
ACTIVE GUARDBAND MANAGEMENT IN POWER7+ TO SAVE ENERGY AND MAINTAIN RELIABILITY

THE AUTHORS PRESENT A MECHANISM THAT REDUCES EXCESS VOLTAGE MARGIN IN MICROPROCESSORS. THEY FIRST DEMONSTRATED THIS MECHANISM IN AN IBM POWER7 SERVER AND PROVED ITS EFFECTIVENESS IN THE POWER7+ PRODUCT. THE MECHANISM REDUCED POWER CONSUMPTION ON THE V_{DD} RAIL BY 11 PERCENT FOR SPEC CPU2006 WORKLOADS WITH NEGLIGIBLE PERFORMANCE LOSS, WHILE INCREASING PROTECTION AGAINST NOISE EVENTS.

••••• Microprocessors must operate reliably across a wide range of environmental conditions and workloads. Selecting an operating voltage high enough to account for worst-case conditions without violating power budgets is a growing challenge. Extra margin in the form of voltage guardband must be included to guarantee proper circuit timing during worst-case voltage droop events, in addition to covering for other variables such as test inaccuracy and aging. However, the voltage droop experienced by the microprocessor's circuits is much smaller under typical workloads and environmental conditions. Therefore, energy is wasted because the higher voltage needed to ensure worst-case operation results in excess timing margin during normal operation.

Instead, it would be preferable to maintain a fixed amount of guardband so that only the necessary energy is expended at any operating condition. To this end, we implemented two cooperating feedback controllers in a prototype Power7 server to

operate the microprocessor with a fixed timing margin.¹ *Critical-path monitors* (CPMs) were previously introduced to Power microprocessors to measure the available timing margin.² The first feedback controller couples these CPMs with a digital phase-locked loop (DPLL) in hardware and prevents timing errors induced by voltage changes that happen on short time scales.³ The second feedback controller, implemented in firmware, adjusts the processor voltage to achieve a desired performance level on a longer time scale. We evaluated many workloads running at a single target frequency and measured a significant reduction in processor power after enabling this active guardband management. We call this technique *undervolting* because it can achieve the same performance level at a lower voltage. The resulting mechanism reduces power consumption for typical workloads and conditions while still allowing worst-case workloads to operate reliably at the maximum frequency.

Charles R. Lefurgy
Alan J. Drake
Michael S. Floyd
Malcolm S. Allen-Ware
Bishop Brock
IBM
José A. Tierno
Apple
John B. Carter
Robert W. Berry Jr.
IBM

This article expands on our prior work by demonstrating a product-ready undervolting architecture running on a prototype Power7+ system. In order to more broadly apply the architecture, we describe a calibration technique for tuning the CPM to work across a wider operating range, from Nominal to Turbo frequencies. We also improve the prior undervolting algorithm to reduce performance loss for workloads that are not steady-state.

Background

Operational voltages in microprocessors are carefully set to ensure correct functionality at each given frequency. Each processor generation has a number of frequency targets, or *sorts*, defined by a target performance level for a given power dissipation. The operating voltage for each processor within a sort is individually determined by a manufacturing test and characterization process on the basis of the chip's intended operating frequency range and environment. For each frequency sort point, a test procedure determines the minimum voltage at which the processor will reliably provide correct operation with some extra operational margin. The procedure first increases the sort frequency by some percentage to add guardband. It then converts the frequency guardband to a voltage guardband by running a self-checking test exerciser program at that higher frequency and reducing the voltage until failure occurs. Finally, the procedure typically adds back a small amount of voltage guardband to the minimum voltage providing correct operation because circuits do not always track with voltage and frequency equally. This voltage and the original sort frequency (without the margin) constitute the operating point of that particular microprocessor in the product. Throughout this article, we typically will express guardband as a voltage margin.

Voltage margins compensate for noise processes and operational variations that directly affect the circuit's latch-to-latch path delay. Critical paths are those circuits with the least available timing margin, such that noise-induced delay changes are sufficient to cause them to fail. The largest noise processes are temperature and voltage changes that result from different workloads.

By adjusting voltage or frequency to track a workload's needs, active guardband management can save energy during low-temperature and low-activity periods while guaranteeing performance during high-temperature and high-activity periods. The faster a system can detect and compensate for a noise event, the more its margins can be reduced to only the amount needed to protect against the fastest random events, testing uncertainty, and aging.

Active guardband management

The architecture for active guardband management (Figure 1) requires three components: a sensor to measure timing margin; a fast, hardware-based timing margin controller that protects against reducing timing margin to unsafe levels; and a performance controller to adjust voltage to convert excess timing margin into energy savings. (For information on another guardband management approach, and how it compares to our approach, see the "Related Work in Reducing Voltage Guardband" sidebar.)

