

IBM Research Report

Wavelet Analysis for Microprocessor Design: Experiences with Wavelet-Based dI/dt Characterization

Russ Joseph¹, Zhigang Hu², Margaret Martonosi¹

¹Department of Electrical Engineering
Princeton University

²IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598



Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Wavelet Analysis for Microprocessor Design: Experiences with Wavelet-Based dI/dt Characterization

Russ Joseph
Dept. of Electrical Eng.
Princeton University
rjoseph@ee.princeton.edu

Zhigang Hu
T.J. Watson Research Center
IBM Corporation
zhigangh@us.ibm.com

Margaret Martonosi
Dept. of Electrical Eng.
Princeton University
mrm@ee.princeton.edu

Abstract

As microprocessors become increasingly complex, the techniques used to analyze and predict their behavior must become increasingly rigorous. This paper applies wavelet analysis techniques to the problem of dI/dt estimation and control in modern microprocessors. While prior work has considered Bayesian phase analysis, Markov analysis, and other techniques to characterize hardware and software behavior, we know of no prior work using wavelets for characterizing computer systems.

The dI/dt problem has been increasingly vexing in recent years, because of aggressive drops in supply voltage and increasingly large relative fluctuations in CPU current dissipation. Because the dI/dt problem has a natural frequency dependence (it is worst in the mid-frequency range of roughly 50-200MHz) it is natural to apply frequency-oriented techniques like wavelets to understand it. Our work proposes (i) an off-line wavelet-based estimation technique that can accurately predict a benchmark's likelihood of causing voltage emergencies, and (ii) an on-line wavelet-based control technique that uses key wavelet coefficients to predict and avert impending voltage emergencies. The off-line estimation technique works with roughly 0.94% error. The on-line control technique reduces false positives in dI/dt prediction, allowing voltage control to occur with less than 1% performance overhead on the SPEC benchmark suite.

1 Introduction

Wavelet analysis techniques have been used in a number of different scientific and engineering applications, ranging from image compression to climate modeling. Despite their broad use, we know of no prior applications of wavelets to microarchitectural behavior.

Wavelet Transforms, like Fourier Transforms, offer a means for summarizing a function's frequency content. While Fourier Transforms work well on periodic functions, they are not as effective, however, at providing concise views of aperiodic situations,

such as the bursty and irregular behavior often seen in computer systems. Wavelet transforms, in contrast, are designed to be able to represent how frequency content changes over time. As a result, they are well-suited to handle the burstiness and “non-stationarity” of computer systems behavior.

The research described here has two parallel goals. First, we set out to examine the utility of wavelet-based techniques in characterizing microprocessor behavior. Second, we wanted to explore accurate and hardware-efficient mechanisms for addressing the dI/dt problem. Voltage regulation and the dI/dt problem serve as the driving example for using wavelets in processor design in this research.

The dI/dt problem—exacerbated both by ongoing increases in CPU current fluctuations and by decreasing CPU supply voltages—has seen increasing attention from computer architects over the past two to three years. This attention is largely due to the increasing difficulties projected for producing cost-effective power supply and voltage regulation systems in upcoming generations of high-performance microprocessors. The dI/dt problem gets its name because it refers to the fact that changes in processor current (amperage is typically denoted as “I”) can lead to voltage fluctuations and circuit errors if the power supply network is inadequately designed.

The dI/dt problem, which refers to time fluctuations in processor current draw, has an inherent frequency dependence that makes it well-suited to wavelet analysis. In particular, the microprocessor’s power supply network can be modeled as a second-order linear system with a frequency response that shows resonance in the mid-frequency range of roughly 50-200MHz. As such, if we can characterize a processor’s current behavior *in that frequency range*, we will be able to generate a good prediction of the likelihood of voltage emergencies.

This paper presents a technique using wavelet analysis to characterize current and voltage behavior in the key frequency subbands relevant to dI/dt . We give a methodology for estimating the likelihood of voltage emergencies, and we present an on-line sensor that uses the most important wavelet coefficients and subbands to compute voltage on the fly.

Overall, the contributions of this work are as follows:

- To our knowledge, we are the first to present an application of wavelet transforms for microarchitectural analysis and design.
- We introduce wavelet analysis in the context of the dI/dt problem, and we show how wavelet representations can be used to automatically classify a program’s susceptibility to dI/dt -induced supply voltage fluctuations.
- We show how wavelet-based characterizations illustrate the interplay of architectural events and power dissipation on different time scales. The presence of cache misses and other events are germane not just to performance issues, but also to the dI/dt problem. This work represents some of the first findings on these phenomena.
- We present a wavelet-based approach for identifying voltage levels at run-time. The wavelet factorization that we propose allows for effective voltage computation with modest hardware cost during execution. Wavelet-based control reduces complexity over previous full convolution methods, while offering superior performance compared to existing pipeline control schemes.

The remainder of this paper is structured as follows. Section 2 gives an overview of wavelet analysis and transforms, and Section 3 gives the needed background on our models for power supply networks and the processor being studied. Section 4 then presents our method for *offline* estimation of voltage emergencies using wavelet analysis. Section 5 follows this with an *online* method for estimating voltage levels using a streamlined version of wavelet convolution. In Section 6, we discuss our work and relate it to other prior work, and in Section 7, we offer conclusions.

2 Wavelet Background

Wavelet analysis is a powerful method of decomposing and representing signals that has proven useful in a broad range of fields. As examples of their broad applicability, meteorologists have used wavelet analysis to study climate changes [3] and physicians have used wavelet based analysis to compress and analyze electrocardiograms [11, 7]. Wavelet based techniques have been shown to asymptotically approach the optimal solutions for important types of problems including signal de-noising and compression [6]. Despite their widespread use in science and engineering, no one has used wavelet analysis for microarchitectural studies. In this paper, we demonstrate that wavelet analysis can be effective in processor design and analysis. In particular, we demonstrate the value of wavelet analysis by applying it to characterization and control of the dI/dt problem.

Wavelet transforms are somewhat similar to Fourier transforms, in that they expose a function's frequency content. A key benefit of the wavelet analysis, however, is its ability to represent how frequency content changes with respect to time. This makes wavelets extremely useful for understanding signals which do not have constant frequency behavior. Recent architectural studies [20] have shown that real applications have complex phase behavior. As consequence, one might expect the current variability of these applications to also change with respect to time. Wavelet analysis is useful for understanding dI/dt issues in processor design for two reasons. First, the current variations that cause voltage fluctuations are sensitive to frequency characteristics which wavelets can capture. Second, these frequency characteristics are localized; they may change as a program moves through its execution and wavelets are well suited to these types of studies.

In this section, we provide a brief overview of the wavelet analysis process. A thorough discussion of the underlying mathematics are beyond the scope of this paper, but the basics presented here are sufficient to understand the dI/dt analysis techniques that we have developed, and we refer readers to other resources for additional information [6].

2.1 Discrete Wavelet Transform

The discrete wavelet transform uses two analysis functions, $\phi(t)$ and $\psi(t)$ to decompose a signal into its wavelet domain representation. These functions allow for the temporal and frequency localization properties that make wavelet analysis powerful. The *scaling function*, $\phi(t)$, is used to capture lower frequency information over long intervals. The *wavelet function*, $\psi(t)$, captures higher frequency information over typically shorter time intervals. Furthermore, there is a one-to-one relationship between $\phi(t)$ and $\psi(t)$ and they are collectively known as a wavelet basis.

For different applications, one can choose different wavelet bases. While different wavelet basis functions are suitable to different types of problems, there is no known optimal wavelet basis, and there is no way to know a priori which wavelet basis

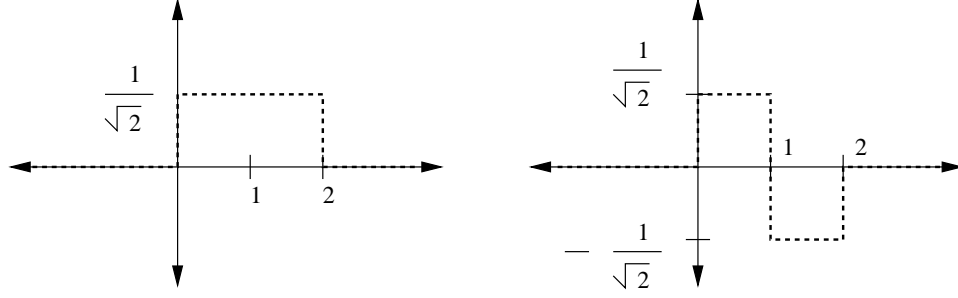


Figure 1. Haar scaling function $\sigma(t)$ (left) and Haar wavelet function $\phi(t)$ (right).

is the best match for the particular signals being studied [6]. In this work, we consider the Haar wavelet basis pictured in Figure 1. Haar wavelets are suitable for dI/dt analysis because they are useful in analyzing the sharp discontinuities that appear in microprocessor current waveforms.

