Thermal-aware Voltage Droop Compensation for Multi-core Architectures

Jia Zhao, Basab Datta, Wayne Burleson and Russell Tessier

Department of Electrical and Computer Engineering
University of Massachusetts
Amherst, MA

{ jiazhao, bdatta, burleson, tessier }@ecs.umass.edu

ABSTRACT

As the rated performance of microprocessors increases, voltage droop emergencies become a significant problem. In this paper, two new techniques to combat voltage droop emergencies are explored. First, a direct connection between temperature and processor clock frequency modulation during voltage droops is established. In general, a higher temperature leads to a lower voltage droop with the same processor activity. Thus, processor frequencies can be reduced less at high temperature in an effort to prevent voltage emergencies. Through experimentation, the benefits of temperature-flexible frequency scaling are explored. Second, processor signatures consisting of performance statistics are used to identify when voltage droop compensation is needed in a multicore environment. The use of an independent on-chip interconnect network allows for the sharing of signatures across cores at run time. Signature sharing in combination with frequency throttling is shown to provide an improvement in average run-time performance in a number of cases for an eight-core multiprocessor.

Categories and Subject Descriptors

C.1.3 [**Processor Architectures**]: Other Architecture Styles.

General Terms

Design, Reliability.

Keywords

Voltage emergency, thermal monitor, monitor network-on-chip

1. INTRODUCTION

As CMOS feature sizes shrink and device integration density increases, on-chip power supply droop caused by large dI/dt becomes an increasing concern. These large current swings coupled with power delivery subsystems that have non-zero impedance cause the supply voltage to fluctuate beyond safe operation margins. These fluctuations eventually lead to incorrect processor computation, necessitating checkpoint-rollback mechanisms to restore correct processor state [1]. Since the design of a low-

impedance power-supply system across a broad range of frequencies is a challenging task, different voltage droop compensation methods have been proposed [2][3][4]. Current techniques to address dI/dt droop can be classified as either design-time or run-time compensation solutions [5]. Design-time solutions are pessimistic and require detailed modeling of the power-supply network [6]. In contrast, dynamic compensation methods can adapt to run-time supply voltage fluctuation [2][3][4].

The most common voltage droop compensation methods include processor frequency scaling in response to low voltage determined by voltage sensors or real-time combinations of processor event metrics. Generally, sensor-based voltage droop compensation methods employ on-chip voltage comparators [7] and analog-to-digital converters [8] for droop detection. Whenever a sensor determines that a local supply voltage has dropped below a predefined threshold, frequency throttling or supply voltage boosting is invoked [2][9]. The combined use of processor event metrics generally provides an early predictor of an impending voltage droop [4].

Previous work has provided a direct link between voltage droop and processor switching activity [2][3]. However, the relationship between these metrics and temperature has not been explored, an important issue since modern processors operate over an extended temperature range from 5°C up to 75°C [10]. Our simulations using a 130nm process indicate that the ambient temperature of a core significantly influences the voltage droop. At elevated temperatures, the same switching activity and processor event metrics draw a reduced instantaneous current, reducing the resulting droop. Experiments show voltage droop decreases by approximately 7.5mV per 20°C increase in a 130nm technology circuit with an operating voltage of 1.5V.

The derived relationship between voltage droop and temperature can be used to adapt the multicore frequencies used for voltage droop compensation. In a series of experiments, adaptive frequency throttling is performed in response to voltage droop signatures calculated locally or collected from adjacent cores. Inter-core signature distribution is performed via a dedicated on-chip monitor network-on-chip (MNoC) [11]. Our experiments show that signature sharing reduces required processor rollbacks, leading to performance improvement. The use of temperature-aware frequency throttling when emergencies

are detected further improves system performance versus a simple, uniform frequency selection.

The rest of this paper is organized as follows. Section 2 provides background on voltage droop signatures and remediation. In Section 3, the experimental relationship between temperature and voltage droop is explored. In Section 4, a technique to determine an appropriate processor frequency in case of detected voltage droop is described. The use of voltage droop signatures in multicores is explained in Section 5. Experimental results from our techniques are described in Section 6. Section 7 concludes the paper.

