The Thesis Committee for Ankita Garg

Certifies that this is the approved version of the following thesis:

Characterization of Voltage Noise in Big, Small and Single-ISA Heterogeneous Systems

APPROVED BY

SUPERVISING COMMITTEE:

Lizy Kurian John, Supervisor

Vijay Janapa Reddi, Co-supervisor

Characterization of Voltage Noise in Big, Small and Single-ISA Heterogeneous Systems

by

Ankita Garg, B.E.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2013

Dedicated to my husband and parents

For their endless love, support and encouragement

Acknowledgments

I would like to express my sincere thanks to Dr. Lizy John and Dr. Vijay Reddi for their guidance and excellent support throughout the duration of my masters.

I wish to thank Jingwen Leng for the immense help in understanding various aspects of power and voltage modeling. I am also thankful to Young-taek Kim for all the technical discussions and feedback on the simulation infrastructure and experiments.

Finally, I would like to thank my husband Deepak Goel, for being extremely supportive throughout my graduation.

Characterization of Voltage Noise in Big, Small and Single-ISA Heterogeneous Systems

Ankita Garg, M.S.Comp.Sci.
The University of Texas at Austin, 2013

Supervisor: Lizy Kurian John

Sensitivity of the microprocessor to voltage fluctuations is becoming a major concern with growing emphasis on designing power-efficient microprocessors. Voltage fluctuations that exceed a certain threshold cause "emergencies" that can lead to timing errors in the processor, thus risking reliability. To guarantee correctness under such conditions, large voltage guardbands are employed, at the cost of reduced performance and wastage of power. Trends in microprocessor technology indicate that worst-case operating voltage margins are not sustainable. Since voltage emergencies occur only infrequently, resilient architectures with aggressive guardbands are needed. However, to enable the exploration of the design space of resilient processors, it is important to have a deep understanding of the characteristics of voltage noise in different system configurations.

Prior research in this area has mostly focused on systems with very few cores. Given the increasing relevance of large multi-core systems, this thesis

presents a detailed characterization of voltage noise on chip multi-processors, consisting of large number of cores. The data indicates that while the worst case voltage droop increases with increase in the number of cores, the frequency of occurrence of the droops is not greatly impacted, emphasizing the feasibility of employing resilient microarchitectures with aggressive voltage margins.

The thesis also presents a comparative study of voltage noise in CMPs consisting of either high-performant out-of-order cores and power-efficient in-order cores. The study highlights that the out-of-order cores experience much larger voltage variations when compared to the in-order cores, but offer a clear advantage in terms of performance. Experiments indicate that in-order configurations that offer equivalent performance to the out-of-order cores result in large energy-delay product, indicating the trade-offs involved in designing for performance, power and reliability.

The thesis also presents a study of voltage noise in single-ISA heterogeneous configurations, to highlight the benefits of such systems towards lowering the worst-case voltage margins, which improve both performance and power. The experimental results indicate that the worst-case voltage droop in such heterogeneous systems lies in between the out-of-order and in-order cores and provide reasonable power-efficiency and performance. Further, the work highlights the importance of exploring the design-space of heterogeneous systems considering reliability as an important design criteria.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Microprocessor design is increasingly constrained by power and performance efficiency [41]. Mobile devices use small in-order cores to achieve better power efficiency, while the high performance servers that use the big out-of-order cores, employ techniques like clock gating and dynamic voltage and frequency scaling, to reduce power consumption. These techniques lead to rapid changes in the supply current over a small amount of time. Due to the parasitic inductance in the power delivery subsystem, these changes in the current can cause large variations in the supply voltage, typically referred to as ***voltage noise***. Significant variations in the supply voltage, called a ***voltage emergency*** [46], can lead to timing errors and operational failures in the microprocessor. Since guaranteeing reliable operation is a fundamental requirement of a microprocessor, chip designers have used voltage margins or guardbands to compensate these voltage variations, to prevent such failures.

Voltage margins are typically around 20% [29] of the nominal voltage on many modern systems. Voltage fluctuations must be contained within this allowed margin. While a voltage margin provides reliability in operation of the chip, the margin has to be designed carefully as it has implications on both

performance and power. The voltage margin affects the peak operating clock frequency, with larger margins lowering the clock frequency as shown in [46]. This reduces performance and also leads to wastage of power. Furthermore, [46] also showed that with technology scaling, the maximum peak-to-peak voltage swing will become double in 16nm when compared to 45nm, which will require even larger margins.

