all the cores, a 1.33 V supply is required, which results in 77 percent power overhead. The diamonds correspond to a per-core voltage scenario, in which each individual core receives a separate voltage to meet the nominal frequency target. The worst-performing core also requires 1.33 V, but the best core requires only a 1.19 V supply.

Finally, the circles represent power with VI. For these points, V_{DDH} is set to 1.33 V and ΔV is equal to 0.3 V. These points can also achieve the desired frequency while dissipating significantly lower power

than the per-core voltage setting because the scheme only needs to raise the voltage for slow blocks within each core. These results show that VI can be an effective voltage and frequency tuning knob to bin the global frequency of a multicore system with process variation to a single value with minimum power overhead.

The significance of VI is that only two power supplies are needed for the entire chip to satisfy the performance needs of individual cores and microarchitectural units within the cores. This solution is far more cost-effective than supplying per-core voltages, which would require 16 separate voltage regulators and power domains. In addition to lower implementation overhead, VI provides much finer-grained voltage control to combat WID process variation, resulting in significant power savings for equivalent performance. However, as Figure 7 shows, there is a power overhead to this technique, which motivated us to consider Revival.

Combining techniques

Combining VL and VI offers several benefits. First, VL can suffer from high IPC costs for certain loops and VI might be a more efficient method for tuning. On the

other hand, applying VI to all units can lead to unnecessary power overhead if VL can be effective. Furthermore, if VL generates a surplus of slack for certain loops, we can apply VI to effectively reclaim that surplus for power savings, while still meeting the desired frequency target.

We began by studying the 13 microarchitectural units in Figure 5 and investigating Revival’s configuration settings before comparing it to other schemes. For latency configurations, we choose VI configurations that allow the chip to meet the *target frequency* (which we defined as the chip’s frequency without process variations) with the lowest power dissipation. We fixed V_{DDH} and V_{DDL} to 1.33 and 1.03 V, respectively. Figure 8 presents the power-performance plot for one typical 16-core chip in our simulation. Because the chip frequency is constant for all configurations, the spread in normalized performance corresponds to IPC differences. The set of points with a normalized performance of 1 is the same set of points we saw in Figure 7, and it represents chips with unmodified pipelines and no additional latency. We see that adding latency generally lets VI choose lower voltage settings that lead to lower overall power. Latency configuration 18 (Lat_cfg#18) is the optimal ED² trade-off across all the configurations.

Finally, we compared Revival’s effectiveness to other techniques in the context of a 16-core processor. We assumed time borrowing between units for all techniques. We compared eight schemes. The first two schemes relax the chip’s frequency:

- *Global frequency* assumes one global power supply for the entire chip. The frequency is set with respect to the nominal global voltage such that the chip will operate at the slowest core’s frequency.
- *Per-core frequency* assumes one global power supply for the entire chip. Each core’s frequency is set with respect to the nominal global voltage and the effects of process variation within that core. Given the different frequencies, this scheme requires synchronization between cores.