The discrete wavelet transform converts a time series into a set of coefficients that form its wavelet representation just as a Fourier transform decomposes a signal into frequency components. For the dI/dt studies in this paper, a suitable signal for analysis would be a cycle by cycle current trace as measured or output by an architectural simulator. The discrete Fourier transform Equation (1) separates this signal $x(t)$ into a single series of coefficients $F[n]$ through multiplication with a complex exponential. Each one of these coefficients corresponds to a specific frequency. Together they represent the spectral structure of $x(t)$ by showing how each frequency component contributes to $x(t)$. However, notice that the Fourier components $F[n]$ are indexed by a single variable, n , which corresponds to frequency and describes the spectral behavior for the entire length of $x(t)$. In other words, the frequency decomposition provided by the Fourier transform is global. Now, consider the wavelet transform equations (2) and (3), which also decompose the original signal $x(t)$ into a sequence of coefficients. Here, some of these coefficients are indexed by two variables, which as we explain shortly, allows them to describe how frequency components change over time. In essence, both the discrete wavelet transform (DWT) and discrete Fourier transform (DFT) represent a signal in terms of coefficients, but the DFT's coefficients describe global frequency behavior whereas the DWT's coefficients describe frequency behavior in a time-localized way.

$$F[n] = \sum_{n=0}^N e^{-j\Omega n t} x(t) \quad (1)$$

$$A[k] = \int_{-\infty}^{\infty} 2^{\frac{j_0}{2}} \phi(2^{j_0} t - k) x(t) \quad (2)$$

$$D[j, k] = \int_{-\infty}^{\infty} 2^{\frac{j}{2}} \psi(2^j t - k) x(t) \quad (3)$$

The wavelet transform produces two types of coefficients. Detail coefficients, $D[j,k]$, as computed by Equation (3), isolate fine-grained characteristics. Approximation coefficients, $A[k]$ as computed by Equation (2), capture coarse-grained features. The

d[0,0]	d[0,1]	d[0,2]	d[0,3]	d[0,4]	d[0,5]	d[0,6]	d[0,7]
d[-1,0]		d[-1,1]		d[-1,2]		d[-1,3]	
d[-2,0]				d[-2,1]			
a[0]				a[1]			

Figure 2. The wavelet coefficient matrix. $a[k]$'s are approximation coefficients and $d[j,k]$'s are detail coefficients.

scaling function, $\phi(t)$, is used to compute the approximation coefficients. The approximation coefficients are indexed by a single variable, k , which correspond to time regions in the original signal $x(t)$. Each approximation coefficient can be thought of as a weighted average of $x(t)$ over a window of size determined by j_0 . The resolution factor j_0 can be selected appropriately for the signal being analyzed. Together the approximation coefficients capture low frequency information about $x(t)$.

While the scaling function and approximation coefficients isolate low frequency behavior, the wavelet function, $\psi(t)$ and the detail coefficients isolate higher frequency components. The calculation described in Equation 3 shows that the detail coefficients are indexed by two variables: j , which corresponds to frequency, and k , which corresponds to time. The j index isolates *time scales*. Increasing values of j identify fine granularity changes in $x(t)$, due to the 2^j time scaling factor. In addition, the k index isolates these frequency effects with respect to time windows which correspond to 2^j .

Together the detail and approximation coefficients capture localized frequency information about the original signal $x(t)$. Figure 2 shows one way to think about the relation between wavelet detail and approximation coefficients. First, the approximation coefficients cover large windows. Second, as the scale index j increases, more coefficients are needed because the granularity becomes finer. This allows analysis to easily be focused on a specific instant in time, an important property when dealing with bursty signals. Together, the wavelet detail and approximation coefficients can represent localized time and frequency effects.

The example in Figure 3 illustrates how the wavelet transform can decompose a signal into coefficients. The wavelet approximation and detail coefficients are computed using (2) and (3).

The discrete wavelet transform has an extremely efficient implementation: the *fast wavelet transform* has an algorithmic complexity of $O(n)$ [6]. Furthermore, wavelet representations are quite sparse. In other words, the majority of the terms in the coefficient matrices (e.g. Figure 2) are either zero or nearly zero. This is a useful property for many applications including (including ours) because it vastly reduces the number of shifts and additions needed to produce a good wavelet based estimate.

2.2 Wavelet Subbands

Wavelet subbands offer a powerful way to visualize wavelet coefficients that are often preferable to direct comparison of the coefficient matrix. Wavelet subbands are actually projections of the wavelet coefficients back into time domain signals. Equations (4) and (5) show how the subbands can be computed from the wavelet coefficients. Each time scale has its own subband signal that corresponds to frequency component of the original signal at that time scale. In terms of wavelet coefficients, a subband represents the contributions of a single row of the coefficient matrix. By adding successive wavelet subbands together, we can build approximations that eventually recreate the original signal. If we choose to ignore some subbands which aren't essential for

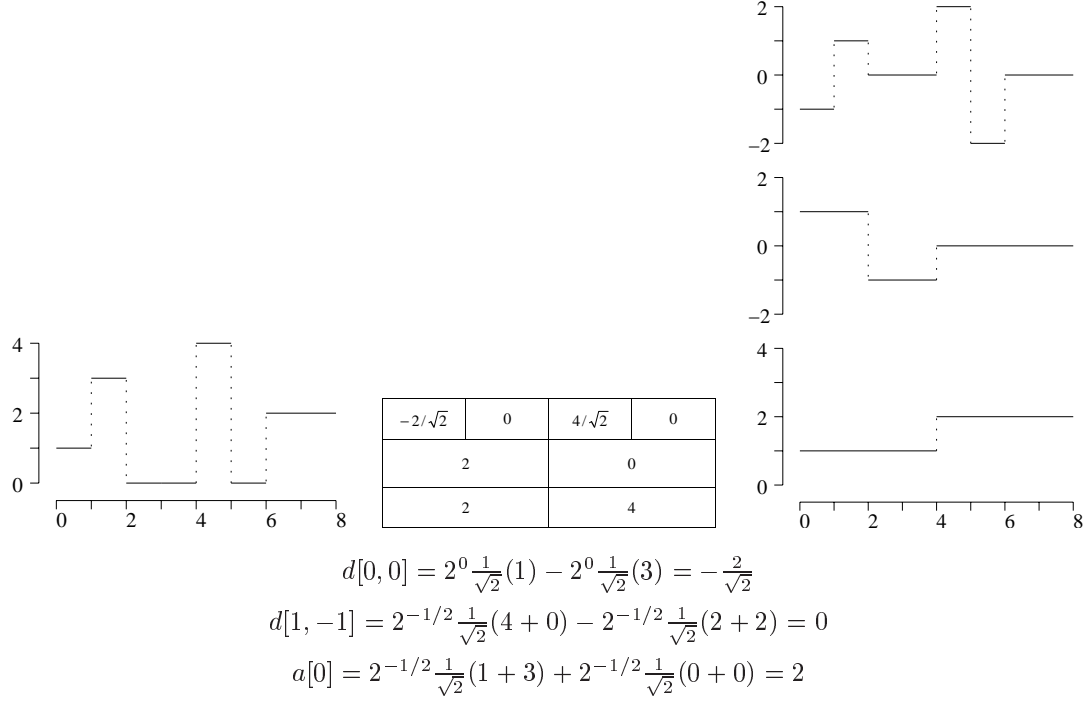


Figure 3. A Haar wavelet analysis example. Left graph is the original waveform, which can be decomposed into an approximation waveform (bottom right), plus detail waveforms on two subbands (middle and top right). The coefficient matrix is shown in the middle.

analysis, then we are effectively filtering the original signal.

$$a(t) = \sum_{k=-\infty}^{\infty} 2^{\frac{j_0 t}{2}} A[k] \phi(2^{j_0} t - k) \quad (4)$$

$$d_j(t) = \sum_{k=-\infty}^{\infty} 2^{\frac{j t}{2}} D[j, k] \psi(2^j t - k) \quad (5)$$

Linear systems properties and wavelet subbands are extremely useful for determining how changes in current directly affect processor voltage levels. For dI/dt analysis, the power supply can be modeled as a linear system in which processor current draw is an input and the resulting voltage of the power supply is an output function [10]. As a consequence, we can first separate the cycle-by-cycle current consumed by the processor, into wavelet subbands. Then we can independently compute the voltage waveform for each subband. Finally, we can add the individual voltage subbands back together with superposition to determine the total voltage behavior. The benefit of this approach is that we can independently determine what impact each wavelet time scale has on the supply voltage, rather than being forced to consider them as a whole. (The power supply network is more sensitive to some frequency ranges and less sensitive to others.) With this knowledge, we can filter out subbands that cannot make a significant impact on the voltage level, simultaneously simplifying our analysis and improving the insights it provides.

2.3 Example: Wavelet Analysis of Processor Current

We now present an illustrative example of wavelet analysis that helps to explain how one can use it to analyze dI/dt sensitivity. Figure 4 shows an application of wavelet analysis on a current waveform taken from the SPEC2000 benchmark gzip. The waveform shown at the top of the figure shows significant current variation over the window. In addition to cycle-by-cycle fluctuations, there are also some larger scale features. In general, high frequency variances do not make a significant contribution to dI/dt , but variations at moderate frequencies do have an impact. Wavelet analysis helps to identify how current fluctuations occur on different time scales. The *scalogram* pictured in Figure 4 is a powerful way to visualize wavelet coefficients and the contribution they make to current variation. Each block in the scalogram corresponds to a detail coefficient in the wavelet transform, note that we do not present approximation coefficients in this case. Large magnitude coefficients are denoted by darker values while small magnitude coefficients are represented by lighter values. The scalogram clearly shows the presence of large scale variation which was also observed in the original signal. Furthermore, the frequency composition of the signal changes with respect to time. This is just a small example to motivate the usefulness of wavelet analysis for the dI/dt problem. In Section 4, we show how wavelets can be used to make quantitative analysis for dI/dt .