Measuring timing margin

The CPMs in Power7+ measure available timing margin and are a refinement of the design implemented in previous Power chips.² Figure 2 shows a block diagram of the CPM and its place in the system control. The CPM consists of a pulse-generation circuit to create the timing edge, an insertion delay used to offset process variation and to adjust timing margin during calibration, a critical-path synthesis block, and a 12-bit edge detector from which 5 bits are forwarded to the DPLL to indicate timing margin. Critical-path delay is approximated with four timing paths in the synthesis block: a low- V_t (threshold voltage) inverter delay path, a low- V_t wire-dominated path, a mid- V_t inverter delay path, and a high- V_t inverter delay path.

The timing margin is measured as a 12-bit thermometer code in the edge detector (Figure 3). The output is a string of 1s followed by 0s where the location of the 1-to-0 transition indicates how far the edge propagated during the cycle. Noise events that reduce timing margin—such as increased voltage, increased temperature, or reduced cycle time—prevent the timing signal from

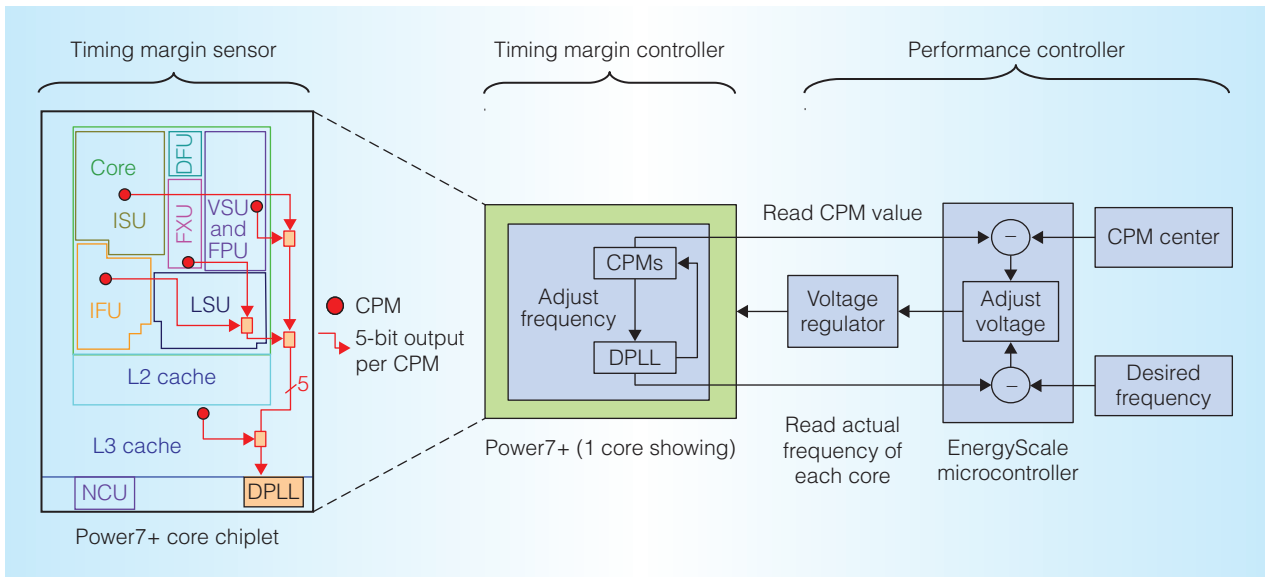


Figure 1. Active guardband management architecture implemented on Power7+. The architecture has three components: a sensor to measure timing margin (left), a timing margin controller (middle), and a performance controller (right). The locations of the embedded critical-path monitor (CPM) sensors are shown in relation to the functional units of the Power7+ processor core. The CPM outputs are connected to the digital phase-locked loop (DPLL) to form a timing margin controller in hardware. A performance controller in firmware then optimizes power efficiency by tuning the voltage. (DFU: decimal floating-point unit; FPU: floating-point unit; FXU: fixed-point unit; IFU: instruction fetch unit; ISU: instruction scheduling unit; LSU: load-store unit; NCU: noncacheable unit; VSU: vector-scalar math unit.)

Related Work in Reducing Voltage Guardband

The best-known technique for adaptively reducing voltage guardband is Razor.¹ Razor adds logic on critical paths to detect soft errors induced by inadequate supply voltage, and rolls back and replays operations when errors occur. This lets Razor more aggressively eliminate guardband than our approach. As a result, Razor reduces power by 33 to 50 percent in a 120 MHz, 130 to 180 nm processor, whereas CPMs reduce power 7 to 18 percent in a 4.144 GHz, 32 nm processor. However, Razor has more impact on area, performance, and overall design and verification effort.

Razor adds an area overhead of 1 to 3 percent. The critical-path monitor (CPM) and digital phase-locked loop (DPLL) circuitry occupy less than 0.06 percent of the core chiplet area (0.025 mm² out of 42.9 mm² in 32 nm) on Power7+. Razor adds a 1 to 3 percent performance penalty

to normal operation, whereas our approach has no normal-mode performance overhead. Finally, and perhaps most importantly, we believe that the CPM approach has less impact on chip design and verification effort because its changes are localized to the CPM and clock source circuits. The design and verification of other existing circuit paths is unchanged. In contrast, Razor involves modifying existing circuit paths, which we expect will have a larger impact on design and verification efforts.