2. BACKGROUND

Processor voltage droop is primarily due to current drawn from the device power supply [2][3] in response to processor activity. A power supply network can be modeled as a second-order linear system [3] with current pulses of different durations used to simulate current drawn from the processor. In general, this work indicates that a single current burst does not lead to a voltage droop, but it does affect the supply voltage for a short period of time. As a result, a sequence of bursts can lead to a large voltage droop if the bursts overlap in time.

Perhaps the most straightforward way to identify voltage droops is to use a voltage monitor to determine when supply voltage drops below a predetermined hard threshold [7][8]. Whenever this voltage drops below the threshold, it is assumed with high probability that processing errors have occurred, requiring a rollback to a previous checkpoint [1]. Since supply voltage can change rapidly and voltage compensation methods often require at least several clock cycles, the use of a voltage monitor to trigger compensation using a hard threshold can be of reduced effectiveness. As a result, thresholds are sometimes set higher than the minimum (a soft threshold) [2][9] to avoid rollbacks at the expense of more frequent compensations.

Signature-based voltage droop predictors are a suitable alternative approach to voltage sensors in determining impending voltage droops [4]. This approach captures sequences of processor control-flow and micro-architectural events as signatures of a voltage emergency. Signatures can be generated from a combination of control flow instructions, cache miss and TLB miss information, and pipeline flush information [4]. When a voltage emergency happens for the first time, the information is captured as a signature. An event history table records the relevant processor events at run time. Comparisons to stored information are subsequently performed during application execution. The use of this information allows for an early prediction of voltage emergencies so that compensation can be performed well in advance of an emergency.

3. TEMPERATURE EFECTS ON VOLTAGE DROOP

Ambient core temperature significantly impacts the current drawing capability of logic and consequently, the peak voltage droop. In deep sub-micron technologies, gate overdrive causes the mobility effect to dominate, leading to a reduction in drawn peak current with a rise in temperature. To assess the thermal influence on voltage droop, a power-delivery model was used in experimentation that is similar to the one used in [12]. The model includes a lumped RL model for the on-die power-grid and RLC constants that match the measured off-chip impedance of the Pentium 4 processor [12].
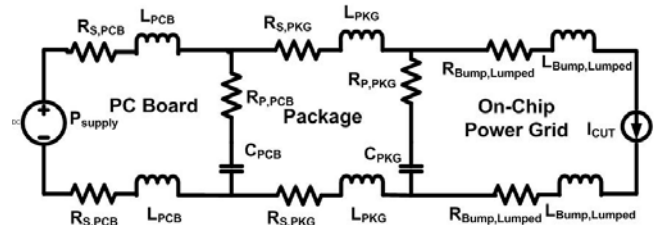

Fig. 1: Power Delivery Model

A 130nm PTM technology model and HSPICE was used in a series of simulations. A previously-used approach [3] was employed to model the effect of extreme instantaneous switching currents on voltage droop. The voltage droop in the power delivery model is triggered by a current-pulse I_{CUT}, as shown in Fig. 1. The width of the pulse is 20 ns [3]. To mimic the processor current behavior for varying ambient temperature, two 15-stage FO5 ring-oscillator circuits with gates 500 times larger than minimum size and which operate at 2 GHz were used. The circuit was first simulated at several different temperatures (Fig. 2) and the peak current values were recorded. These current values were then used to set the amplitudes of the current-pulse I_{CUT} and measure the corresponding maximum undershoots of the on-die power-supply.
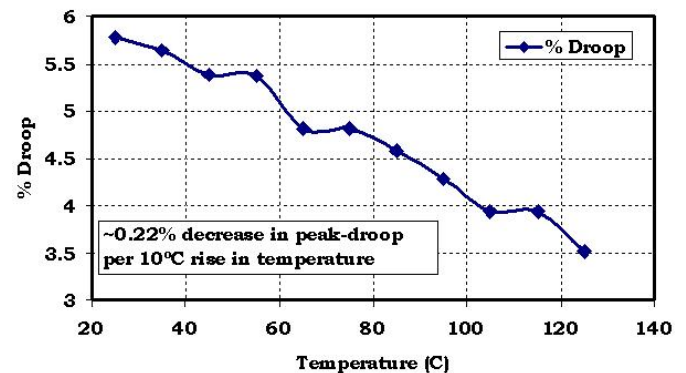