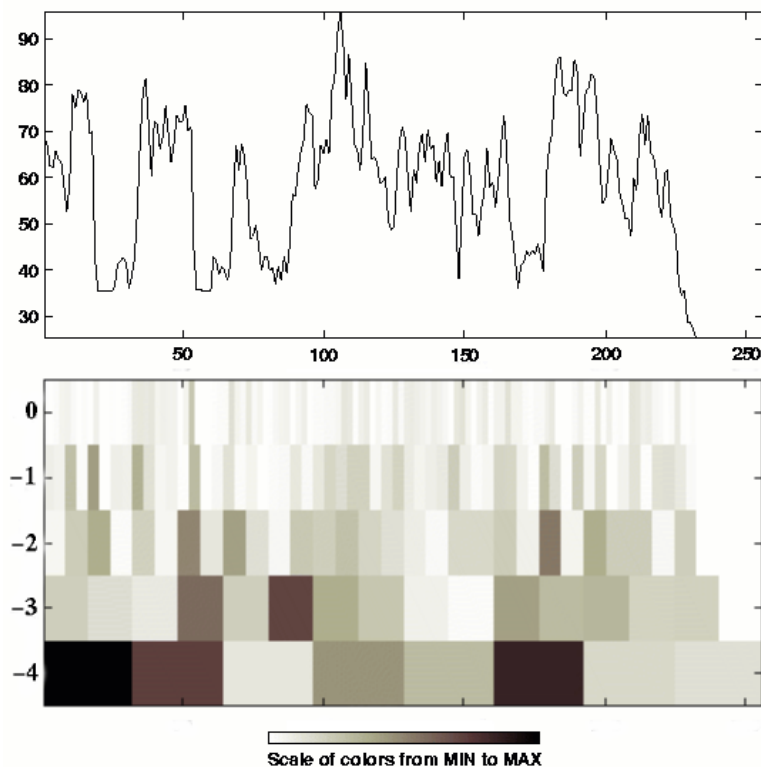


Figure 4. Current waveform (top) and scalogram (bottom) for a 256 cycle window in the SPEC2000 benchmark gzip.

3 Power Supply and Processor Models

Power supply design for high-performance processors is an extremely difficult task, and looming technology trends dictate that it will only become more taxing in terms of cost, design time, and overall complexity. The power supply network must supply the large amounts of current that high performance processors need while maintaining a stable supply voltage. It is crucial that a stable supply voltage be maintained, because circuits may encounter timing or noise-induced error if the reference levels stray outside of a +/-5% voltage range [2]. To do this, designers must limit the amount of impedance in the system. However, the more sophisticated power delivery systems which are required to achieve lower impedances are expensive and complex. This is troubling because future processors will demand even lower supply impedances [18].

Recent research has shown that microarchitectural voltage control can reduce the burden of traditional power supply design [9, 12, 16, 17]. Our research here extends on this prior work by using wavelet analysis to estimate voltages and predict voltage emergencies. In this section, we first describe models for the power supply network and microprocessor that help us to explore how program characteristics and hardware design impact current dissipation and voltage oscillations.

3.1 Power Supply Model

The power supply network has significant parasitic impedance that can produce large voltage ripples that have an adverse effect on reliability and performance. While power supply designers take great effort to limit impedance, non-negligible amounts of inductance, capacitance, and resistance remain. To reduce the impact of these parasitics on the on-chip voltage levels seen by devices, designers try to reduce the resistivity in the supply network by increasing the number of package pins devoted to Vdd/Gnd and by improving the on chip power grid [2]. To reduce the impact of the inductance, large decoupling capacitors are placed at various points throughout the power supply network [21, 10]. Nonetheless, it is increasingly difficult and costly to reduce impedance further. This is particularly true at the mid-frequency range from 50-200MHz.

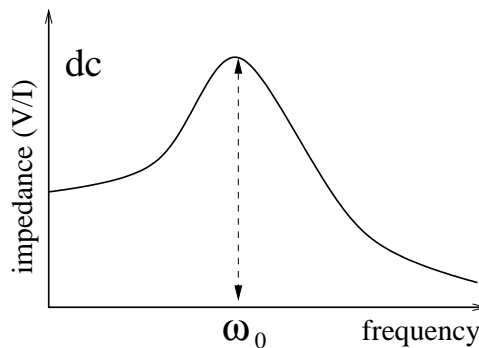


Figure 5. Frequency response of a second-order linear system, which models a typical power supply system.

For the most pressing dI/dt concerns in the mid-frequency range, a second order linear system is an appropriate model [10]. A linear model is a reasonable abstraction because the circuit elements responsible for this mid-frequency noise are all linear, i.e.

resistors, capacitors, and inductors. The only non-linear element within the supply network is the voltage regulator module, which acts on much lower frequency ranges; for the purposes of mid-frequency noise simulation, it can be modeled as a combination of linear elements [2]. Figure 5 shows the frequency response of a second-order linear system. Current fluctuations that are near the resonant frequency ω_0 are amplified and could lead to large voltage fluctuations. In essence, the second-order model captures the the power supply model’s behavior as a bandpass filter, its dominant characteristic.

In this paper, we model the power supply network as a second-order system and calculate the maximum impedance necessary to keep the voltage level within +/-5% of Vdd under a worst-case execution sequence as in [12]. We note that commercial microprocessor designers often benchmark the adequacy of their supply networks with custom crafted microbenchmarks [1], so this seemed a reasonable approach. The maximum amount of impedance that still keeps voltage ripples within +/-5% is known as target impedance [21]. Less capable power supply networks that need the help of architectural control in addition to traditional regulation are characterized by larger impedance values. For example, 150% target impedance refers to a system where the power supply network has 1.5x the standard impedance, and therefore will see voltage faults if microarchitectural control is not implemented. If microarchitectural techniques can eliminate voltage faults on a system with a 150% target impedance power supply, we say that we have reduced dI/dt by 33%.

Because we use a linear system representation, the convolution operation is used to calculate voltage levels as a function of current over time. The convolution operation (Equation 6) computes the instantaneous voltage as a function of the amperage consumed at current and previous cycles. The time shifted values of the current, $i(t)$ are weighted by the impulse response $h(t)$, which captures the complete behavior of a linear system [13]. We use a direct application of it to simulate voltage levels and a simplified version to approximate voltage in hardware. The authors of [9] first proposed convolution to simulate dI/dt noise, and we employ the same general tactic. With the use of convolution and the linear model for the power supply, we were able to compute the voltage as a function of time given an input current waveform.

$$v(t) = \sum_{i=-\infty}^{\infty} i(t-i) * h(i) \tag{6}$$

3.2 Processor Model and Benchmarks

For our processor model, we used Wattch [4], a widely used architectural power simulator based on SimpleScalar [5]. We modified Wattch to simulate a 3.0GHz processor with a nominal Vdd of 1.0V executing the Alpha 21264 architecture. Table 1 presents the parameters we used. We modified Wattch/SimpleScalar to model the performance/energy impact of deep pipelines including multiple fetch and decode stages. We also updated Wattch to spread the power usage of pipelined structures over multiple stages. To compute per-cycle current, we divided the per-cycle power from Wattch by the supply voltage. For our choice of Vdd = 1.0V, one watt of power consumed corresponds to one ampere of current drained. When the supply voltage drops, the current consumed by devices on chip actually decreases, so the active elements on chip may actually dampen the voltage ripples somewhat. However, the same assumptions are used by power supply designers in early stage planning [2], and are considered good, conservative estimates.

For evaluations, we use all 26 SPEC integer and floating-point benchmarks. To ensure that we observed representative behavior, we used simulation points presented in [20]. These simulation points were automatically chosen to capture as much of the true program behavior as possible while reducing simulation time.

Execution Core	
Clock Rate	3.0 GHz
Instruction Window	80-RUU, 40-LSQ
Functional Units	4 IntALU, 1 IntMult/IntDiv 2 FPALU, 1 FPMult/FPDiv 2 Memory Ports
Front End	
Fetch/Decode Width	4 inst, 4 inst
Branch Penalty	12 cycles
Branch Predictor	Combined: 4K Bimod Chooser 4K Bimod w/ 4K 12-bit Gshare
BTB	1K Entry, 2-way
RAS	32 Entry
Memory Hierarchy	
L1 I-Cache	64KB, 2-way, 3 cycle latency
L1 D-Cache	64KB, 2-way, 3 cycle latency
L2 I/D-Cache	2MB, 4-way, 16 cycle latency
Main Memory	250 cycle latency

Table 1. Processor Parameters

4 Wavelet Variance Characterization

In this section we propose an offline methodology that uses wavelet variance to automatically characterize a program and system’s dI/dt behavior and to estimate its impact on supply voltage levels. Similar approaches have been used in other fields to study physical phenomena such as the albedo of pack ice and ocean shear [19]. In our case, we can identify how changes in current over different time scales impact the voltage level seen by the processor. We do so by applying wavelet transforms and characterizing the resulting wavelet coefficients. Rather than merely classifying the magnitude of the dI/dt swings, we also provide a means to estimate the ultimate impact on voltage seen by the processor. This is an benefit for architects because microarchitectural dI/dt control schemes have been proposed [9, 16, 12, 17], but until now, there have been no methodologies that have offered a way to characterize dI/dt behavior (a cause of the inductive noise) and relate it to problematic supply voltages oscillations (the effect of the inductive noise). We are the first to propose an approach that directly relates the two, and we make extensive use of wavelet properties to do so.

The time scale decomposition of wavelets are useful for understanding how dI/dt activity influences voltage levels because we can separately address different frequency components of the processor’s current waveform that have dissimilar effects on voltage. For example, the power supply impedance acts as a bandpass filter which amplifies the current fluctuations near its resonant frequency while filtering some perturbations that occur at lower and higher frequencies. Wavelet time scales correspond to different frequency ranges, so by applying wavelet transforms, we can abstract the important frequency content in a straightforward and computationally inexpensive manner.

The temporal localization of wavelet analysis allows us to independently characterize different time phases of program execution and assess their individual impact on the voltage level. This is an important ability since real programs have been shown to possess complex phase behavior [20]. Furthermore, wavelets allow us to localize our analysis so that we can focus on not just the

frequency content of the processor current waveform, but how this frequency content changes with respect to architectural events such as cache misses and branch mispredictions. We demonstrate this in Section 4.3.

4.1 Relating Wavelet Variance to Voltage Variance

As Section 3 described, large amounts of current variation near the resonant frequency of the power supply system are problematic because they can adversely affect the supply voltage levels seen by the processor. This is a concern because stable voltage levels are critical for reliability and performance concerns. While it is intuitive that large changes near the resonant point are problematic, it is harder to quantify the direct impact these fluctuations have on the ultimately crucial measure—supply voltage. In one possible scenario, some current fluctuations at the resonant frequency might be reasonable, but the cumulative effect of fluctuations slightly above and below this resonant frequency could act together to push the voltage outside of a safe operating range.