Reference

1. S. Das et al., "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance," *IEEE J. Solid-State Circuits*, Jan. 2009, pp. 32-48.

propagating as far into the edge detector, and so the 1-to-0 transition moves to the left. A new measurement is taken each cycle, allowing the CPM to track fast-moving events. Locating CPMs in the microprocessor's power-dense regions allows the CPMs to experience the most extreme voltage and temperature variations on the processor. In Power7+,

each core has five CPMs. The CPM outputs for each core are logically combined by AND gates so that the CPM indicating the least timing margin will dominate.

Protecting timing margin

Each Power7+ core has a DPLL, providing per-core dynamic frequency scaling.⁴

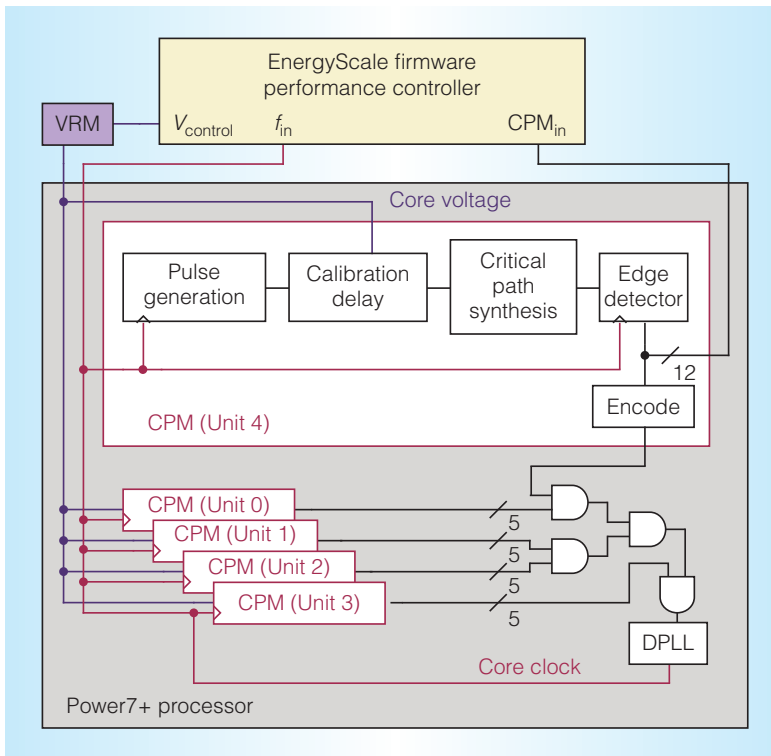


Figure 2. CPM block diagram for a processor core. The CPM consists of a pulse generation circuit feeding into an edge detector through a critical-path synthesis circuit and a calibration delay for tuning. The encoded edge detector output from the CPM is sent to the DPLL to control the clock frequency. The DPLL responds to the CPM which indicates the least amount of timing margin. (VRM: voltage regulator module.)

directly control the DPLL and together form the timing margin controller. The 5-bit CPM output gives the DPLL a per-cycle measurement of the timing margin. The DPLL adjusts frequency dynamically to maintain a CPM output of 11100. When the output edge moves to the left (due to reduced voltage or increased temperature, for example), the DPLL will reduce frequency. This increases the amount of time the edge has to propagate, causing it to move back toward the centered position on the following samples. Conversely, the CPM indicates increased timing margin as a shift in the edge position to the right, which causes the DPLL to speed up until the edge is centered again. The DPLL responds to sudden changes in timing margin with frequency steps up to 7 percent in less than 10 ns and can slew at 50 MHz/ μ s when tracking longer-term changes.

Calibrating CPMs

Proper calibration of the CPMs is crucial to obtaining the desired timing margin. During production of Power7+, a test procedure performs calibration at two primary operating points for that sort: Nominal and Turbo. At each operating point, the corresponding voltage and frequency is set and the CPM insertion delay is adjusted while running a high-power exerciser program until the edge is centered in the CPM output window. The timing margin controller is then enabled and resultant frequency

To improve the latency of frequency changes in response to operating condition changes, we added a secondary operating mode to the DPLL. In this mode, the CPM outputs

