Control Techniques to Eliminate Voltage Emergencies in High Performance Processors

Russ Joseph   David Brooks[†]   Margaret Martonosi

Department of Electrical Engineering
Princeton University
{rjoseph,mrm}@ee.princeton.edu

[†]Division of Engineering and Applied Sciences
Harvard University
dbrooks@eecs.harvard.edu

Abstract

Increasing focus on power dissipation issues in current microprocessors has led to a host of proposals for clock gating and other power-saving techniques. While generally effective at reducing average power, many of these techniques have the undesired side-effect of increasing both the variability of power dissipation and the variability of current drawn by the processor. This increase in current variability, often referred to as the dI/dt problem, *can cause supply voltage fluctuations. Such voltage fluctuations lead to unreliable circuits if not addressed, and increasingly expensive chip packaging techniques are needed to mitigate them.*

This paper proposes and evaluates a methodology for augmenting packaging techniques for dI/dt with microarchitectural control mechanisms. We discuss the resonant frequencies most relevant to current microprocessor packages, produce and evaluate a "dI/dt stressmark" that exercises the system at its resonant frequency, and characterize the behavior of more mainstream applications. Based on these results plus evaluations of the impact of controller error and delay, our microarchitectural control proposals offer bounds on supply voltage fluctuations, with nearly negligible impact on performance and energy. With the ITRS roadmap predicting aggressive drops in supply voltage and power supply impedances in coming chip generations, novel voltage control techniques will be required to stay on track. Our microarchitectural dI/dt controllers represent a step in this direction.

1 Introduction

Supply voltage fluctuations have emerged as a serious cause for concern in high performance processor design. These perturbations, sometimes known as "ground bounce", occur when the processor demands rapid changes in load current over a relatively small time scale. Since the power delivery system has substantial parasitic inductance, this current variation produces voltage ripples on the chip's supply lines. This is significant because if the supply voltage rises or drops below a specific tolerance range, the CPU may malfunction. This fundamental challenge is known as the *dI/dt problem* since the magnitude of these voltage ripples are affected by the instantaneous change of current with respect to time.

At present it is difficult to design a high quality, low impedance power supply system, and industry trends may compound the difficulty in the near future. To see why, first consider that the goal of power supply design is to satisfy de-

mands in load current in a timely fashion, while maintaining a steady reference voltage. This is difficult in practice because real materials add significant amounts of parasitic impedance. The equation $\Delta V = Z \Delta I$ concisely summarizes how current variation (ΔI) and impedance (Z) affect the deviation in supply voltage (ΔV).

Across successive generations of high performance processors, the maximum device current is expected to increase [21]. At the same time, a wide array of dynamic optimizations are being proposed to reduce the average power by implementing low energy modes where power and current are reduced by disabling idle resources. Taking both factors into consideration, the maximum current swing (ΔI) will likely increase. In the same time frame, supply voltages will decrease as transistors are scaled [21]. This will decrease the allowable voltage ripple (ΔV) as well. With progressively larger current swings and smaller tolerable voltage variation, it is clear that the unwanted impedance must be decreased accordingly.

Figure 1 shows the trends in relative supply network impedance for cost-performance and high-performance systems as extracted from the 2001 ITRS roadmap [21]. There are two trends to focus on in this figure. First, to enable desired trends in feature size and supply voltage, a supply network's target impedance must drop rapidly, at roughly 2x every 3-5 years. Achieving these aggressive impedance targets in a cost-effective manner will be extremely challenging. The second trend to note is that the relative difference between target impedances of the cost-performance and high-performance systems is shrinking. The expense of sophisticated power-supply systems may quickly become prohibitive for the cost-performance systems.
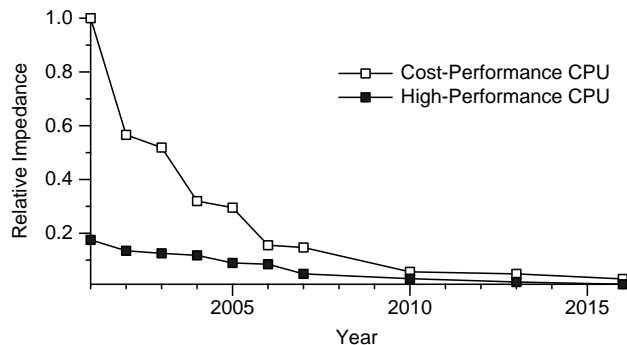

Figure 1: Relative Impedance Trends (from ITRS data)

To reduce total supply system impedance, contemporary distribution networks are first structured to minimize resis-

tance and inductance in the multi-tiered power and ground paths leading from the voltage regulator to the motherboard, package, and finally die. Then large amounts of capacitance are strategically placed throughout the network to counter-act the remaining inductance [23]. To meet even stricter impedance guidelines, more sophisticated supply designs will be required, increasing both complexity and cost. It is important to note that all of the decisions made to meet the necessary electrical parameters of the system must also be compatible with the mechanical and thermal constraints as well. These additional packaging adjustments are vexing not only because they are expensive, but also because they must protect against a *worst-case* possibility that is approached very infrequently in real workloads.

Rather than relying solely on packaging heroics to solve dI/dt, another alternative is to consider an approach that augments reasonable packaging techniques with microarchitectural approaches. This paper demonstrates that effective microarchitectural control of processor current can maintain safe operating voltages with almost no performance or energy impact. Specifically, this work makes several key contributions:

- We characterize the dI/dt behavior of current chips running both current benchmarks as well as extreme-case "stressmarks" and discuss the relevant behavior and time constants in need of control.
- We show the utility of framing the dI/dt and voltage swing problems in terms of linear systems and control theory in order to use numerical techniques to guide our choice of response policies and mechanisms.
- We characterize voltage fluctuations from a micro-architectural standpoint to identify many of the underlying issues and understand how inadequacies in power supply design interplay with the frequency and severity of these fluctuations.
- We examine simple micro-architectural control policies that can eliminate the undesirable voltage transitions. Specifically we independently analyze how effective sensing mechanisms must be in identifying near failure and what actions are appropriate for nullifying the danger. With control theory, we achieve bounds that guarantee our mechanisms are suitable.

The rest of this paper is structured as follows: Section 2 gives an overview of microprocessor supply networks, how they can be modeled, and how the networks respond to different characteristic current fluctuations. Section 3 analyzes power supply issues from a micro-architectural perspective, showing how current fluctuations can modeled at the microarchitectural level, and how software might lead to these current fluctuations. In Section 4, we examine how a simple threshold controller can be used to steady the supply voltage and discuss voltage sensor design issues. Section 5 then focuses on microarchitectural actuator designs, offering both performance and energy evaluations. Section 6 provides a discussion of our findings and offers possible modifications and future directions. In Section 7, we examine how the policies in this paper relate to previous research and finally, we offer a summary in Section 8.

2   Overview of Processor Current/Voltage Swings

As little as ten years ago, most microprocessors exhibited relatively little variation in the power they dissipated or the current they drew [24]. Their average power was close to their maximum power because they employed relatively few techniques to clock-gate units or switch to idle modes to save power where possible.

As power and thermal issues have become increasingly prominent, however, power saving modes have become increasingly common. The use of these modes has increased the variability of power dissipation and current drawn by current microprocessors. Variations in the current required by the processor over time are referred to as the *dI/dt problem* because current is typically denoted by the symbol I. Sudden increases in the current-draw are problematic because they can cause the supply voltage to dip. (This is akin, on a different scale, to the brownouts a building may experience when an occupant turns on a power-hungry appliance.)