Wavelet analysis can be applied together with traditional statistical measures such as mean and variance to describe how processor current draw varies with respect to both time and frequency, and more importantly how likely these current fluctuations are to cause voltage faults. The variance, σ_x^2 , is an approximate measure of the “spread” of the data points and is the preferred metric for describing how closely the data points are clustered around the mean value. For our analysis, we wish to determine how frequently and widely the voltage varies from its typical value. Large voltage variances are likely to require large amounts of dI/dt control and subsequently see significant performance degradations and energy increases. At the same time, smaller variances are less of an issue for concern since they suggest that dI/dt control is infrequently required. Note that variance does not give absolute bounds on voltage levels, but rather can be used to assess the probability that the voltage strays outside of normal, non-controlled voltage range. In our studies we first perform wavelet transforms on the current consumed by the processor, and relate statistical properties of wavelet coefficients to the corresponding voltage variance to quantify how much execution time a particular benchmark is likely to spend in controlled regions. This gives us an understanding of how difficult it would be for a given dI/dt control strategy to keep the voltage level stable while minimizing performance and energy impact.

To understand how wavelet representations might be used to characterize current consumption, we performed a series of experiments to determine if there were any significant statistical trends. Using the performance and power model described in Section 3, we executed all 26 SPEC benchmarks and examined the current variation within small window segments of 32, 64, and 128 cycles. These window sizes are long enough to offer good statistical sampling properties, but are short enough to localize behavior into the time frames relevant for dI/dt: tens to hundreds of cycles. Following established statistical procedure, we chose these windows at random intervals throughout the execution of the benchmarks. Our experiments led to two major observations:

- In a significant fraction of execution intervals, cycle-by-cycle processor amperage has a probability distribution that is approximately Gaussian.
- The remaining fraction of execution intervals have very low current variance, and therefore are less likely to be problems for dI/dt.

Figure 6 shows that for time windows relevant to dI/dt induced voltage variations, 27% to 39% of execution intervals have current distributions that can be classified as Gaussian. For this Gaussian classification, we applied the Chi-Squared Goodness of Fit test with 95% significance [14]. This is a commonly used statistical test, and its purpose is determine if a data sample comes from a particular type of distribution. In this case, we tested for a normal distribution with the same mean and variance as the sample window data. For 32 cycle windows, integer and floating point benchmarks have 27% to 30% Gaussian classification rate. As the window size increases, the acceptance rate increases to a large degree for integer benchmarks, but not as much for floating-point benchmarks. One possibility for this difference is the larger number of memory stalls in the floating-point benchmarks.

Figure 7 shows the average current variance for the remaining 61% to 73% of execution windows that were not identified as Gaussian. The average current variance for the non-Gaussian windows is quite low overall and is also much lower than the overall benchmark average. This suggests that the non-Gaussian windows contribute little to the overall current variance. Consequently, they will have less of an impact of voltage levels, so efforts are better spent obtaining good voltage estimates for the Gaussian window segments. Furthermore, when Gaussian signals (such as the cycle-by-cycle current consumption of the processor) are input to a linear system (such as the power supply network), the output is also Gaussian [8]. This gives us a way to relate current consumption to voltage levels in the important, frequently occurring cases. We focus on Gaussian window segments in the remainder of this section.

Since voltage variance on different wavelet decomposition levels often differs by orders of magnitude, we can ignore those wavelet levels that have small impact while estimating voltage variance. Figure 8 shows the error incurred when estimating voltage variance using only 4 out of the 8 total decomposition levels. Across all the benchmarks, the error is consistently small, ranging between 0.1% to 1.6%.

There are two necessary conditions for large dI/dt induced voltage swings: (1) a large variance on time scales that correspond to the resonant period and (2) pulse patterns that can build constructive interference in the power supply network. With wavelet analysis, we can easily identify large variations on different time scales and problematic current consumption patterns. To compute dI/dt induced voltage variance, we developed a statistical model that used wavelet scale variance to determine the current variation at different time scales and correlation between adjacent wavelet detail coefficients to identify pulse patterns. Specifically, we performed a series of experiments that allowed us to isolate the effects that wavelet variance and correlation had on each detail scale level. This provided us with multiplicative factors that we used to relate current variation to voltage variation. Our method has the following steps:

1. We first compute the DWT of a window segment of 256 cycles. This window length was chosen because it could capture current variations on the range of tens to hundreds of cycles that are known to be important for dI/dt .
2. The second step is to determine the variance of each wavelet scale. The intuition is that large current variances on a wavelet scale could translate to large voltage variances, especially if that time scale corresponds to the resonant frequency. The variance calculation is straightforward because of Parseval's Equation [6], which says that the variance of the wavelet subband for scale j , is equal to the sum of squared detail coefficients on that scale.

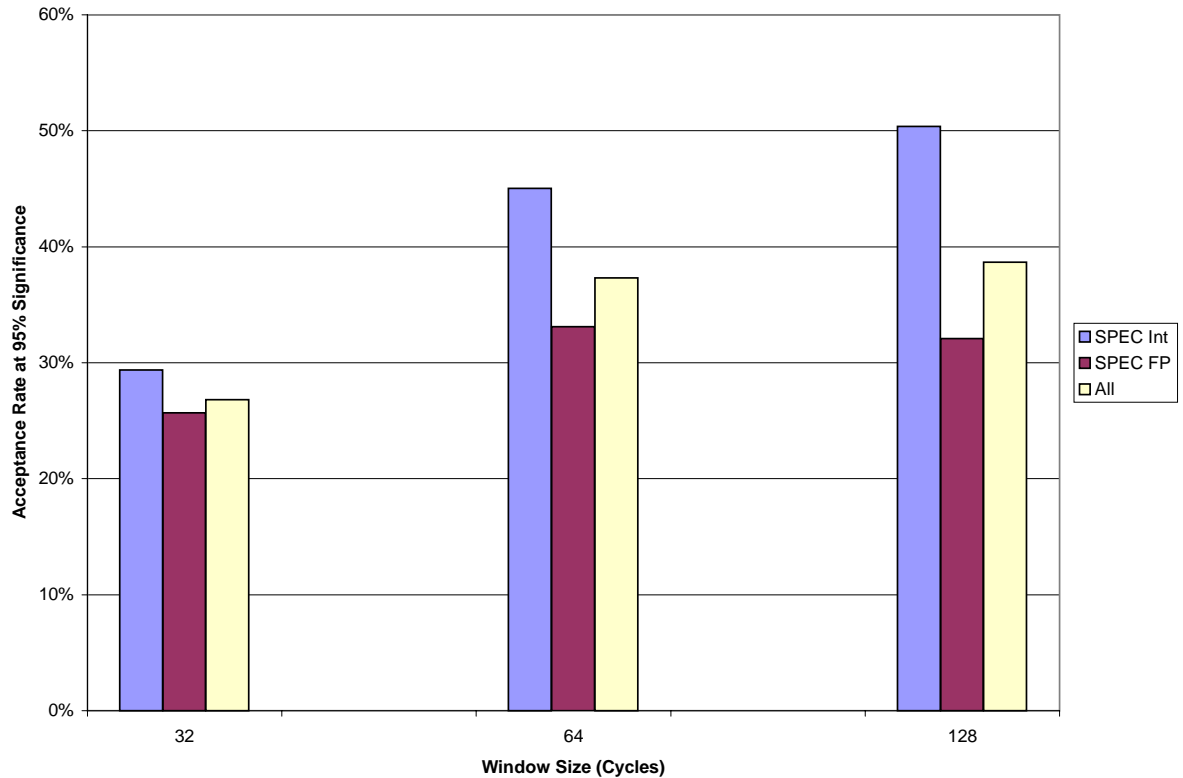


Figure 6. Acceptance rate for Chi-Sq Gaussian test at 95% significance. These graphs show the percentage of 32,64, and 128 cycle execution windows that qualify as displaying Gaussian behavior in per-cycle current dissipation.

3. Next, we compute the correlation between adjacent detail coefficients on a given scale. The correlation computation allows us to identify patterns that could be harmful for dI/dt . In essence, strong positive or negative correlations correspond to pulse signals, which could build resonance in the supply network. The model that we developed allows us to factor the adjacency correlation into our estimates of voltage variance.
4. We compute an estimate for the voltage variance on each wavelet scale using the quantities from preceding steps. Under this model the correlation between adjacent coefficients on a particular scale determines a multiplicative factor between current variance and the voltage variance contributed by that scale.
5. Finally, we applied a Gaussian model to determine the probability of observing different voltage levels. The Gaussian model takes two parameters: estimated voltage mean and estimated voltage variance. The voltage mean is just the IR drop across the power supply network, and we can estimate this by multiplying the average current over the 256 cycle window by the power supply resistance. The estimated voltage variance is the sum of the individual voltage variance contributions on the different wavelet scales, as calculated in steps 1-4 above.

The Gaussian model gives us the probability that the supply voltage seen by the processor is above or below a specific level. The control thresholds used by a microarchitectural voltage regulator would serve as interesting comparison points since they would give an indication of the frequency with which a dI/dt control scheme would be invoked on a particular benchmark.