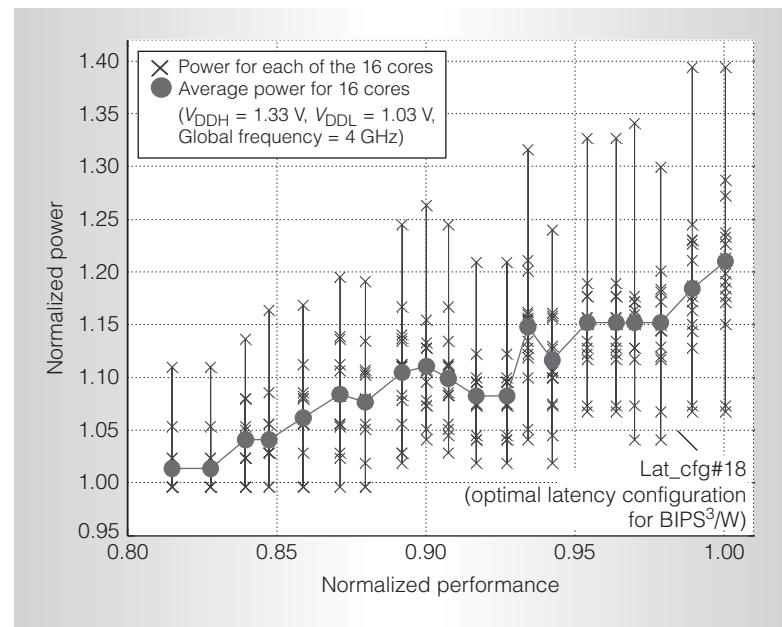


Figure 8. Power versus performance of Revival.

All the remaining schemes attempt to meet the chip’s target frequency, but they could also target any other frequency by adjusting voltages:

- *Global voltage* assumes a single, global power supply for the entire chip, but it raises this voltage to meet the target frequency. For the chip to operate at this frequency, the global voltage must be set with respect to the entire chip’s worst-case critical path delay.
- *Two-voltage domains* is similar to the global-voltage scheme, but it provides two-voltage domains in an attempt to have similar implementation complexity to VI, which requires two voltages. This scheme is slightly more flexible; because half of the cores are tied to different domains, the voltage is now set separately according to the worst-case critical path within each of the two domains.
- *Per-core voltage* assumes that each core has a separate, independent voltage domain to meet the target frequency. This scheme lets each core choose its own optimal voltage according to process variation, but the hardware cost might be prohibitively high.

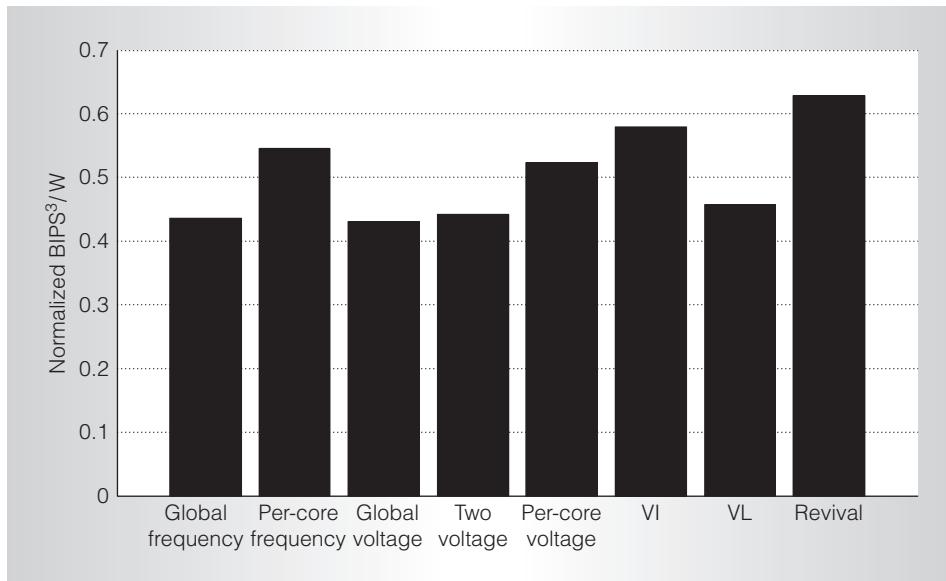


Figure 9. All techniques for the average of 50 chips.

- *Variable latency* applies the VL scheme in isolation, attempting to meet the target frequency. If a unit can't meet timing, extra latency is inserted.
- *Voltage interpolation* applies the VI scheme in isolation, attempting to meet the target frequency.
- *Revival* applies both VL and VI, attempting to meet the target frequency. If a core can't meet timing, we explore all latency configurations and all possible VI settings while optimizing for energy delay². V_{DDH} and V_{DDL} are fixed.