Fig. 2: Temperature impact on voltage droop (%) in 130nm

Fig. 2 illustrates the percentage of voltage droop with respect to a 1.5V supply voltage observed at different temperatures. In general, high current peaks observed at lower temperatures lead to larger voltage droops versus those experienced at higher temperatures. As shown in Fig. 2, a 20°C temperature rise reduces the voltage droop by around 0.5% with respect to the supply voltage. This value is significant since in contemporary droop-management algorithms [4][13] voltage droops greater than 4% of the supply voltage are assumed with high probability to cause processing errors and require system rollbacks.

The effect of process variations on the thermal characteristics of voltage droop was also studied by assuming a Gaussian distribution for the effective channel-length (L_{eff}) ($3\sigma=15\%$) in the previously-described ring-oscillator circuit. L_{eff} is chosen to model process variations as it is known to have significant deviation in the deep sub-micron nodes. One hundred Monte Carlo simulations were performed using the ring-oscillator circuit. A collection of supply voltage droops were determined for four temperatures over a random set of effective channel lengths.
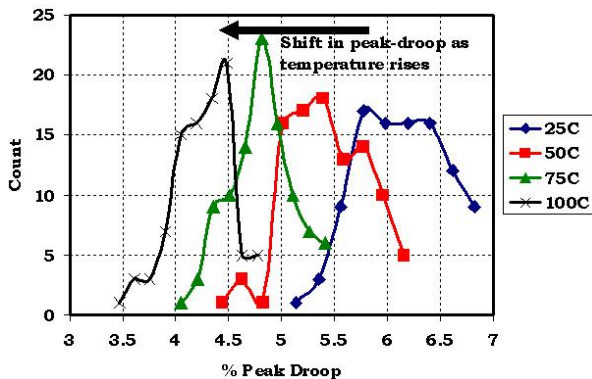

Fig 3: Histogram of voltage droop (%) experienced with $3\sigma=15\%$ variation in L_{eff} at four temperatures (obtained over 100 Monte-Carlo runs). The peak droop decreases with a rise in temperature.

Fig. 3 shows the results of this simulation. The x-axis in the figure is the voltage droop in the power supply network caused by the ring-oscillator circuit. The y-axis indicates the number of cases (out of 100) that cause a given voltage droop at a specific temperature. Fig. 3 indicates that regardless of the process-corner occupied by the core, a higher temperature significantly reduces the average droop experienced by the power supply network. As seen in Fig. 3, the peak voltage droop for the maximum number of circuits at a given temperature decreases as the temperature increases. In general, current generation multicore processors experience a wide thermal operating range and the possibility of anisotropic heat flow. These factors provide sufficient motivation to perform a reevaluation of traditional droop compensation techniques in multicore designs with non-uniform die temperature distribution.

4. THERMAL-AWARE ADAPTIVE VOLTAGE DROOP COMPENSATION METHOD

Frequency throttling is perhaps the most frequently used remediation technique [3] in the face of power supply voltage droops. This approach lowers the number of current bursts within a fixed time span. The more a processor core operating frequency is reduced from its standard operating frequency, the less likely a power supply voltage droop is to occur.

Based on the temperature influence explained in Section 3, an adaptive frequency throttling approach to avoid voltage droop is presented. When the temperature is higher, the frequency is reduced less, providing an overall performance gain while avoiding a voltage emergency. Since modern processors support rapid, fine-grained frequency changes [14], a range of target frequencies can be supported. The target low frequency for each temperature can be determined early in processor execution during a tuning phase. The frequency range information is then stored in a table indexed by temperature range. When frequency throttling is required, the instantaneous temperature measured by thermal monitors is used as an index into the table to select the proper frequency.

The throttling frequency tuning phase requires a series of steps, including the use of an on-chip voltage monitor and processor rollback mechanisms [1]. Initially, a processor is operated at its standard operating frequency F_0. A series of additional lower frequencies $F_1 > \ldots > F_{n-1} > F_n$ are identified. During the tuning phase, the standard frequency is initially reduced by one step to F_1 for the measured temperature. If an emergency occurs, processor rollback is performed and the next lower frequency is used during the next emergency. Eventually, a stable frequency which avoids rollbacks is determined and stored in the frequency table. In many cases, the table values can be predetermined once at processor fabrication and preloaded to avoid the need for tuning every time the processor is used.