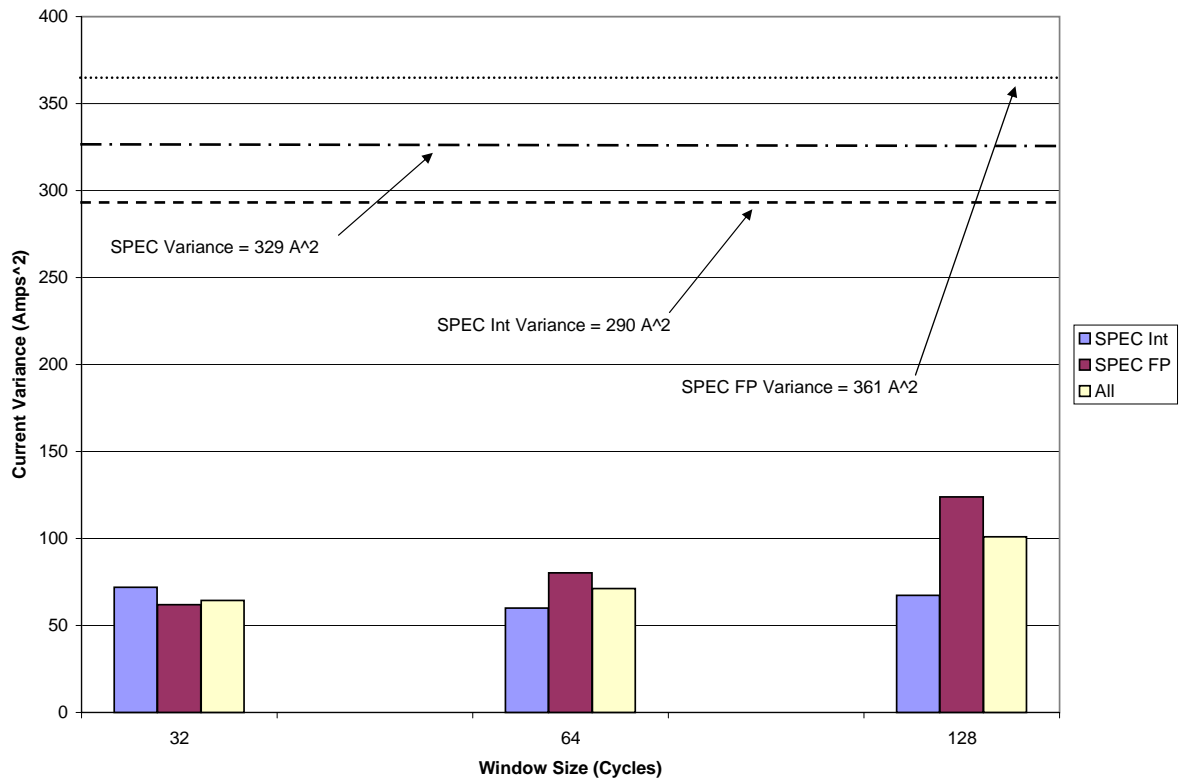


Figure 7. Mean current variance for non-Gaussian window segments. The low variance compared to the overall variance indicates that focusing on Gaussian intervals should provide a good overall estimate.

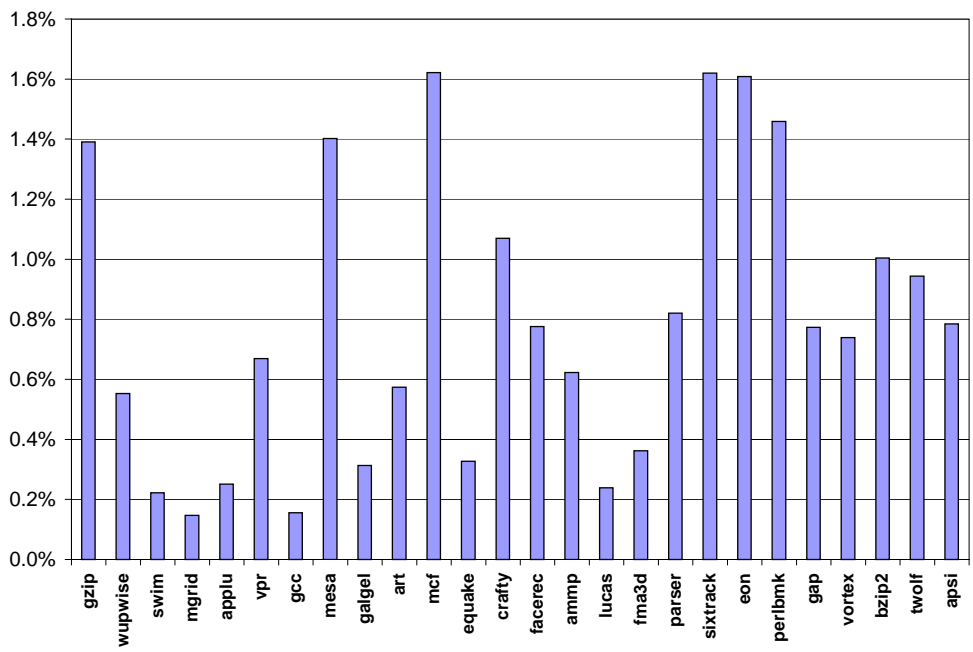


Figure 8. Error of variance estimate using only 4 decomposition levels.

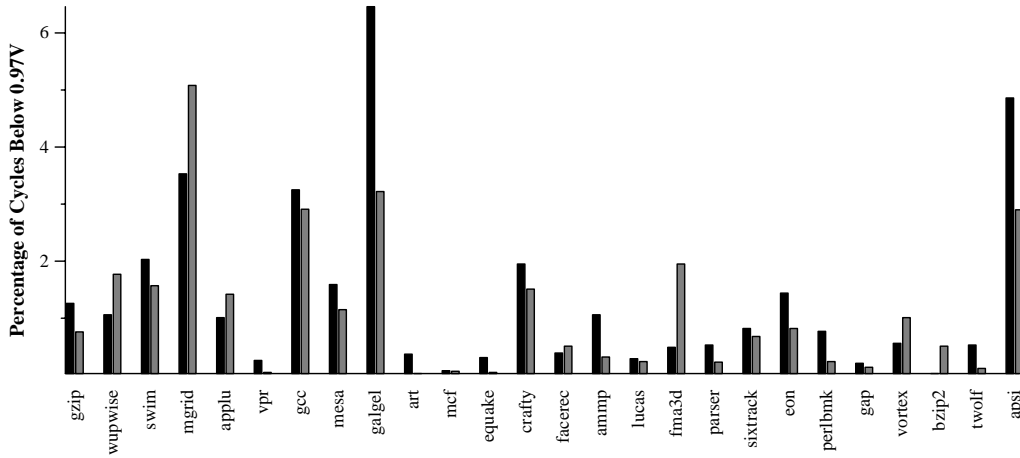


Figure 9. Estimated percent of cycles below control point compared to observed number of cycles.

4.2 Results : Voltage Characterization

Using the voltage estimation scheme outlined in the previous section, we profiled SPEC 2000 benchmarks to estimate the severity of dI/dt induced voltage variation. One of the uses of voltage profiling is to gauge the severity of voltage oscillations so that we can estimate the how often a given program will require dI/dt control. Some of the experiments that we present later in this paper suggest that voltage levels below 0.97V would need to be controlled to prevent voltage low faults. In Figure 9, we compare the percentage of execution cycles actually spent below 0.97V to the estimated percentage of cycles spent below that point using the scheme described in Section 4.1. Overall the root mean square for error is 0.94%. Figure 9 shows that while our estimates do not exactly determine the number of cycles spent below 0.97V, they do a good job at determining whether or not a benchmark might be problematic for dI/dt . For example it identifies mgrid, gcc, galgel, and apsi as benchmarks that spend at least 3% percent of their execution below 0.97V. It also identifies benchmarks such as vpr, mcf, equake, and gap which spend less than 0.5% of their execution time below this control point. Overall, wavlet voltage estimates are useful for identifying the severity of voltage variations.

4.3 Results : Relating Voltage Variation to Architectural Events

One of the interesting aspects of offline, wavelet-based estimates is that they can be used to offer insights as to the impact of different microarchitectural events on voltage levels and voltage variability. As an example of this, we characterized 26 SPEC benchmarks regarding their voltage variance and we compared it to several microarchitectural events. The clearest relationship was between L2 cache misses and voltage variance. Our variance analysis of wavelet window segments shows that low L2 cache misses correlates strongly with Gaussian voltage distributions.

In particular, Figure 10 shows voltage histograms four benchmarks (gzip, crafty, mesa, and eon) which have few L2 cache

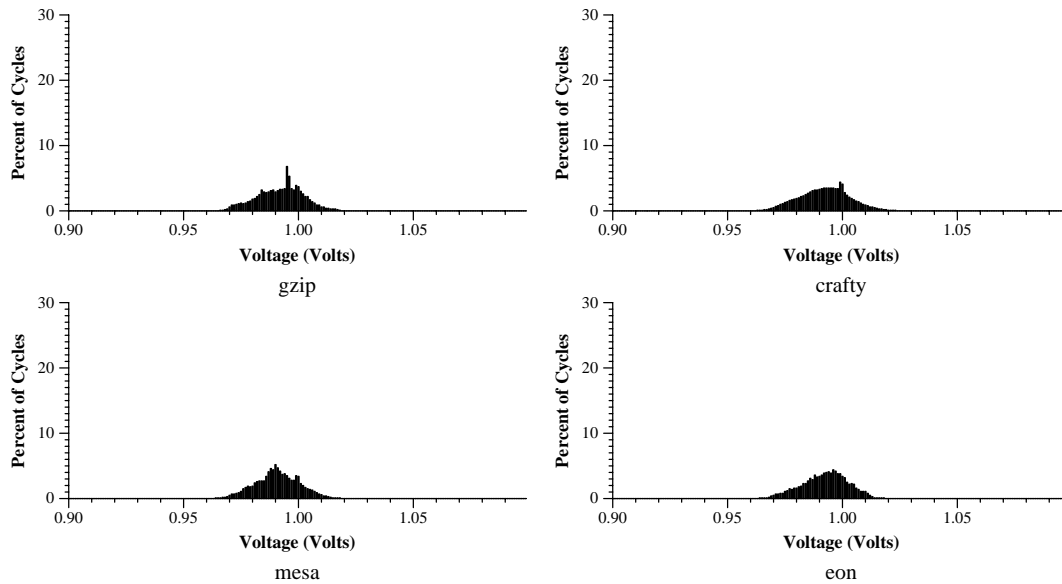


Figure 10. Histogram of cycles spent at different voltage levels for four SPEC benchmarks (gzip, mesa, crafty, eon) with few L2 misses. These voltages are distributed in an approximately Gaussian manner.

misses. Visually, one sees that the voltage profiles for these benchmarks have approximately Gaussian shapes. In contrast, Figure 11 shows the histograms for four benchmarks (swim, lucas, mcf, and art) which all have high L2 miss rates. All four benchmarks show prominent spikes at the nominal supply voltage level 1.0V, and do not exhibit a Gaussian shape.