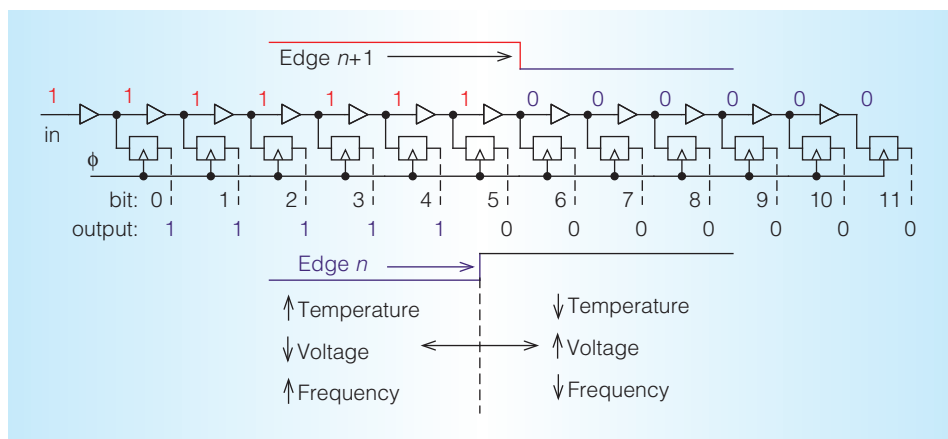


Figure 3. Detailed view of the edge detector block in each CPM. The timing margin is measured as a 12-bit thermometer code in the edge detector.

measurements are taken at this and adjacent CPM insertion delay settings ($+/-1$, $+/-2$, and so on). The test procedure calibrates each critical path and CPM instance independently. Ideally, the delay setting to achieve a target frequency for each critical path is the same at both operating points, indicating the CPM perfectly tracks the processor's frequency response across the operating voltage range. In reality, there are differences due to process variation, tuning errors, discrete delay step sizes, and localized operating point effects. During system boot, firmware selects the CPM synthesis path whose calibration is closest to the desired sort frequency and which has the smallest difference in delay setting from Turbo to Nominal, yet does not reduce the frequency guardband by more than 2 percent of the target frequency across the operating range.

We traditionally verify microprocessors by executing a suite of heavy exerciser programs in a system with the full operational guardband, and then checking again at a stress bias (with lowered voltage and increased frequency) to ensure reliable operation with reduced margin. However, with the timing margin controller enabled, the traditional manufacturing system test procedure can no longer apply the same stress bias because the frequency of the DPLL is controlled by the CPM output and thus lowers with the voltage to maintain a constant timing margin. Instead, we created a new test process that also calibrates the CPMs with less timing margin during manufacturing test, where we remove approximately two-thirds of the guardband. The system test stress procedure then runs the exerciser suite at this lower margin setting to ensure functionality at a stress bias equivalent to the traditional testing. Measurements of Power7+ systems show that the timing margin controller can be reliably calibrated to track maximum frequency to within 1.5 percent across the range from Nominal to Turbo.

Converting excess timing margin to energy savings

With the timing margin controller holding the timing margin constant and protecting against short-term noise, we are able to

adjust the microprocessor's voltage setting to save energy. When the system isn't running under worst-case conditions, the timing margin controller overclocks the microprocessor to remove excess timing margin. We added a performance controller to the server's firmware that detects when the average clock frequency attained by the processor's timing margin controller is above the level promised by the energy policy selected by the customer for that server. When this occurs, the performance controller reduces voltage to save energy.

The performance controller, shown in Figure 1, is implemented in an external microcontroller. It takes a desired frequency target as input. During each 32-ms controller interval, the performance controller measures the DPLL's average frequency output and compares this to the target frequency. If the measured frequency is higher than the target frequency, the controller lowers the voltage. The CPM senses this voltage reduction as a loss of timing margin, which causes the DPLL to lower its output frequency toward the frequency target. Conversely, if the measured frequency is lower than the target frequency, the controller raises the voltage. This results in additional timing margin, which causes the DPLL to raise its output frequency toward the frequency target. Each Power7+ core has independent DPLLs, and its CPMs are calibrated to a common frequency during manufacturing test. Because all cores share the same voltage, the performance controller adjusts voltage so that each core runs at least as fast as its target frequency. However, we cap each DPLL's maximum frequency at no more than 8 MHz above the target frequency. This limits the timing margin controller from wasting energy by running faster than the desired frequency, but allows the performance controller an 8-MHz over-clocking window above the target frequency to sense opportunities for voltage reduction.

Experimental results

We performed experiments on prototype systems with Power7 and Power7+ chips to validate the CPMs and controllers. Onboard power sensors provided accurate measurements