Thus, state-of-the-art microprocessors demand sophisticated power supply networks that can provide a very stable supply voltage while delivering a wide range of load currents. The supply voltage must be held at a constant, safe operating level so that on-chip logic and memory function correctly. Spikes, or overshoots, in supply voltage can cause voltage breakdown or thermal problems that literally burn the chip. On the other hand, transient dips, or undershoots, in supply voltage can cause incorrect values to be calculated or stored, leading to lasting errors in application program results. A processor may draw a large amount of current during computation intensive periods and smaller amounts when idle, e.g., waiting for I/O or memory requests to be fulfilled. The voltage must be held constant despite these rapid current swings.

2.1   Power Supply Networks: Basics

In order to build a microprocessor and power supply network in which voltage is sufficiently insensitive to microprocessor current draw, we clearly need a way to reason about the relationship of voltage to current. While modern-day microprocessors are obviously highly-complex systems, electrical models are frequently used that approximate them (or portions of them) in terms of linear circuit theory and Ohm's Law. Ohm's Law states that voltage is equal to current multiplied by a complex impedance, Z. The impedance of the supply network is a function of frequency. To reduce voltage fluctuations, the supply network must maintain a low impedance throughout the frequency range where processor current varies. In essence, a low target impedance will guarantee that the supply voltage stays within its allowable range regardless of the processor's current swings. Thus, *target impedance* has emerged as a de facto standard for evaluating the efficacy of a power supply system.

In practice, it is challenging to achieve the necessary target impedance. As supply voltages decrease, the absolute voltage swings allowed also decrease, and thus target impedances must also get smaller. In particular, non-negligible parasitic resistances and inductances in the materials used to build the power supply system can hinder efforts. As the load current changes, the resistances produce an IR drop, and the inductance creates $L\frac{dI}{dt}$ voltage ripples.

The effective resistance can be reduced by increasing the number of power supply pins, leaving fewer available for I/O. The parasitic impedances present more vexing problems in the form of voltage ripples at broad frequency ranges. Voltage regulators in modern computers have active elements that
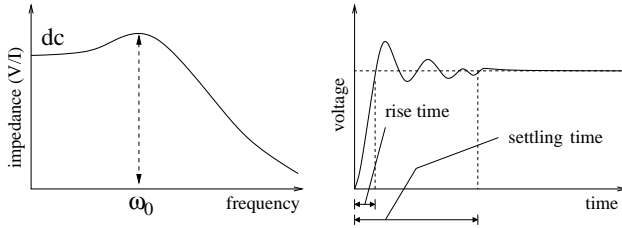
Figure 2: Frequency and transient response of a second order linear system.

can eliminate some of the very low frequency noise. Unfortunately these modules are only effective up to 1kHz. Beyond that range, designers carefully select and position decoupling capacitors on the motherboard, inside the package and on the die to minimize the inductive noise. Typically, very large amounts of capacitance are needed to meet target impedance goals. This increases total packaging costs and complexity.

2.2 Power Supply Networks: Modeling

Thorough evaluation of a candidate supply network involves the construction and simulation of an intricate electrical model. At the final design stages, this could include complicated 2D and 3D electromagnetic field solvers to develop detailed models for the network components. However, earlier stage analysis can be eased by use of a second-order linear model. Second-order systems are appealing because they are simple enough to reduce the computational burdens of simulation, but yet have been shown to be effective for early-stage exploration of power supply designs [10]. In addition, second-order linear system models dovetail very naturally with the large body of well-established control theory techniques [7].

Figure 2 shows canonical frequency response and transient response plots for an underdamped second-order linear system. The graph on the left plots the system's impedance as a function of frequency. The key design criterion, target impedance, is the maximum value of this curve. When modeling power supplies as second-order linear systems, the target impedance occurs at the system's resonant frequency, ω_0 since these systems are underdamped in practice.

The graph on the right in Figure 2 shows how voltage varies in response to a step increase of current in the system. In the parlance of basic linear systems theory, this graph represents the *step response* of the system and is calculated by computing the convolution of the input current waveform with the power supply network's impulse response [13]. The voltage swings up at first, overshoots the target, and then after some settling time eventually reaches the true target voltage. These overshoots and ringing are the phenomena we seek to control.

In this paper, we have implemented a second-order linear model using MATLAB [18]. In particular, the model captures the DC resistance and the peak impedance in the frequency range from 50MHz-200MHz. Our power supply system parameters are consistent with published analysis such as [26] which examines the Alpha 21364 package. More generally, however, this 50-200MHz mid-frequency range is regarded as the most troubling for several gigahertz-and-beyond CPUs due to large inductances in the package. For this reason, we focus primarily on that frequency range. The DC resistance

of 0.5mΩ and resonant frequency of 50MHz used in our analysis is representative of the power supply system for modern 3GHz microprocessor operating at 1.0V. We vary the target impedance to evaluate the effects that it can have on voltage levels and the potential for the voltage control policies in this paper.

2.3 System Responses

To build intuition about how voltage varies with different changes in current, Figures 3, 4, 5, and 6 present a sequence of voltage responses to different current draws. We use these to build intuition about how events occurring at the microarchitectural level may (or may not) translate into undesirable voltage fluctuations.
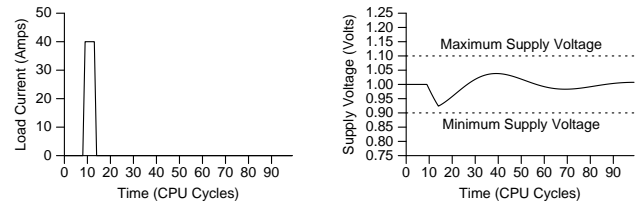

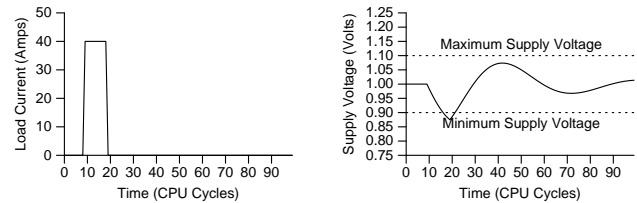Figure 3: Response to a narrow current spike.

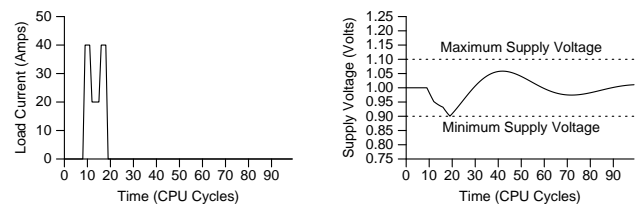Figure 4: Response to a wide current spike.

Figure 5: Response to a notched current spike.

Figure 3 shows a brief spike of increased current demand introduced in the system at time 9 and lasting for a duration of 5 CPU cycles. The spike causes the voltage to dip slightly but the spike's duration is short enough that the network begins to recover before the minimum voltage threshold is crossed. After a short settling period, voltage returns to its original value.

In contrast, Figure 4 shows a second spike of similar magnitude, but with a longer duration—10 cycles. In this case, the duration of increased current draw is long enough to pull the voltage down below the desired minimum voltage threshold.

These two simple examples already highlight a few items of interest to microarchitects. Foremost, for the frequency responses and packaging profiles of current chips, single-cycle or very narrow current spikes are not the main problem to focus on in terms of supply voltage regulation. Narrow current spikes are over quickly enough that they do not draw down supply voltage, even in only modestly regulated

Load Current (Amps) — 50 40 30 20 10 0 — Time (CPU Cycles): 0 10 20 30 40 50 60 70 80 90 100 110

Supply Voltage (Volts) — 1.25 1.20 1.15 1.10 1.05 1.00 0.95 0.90 0.85 0.80 0.75 — Maximum Supply Voltage — Minimum Supply Voltage — Time (CPU Cycles): 0 10 20 30 40 50 60 70 80 90 100 110
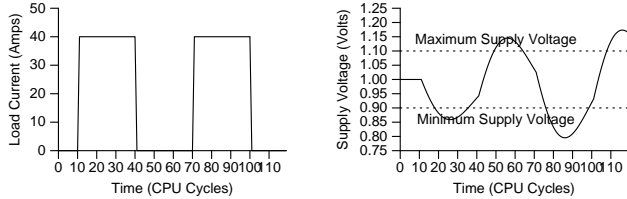
Figure 6: System response to pulses at resonant frequency.

systems. In other words, very short bursts of activity can be tolerated without significant effects on the voltage level. This fact can be exploited by a micro-architectural controller by allowing slightly greedy initial responses for low to high power transitions. Consider a processor that is waiting for a high-latency memory request to be satisfied and is at a low power state with most of its execution units de-activated. When the memory request is satisfied, new ready instructions can be executed immediately, causing a sharp current increase. A micro-architectural voltage controller can allow this behavior—initially assuming that the burst of activity will be relatively short—and not hinder performance. If the burst is indeed short, then no harm is done since the voltage ripple will be small. This could yield significant performance benefits over a more pessimistic policy that slowly re-activated execution units to lessen the impact of the swing.