We again considered 50 16-core chips affected by process variation, applying all the schemes to all the chips. Figure 9 plots the average ED² (BIPS³ per watt [BIPS³/W]) results for all 50 chips. The global-frequency and global-voltage schemes perform poorly because they incur large performance and power overhead, respectively. The two-voltage technique offers only small benefits. The per-core voltage scheme is about 26 percent better than the global scheme because each core can choose an optimal voltage. VL in isolation is hampered by tight loops. Although it performs better than the global-frequency and global-voltage schemes, it doesn't do as well as the per-core voltage case. In contrast, VI outperforms

the per-core voltage scheme and is 35 percent better than the global scheme.

Revival performs the best and improves BIPS³/W by 47 percent. Because Revival needs only two power supplies and modest amounts of additional hardware for extra latches and voltage selection, it's the most favorable scheme.

A remaining challenge for postfabrication tuning techniques such as VI and VL revolves around the ability to efficiently test different possible configurations and converge upon the best setting. We plan to explore how various aspects of postfabrication testing can affect the final settings and resulting deviations from optimal solutions. Moreover, we anticipate long-term testing and tuning in the field can improve the design and ways to compensate for time-varying sources of variations such as temperature and aging.

MICRO

Acknowledgments

This work was supported by US National Science Foundation grants CCF-0429782 and CCF-0702344 and a gift from Intel. We thank United Microelectronics Corporation for chip fabrication and the anonymous reviewers for their detailed comments and suggestions.

References

1. K. Bowman, S. Duvall, and J. Meindl, "Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration," *J. Solid-State Circuits*, vol. 37, no. 2, Feb. 2002, pp. 183-190.
2. X. Liang and D. Brooks, "Mitigating the Impact of Process Variations on Processor Register Files and Execution Units," *Proc. 39th IEEE Int'l Symp. Microarchitecture* (Micro 06), IEEE CS Press, 2006, pp. 504-514.
3. S. Ozdemir et al., "Yield-Aware Cache Architectures," *Proc. 39th IEEE/ACM Int'l Symp. Microarchitecture* (Micro 06), IEEE CS Press, 2006, pp. 15-25.
4. A. Tiwari, S.R. Sarangi, and J. Torrellas, "Recycle: Pipeline Adaptation to Tolerate Process Variation," *Proc. Int'l Symp. Computer Architecture* (ISCA 07), ACM Press, 2007, pp. 323-334.
5. R. Teodorescu et al., "Mitigating Parameter Variation with Dynamic Fine-Grain Body Biasing," *Proc. 40th IEEE Int'l Symp. Microarchitecture* (Micro 07), IEEE CS Press, 2007, pp. 27-42.
6. X. Liang, D. Brooks, and G.-Y. Wei, "A Process-Variation-Tolerant Floating-Point Unit with Voltage Interpolation and Variable Latency," *Proc. IEEE Int'l Solid-State Circuits Conf.*, IEEE Press, 2008, pp. 404-405.
7. E. Borch et al., "Loose Loops Sink Chips," *Proc. 8th Int'l Symp. High-Performance Computer Architecture* (HPCA 02), IEEE CS Press, 2002, pp. 299-310.
8. A.J. Bhavnagarwala, X. Tang, and J.D. Meindl, "The Impact of Intrinsic Device Fluctuations on CMOS SRAM Cell Stability," *IEEE J. Solid-State Circuits*, vol. 36, no. 4, Apr. 2001, pp. 658-665.
9. A. Agarwal et al., "A Process-Tolerant Cache Architecture for Improved Yield in Nanoscale Technologies," *IEEE Trans. Very Large Scale Integration Systems*, vol. 13, no. 1, Jan. 2005, pp. 27-38.
10. X. Liang, G. Wei, and D. Brooks, "Process Variation Tolerant 3T1D-based Cache Architectures," *Proc. 40th IEEE Int'l Symp. Microarchitecture* (Micro 07), IEEE CS Press, 2007, pp. 15-26.