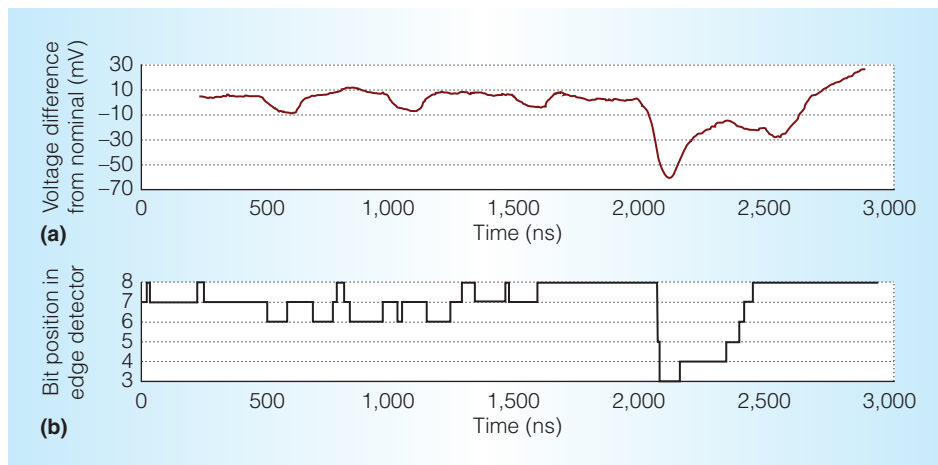


Figure 4. Response to worst-case noise event with timing margin controller disabled. Injected instantaneous droop event (a); measured CPM response to a droop event similar to that seen in part a, with the timing margin controller disabled (b). The CPM bit position closely follows the measured voltage over time and shows a significant loss of timing margin as it falls well below position 6.

for the total system, the microprocessor V_{DD} rail, and fans. Digital thermal sensors associated with each CPM measured the temperature. The server's firmware controlled the processor temperature by dynamically adjusting fan speed, and the ambient temperature was between 28 and 30° C.

Timing margin controller validation

We validated the timing margin controller on a prototype Power7 system and focused on workload transients, which happen faster than the power supply voltage can be adjusted. We have observed workload-induced voltage droops in this system with a time period of about 50 ns. In contrast, system voltage regulators take several microseconds to adjust voltage by even 1 percent. Fortunately, the timing margin controller in the processor provides a faster response within nanoseconds.

Figure 4a shows an illustrative severe voltage droop event captured from an internal chip voltage sense line. The droop event begins around 2,000 ns into the trace. We removed the load line from the voltage regulator module to enhance the effect of the induced droop so that we could measure the timing margin control system's response. We ran a steady-state maximum power workload at a nominal frequency, voltage, and

temperature and calibrated the CPMs at these same conditions to maximize visibility. Figure 4b shows a cycle-by-cycle trace of another instance of the voltage droop as sensed by the CPMs on one core of the chip when the timing margin controller was disabled, also aligned to begin at 2,000 ns. Because the CPM calibration is centered on edge position 6, the 5-bit output causes the trace visibility to clip at bits 3 (all 0s) and 8 (all 1s).

Figure 5a shows the timing margin controller responding instantly to a similarly induced voltage droop. We programmed the DPLL's frequency cap to 3,360 MHz because of our particular test bench setup. Figure 5b is a trace of CPM edge readings for the same droop event. The minimum reading of 5, an excursion of one edge position below the calibration center, demonstrates that most of the timing margin is preserved. The reading saturates at 7 when the voltage is above nominal because the frequency cap prevents the DPLL from over-clocking, resulting in extra timing margin.

In conclusion, the timing margin controller protects the processor from timing failure due to rapid workload transients. The impact of the timing margin controller is a reduction in clock frequency and performance, but with continued safe operation of the chip. Only by adding the performance controller

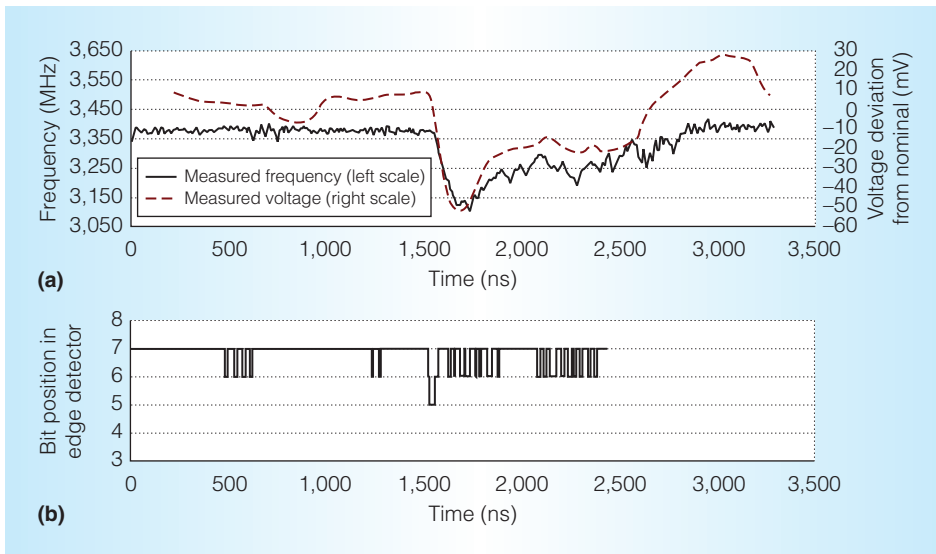


Figure 5. Response to worst-case noise event with timing margin controller enabled. Measured frequency response to injected droop event with timing margin controller enabled (a); CPM bit position in response to the same droop event seen in part a, with timing margin controller enabled (b). The CPM bit position hardly falls below the calibrated setpoint at position 6, meaning the timing margin is almost fully preserved as the frequency drops in response to the voltage droop.

can voltage adjustments be used as a means to manage the available timing margin and preserve performance for a given operating condition.