If the current burst turns out to have a more significant duration, then the voltage controller will have to retreat from its initial, greedy decision and take action to avoid a voltage emergency. For example, Figure 5 depicts a scenario in which input current initially spikes high, but then is forced downward (for example, by disabling functional units or throttling instruction issue). This notched wide spike demonstrates that it is possible to recover from a burst of high activity, by temporarily decreasing the current and giving the supply network a chance to recover. The notch represents the system's microarchitectural control kicking in to keep the supply voltage within the specified range.

A second observation from these figures concerns sensor delay. Since sustained current bursts are problematic for the voltage level, but short bursts can be tolerated, the voltage sensor and control actuator can have some modest amount of delay and still be effective. This is important since most real microarchitectural control implementations will likely require a few cycles to detect problems and begin to respond. Sections 4 and 5 study this delay in more detail.

Finally, the worst-case input can also be deduced from the second-order linear analysis. As shown in Figure 2 the power supply network has a certain resonant frequency and characteristic settling time. The worst-case current swing occurs when transient currents produce large current swings at the resonant frequency. In Figure 6, we show this *dI/dt stressmark* effect by stimulating the power supply network with a train of 30-cycle-wide pulses on a 60 cycle period. This 60 cycle period corresponds to a 50MHz resonant frequency at a 3GHz CPU clock frequency. The first pulse is wide enough to drop the supply voltage below its minimum voltage level. The second pulse is even more dangerous and results in an even greater voltage ripple. This is essentially because the input signal matches the natural frequency and allows some resonance to build up from the first pulse. When the second pulse approaches, its individual effect is superimposed with

the resonant echo to produce larger voltage variation.

3 Mapping to Processors and Applications

Having introduced voltage regulation and dI/dt problems using abstractions from linear system and control theory, we now turn to characterizing real application and architecture behavior, in order to better frame the control problem we face. One can view a program's execution as a progression of current steps, upwards or downwards, of varying widths. As introduced in Figures 3 and 4, many of these steps will either be short enough or narrow enough to not pose a problem for supply voltage regulation. In a processor with aggressive clock gating, we would expect power consumption to vary considerably as programs execute. Cache misses and fills, branch mispredictions, and natural variances in ILP could all account for variances in processor power and current. With significant inductance in the power supply system, these current fluctuations can cause voltage surges and dips as described in the preceding section.

The frequency and severity of these voltage anomalies depends heavily on the design of the power supply network. Since the greater dampening factors increase cost and design complexity, the principle motivation for micro-architectural control techniques is to achieve the safety of the higher dampening ratios with simple and cost effective policies. It is worthwhile to profile real programs under varying parameters to determine what range of dampening ratios are suitable for micro-architectural control.

3.1 Microarchitectural Modeling Methodology

To measure voltage levels, performance, and energy in our micro-architectural simulations, we explored a technique similar to [9]. We started from Wattch [5], an architectural level power simulator based on the widely used Simplescalar Toolset [6]. Wattch models power consumption on a structural level, identifying the usage and activity of micro-architectural structures to generate per cycle processor power estimates which we directly translate into current figures. The processor configuration is presented in Table 1.

Execution Core	
Clock Rate	3.0 GHz
Instruction Window	256-RUU, 128-LSQ
Functional Units	8 IntALU, 2 IntMult/IntDiv
	4 FPALU, 2 FPMult/FPDiv
	4 Memory Ports
Front End	
Fetch/Decode Width	8 inst/ 8 inst
Branch Penalty	10 cycles
Branch Predictor	Combined - 64Kb Chooser
	64Kb Bimodal and 64Kb Gshare
BTB	1K Entry
RAS	64 Entry
Memory Hierarchy	
L1 D-Cache	64KB, 2-way
L1 I-Cache	64KB, 2-way
L2 I/D-Cache	2MB, 4-way, 16 cycle latency
Main Memory	300 cycle latency

Table 1: Processor Parameters

From the MATLAB models discussed in the previous section, we know the impulse response for the power supply network and elementary signal processing techniques, namely convolution summation, allow us to calculate the processor

voltage supply as a function of time. Essentially, this operation consists of convolving the trace of per-cycle current estimates produced by Wattch with the MATLAB-generated impulse response. This convolution, consisting of point-wise multiplications and a final sum, generates a per-cycle view of the supply voltage as previously demonstrated in [9]. Figure 7 shows how our voltage simulation interacts with Wattch.
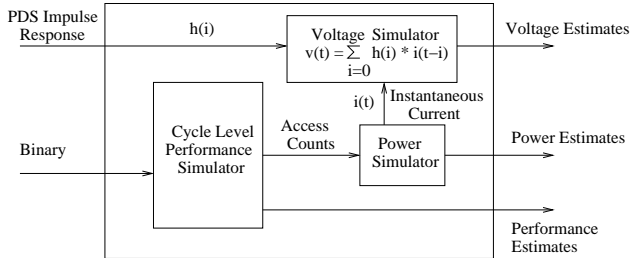

Figure 7: Voltage simulation

Finally, we note that we made several modifications to Wattch to improve the accuracy of the current simulation. First, we used scaling factors from [21] to tune our Wattch model for a 3GHz processor with a nominal supply voltage of 1.0V. We assume that a capable voltage regulator can maintain the ideal supply level of 1.0 V when the processor is at its minimum power level. Since Wattch and Simplescalar do not accurately model the impact of pipeline refill costs following branch misprediction (and since we feared that this effect could represent a significant current swing) we added additional pipeline stages to account for the super-pipelined fetch and decode stages. We assumed that the processor was capable of clock-gating the functional units, writeback bus, and caches. Furthermore, we made modifications to improve the per-cycle power computations, spreading the energy of multiple cycle operations, such as floating point execution over several cycles. This avoids the overestimation of current swings that might occur if the power were accounted for all at once.

3.2 Building a dI/dt Stressmark

For some of our results, we wish to examine controller behavior on extreme-case software that stress-tests the system. We start by taking the worst-case example from Section 2.3 and showing how to map it into a piece of software whose current draw versus time displays a similar, nearly square-wave, pattern. Figure 8 shows the main loop body of our resulting "dI/dt stressmark": a snippet of Alpha assembly code that produces periods of high and low activity when executed on our target platform. The loop body starts with a period of very low activity (and low current draw) because the divide (divt) operations produce long stalls. Following this low-current period is a high-current period in which dependent instructions store the floating point result to memory, reread it, and then store it to integer registers. (Dependencies are depicted via dotted arrows.) To exacerbate the power shift, operand values are chosen to produce the maximum possible transition activity as results are read and written. The number of instructions in the loop is chosen so that its execution time will closely match the resonant period of the power supply network, mimicking the worst-case resonance previously shown in Figure 6.