Moving from a visual level to a statistical level, Figure 12 shows a statistical test of “gaussian-ness” applied to execution windows from the 26 SPEC benchmarks. In particular, we used a Chi-Square test at 95% significance to check for gaussian behavior in execution windows of 64-cycles in each benchmark. The benchmarks with high L2 cache misses are the least likely to show gaussian behavior in voltage. This is intuitive because these benchmarks tend to spend long periods of time waiting for L2 misses being serviced, followed by spikes of activity when the data returns. In contrast, programs with fewer cache misses have smoother execution profiles and thus are closer to gaussian in their current and voltage profiles.

5 Wavelet Based dI/dt Control

In the previous section, we demonstrated that off-line wavelet based statistical models can help to characterize dI/dt behavior and determine when problematic current fluctuations will influence voltage levels inside the processor. In this section, we focus our attention on on-line dI/dt control. In particular, we demonstrate a wavelet-based voltage monitor that can determine how close the processor is to a voltage fault by tracking current variations.

Previous work on architectural control techniques to limit inductive noise have had one of two fundamental strategies: (1) directly or indirectly monitor the *voltage level* and use the voltage level to trigger a reactive microarchitectural control mechanism [9, 12] or (2) estimate the *current* consumed by the processor by tracking microarchitectural events and maintaining an invariant on the allowable change in current over a relevant time window [17]. Under both of these approaches, normal execution operations must be suspended to avoid a voltage faults, but this may have an adverse effect on performance and energy-efficiency. For

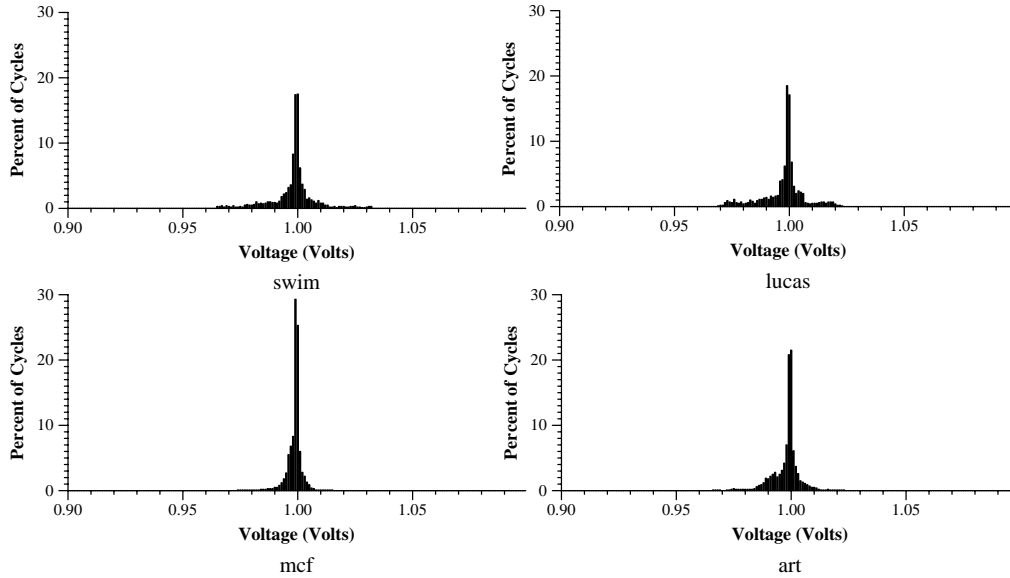


Figure 11. Histogram of cycles spent at different voltage levels for four SPEC benchmarks (swim, lucas, mcf, art) with many L2 misses. These voltages do not exhibit any Gaussian qualities.

example, if a control point is reached, both types of control mechanisms stall instruction issue to prevent the voltage from dropping below the minimum value. This decreases the current draw, so that the voltage will not sink further, but it may reduce performance since ready instructions are not being issued. Conversely, rising voltages are the result of very low current draws. In this case, both control strategies issue no-ops to increase the current consumption.

Control techniques based on voltage monitors can have relatively small performance and energy impacts, but accurate voltage monitoring can be difficult to implement. Since voltage-based monitors directly track the quantity that ultimately determines whether or not an error can occur, voltage monitoring schemes are unlikely to induce a false positive, e.g. stall instruction issue when a voltage emergency is not imminent. Due to this, control is only likely to be initiated when it is necessary, minimizing performance and energy impact. On the other hand, the complexity of previous voltage sensing proposals is high. In [12] the authors suggest using an analog circuit to sense voltage levels. While today’s chips have increasing amounts of analog circuits, the added complexity of integrating a mixed analog/digital design on die might be problematic. Another recent proposal using a convolution based voltage monitor, suffers from implementation difficulties as well. The problem with this approach is that a large number of convolution terms are needed to accurately track voltage level and such hardware is difficult to build with 1-2 cycle delays.

Control schemes that monitor current consumption are easier to build. For example, in [17], the authors propose a mechanism called *pipeline damping*, where the hardware maintains the current consumed over a sufficiently long history. They impose a restriction on the difference in current between cycles of a specified window length. By choosing a sufficiently small delta, they can bound the maximum dI/dt swing. The hardware complexity to implement this is small, but this scheme may produce a significant number of false positives.

The wavelet-based control scheme that we present here is designed to have few false positives and to have an efficient imple-

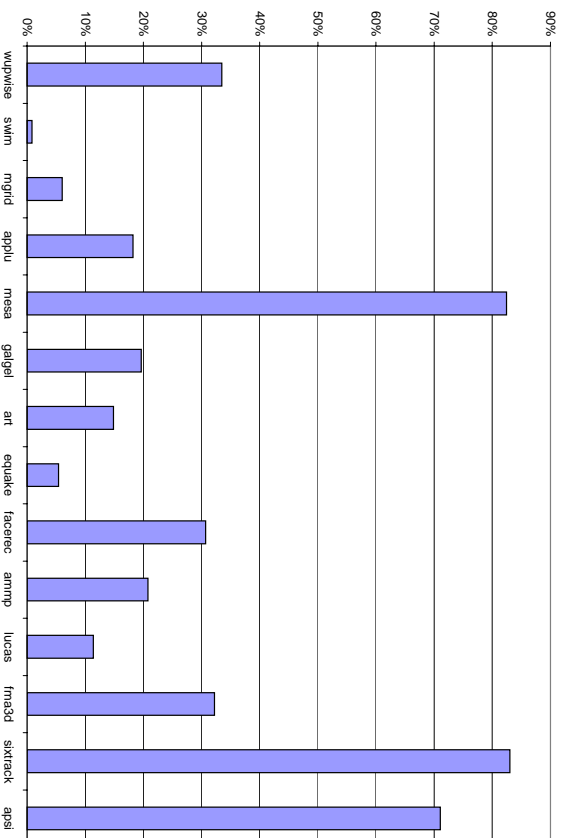
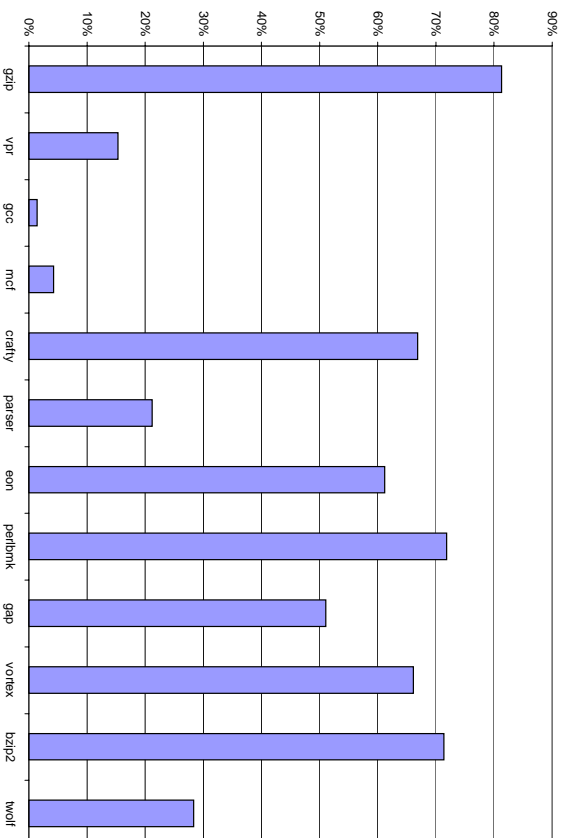


Figure 12. Percentage of 64 cycle execution windows in which the cycle-by-cycle current consumption was determined to be Gaussian under the Chi-Square 95% significant test. We present SPEC integer (top) and floating-point (bottom).

mentation. It allows the microarchitecture to efficiently track voltage levels at run-time, allowing for a dI/dt controller that avoids voltage emergencies without compromising performance or energy-efficiency. The wavelet representation significantly decreases hardware complexity so that we can achieve the higher accuracy of a voltage monitor, but with a more feasible implementation than previously proposed convolution voltage monitors [9].

5.1 Wavelet-Based Voltage Monitors

Our wavelet-based voltage monitor is more efficient because the wavelet representation that we use can reduce the number of terms that appear in the computation, providing a more efficient means to track the voltage level. Our approach is based on wavelet subband convolution [22].