5. MULTICORE VOLTAGE COMPENSATION USING SIGNATURE PREDICTION

Recently, it has been determined [4] that the use of voltage signatures to determine impending voltage emergencies yields better system performance than the use of information from voltage sensors. In the signature-based voltage droop detection method, a shift register is used to store the event history of control flow instructions, cache and TLB misses, and pipeline flushes. The first time a voltage emergency is detected, the information in the event history table is captured to form a signature which is stored in a compressed signature table [4]. Subsequent event histories are compared against the signature to predict upcoming emergencies. Once an impending emergency is detected, the target processor frequency is located in the frequency table. A voltage droop sensor is required for each core to locate initial emergencies. Fig. 4 illustrates the thermal-aware signature based voltage compensation module in each core of a multicore system. A frequency table and thermal monitors are included in this structure as explained in Section 4.

A new contribution of this work is the use of signatures in a *multicore* processing environment, a key limitation of [4]. Here, we consider signature sharing across processors which execute some common basic code blocks. This situation exists when a program supports multicore execution, as seen in the next section. By using a signature originally detected in an alternative block, a processor is relieved from undergoing a sensor-based voltage emergency to detect the signature. In Section 6, the results

of this sharing are quantified. In general, signatures are shared as soon as they are identified.
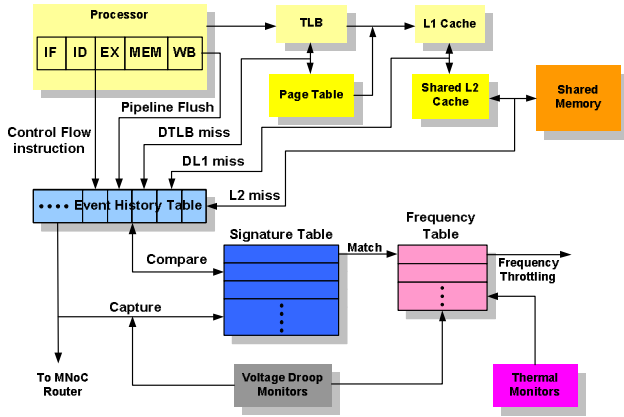

Fig. 4: Thermal-aware signature based voltage compensation module in each core. Adapted from [4]
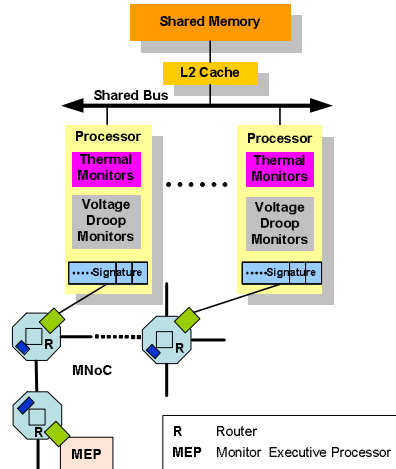

Fig. 5: MNoC used for thermal-aware voltage compensation

Although signature information could be shared using the main multi-core interconnect, for this work we assume that a dedicated on-chip interconnect for monitor and other low-bandwidth data is used. This monitor network-on-chip (MNoC) [11] has previously been shown to provide low latency transport for monitor information while consuming a minimal amount of on-chip resources. As shown in Fig. 5, a signature captured in a core is sent to an MNoC router. A central control module, the monitor executive processor (MEP), collects signature packets from the cores and synchronizes signature tables across the cores.

6. EXPERIMENTAL RESULTS

A modified SESC multiprocessor architectural simulator is used to evaluate performance for our system. The multiprocessor setup is shown in Table 1. The processor power model used by SESC is based on Wattch [15]. The cache power model is based on CACTI and the processor architecture is modeled after an Alpha21264 with a MIPS ISA. The per-core voltage droop is calculated from the convolution of the cycle-accurate power consumption with the impulse response to the power supply network [16]. The power delivery model is based on the Alpha21264/21364 [3]. SPLASH 2 benchmarks are used

for the evaluation. Voltage supply fluctuations of greater than 4% are viewed as voltage emergencies [4]. The processor events that are monitored and compared against signatures include control flow instructions captured in the issue stage, cache and TLB misses.