Xiaoyao Liang is pursuing a PhD in electrical engineering at Harvard University. His research interests include computer architecture and VLSI design, focusing on joint circuit and architecture solutions to combat process variability. Liang has an MS in electrical engineering from Stony Brook University.

Gu-Yeon Wei is an associate professor of electrical engineering at Harvard University. His research interests include mixed-signal VLSI design for wireline data communication, energy-efficient computing devices for sensor networks, and collaborative software, architecture, and circuit techniques to combat variability in nanoscale technologies. Wei has a PhD in electrical engineering from Stanford University.

David Brooks is an associate professor of computer science at Harvard University. His research interests include architectural and software approaches to address power, thermal, and reliability issues for embedded and high-performance computing systems. Brooks has a PhD in electrical engineering from Princeton University.

Direct questions and comments about this article to David Brooks, School of Engineering and Applied Sciences, Harvard Univ., Cambridge, MA 02138; dbrooks@eecs.harvard.edu.

For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/csdl>.

RUNNING IN CIRCLES LOOKING FOR A GREAT COMPUTER JOB OR HIRE?

The IEEE Computer Society Career Center is the best niche employment source for computer science and engineering jobs, with hundreds of jobs viewed by thousands of the finest scientists each month - **in Computer magazine and/or online!**

 **careers.computer.org**
<http://careers.computer.org>

- Software Engineer
- Member of Technical Staff
- Computer Scientist
- Dean/Professor/Instructor
- Postdoctoral Researcher
- Design Engineer
- Consultant



The IEEE Computer Society Career Center is part of the *Physics Today* Career Network, a niche job board network for the physical sciences and engineering disciplines. Jobs and resumes are shared with four partner jobboards - *Physics Today* Jobs and the American Association of Physics Teachers (AAPT), American Physical Society (APS), and AVS: Science and Technology of Materials, Interfaces, and Processing Career Centers.

 IEEE
computer society

Advertising Information January/February 2009

IEEE Micro

Advertising Personnel

Marion Delaney
IEEE Media, Advertising Dir.
Phone: +1 415 863 4717
Email: md.ieeemedia@ieee.org

Marian Anderson
Sr. Advertising Coordinator
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: manderson@computer.org

Sandy Brown
Sr. Business Development Mgr.
Phone: +1 714 821 8380
Fax: +1 714 821 4010
Email: sb.ieeemedia@ieee.org

Advertising Sales Representatives

Recruitment:

Mid Atlantic
Lisa Rinaldo
Phone: +1 732 772 0160
Fax: +1 732 772 0164
Email: lr.ieeemedia@ieee.org

New England
John Restchack
Phone: +1 212 419 7578
Fax: +1 212 419 7589
Email: j.restchack@ieee.org

Southeast
Thomas M. Flynn
Phone: +1 770 645 2944
Fax: +1 770 993 4423
Email: flyntom@mindspring.com

Midwest/Southwest
Darcy Giovingo
Phone: +1 847 498-4520
Fax: +1 847 498-5911

Email: dg.ieeemedia@ieee.org

Northwest/Southern CA
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Japan
Tim Matteson
Phone: +1 310 836 4064
Fax: +1 310 836 4067
Email: tm.ieeemedia@ieee.org

Europe
Hilary Turnbull
Phone: +44 1875 825700
Fax: +44 1875 825701
Email: impress@impressmedia.com

Product:

US East
Joseph M. Donnelly
Phone: +1 732 526 7119
Email: jmd.ieeemedia@ieee.org

US Central
Darcy Giovingo
Phone: +1 847 498-4520
Fax: +1 847 498-5911
Email: dg.ieeemedia@ieee.org

US West
Lynne Stickrod
Phone: +1 415 503 3936
Fax: +1 415 503 3937
Email: ls.ieeemedia@ieee.org

Europe
Sven Anacker
Phone: +49 202 27169 11
Fax: +49 202 27169 20
Email: sanacker@intermediapartners.de