Wavelet convolution provides an effective way to determine voltage levels because it can reduce the number of convolution terms. This is possible because the wavelet representation of current can sort individual cycle-by-cycle current terms into groups of coefficients that have similar impact on the voltage level. We can safely omit groups of coefficients that have little impact on the voltage level. For example, higher frequency detail subbands have little impact on the supply voltage since they lie above the resonant frequency. Since a small number of terms are most responsible for the voltage level, we can get good accuracy by calculating only with them.

To identify the coefficients which have the most impact on voltage level, we order the coefficients by decreasing magnitude. (Coefficients with large negative or positive values can have more impact on the voltage level than coefficients with values closer to zero.) Once coefficient terms are sorted in descending magnitude, we need to assess how many are needed to achieve an acceptable error rate. Larger error rates result in more conservative threshold values. Subsequently, these more conservative threshold values could lead to an increased number of false positives, and hence more performance and energy degradation. Clearly, reduced complexity favors a smaller number of wavelet coefficients and hence a smaller number of convolution terms.

To put the relationship between error levels and number of convolution terms in perspective, Figure 13 plots the maximum error possible when using an increasing number of wavelet convolution terms. We plot the error in Volts for different values of power supply impedance. (Recall that 100% target impedance is perfectly protected against voltage emergencies, while increasing percentages denote poorer voltage regulation.)

For all supply impedances, the error is very large when the coefficient count is small, and it decreases at a reasonable rate, approaching the 0.02V for coefficient counts of 9, 13, and 20 for 125%, 150%, and 200% target impedance levels. More coefficients are needed for the 200% case because the voltage fluctuations are large and therefore more difficult to summarize. Nonetheless, even 20 coefficients is small compared to the hundreds of terms present in the standard convolution equation. Our studies suggest that values of around 0.02 V (20mV) of error are small enough to still allow for protection against voltage faults with little impact on performance or energy.

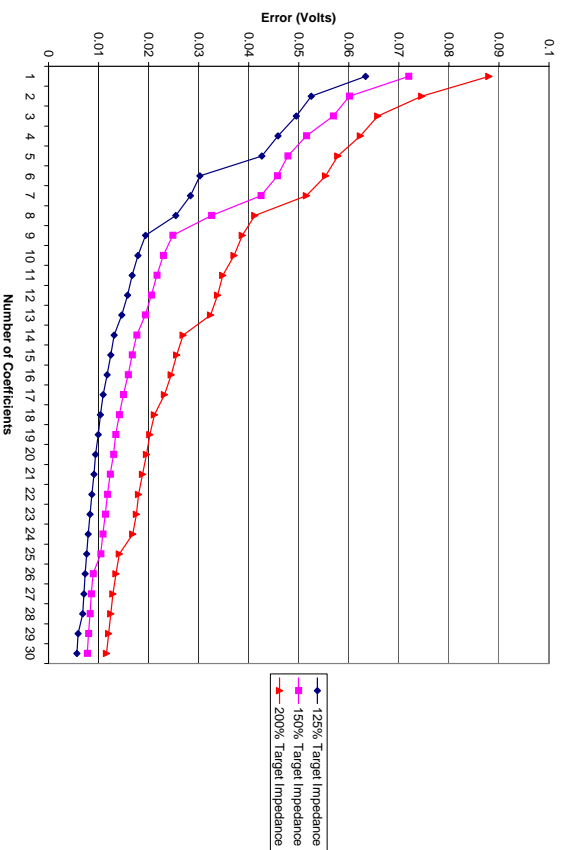


Figure 13. Maximum voltage estimation error for wavelet-based voltage monitors as the number of wavelet convolution terms is increased.

5.2 Implementation

Wavelet convolution reduces the number of coefficients that need to be tracked for accurate voltage monitoring, which makes it feasible to build a monitor in hardware. Since fewer coefficients actually contribute to the voltage computations, the number of shifts and adds needed to produce the result is significantly decreased. As Figure 13 shows, the total number of terms can be significantly reduced with tolerable amount of error. We now focus on a possible implementation strategy of a wavelet voltage monitor.

Wavelet voltage monitors are responsible for three separate tasks: (1) tracking how cycle by cycle current values interact to produce the subband coefficients, (2) determining aggregate effect of individual coefficients on the current voltage level, and (3) deciding whether action must be taken to ensure that the voltage does not reach a fault level. We briefly outline the process for implementing these three functions below.

The first task is to convert the cycle by cycle current consumption into coefficients that correspond to the subband signals. Because of their regularity, Haar wavelet subbands and their contribution to the voltage level can be computed efficiently in hardware by a series of shift registers. As Figure 1 shows, the Haar wavelet function is comprised of a pair of positive and negative pulses. As a result, we can compute the value of the subband signal by first taking the sum of cycle-by-cycle current values that intersect the first positive pulse. Next, we take the sum of the values that intersect with the negative pulse. Finally, we subtract the difference. For approximation coefficients, the process is even simpler since there is a single positive pulse in the

scaling function as shown in Figure 1. An efficient implementation does not need to sum all of the points on every cycle because we can determine exactly how the term changes from cycle to cycle. Consider the shift register implementations of detail and approximation terms in Figure 14. As new current values appear, they track the change in the point-wise sum by adding and subtracting values as they move in and out of regions of the wavelet and scaling functions.

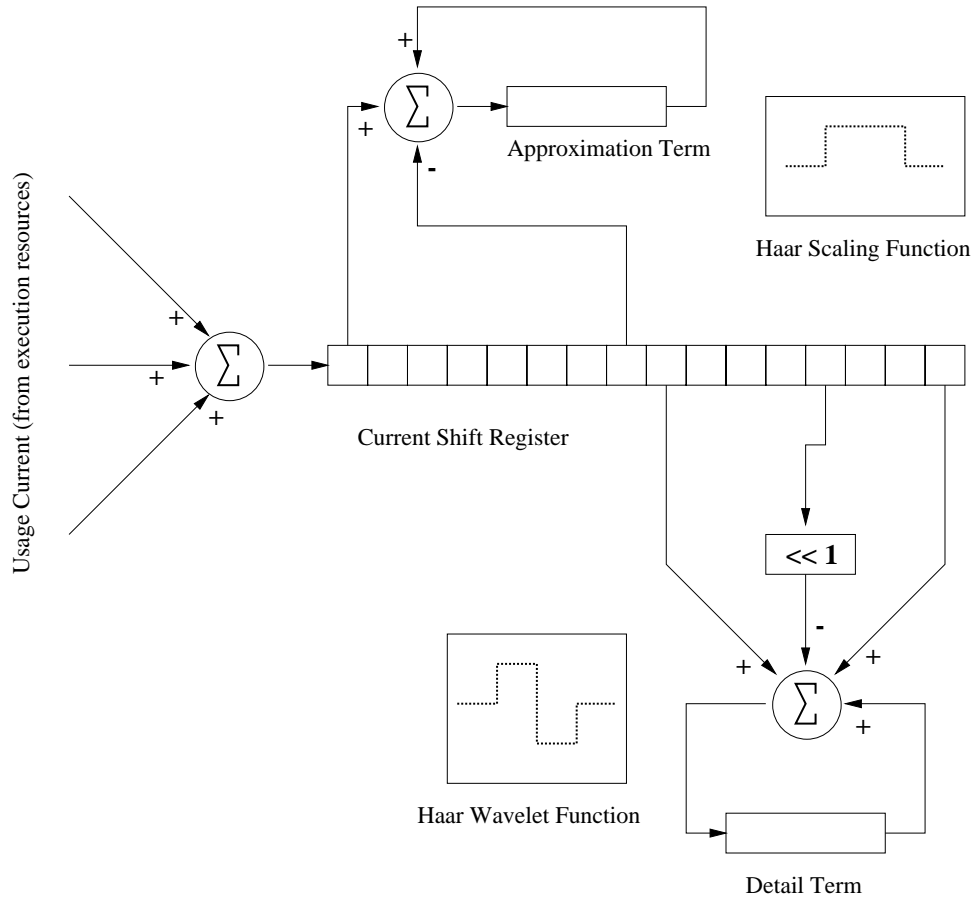


Figure 14. Shift register implementation of Haar detail and approximation term computations with two coefficients.

The second step is computing the contribution that each subband term makes on the power supply. These subband convolution products are independent operations and can be performed in parallel, accelerating computation. Furthermore, they are multiplications with a constant term, so the circuit implementation would most likely be optimized into shifts. The final step in the calculation is to perform a column addition on the terms. This step can take advantage of prior hardware proposals for low-latency column addition [15].

Once the voltage has been approximated, a comparator determines whether or not it has exceeded either the high or low control point. If the approximated voltage is below the low control point, instruction issue is stalled. This reduces the power so that a voltage fault can be avoided, but may have an adverse effect on performance. When the voltage exceeds the high control point, no-ops are issued to functional units to increase the current consumption and prevent a voltage high fault.

5.3 Results

To gauge the relationship between the number of wavelet convolution terms needed and the net impact that a wavelet based controller would have on performance and power, we performed a series of experiments where we varied the voltage control points. For our power supply model, we used 150% target impedance because this corresponds to the 33% dI/dt reduction described in [17]. As Figure 13 illustrates, an increasing number of convolution terms increases the overall accuracy. Improving accuracy allows threshold settings which are likely to initiate control only when needed.

Figure 15 shows the slowdown on SPEC benchmarks for different control threshold tolerances. The tolerances indicate where the voltage control point is with respect to the actual voltage fault point. For example, a threshold setting of 10mV means that the voltage low control point is reached at 0.96 V, exactly 0.01 V (10mV) above the minimal allowed voltage of 0.95 V. Likewise the voltage high control point would appear at 1.04 V, 10mV below the maximum allowed voltage of 1.05V. Figure 15 shows that for optimistic threshold settings, such as 10mV, the performance impact of wavelet-based control is almost negligible. The mean slowdown is around 0.01%. As threshold settings become more conservative and performance degradation increases, but the maximum slowdown is around 2%. This compares favorable to the maximum value of 22% given in [17].