Table 1. Experimental Setup

Simulator	SESC multiprocessor simulator
Technology	90 nm
Num of processors	8, 16
Frequency throttling	f_{normal}=2GHz, f_1=1.3GHz, f_2=1GHz
Benchmarks	SPLASH 2
Processor configuration	
I-cache	64KB, 4-way
D-cache	64KB, 8-way, 2 cycles
Branch Predictor	Hybrid
Branch Target Buffer	4K entries, 16-way
Instruction Queue	16 entries
Retirement Order Buffer	176 entries
Load/Store Buffers	56/56 entries
L2 Cache	1MB, 8-way, 10 cycles

Table 2. Signature summary (per-core)

Test bench	Case	Sign num.	Sign num. reduction (%)	Local sign match (%)	Global sign match (%)
Ocean	Local only	15,989	21.1	100	0
	Global sharing	12,612		23	77
LU	Local only	55	0	100	0
	Global sharing	55		92	8
Fmm	Local only	11,696	16.6	100	0
	Global sharing	9,755		21	79
Radix	Local only	34,218	5.8	100	0
	Global sharing	32,230		52	48

In an initial set of experiments, an 8-core system was simulated using SESC. Like [4], each core includes a signature-based compensation module and frequency table, as shown in Fig. 4. Each signature includes a collection of 32 events. Signatures are stored in a 16K entry event history table for every core. Both signatures generated locally (within the core) and globally (generated by another core and transferred via MNoC) can be stored in the table. Since signatures are tied to specific instruction sequences, the same code must be executed by multiple cores in the case of a global signature match. The MNoC interface follows the setup shown in Fig. 5.

Table 2 illustrates the fraction of local versus global signature matches seen on average by every core for four SPLASH 2 benchmarks. For comparison, the total number of signatures generated per-core for the global sharing

approach is compared to the total when only local signatures are used. The amount of sharing varies widely across benchmarks based on the amount of per-core code sharing. Table 2 also illustrates the average per-core reduction in signature generation on a per-benchmark basis. For the Ocean benchmark, the total number of generated signatures is reduced by more than 20% since signatures are shared across cores. However, the LU benchmark has little code sharing across cores so signature sharing provides little benefit.

Since initial signature detection requires a system rollback, a penalty of 100 to 1000 cycles is paid [4]. Fewer required signature detections leads to improved performance because required rollbacks are reduced. An intermediate value of a 500 cycle rollback penalty was used in our experimentation along with a frequency throttling period of 10 cycles [1]. The performance penalty to change the system frequency is assumed to be trivial [17] and is not included in the experiment. To evaluate the benefit of global signature sharing, the performance of four SPLASH 2 benchmarks is considered.

Table 3. Performance comparison versus global prediction

Test bench	Execution time (ms)		Execution time reduction (%)
Ocean	Sensor based	32.98	-24.62
	Local prediction only	24.86	-2.82
	Global prediction	24.16	
LU	Sensor based	24.96	-31.21
	Local prediction only	17.17	2.33
	Global prediction	17.57	
Fmm	Sensor based	24.92	-24.76
	Local prediction only	19.01	-1.37
	Global prediction	18.75	
Radix	Sensor based	26.01	-27.10
	Local prediction only	18.96	-2.43
	Global prediction	18.50	

Table 3 illustrates the relative performance of applications executed with global sharing versus the local-only signature and sensor-only approaches for voltage droop remediation. Table 3 indicates that the local, signature-based predication allows for about a 30% improvement in performance versus a sensor-only approach since rollbacks are required less frequently. This result is consistent with previously reported results [4]. Global sharing provides a benefit for three benchmarks (Ocean, Fmm and Radix) and a drawback for one benchmark (LU). The LU benchmark uses very few signatures and in general they are best isolated in the core which generated them. The reduced performance for global sharing is a result of false droop detection in some cases which offsets the reduced initial detection of signatures. Thus, the use of global signature sharing does not reduce the total number of generated signatures for this benchmark. In general, however, additional global sharing leads to a larger benefit.