Voltage speculation

The performance controller selects a voltage to allow the DPLL to attain a specific target clock frequency. We call this *voltage speculation* because the precise ideal voltage is unknown and changes in real time with the chip activity level and temperature. Mispredicting the ideal voltage results in either overclocking (reduced energy efficiency) or underclocking (reduced performance). We investigate two voltage speculation algorithms that make different tradeoffs for energy efficiency and performance.

The first algorithm is an ad hoc controller that changes voltage up or down by one voltage table step at a time to adjust the measured frequency toward the target frequency. The Power7+ voltage table has a voltage for each 28-MHz frequency step. We found the ad hoc controller has insignificant performance impact on the SPEC CPU2006 suite overall. However, some individual workloads experience measurable performance loss.

This is due to the controller slowly single-stepping up through the voltage table when frequency falls below the target frequency. This typically affects workloads that frequently have a high rate of change in instructions processed or have CPU idle periods due to I/O access. The minimum time between consecutive voltage adjustments is 96 ms (3 controller intervals of 32 ms). This ensures the frequency measured due to a voltage change has settled before selecting a new voltage.

We developed a performance-aware controller to reduce performance loss for workloads that did not do well with the ad hoc method. The key idea is to not probe the limits of undervolting as aggressively as the ad hoc method. However, this comes at the cost of slightly reduced energy efficiency. This controller measures minute reductions in clock cycles lost to undervolting and attempts to make them up before further undervolting. When the measured frequency is too low, this controller uses the frequency error to proportionally change voltage in a single step, so it settles much faster. Additionally, voltage may be adjusted at every controller interval (without waiting the typical 96 ms) to stop the frequency reduction as

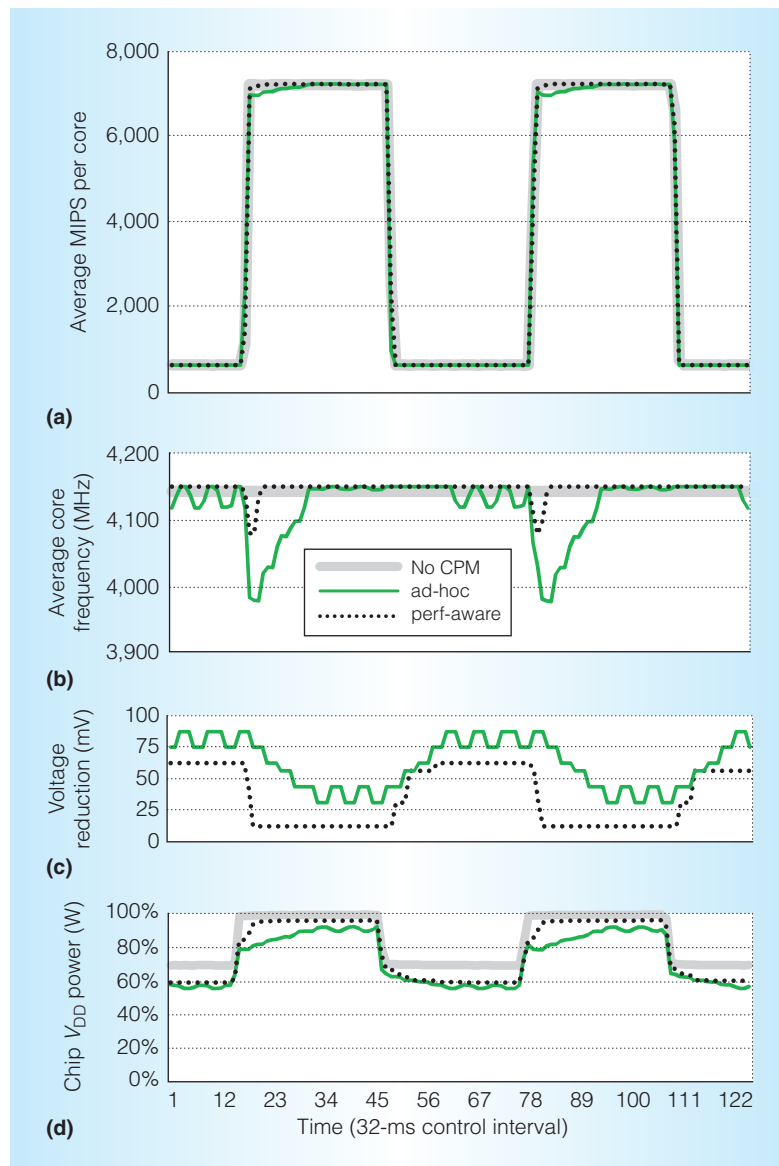


Figure 6. Voltage speculation algorithms. From top to bottom, the traces show the average million instructions per second (MIPS) per core (a), average core frequency (b), voltage reduction (c), and chip V_{DD} power as a percentage of maximum power (d). The performance-aware controller has better performance than the ad hoc controller, but saves less energy.

soon as possible. When the frequency is too high, the controller uses the 12-bit CPM edge detector reading compared to the CPM center as a guide for how much to adjust the voltage. This allows several down voltage steps in a single controller interval. Without the CPM edge detector reading, the controller can see only that it is going at most 8 MHz too fast due to DPLL frequency capping and can't accurately convert

this to an ideal-size voltage reduction. Furthermore, any change in the target frequency selected by the power management policy is instantly accounted for by consulting the traditional voltage table to estimate the voltage change on the basis of the amount of target frequency change.