Obviously, such extreme-case power stressmarks must be


```
    ldt     $f1, ($4)    ◄----┐
    divt    $f1, $f2, $f3         │
    divt    $f3, $f2, $f3         │
    stt     $f3, 8($4)            │
    ldq     $7, 8($4)            │
┌─► cmovne  $31, $7, $3          │
└─► stq     $3, $(4)            │
    stq     $3, $(4)            │
    stq     $3, $(4)            │
    ...                          │
    stq     $3, $(4)    ◄----┘
```

Figure 8: Loop body for $\frac{dI}{dt}$ stressmark.

crafted with significant knowledge about the power, packaging, and timing characteristics of the processor being targeted. Furthermore, the task is made more difficult by the fact that adding instructions to manipulate operands or increase functional unit activity can affect the loop timing and move it off the resonant frequency. To test how closely our stressmark software approaches the theoretical worst-case effect, we ran the the stressmark software through an architecture-level power simulator to generate a time-varying current profile. We then input this current profile into our second-order linear systems model to see how voltage would be impacted.
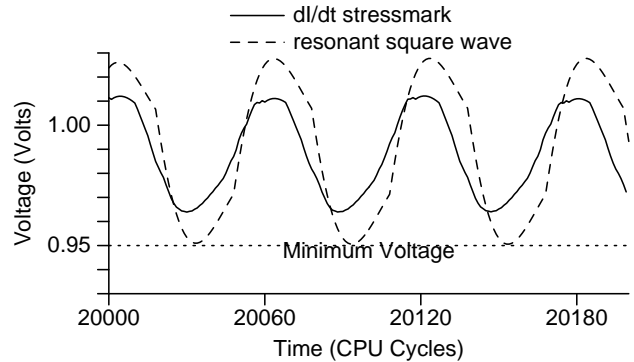

Figure 9: Maximum height pulse at resonant frequency versus $\frac{dI}{dt}$ stressmark.

As shown in Figure 9, the voltage swings induced by the stressmark are not as extreme as the true worst-case, but are nonetheless severe enough to stress-test a system's voltage control capability. We present results for this stressmark, in addition to SPEC, in the benchmark studies in later sections of this paper.

3.3 Characterizing the SPEC Benchmarks

We next wish to explore how behavior in the SPEC benchmarks compares to the more extreme case previously explored. Using the microarchitectural modeling techniques described in Section 3.1, we simulated all 26 SPEC2000 benchmarks for 200 million instructions after skipping the first billion instructions.

Recall that target impedance represents the impedance value that will keep the voltage within a specified range. Impedance values equal to or lower than the target impedance are desired, but are expensive to achieve through packaging alone. Impedance values greater than the target impedance are simpler and cheaper to achieve, but may allow the voltage swings to be undesirably large.

	Percent of Target Impedance			
	100%	200%	300%	400%
Benchmarks w/ Voltage Emergencies	0	0	1	14
Emergency Frequency (Average)	0%	0%	roughly 0%	<0.00003%
Emergency Frequency (Maximum)	0%	0%	roughly 0%	0.0005%

Table 2: Voltage Emergencies on SPEC2000 Benchmarks

In Table 2, the leftmost column gives benchmark characteristics if the achieved system impedance were equal to (i.e., 100% of) target impedance. Proceeding rightward from there, the columns show what happens as the system impedance is larger (and thus less desirable) multiples of target impedance. Voltage emergencies are defined as instances where voltage swings greater than 5% occur. By definition, voltage emergencies cannot occur if the target impedance is met, so the leftmost column indicates that none of the SPEC benchmarks have voltage emergencies. As one moves rightward, towards cheaper but higher-impedance power supply networks, the incidence of benchmarks with voltage emergencies increases somewhat. Nonetheless, the SPEC benchmarks show behavior that is much less taxing than that of the stressmark and in fact an impedance that is 200% of the target impedance is still good enough to have 0 voltage emergencies across all of SPEC.

Figure 10 rounds out the benchmark characterization by showing (for the 100% impedance case) how voltages distribute themselves across the possible range of values. Although the 100% target impedance case means that the voltage is never out of spec, the distributions are interesting because they show the degree of voltage variation the different applications induce. The benchmark ammp, for example, has poor cache performance with many stall cycles and low IPC. It rarely sees large current or power variations, and as a result, its voltages tend to be quite stable. In contrast, swim is a benchmark with moderately low IPC, but with more variations in its behavior. As a result, its voltage distribution shows that it spends more time at different voltage levels.

In the discussions that that follow, we focus on the 200% impedance case. In this scenario, a potentially lower cost and complexity packaging solution is augmented with a hardware control mechanism which we introduce in Section 4. This combination is used in lieu of a more sophisticated and expensive packaging solution that could guarantee safe operation on its own. As Table 2 demonstrates, the SPEC benchmarks still do not produce voltage emergencies under this impedance, but we note that our stressmark does. We conducted experiments with both real benchmarks and our stressmark to determine how the control policies affect real application performance, verify that they meet the intended voltage specifications, and offer likely worst-case bounds on execution-time and energy increase.

4 Exploring Microarchitectural Control: Sensor Design and Evaluation

Voltage emergencies are an example of a worst-case design constraint: no emergencies can be tolerated, and a microarchitectural regulator must offer guarantees on voltage regulation. While heuristic strategies might be able to quell voltage fluctuations under many operating conditions, it is difficult to bound their behavior. On the other hand, crafting a regulator under the guidelines of control theory offers significant benefits. As we demonstrate in this paper, worst-case bounds are possible with such an approach. Furthermore, the design and analysis procedure can be significantly stream-lined, reducing both cost and complexity.

In this section, we propose a simple threshold control strategy that can be used to eliminate voltage emergencies and we discuss the implications of building a sensor mechanism appropriate for this control strategy. By working within the established framework of control theory, we benefit in several ways. First, we can easily identify the maximum voltage ripple and verify that it is within the allowable range. In addition, we can separately evaluate the performance and energy impact of different micro-architectural strategies since we have already guaranteed correctness.

4.1 Threshold Control

This paper proposes the use of threshold control for dI/dt. Rather than measure a value exactly, threshold controllers operate by sensing transitions from one range of a value to another range, and triggering actions accordingly. Because we need only sense voltage ranges, rather than precise voltage values, the components of the control mechanism are simpler. We believe that they could be easily implemented with reasonable delay in a real processor.

In our proposed controller, a simple voltage sensing mechanism communicates directly with the actuator logic which cooperates with the existing pipeline control and clock gating logic to disable or enable processor units as needed. The sensor's only function is to determine whether or not the processor is dangerously close to a voltage emergency. In particular, it registers one of three possible output values to the compensation logic: Voltage Low, Voltage Normal, and Voltage High. This mechanism could be significantly easier to implement than a sensor which samples and digitizes the voltage level in an attempt to determine exactly how significantly it deviates from the standard level. Here we only wish to determine whether or not the voltage is relatively high or low. When the voltage surpasses some predetermined threshold, it signals the compensation logic, which responds by stimulating the actuator. The actuator temporarily suspends the processor's normal operation and performs some set of tasks to quickly raise or lower the voltage back to a safe level. There are several micro-architectural actions that could serve as actuation mechanisms; they are discussed in Section 5 which follows. Once a normal voltage level has been restored, the processor transitions back into normal operating mode and standard execution resumes.