The measured average per-core injection rate of signatures into MNoC for an 8-core system is less than 1 per 100 clock cycles for all four benchmarks. Experiments with a modified Popnet interconnect simulator for MNoC verified a low network latency of less than 15 cycles for all inter-core signature transfers [18].

Table 4. Adaptive frequency cases

Temperature range	Frequency table cases				
	1	2	3	4	5
20-40°C	1GHz	1GHz	1GHz	1GHz	1.3GHz
40-60°C	1GHz	1GHz	1GHz	1.3GHz	1.3GHz
60-80°C	1GHz	1GHz	1.3GHz	1.3GHz	1.3GHz
80-100°C	1GHz	1.3GHz	1.3GHz	1.3GHz	1.3GHz

Table 5. Performance results for multicore systems

Test bench	Case	Performance (ms)	Performance benefit (%)	Performance (ms)	Performance benefit (%)
		8 core system		16 core system	
Ocean	Case 1	24.16		23.95	
	Case 2	22.68	6.13	22.70	5.22
	Case 3	21.62	10.05	22.55	5.85
	Case 4	21.62	10.05	21.73	9.27
	Case 5	21.59	10.06	21.65	9.60
LU	Case 1	17.56		18.75	
	Case 2	15.90	9.45	16.47	12.16
	Case 3	15.32	12.76	15.71	16.21
	Case 4	15.32	12.76	15.68	16.37
	Case 5	15.12	13.89	15.68	16.37
Fmm	Case 1	18.75		14.37	
	Case 2	17.93	4.37	13.98	2.71
	Case 3	17.93	4.37	13.98	2.71
	Case 4	17.85	4.80	13.98	2.71
	Case 5	17.80	5.07	13.98	2.71
Radix	Case 1	18.50		13.00	
	Case 2	17.80	3.78	12.26	5.69
	Case 3	17.80	3.78	12.26	5.69
	Case 4	17.80	3.78	12.26	5.69
	Case 5	17.77	3.95	12.26	5.69

In a second experiment, the effect of temperature on multicore voltage droop remediation is considered. The same experimental setup in terms of signature-based voltage compensation modules and MNoC interfaces as the first experiment is used. In this experiment four temperature ranges are considered, as shown in Table 4. Five separate cases are considered for frequency scaling based on temperature. In the first case, processor frequency is reduced from 2 GHz to 1 GHz if an impending voltage droop is detected, regardless of temperature. In the fifth case, the frequency is reduced from 2 GHz to 1.3 GHz. For cases 2, 3, and 4 a mix of frequencies is considered based

on the temperature. These middle cases best illustrate the benefit of temperature-aware frequency toggling.

In this second experiment, system performance for 8 and 16 cores under the five cases are explored. The instantaneous temperature is not generated by Hotspot but, rather, is controlled by user input to fall within a range of 20°C to 100°C.

As shown in Table 5, temperature-aware adaptive frequency throttling leads to improved performance versus cases in which frequency is always reduced by 50% (case 1). The total instructions executed by the 8 and 16 core systems are up to 100 million and 200 million, respectively. For Ocean, Fmm and Radix benchmarks, the performance benefits are less than 11%, while for the LU benchmark, performance benefits of greater than 15% are seen. Implementations of the first three benchmarks are nearly evenly distributed across the cores, thus activity on all cores is roughly equivalent. For the LU benchmark, one core executes more instructions than other cores, making it more susceptible to voltage droop. Since frequency changes are more likely in this core, the adaptive frequency approach provides a greater benefit.

7. CONCLUSIONS

In this paper, the influence of processor temperature and shared performance signatures on supply voltage droop is explored. Experiments show that with the same processor activity, a higher temperature leads to a lower voltage droop. Based on this result, a thermal-aware adaptive frequency throttling method is proposed. This method is combined with signature-based voltage emergency prediction methods to combat voltage droop emergencies. Experiments also show that signature sharing provides benefits for three out of four benchmarks and thermal-aware frequency throttling provides significant benefits (>5%) for all tested benchmarks.

8. ACKNOWLEGEMENTS

REFERENCES

[1] M. Gupta, K. Rangan, M. Smith, G. Wei, and D. Brooks. "DeCoR: A Delayed Commit and Rollback Mechanism for Handling Inductive Noise in Microprocessors," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA-14)*, pp. 381-392, 2008.