We implemented active guardband management in a prototype IBM Power 780 server with four Power7+ microprocessors. The server has 32 cores total with 4 hardware threads each and a 128-Gbyte DRAM memory. The performance controller is implemented in a prototype version of the EnergyScale firmware, which runs on an independent microcontroller and is responsible for the system's power management.³ Processor clock frequency is only changed by the timing margin controller (in hardware), whereas the performance controller adjusts the voltage. EnergyScale's Dynamic Power Saver policy sets the target clock frequency according to system utilization. The SPEC CPU2006 workloads typically run at the maximum (Turbo) frequency of 4,144 MHz, except during short periods of disk I/O when utilization drops.

Figure 6 illustrates key behaviors of the voltage speculation algorithms using a synthetic workload. For comparison, we show the baseline system, which does not use CPMs; in this case, EnergyScale selects the voltage using the standard voltage-frequency pairs derived during the manufacturing test process. The top trace shows 4 seconds of a constant low million-instructions-per-second (MIPS) background workload combined with a high-MIPS workload that turns on and off every second. The workloads are replicated across all cores and are synchronized. The baseline trace and undervolting traces are aligned to start at the same time. The x-axis shows each algorithm control interval of 32 ms.

The ad hoc method reduces the average MIPS by 1 percent largely due to di/dt induced voltage droop when the high-MIPS workload starts. The performance-aware method, on the other hand, attains 100 percent of the baseline system's MIPS rate without CPMs. The clock frequency trace shows the DPLL slowing down from the Turbo frequency to respond to guardband losses. The ad hoc method causes frequency to drop by



Figure 7. Comparison of undervolting algorithms to CPM disabled. The top graph shows the workload’s average running time (lower is better). The bottom graph shows the average Power7+ V_{DD} rail power (lower is better). The number after each benchmark name signifies the number of hardware threads on each core running the workload.

over 3 percent until the voltage rises. The performance-aware algorithm has a smaller drop in frequency because it doesn’t attempt to reduce voltage as far. Also, it returns to the Turbo frequency sooner because it can proportionally move voltage. The sawtooth pattern in the ad hoc trace is due to its probing the limits of undervolting for the target frequency. The voltage trace shows the ad hoc algorithm hunting up and down for a voltage that yields the Turbo frequency. The performance-aware method reduces voltage until the CPM reading is centered, and doesn’t probe further. Finally, the power trace shows the power reduction on the Power7+ V_{DD} rail. The ad hoc method reduces average power 14 percent while the performance-aware method reduces average power 8 percent.

Saving energy

We measured the SPEC CPU2006 benchmark suite on the Power7+ system to understand how our controllers would perform under significant workloads. Figure 7

compares the performance and power consumption of the voltage speculation methods to the baseline system without CPMs. Each data point is the average value of 11 runs. We ran CPU2006 in a “rate” mode with 1 to 4 copies of the benchmark running on each core, which has 4 hardware threads. We selected the number of threads for each workload to maximize the SPECrate performance for the baseline system with CPMs disabled. The number after the benchmark name in Figure 7 indicates the number of threads on each core running a copy of the workload.

Both methods reduce Power7+ chip V_{DD} power significantly across all workloads. On average, the ad hoc method reduces power by 13 percent, while the performance-aware method reduces power by 11 percent. The DC power of the entire server reduces by 7 and 6 percent, respectively. For the server, the power reduction comes mostly from the reduction in power on the chip V_{DD} rail. However, a small portion comes from reduced fan power because the reduction in

processor power allows the fan controller to maintain the normal processor 70° C operating point at a reduced fan speed.

The average performance improvement across the CPU2006 suite is 0.03 percent for the ad hoc method and 0.19 percent for the performance-aware method. Two workloads, *mcf* and *perlbench*, stand out with performance decreases of 0.9 and 1.7 percent, respectively, for the ad hoc method. The performance-aware method improved both cases. In the context of high-performance servers, power-reducing optimizations are used carefully so that workload performance is not harmed. Therefore, even a 1-percent change in performance becomes important when deciding when to deploy energy-saving techniques in production servers.

In a high-performance context, we believe the performance-aware controller is the most appropriate to deploy because it gives up less performance than the ad hoc method for some workloads. For systems such as mobile devices where battery life is important, the ad hoc controller might make more sense because it has a slightly improved power reduction and similar performance overall.