Figure 11 shows how a micro-architectural controller can improve the voltage level. At the beginning of the trace, the processor voltage is close to the ideal 1V. During a
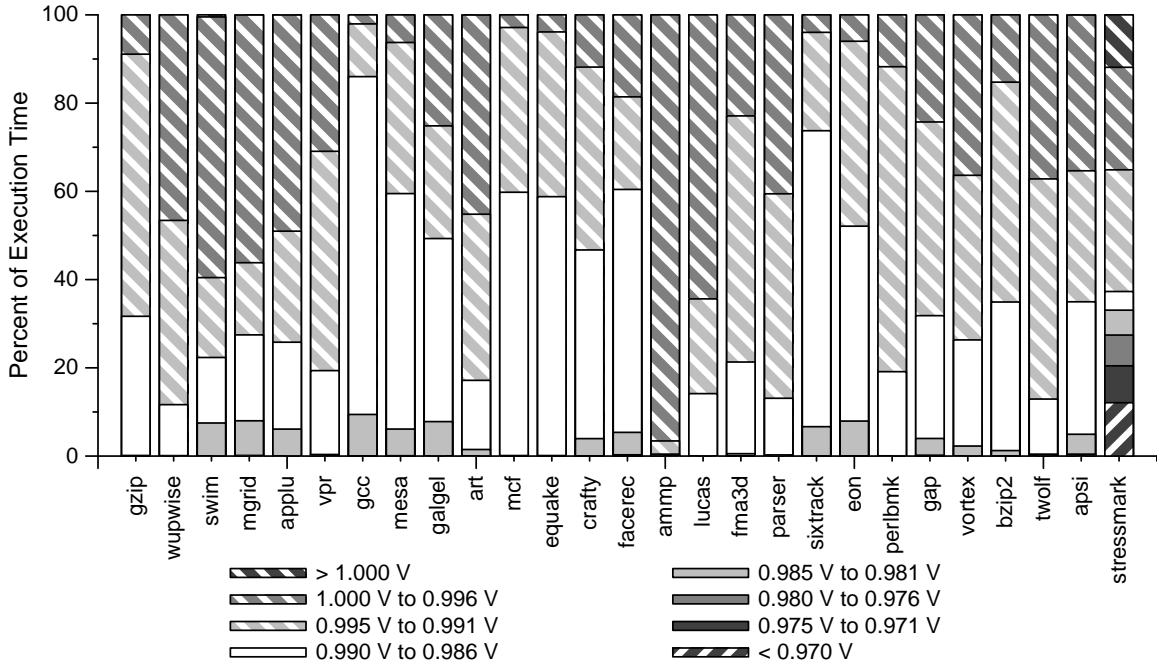
Figure 10: Voltage distributions for the SPEC2000 benchmarks and our stressmark for the case when the peak impedance matches the target. Note that some benchmarks (e.g., ammp) have exceptionally stable voltage with few dips, while other benchmarks (e.g., galgel) vary across a wider range of voltage levels.
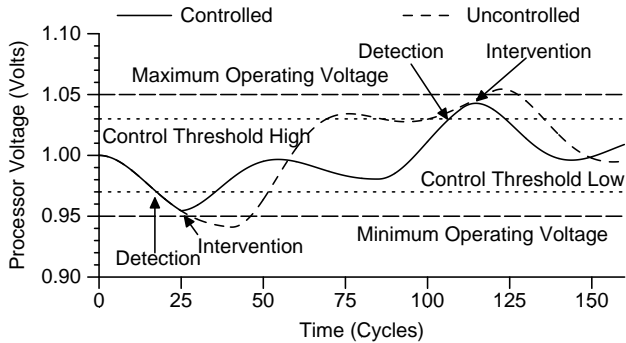

Figure 11: A simple threshold controller in action.

brief period of activity, the voltage level rapidly decreases. Unchecked, this behavior would lead to a voltage emergency. A threshold controller could however sense the rapid drop in voltage and respond, avoiding the emergency and allowing some time for recovery. A similar sequence of events would take place if the voltage rose above a threshold, and the actuator would respond with a different mechanism to effectively calm the voltage peak.

The subsections that follow discuss the implementation of the sensor mechanism, and some key design decisions for the threshold controller overall.

4.2 Sensor Mechanism

In [9] the authors proposed on the fly voltage computation using convolution hardware. While this would yield an accurate voltage reading, it involves a series of tens or hundreds of multiply-accumulates; thus would be difficult and energy-intensive to implement hardware for this that would produce

the answer within the few cycles needed for effective operation.

We believe that there are existing circuit level voltage sensing techniques that could be used for detecting voltage emergencies. In particular, analog circuit designers commonly employ bandgap references which rely on properties of silicon to provide a stable reference voltage [1, 2, 11]. By nature these low-noise voltage references are not sensitive to temperature or supply variations and could be used for comparison with the fluctuating power supply [15]. Another possible alternative are detector circuits based on buffer delay lines or inverter chains. These devices rely on relationships between voltage supply level and transistor switching speeds and have been used to regulate dynamic voltage scaling implementations [14]. These types of techniques could be used to provide fast threshold detection with roughly 1-2 cycles latency.

4.3 Setting Thresholds and Bounding Voltage Swings

The choice of how to set voltage-high and voltage-low thresholds is at the core of our control implementation. For example, the voltage-low threshold obviously has to be high enough to guarantee that once the sensor detects the system has crossed this threshold, there is time to actuate an effective response. If the threshold is set too conservatively, however, it could trigger many false alarms when there is no immediate danger. This could potentially harm performance if the voltage mediation includes deactivating some pipeline resources. There is similar difficulty in choosing the correct voltage-high threshold; it must be set to allow effective responses, but setting it too conservatively may waste energy. This is because the actuator's response to a voltage-high level
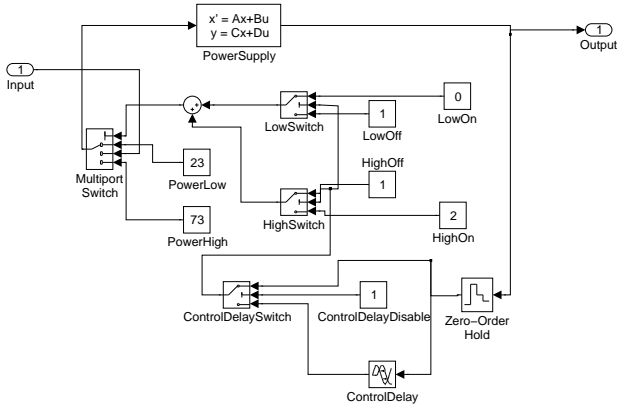
Figure 12: Simulink model of control mechanism.

may enable inactive resources to temporarily raise the current draw and lower the voltage.

Ultimately both the sensor and actuator have an impact on the controller's efficacy, since their actions (and their delays) impact whether a response is timely and effective. In this section, we separate the issues by examining sensor properties assuming an ideal actuator. Section 5 then focuses on the microarchitectural issues of building real actuators.

One of the advantages of our control theoretic view of the problem is that we can very methodically choose appropriate threshold levels given different assumptions about (i) acceptable voltage fluctuations, (ii) sensor delay, and (iii) sensor error. Figure 13 outlines our methodology for exploring microarchitectural voltage control.

First we analyze both the power supply system and processor model. We are specifically interested in finding worst-case scenarios. In particular, we examine the power supply system to find the resonant frequency and peak impedance. We also examine the processor power model to find minimum and maximum power values. To identify optimal emergency thresholds, we relied on MATLAB/Simulink, software packages which are used widely in the control engineering community to analyze system characteristics [18]. With the information from our analysis, we can generate a suitable system model and true worst-case waveform in Matlab/Simulink. Then under Matlab/Simulink we analyze the model with the worst-case waveform to find the appropriate voltage high and low thresholds to guarantee that voltage stays within the intended range. Using the methodology described in Section 3, we simulate processor voltage and performance under Wattch, using the control thresholds produced by Matlab/Simulink.

Figure 12 shows our Simulink model of the controller. By varying parameters on the model, we use Simulink to solve for the voltage thresholds that guarantee stability and system integrity while minimizing performance and energy impact. We can determine specifically how sensor delays and errors affect the voltage threshold. Controller delay is accounted for via the "ControlDelay" modules at the bottom of the diagram. Although not illustrated in the diagram, we also considered the effect of sensor error in our analysis and show our results in Section 4.5.