[2] E. Grochowski, D. Ayers, and V. Tiwari, "Microarchitectural Simulation and Control of di/dt-induced Power Supply Voltage Variation," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA-8)*, pp. 7-16, 2002.

[3] R. Joseph, D. Brooks, and M. Martonosi. "Control Techniques to Eliminate Voltage Emergencies in High-Performance Processors," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA-9)*, pp. 79-90, 2003.

[4] V. Reddi, M. Gupta, G. Holloway, M. Smith, G. Wei, and D. Brooks. "Voltage Emergency Prediction: A Signature-Based Approach To Reducing Voltage Emergencies," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA-15)*, pp. 18-27, 2009.

[5] M. Holtz, S. Narsimhan and S. Bhuniya, "On-die CMOS Voltage Droop Detection and Dynamic Compensation," in *Proc. Great Lakes Symposium on VLSI (GLSVLSI'08)*, pp. 35-40, 2008

[6] H. Su, S. Sapatnekar and S. Nasif, "An Algorithm for Optimal Decoupling Capacitor Sizing and Placement," in *Proc. International Symposium on Physical Design (ISPD'02)*, pp. 68 -73, 2002

[7] T. Nakura, M. Ikeda and K. Asada, "Preliminary Experiments for Power Supply Noise Reduction using Stubs," in *Proc. Asia-Pacific Conference on Advanced System Integrated Circuits (AP-ASIC'04)*, pp. 286-289, 2004.

[8] E. Alon, V. Stojanovic and M. Horowitz, "Circuits and Techniques for High Resolution Measurement of on-chip Power Supply Noise," in *Digest of Technical Papers of Symposium on VLSI Circuits*, pp. 102-105, 2004

[9] E. Alon and M. Horowitz, "Integrated Regulation for Energy-efficient Digital Circuits," in *Journal of Solid-State Circuits (JSSC)*, pp. 1795-1807, 2007.

[10] "Intel Pentium 4 Processor in the 478-pin Package at 1.40 GHz, 1.50 GHz, 1.60 GHz, 1.70 GHz, 1.80 GHz, 1.90 GHz, and 2 GHz Datasheet", http://download.intel.com/design/10/datashts/24988703 .pdf

[11] S. Madduri, R. Vadlamani, W. Burleson and R. Tessier, "A Monitor Interconnect and Support Subsystem for Multicore Processors", in *Proc. Design Automation and Test in Europe Conference (DATE'09)*, pp. 761-766, 2009.

[12] M. Gupta, J. Oatley, R. Joseph, G. Wei, and D. Brooks. "Understanding Voltage Variations in Chip Multiprocessors Using a Distributed Power-Delivery Network," in *Proc. Design, Automation and Test in Europe Conference (DATE'07)*, pp. 1-6, 2007.

[13] M. Gupta, V. Reddi, G. Holloway, G. Wei and D. Brooks, "An Event-Guided Approach to Handling Inductive Noise in Processors," in *Proc. Design, Automation and Test in Europe Conference (DATE'09)*, pp. 160-165, 2009.

[14] S. Herbert and D. Marculescu, "Variation-Aware Dynamic Voltage/ Frequency Scaling," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA-15)*, pp. 301-312, 2009.

[15] D. Brooks, V. Tiwari and M. Martonosi, "Wattch: A Framework for Architectural-level Power Analysis and Optimizations," in *Proc. International Symposium Computer Architecture (ISCA'00)*, pp. 83-94, 2000.

[16] K. Hazelwood and D. Brooks. "Eliminating Voltage Emergencies via Microarchitectural Voltage Control Feedback and Dynamic Optimization," in *Proc. International Symposium on Low-Power Electronics and Design*, pp. 326-331, 2004.

[17] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. International Symposium on High-Performance Computer Architecture (HPCA'01)*, pp. 171-182, 2001

[18] R. Vadlamani, J. Zhao, W. Burleson and R. Tessier, "Multicore Soft Error Rate Stabilization Using Adaptive Dual Modular Redundancy", in *Proc. Design Automation and Test in Europe Conference (DATE'10)*, 2010.