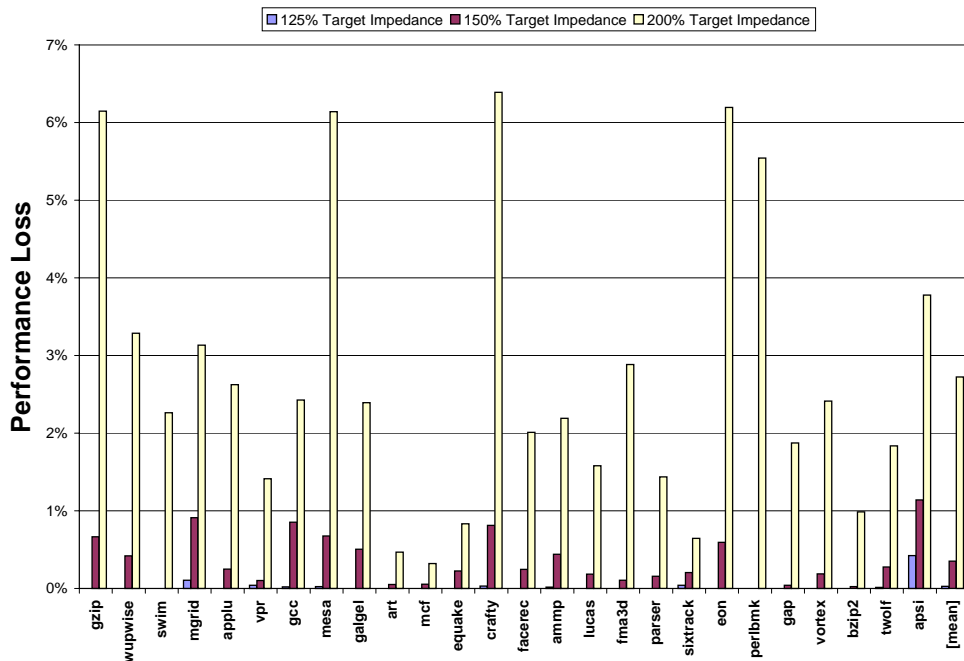


Figure 15. Performance loss under dI/dt control as a function of control threshold settings.

6 Discussion and Related Work

The topic of dI/dt control began to see attention at the microarchitectural level only in the past two to three years. Essentially there are two main parts to any microarchitectural dI/dt controller. The first part is the sensing mechanism used to determine when trouble is imminent. The second part is the actuation or control mechanism used to take action in order to keep the system's

voltage under control. Table 2 presents a summary of how different proposals (including the one described here) compare on different issues.

While the name “dI/dt problem” refers to current fluctuations, it is ultimately the *voltage* fluctuations induced by current changes that are problematic in high-performance microprocessors. Thus, in building a sensing mechanism for dI/dt, one can choose between sensing current, sensing voltage, or sensing some proxy of the two and doing estimation calculations. Current or voltage sensors can be built as analog devices. Current sensors are more readily buildable, while supply voltage sensors are more difficult due to the fact that they are trying to measure V_{dd} itself, though all other on-chip logic typically treats V_{dd} as the bedrock reference value on the chip.

Some prior work has looked at estimation-based proxies for current and voltage. In particular, pipeline damping [17] proposes using current estimation over time windows to determine whether to engage voltage control. While this method is relatively simple to implement, it has the potential for high false-positive rates. High false-positive rates mean that voltage control mechanisms must be engaged more frequently, which leads to potentially large performance and energy impact as well. (Their paper mentions performance slowdowns as large as 22% for SPEC benchmarks, which are not significant dI/dt stressors.) The convolution-based methodology proposed by Grochowski et al. [9] has the potential to be more accurate in its cycle-by-cycle voltage estimates and thus have a lower false positive rate. On the other hand, it is difficult to build a single-cycle implementation of the convolution circuit they propose. Our work offers the low false-positive rate of an accurate sensing circuit, with an easier implementation than full-blown convolution hardware.

	Analog Voltage Sensing Circuit Joseph et al. HPCA9	Full Convolution Grochowski et al. HPCA8	Pipeline Damping Powell et al. ISCA03	Wavelet Convolution This proposal
False Positive Rate	low	low/medium	potentially large	low/medium
Performance/Energy Impact	low	low	potentially large	1-6.5%
Implementation Complexity	Requires analog circuit	single cycle difficult	modest modest	between delta and convolution
Control Stability	good if delay small	multicycle may be unstable	sensitive to current estimates	good with sufficient coefficients
Sensor/Controller Delay	low	high	modest	between delta and convolution

Table 2. Qualitative comparison of microarchitectural dI/dt proposals, including the wavelet-based method presented here.

7 Summary

Wavelet analysis is a powerful method of decomposing and representing signals in both the frequency and time domains. Compared to traditional Fourier analysis, wavelet analysis has the following advantages:

- Wavelet analysis can analyze signals that contain discontinuities and sharp spikes.
- Wavelet analysis can analyze non-stationary signals whose frequency behavior varies with time.
- Wavelet coefficient matrices are typically sparse. Most coefficients are zero or near zero, so that a small group of coefficients

can represent a signal fairly well.

- Wavelet analysis is computationally efficient. A fast wavelet transform can be done in $O(N)$ time.

While wavelet analysis has been widely applied in science and engineering, no published work has shown its application in the computer architecture field. In this paper, we propose the application of wavelet analysis in microprocessor design, and specifically we show how to use wavelets to characterize and estimate voltage variation on chip.

We start with an introduction of basic concepts of wavelet analysis, and use an example to show how a waveform can be decomposed into approximation and detail waveforms. We describe how a signal can be divided into subbands, each representing the frequency component of the original signal at a particular time scale. We then briefly explain the power supply system model used in this paper, and show how voltage can be calculated through convolution between current and impulse response of the power supply system.

Our first application of wavelet analysis is to characterize the voltage variance of a particular program workload. Voltage variance is a measure of how cycle-by-cycle voltage values spread around the nominal voltage. Large voltage variances are undesirable because they are indicative of dI/dt problems which lead to reliability issues. To calculate voltage variance we first determine the variance of each wavelet subband. Since the subbands closest to the resonant frequency have the greatest impact on voltage variance, omitting other subbands only has negligible impact on the accuracy of our wavelet voltage variance calculation. This not only reduces the computational complexity, but also offers insights into how program behavior can affect voltage variance.

Our second application of wavelet analysis is a wavelet-based voltage monitor that is more computationally efficient than a full convolution. This is possible because only a few wavelet coefficients are needed to achieve a reasonable accuracy. Online dI/dt control based on a wavelet-based voltage monitor can eliminate voltage emergencies while limiting performance loss to a few percent. Because of the regularity of the Haar wavelet, the coefficients can be computed efficiently using a few shift registers and constant adders.

In summary, this paper represents a first attempt to apply wavelet analysis to the computer architecture field. Because of its power to represent bursty signals and sharp spikes, as well as its computational efficiency, wavelet analysis can be a powerful aid to computer architects in understanding and analyzing complex program and microprocessor behavior.

References

- [1] P. J. Bannon. Personal communication, 2002.
- [2] D. Blaauw, R. Panda, and R. Chaudhry. Design and analysis of power distribution networks. In A. Chandrakasan, W. J. Bowhill, and F. Fox, editors, *Design of High-Performance Microprocessor Circuits*, pages 499–522. IEEE Press, 2001.
- [3] K. M. Bolton, E.W. and J. M. Lilly. A wavelet analysis of plio-pleistocene climate indicators: A new view of periodicity evolution. June 1995.

- [4] 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.
- [5] D. Burger, T. M. Austin, and S. Bennett. Evaluating future microprocessors: the SimpleScalar tool set. Tech. Report TR-1308, Univ. of Wisconsin-Madison Computer Sciences Dept., July 1996.
- [6] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms : A Primer*. 1998.
- [7] C. C. Gamo, P. Gaydecki, A. Zaidi, and A. Fitzpatrick. An implementation of the wavelet transform for ecg analysis. In *First IEEE Conference on Advances in Medical Signal and Information Processing*, September 2000.
- [8] A. L. Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley, 1994.
- [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. *IEEE Transactions on Advanced Packaging*, 22(3):240–248, August 1999.
- [11] M. Hilton. Wavelet and wavelet packet compression of electrocardiograms. *IEEE Transactions on Biomedical Engineering*, 44(5), May 1997.
- [12] R. Joseph, D. Brooks, and M. Martonosi. Control techniques to eliminate voltage emergencies in high performance processors. In *Proc. of the 9th International Symposium on High Performance Computer Architecture (HPCA-9)*, February 2003.
- [13] T. Kailath. *Linear Systems*. Prentice-Hall, 1980.
- [14] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and' Sons, 8th edition, 1999.
- [15] Z. Luo and M. Martonosi. *Using Delayed Addition Techniques to Accelerate Integer and Floating Point Arithmetic on FPGAs*, volume 3526. November 1998.
- [16] 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.
- [17] M. D. Powell and T. N. Vijaykumar. Pipeline damping: A microarchitectural technique to reduce inductive noise in supply noise, June 2003.
- [18] Semiconductor Industry Association. International Technology Roadmap for Semiconductors, 2001. <http://public.itrs.net/Files/2001ITRS/Home.htm>.

- [19] Serroukh, A. T. Walden, and D. B. Percival. Statistical properties and uses of the wavelet variance estimator for the scale analysis of time series. *Journal of the American Statistical Association*, 95(450), June 2000.
- [20] T. Sherwood, E. Perelman, G. Hamerly, and B. Calder. Automatically characterizing large scale program behavior. In *Proc. Tenth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, Oct 2002.
- [21] 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.
- [22] P. P. Vaidyanathan. Orthonormal and biorthonormal filter banks as convolvers, and convolutional coding gain. *IEEE Transactions on Signal Processing*, 41(6), June 1993.