Table 3 shows a collection of Simulink threshold values collected for sensor delay values ranging from 0 cycles to 6 cycles. The 200% impedance setting presumes that voltages are allowed to fluctuate well beyond an allowable plus/minus
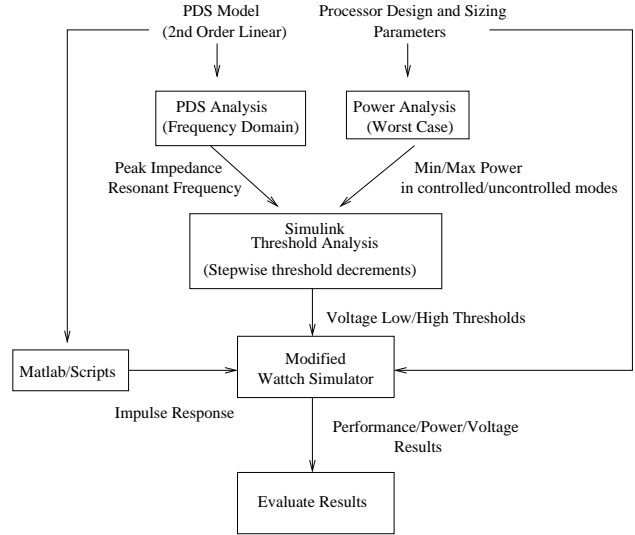

Figure 13: Design flow for microarchitectural voltage control.

Delay (cycles)	Low Threshold (V)	High Threshold (V)	Safe Window (mV)
0	0.956	1.05	94mV
1	0.956	1.017	61mV
2	0.960	1.017	57mV
3	0.962	1.017	55mV
4	0.966	1.017	51mV
5	0.971	1.017	46mV
6	0.976	1.017	41mV

Table 3: Voltage thresholds under delay for 200% impedance

5% from the nominal value. The figure demonstrates that as sensor response degrades, the operating voltage range shrinks. This is intuitive because when detection of voltage levels is slow, the control system must be conservative in order to guard against the possibility that the system transitions into an emergency before true detection and response can occur. As the delay increases, so does the uncertainty in voltage level. To account for this, the control theoretic bounds narrow the operating range in order to guarantee the voltage specification.

4.4 Effect of Sensor Delay

To examine the effect of sensor error and delay on both energy and performance, we modified Wattch to monitor the voltage level and trigger activation and de-activation of processor components to implement our ideal actuator. We note that none of the actuator mechanisms alter the program correctness since the processor does not drop instructions that have temporarily stalled, nor are incorrect values stored when extra execution resources are activated.

We consider the effects on processor performance and energy due to sensor delays ranging from 0 to 6 cycles. Figures 14 and 15 plot sensor delay's impact on performance and energy. In particular, they plot performance and energy degradation for the average of the eight SPEC2000 benchmarks which showed some voltage variation (swim, mgrid, gcc, galgel, facerec, sixtrack, and eon) as well as the stressmark described in Section 3.2. These figures show that while the SPEC benchmarks are largely unaffected by increases in sensor delay, the

performance loss and energy increase of the dI/dt stressmark is significant. Recall, however, that the stressmark is a scenario contrived to be nearly worst-case. While the system must be built to guard against worst-case behavior, the expected performance impact on real applications is typified by the results shown here for SPEC.
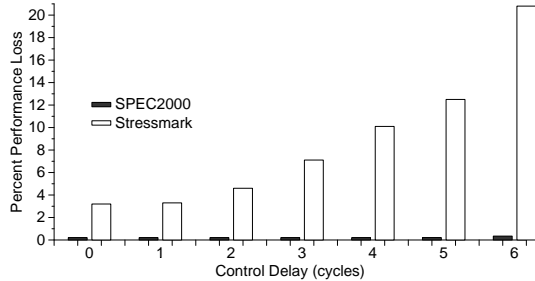

Figure 14: Impact of sensor delay on performance.

4.5 Effect of Sensor Error

Sensors of all kinds exhibit error in their readings which can affect the performance of feedback control systems. In this section we quantify the performance and energy impact of error in the voltage sensing mechanism used in the feedback control. To account for this error, we introduced white noise into the simulated voltage readings using a random number generator. We consider the effect of sensing error by introducing noise with magnitude in the range of 10mV to 25mV and examining the effect on performance and energy. To compensate for potentially inaccurate readings, the voltage high and voltage low thresholds in Table 3 have to be modified to account for the sensing error by correspondingly lowering and raising the threshold by the potential error. Thus we would expect that both performance and energy might suffer if the sensing error grows too large.

Figure 16 agrees with this conjecture. The plot shows the mean performance loss and energy increase of the same SPEC benchmarks from the previous section when sensor error is increased. We see that small threshold errors (less than 15mV) have a negligible effect on both performance and power. However, as the error increases, the operating windows decrease and both performance and energy suffer.
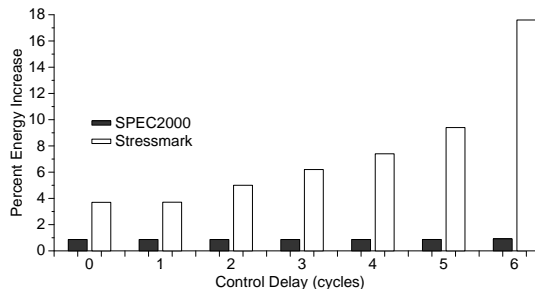

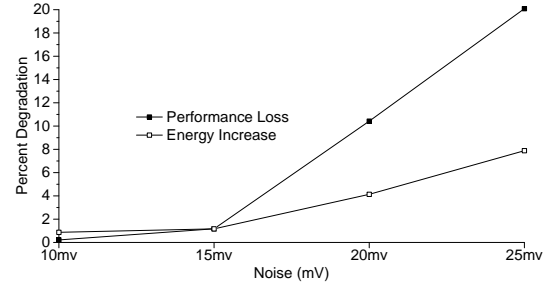Figure 15: Impact of sensor delay on energy.

Figure 16: Impact of sensor error on performance and energy.

5 Exploring Microarchitectural Control: Actuator Design

Although the sensor is responsible for determining when an emergency is about to occur, an equally important task is responding to the crisis to avoid the emergency. In control systems terminology, the agent that intervenes is known as an actuator. In the previous section, we explored the relationship between sensor properties and performance/energy. This was achieved by assuming an ideal actuator and varying parameters of the sensor. We now turn to considering microarchitecture-based actuator design in this section.

There are several micro-architectural techniques that may be useful for an actuator. Any quick-acting mechanism that can quickly lower the processor current to avoid a voltage low emergency and increase the processor current to prevent a voltage high emergency could be useful. For example, electrical solutions like voltage scaling can significantly reduce the processor power; unfortunately, the time scales needed for such transitions are fairly large. As previously demonstrated, voltage control needs to act within 1-5 cycles.

One simple and fast-acting architectural approach is to use clock-gating of processor resources for voltage control. For example, when the processor voltage sensor indicates a "voltage low" level, active processor units could be deactivated, quickly lowering the processor current draw and power dissipation, thereby allowing the voltage level to recover. In a similar vein, beyond a voltage high threshold, disabled execution resources can be fired up in extra activity to quickly increase the processor's current draw and again allow a recovery.

Thus, a very central design decision for the actuator is which execution resources should be controlled by it. This is important since it affects performance, energy and the ability to allow the processor voltage to recover. In the remainder of this section, we consider how these design decisions affect the performance and energy behavior.

5.1 Granularity of Hardware Actuation

In some prior work [4, 17], the processor front-end is throttled either to reduce energy or to improve a thermal profile. And obviously, existing processors already make extensive use of functional unit clock-gating for energy reduction [8]. Here we propose leveraging and slightly augmenting localized control of pipeline units to serve as an actuation mechanism to regulate processor current and voltage.