Active guardband management provides a new capability that allows circuits to operate at a nearly constant, worst-case timing margin. Real-time sensing of the available timing margin by CPMs and fast protection by the DPLL ensure reliability. Furthermore, voltage speculation allows momentary slack in the calibrated timing margin to be converted into a power reduction, and provides additional resilience to noise, which enhances the system's reliability. Because our active guardband management solution is effective and commercially feasible, it is now being employed in several Power7+ server models, allowing them to offer even more energy-efficient operation.

MICRO

Acknowledgments

Philip Restle, Alexander Rylyakov, Daniel Friedman, and Daniel Beece helped lay the groundwork for the CPM-clock feedback control loop. Richard Willaman collected the instantaneous droop scope data. This work was supported in part by DARPA under contract HR0011-07-9-0002.

References

1. C. Lefurgy et al., "Active Management of Timing Guardband to Save Energy in POWER7," *Proc. 44th Ann. Int'l Symp. Microarchitecture*, ACM, 2011, pp. 1-11.
2. A. Drake et al., "A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor," *Proc. Int'l Solid-State Circuits Conf. (ISSCC 07)*, IEEE, 2007, pp. 398-399.
3. M. Floyd et al., "Introducing the Adaptive Energy Management Features of the Power7 Chip," *IEEE Micro*, Mar./Apr. 2011, pp. 60-75.
4. J.A. Tierno, A.V. Rylyakov, and D.J. Friedman, "A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI," *IEEE J. Solid-State Circuits*, Jan. 2008, pp. 42-51.

Charles R. Lefurgy is a research staff member and master inventor at the IBM Austin Research Lab, where he focuses on power management for servers and data centers. He contributed to the power management firmware in Power6 and Power7 servers and the IBM Active Energy Manager product used to monitor data center power. Lefurgy has a PhD in computer science and engineering from the University of Michigan. He is a senior member of IEEE and the ACM.

Alan J. Drake is a research staff member at the IBM Austin Research Lab. His research focuses on developing sensors that measure timing margin in real time in microprocessors. Drake has a PhD in electrical engineering from the University of Michigan. He is a member of IEEE.

Michael S. Floyd is the architect and lead for the Power7+ EnergyScale design at the IBM Austin Server and Technology Group. His work has focused on IBM server development, including hardware bring-up, test, debug, and RAS, along with design, lead, and microarchitect roles on the Power4, Power5, and Power6 processors and support chips. Floyd has an MS in electrical engineering from Stanford University. He is an IBM master inventor.

Malcolm S. Allen-Ware is a distinguished engineer and master inventor reporting to

the director of the IBM Austin Research Lab. His research interests include memory sub-systems; IBM flash storage systems; and dynamic power, performance, and reliability optimization across all IBM systems using Intel, Power, and mainframe z processors. Allen-Ware has an MS in communications theory, digital signal processing, and computer architecture from North Carolina State University.


Bishop Brock is a senior scientist in the IBM Systems and Technology Group. His work on the Power7 project has included the development of hardware verification, micro-architectural validation, and power and performance prediction methodologies for power management hardware and firmware systems. Brock has an MS in computer science from the University of Texas at Austin. He is a member of the ACM.

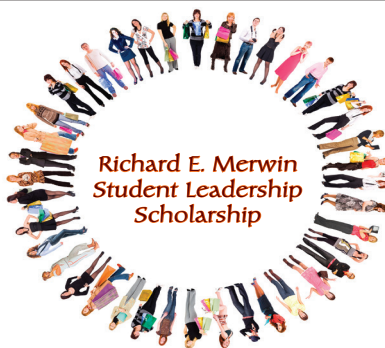
José A. Tierno is a research scientist at Apple. His research interests include self-timed digital circuits and digital replacement of analog circuits. Tierno has a PhD in computer science from the California Institute of Technology.

John B. Carter leads the Future Systems half of IBM Research Austin, with an emphasis on energy-efficient system design, distributed systems, runtime environments for cloud-delivered mobile services, storage, and data center networking. Carter has a PhD in computer science from Rice University. He is a senior member of IEEE.

Robert W. Berry Jr. is the system characterization lead for Power systems at the IBM Austin Server and Technology Group. His previous work has focused on validation for PowerPC processors and design and test of mainframe multichip modules. Berry has an MSE in computer engineering from Syracuse University. He is an IBM master inventor.

Direct questions and comments about this article to Charles R. Lefurgy, IBM, 11501 Burnet Rd., Austin, TX 78758; lefurgy@us.ibm.com.

 Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.



IEEE Computer Society is offering \$40,000 in student scholarships, from \$1,000 and up, to recognize and reward active student volunteer leaders who show promise in their academic and professional efforts.

Graduate students and undergraduate students in their final two years, enrolled in a program in electrical or computer engineering, computer science, information technology, or a well-defined computer-related field, are eligible. IEEE Computer Society student membership is required.

Apply now! Application deadline is 30 September 2013. For more information, go to www.computer.org/scholarships, or email patricia.edwards@computer.org.

To join IEEE Computer Society, visit www.computer.org/membership.

IEEE  computer society

 IEEE