When considering which execution units should be activated/deactivated, there are several interesting issues. First, proper control requires that sometimes we want to turn off a

unit in use (to reduce current draw to recover from a voltage-low state) while at other moments, we want to fire up an idle unit to smooth out a sudden dip in current draw and recover from a voltage-high state. (We refer to these extra voltage-control uses of idle units as "phantom firings".) Thus, the units we choose for actuation should be able to be both fired-up or disabled without affecting program correctness. Another related issue is ease of control. Some fairly self-contained execution resources like functional units are much easier to envision turning on and off quickly, while larger and more complicated structures like issue queues and re-order buffers may be more challenging to clock-gate or phantom-fire at a fine granularity.

Clearly different pipeline structures have different power consumptions; turning on or off a higher-power resource can be a quicker but more heavy-handed power control approach. This heavy-handedness can cost extra energy (for example, when phantom-firing a high-energy unit solely for voltage control).

Finally, different pipeline structures have different contributions to overall performance. This means that some ordinarily attractive high power structures should not be disabled because they are simply too essential to performance.

In this paper, we evaluate three levels of actuation granularity. The first level, functional unit (FU) control, allows the actuator to clock-gate or phantom-fire all of the functional units on a given cycle. To extend the scope of control, we also consider clock-gating/phantom-firing caches. We note that these operations still preserve cache state, and do not modify the state or content of cache lines. They merely disable or enable the clock signal to cache structures. A medium-granularity approach is FU/DL1 control, in which functional units plus the level-one data cache are used as the regulation mechanism. Finally, the coarsest-granularity, FU/DL1/IL1, regulates using the block of functional units plus level-one data cache plus level-one instruction cache. The controller success as well as performance and energy implications are examined in subsections to follow. In our analysis, we assume that a drop below the low voltage threshold deactivates all of the controlled units until the voltage level is above the threshold again. In a similar fashion, a rise above the threshold activates all of the controlled units until the voltage has recovered.

In our research, we have also examined other resource possibilities, but these three were particularly promising under our Matlab/Simulink analysis. Furthermore, they seem implementable with reasonable changes from existing microprocessor pipeline control. We address other control policy and mechanism variations in Section 6.

The subsections that follow assess the performance impact, and energy impact of the possibilities we have outlined here. We consider the eight SPEC2000 benchmarks that had the most challenging voltage emergencies from our characterization in Section 3.3.

5.2 Actuation Performance Impact

The results of the three proposed actuation mechanisms are shown in Figures 17. Of the three proposed actuation mechanisms, we have found that solely controlling the functional units (fixed and float pipelines) is unsuccessful. The fine granularity of this technique means that it does not have the necessary leverage to reshape voltage quickly. For small controller
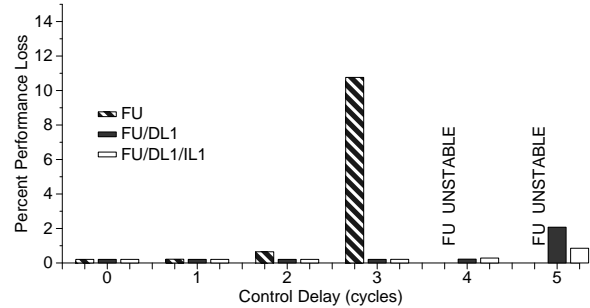
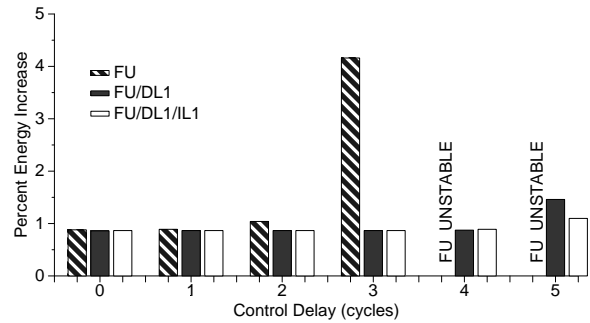Figure 17: Impact of guarded actuator delay on performance for SPEC2000.

Figure 18: Impact of guarded actuator delay on energy for SPEC2000.

delays, it is usable, but the technique becomes unstable for controller delays of three or more. Even in the range when this type of controller is stable, the performance loss can become significant.

For the other two strategies, the actuation is effective enough that it results in almost no performance loss as long as controller delay can be kept to four cycles or less. Performance loss was less than 2% for both FU/DL1 and FU/DL1/IL1.

We also evaluated the power stressmark to provide a partial sanity check for the efficacy of actuation mechanisms and bounds on potential performance loss. As expected, we witnessed more extreme performance losses, but voltage emergencies were protected. With very large delays of five cycles the performance loss was 24.5% for FU/DL1 and 23.2% for FU/DL1/IL1 compared to less than 2% for SPEC2000. But with zero cycles of control delay, the power stressmark experienced slightly less than a 6% performance drop. Nonetheless, these performance drops are acceptable for an unlikely, nearly-worst case scenario.

5.3 Energy Impact

We now consider the additional energy overhead that is incurred by the dI/dt controller mechanism. Extra stalls introduced by the actuator to eliminate voltage low emergencies will increase the total execution time, and subsequently increase total energy. Also in the case of voltage overshoot, additional power is burned by phantom-firing.

Figure 18 shows the impact of the actuator mechanisms on SPEC. The energy overhead tends to be less than 1%. Energy increases slightly with larger controller overheads. As expected, the energy increase of the stressmark was higher than that for SPEC. Even so, energy increases are fairly mod-

est (less than 5%) for 0 cycles of control delay, increasing to 22% energy increases with the extreme value of 5 cycles of control delay.

6   Discussion and Future Work

In proposing microarchitectural control mechanisms for the dI/dt problem, this paper represents an important first step in a complex issue. Future work can build off this foundation in several ways.

First, as microarchitects, it is natural for us to consider conducting more detailed studies of even more actuation mechanisms. In addition, one might want to consider using different actuation mechanisms for voltage-high and voltage-low emergencies. This asymmetry could exploit the fact that some CPU units are better suited for easy clock-gating (for the more common voltage-low emergencies) while other units are easier to control for phantom-firings (for the less common voltage-high emergencies). Likewise, more detailed sensor studies and circuit designs will help to move this research into widespread use. Another issue pertains to processor recovery from voltage control actuation. That is, the CPU must throw away results from phantom-firings and restart instructions as needed. In this paper, we assumed that the control logic could protect necessary state and recover without back-tracking or completely re-starting instruction execution. Other possibilities include re-playing instructions or flushing the pipeline if execution cannot resume mid-stream. We performed some initial experiments which show similar performance/energy results with these options, but further exploration may prove interesting.

It is tempting to consider exploring a variety of other, more sophisticated control approaches, such as the P-I-D controllers used in some previous work [22, 16]. We note here, however, that our initial explorations with more P-I-D controllers for dI/dt control raised some concerns. First rather than a simple High/Normal/Low voltage status, P-I-D controllers need a more definitive voltage reading to determine how to respond. This might significantly increase complexity or latency, which is problematic since very short turnaround times are crucial. Secondly, a textbook digital P-I-D controller would require a series of additions and multiplications based on previous voltage readings to determine a response. Again, this would likely increase the control delay, impacting performance. Work on other control algorithms may, however, prove more fruitful.

Another key area of future research will lie in improving the locality at which we model dI/dt effects. Local power supply swings in different chip quadrants can be an important issue to consider, in addition to the more global effects considered here.

Finally, we consider the second-order linear models from this study to be exceptionally appropriate for the hybrid architecture/circuits research we have discussed here. They are, however, somewhat more abstract than the more detailed circuit models that packaging engineers typically rely on for later-stage design. Space constraints prevented us from including extensive validations between different levels of modeling, but we feel that such comparisons are important long-term.

7   Related Work

Until recently most research on power-aware, high performance computing targeted reduction in average power. While reduction in average power can translate into better energy-efficiency and longer battery life for mobile computing, there are a number of other related issues also in need of attention.

In this paper, we investigate the potential benefits of voltage control, which is closely coupled to energy reduction strategies. Many high-performance processors switch operating modes, disabling and re-enabling architected structures to improve energy-efficiency. However, this can cause dramatic swings in processor current and as a result, dangerous supply voltage fluctuations. In [25], the authors presented compilation techniques to mitigate the voltage fluctuations. Specifically, they scheduled instructions to minimize the number of rapid changes in processor power level. In [19] a shift register based technique was employed to gently step functional unit power up and down to reduce the maximum current swing.

Full micro-architectural voltage control was proposed in [9]. This paper introduced regulation of processor voltage via activation and de-activation of functional units. The authors assumed that voltage could be tracked by performing a series of computations. A direct implementation of their voltage calculation would be difficult, however, given the small time-window to respond to a voltage crisis.

Another well known consequence of the increasing performance in high performance processors has been the rapid rise of thermal density. This increases the burden on packaging materials to redistribute heat and the cooling system to dissipate it. This is a considerable challenge even for state-of-the-art packaging and cooling technologies and ultimately increases total system cost [3]. Thermal control systems have been evaluated [4, 22] and implemented [12, 20] to regulate processor temperature with micro-architectural techniques. A key difference is that thermodynamics of modern CPUs have much larger time constants than the electrical systems associated with voltage control. This makes delay less of an issue in thermal control than it is in voltage control. Brooks and Martonosi evaluated several micro-architectural response mechanisms under a threshold thermal control policy. Skadron *et al.* introduced use of formal control theory for temperature regulation [22]. They demonstrated that a PID controller in concert with clever instruction fetch throttling could produce even better results. Furthermore, they also presented an improved thermal model and demonstrated that control theory could be used to prove bounds on the temperature regulator's performance. Subsequent work by Lu *et al.* applied control theory techniques to Dynamic Frequency and Voltage Scaling as well [16].

Our work examines how control theory can be applied to voltage regulation. Like Skadron *et al.*, we take advantage of the bounds that control theory provides to ensure that processor dynamics stay within their intended operating range. In addition, we use control theory to determine what are suitable thresholds. We also generalize the control mechanisms presented in [9] to examine other micro-architectural policies. We also present a characterization of the dI/dt problem from a micro-architectural perspective. We feel that if micro-architects are to contribute in the effort to reduce inductive noise, they need accessible and accurate models and paradigms for the power distribution network so that they can

focus on the important issues.

8 Conclusions

Increasingly aggressive design points for microprocessor supply voltage and power supply impedance are predicted in upcoming generations on the SIA ITRS roadmap [21]. With impedances requiring 2X improvements every roughly 3-5 years, voltage and dI/dt regulation based solely on packaging techniques may become prohibitively expensive in upcoming processor generations. Furthermore, this extra cost and complexity would guard against an infrequently occurring worst-case.

With these trends in mind, this paper has proposed microarchitectural mechanisms for microprocessor voltage and current control. By using control theory and linear systems theory as foundations for our work, our methodology for designing a control system offers worst-case bounds on its behavior. Furthermore, the systems theory steps we take make designing a system with desired voltage swings a methodical, rather than trial-and-error, process.

Examining the frequency response curves and packaging constraints from real processors also allows us to construct and evaluate a "dI/dt stressmark" with behavior that resonates at the worst-case frequency of the processor package. We can then compare it to the behavior of less taxing SPEC2000 benchmarks.

Overall, we find that microarchitectural techniques for dI/dt control actuation are feasible. Given the 50-200MHz frequency range that is most problematic, microarchitectural control can be built with delay values that are sufficiently small to allow safe operation. While the dI/dt stressmark sees performance/energy impact on the order of 20% from microarchitectural control, the impact on mainstream applications is nearly negligible. Overall, we view these techniques as increasingly important assists to packaging-level power supply regulation.

9 Acknowledgements

References

[1] P. E. Allen and D. R. Holberg. *CMOS Analog Circuit Design*. Holt, Reinhart, and Winston, 1987.

[2] H. Banba et al. A CMOS bandgap reference circuit with sub-1V operation. *IEEE Journal of Solid-State Circuits*, 34(5):670–674, May 1999.

[3] S. Borkar. Design challenges of technology scaling. *IEEE Micro*, 19(4):23–29, July 1999.

[4] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the Seventh International Symposium on High-Performance Computer Architecture (HPCA-7)*, January 2001.

[5] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th International Symposium on Computer Architecture*, June 2000.

[6] D. Burger and T. M. Austin. The SimpleScalar Tool Set, Version 2.0. *Computer Architecture News*, pages 13–25, June 1997.

[7] R. C. Dorf. *Modern Control Systems*. Addison-Wesley, 1967.

[8] M. Gowan, L. Biro, and D. Jackson. Power considerations in the design of the Alpha 21264 microprocessor. In *35th Design Automation Conference*, 1998.

[9] E. Grochowski, D. Ayers, and V. Tiwari. Microarchitectural simulation and control of di/dt-induced power supply voltage variation. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture (HPCA-8)*, February 2002.

[10] D. J. Herrell and B. Beker. Modeling of power distribution systems for high-performance microprocessors. 22(3):240–248, August 1999.

[11] W. Holman. *A Low Noise CMOS Voltage Reference*. PhD thesis, Georgia Insitute of Technology, 1994.

[12] Intel Corp. Intel Pentium 4 thermal management, 2002. http://www.intel.com/support/processors/pentium4/thermal.htm.

[13] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.

[14] J. Kim and M. Horowitz. An efficient digital sliding controller for adaptive power supply regulation. In *Proceedings of IEEE Symposium on VLSI Circuits*, July 2001.

[15] E. Kussener et al. New regulated voltage down converter based on modified bandgap cells. In *Proceedings 26th European Solid-State Circuits Conference*, September 2000.

[16] Z. Lu et al. Control-theoretic dynamic frequency and voltage scaling for multimedia workloads. In *Proc. International Conference on Compilers, Architectures, and Synthesis for Embedded Systems*, Oct 2002.

[17] S. Manne, A. Klauser, and D. Grunwald. Pipeline gating: Speculation control for energy reduction. In *Proceedings of the 25th International Symposium on Computer Architecture*, pages 132–41, June 1998.

[18] MathWorks Corp. Matlab and simulink software packages, 2002. http://www.mathworks.com/.

[19] M. D. Pant, P. Pant, D. S. Wills, and V. Tiwari. Inductive noise reduction at the architectural level. In *Proceedings of the Thirteenth International Conference on VLSI Design*, January 2000.

[20] H. Sanchez et al. Thermal management system for high performance PowerPC microprocessors. In *Proceedings of CompCon '97*, Feb. 1997.

[21] Semiconductor Industry Association. International Technology Roadmap for Semiconductors, 2001.

[22] K. Skadron, T. Abdelzaher, and M. R. Stan. Control-theoretic techniques and thermal rc-modeling for accurate and localized dynamic thermal management. In *Proceedings of the Eighth International Symposium on High-Performance Computer Architecture (HPCA-8)*, February 2002.

[23] L. D. Smith, R. E. Anderson, D. W. Forehand, T. J. Pelc, and T. Roy. Power distribution system design methodology and capacitor selection for modern cmos technology. *IEEE Transactions on Advanced Packaging*, 22(3):284–291, August 1999.

[24] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. *IEEE Transactions on VLSI Systems*, 2(4):437–445, December 1994.

[25] M. C. Toburen. Power analysis and instruction scheduling for reduced di/dt in the execution core of high-performance microprocessors. North Carolina State University. Master's Thesis, Aug. 1999.

[26] M. Tsuk et al. Modeling and measurement of the alpha 21364 package. In *Proc. 2001 IEEE Topical Meeting on Electrical Performance of Electronic Packaging (EPEP)*, Oct. 2001.