Many techniques have been proposed in the literature, both in hardware and software, for reducing the worst case voltage drop, allowing narrower guardbands. A significant number of these techniques have been proposed for resilient architectures [34] [20] [24] [23] [42] [12] [13], that have error recovery circuitry to recover from transient faults and timing violations, to reduce the occurrences of large voltage variations, to reduce the impact of added reliability on performance. Thus, with and without resilient processor designs, reliability will be an important microprocessor design criteria.

Efforts to improve the energy efficiency of high performance processors has led to another class of multi-cores, called heterogeneous multi-cores. Heterogeneous systems may consist of different-ISA cores, like CPU and GPU, e.g, Intel's Sandy Bridge [2], AMD's Fusion [1], NVidia's Tegra [4], or single-ISA multi-core system design proposed by Rakesh et al [35], adopted by commercial products like ARM's big.LITTLE [5] comprising of big out-of-order cores and small in-order cores. [10][16][18] [21][32] illustrates that such heterogeneous multi-cores can provide energy efficiency when workloads are appropriately scheduled on the most suitable cores. Thus, heterogeneous system architec-

ture is a promising design, especially for the power constrained mobile devices. Given the importance of reliability, it is necessary to understand how voltage noise impacts the design of heterogeneous systems.

In this thesis, I present a detailed characterization of voltage fluctuations in a multi-core system with large number of cores and present a comparative study of voltage fluctuations in the high performant out-of-order and small, energy-efficient in-order cores. The study also characterizes voltage noise in single-ISA heterogeneous systems. The results indicate that while the big cores yield better performance, they require extremely large voltage guardbands, compared to the smaller, in-order cores, leading to reduced power-efficiency. The study also indicates that the heterogeneous systems not only provide energy-efficiency, they allow reduced voltage margins, thus effectively increasing performance.

The main contributions of this thesis are:

1. pvSim, a simulation framework that can be used to study power and voltage variations in multi-core systems at a cycle-level granularity.

2. The study characterizes voltage variations in larger multi-core systems and shows that the magnitude of the worst-case voltage droop increases as the number of cores increase.

3. The thesis presents a comparative analysis of voltage noise in high-performant out-of-order cores and the power-efficient in-order cores. Ex-

perimental results indicate that the out-of-order cores suffer larger voltage variations when compared to the in-order cores, requiring an order magnitude larger voltage guardbands.

4. The study compares voltage fluctuations on equivalent-TDP configurations consisting of big, small and a combination of big and small cores, to demonstrate that single-ISA heterogeneous systems achieve better energy-efficiency and reduced worst case voltage fluctuations.

Chapter 2

Background

This chapter provides a brief description of voltage noise in microprocessors and modeling of power delivery networks, as it will enable better understanding of the topics discussed in the thesis.

2.1 Voltage Noise

The power delivery network (PDN) of a microprocessor is responsible for providing steady voltage to the processor. It consists of inductive and resistive elements on the motherboard (MB), package, and die. The rate of change of current in the circuit, either due to clock gating or workload activity, leads to a proportional variation in the supply voltage, which comprises of the drop across the parasitic resistance (IR drop) and the inductive components (commonly referred to as inductive noise or ***di/dt*** in the literature). Low-level circuit techniques use a hierarchy of on-die, on-chip and off-chip decoupling capacitors (decap) to reduce the effects of inductive noise, by reducing the overall impedance. However, power supplies with such decoupling capacitors still suffer from high impedance as it is difficult to completely nullify the impact of inductive noise in the network. Current variations at the frequency of such

peak impedance result in supply voltage variations.

Several resonant frequencies like, the first, second and third droop are possible in the circuit, depending on the interaction of the inductances and capacitances at different levels. For instance, as highlighted in [31], the prominent resonant frequency is the first droop, which occurs due to the interaction of package and on-die inductance with on-die decap. While the second and third resonant frequencies lead to supply voltage fluctuations, the magnitude of such fluctuations are smaller when compared to those induced by the first droop. The first droop for many PDNs is typically in the range of 50-200MHz. Repeating pattern of low to high current variations at the frequency of the first droop of the PDN, results in large voltage fluctuations, potentially causing timing faults and affecting long-term reliability of the system. To prevent such side-effects, processors are usually designed to tolerate such worst-case voltage swings, by employing operating voltage margins that are about 20% of the nominal supply voltage.

2.2 Modeling Power Delivering Network

In a microprocessor, the supply voltage is provided by a PDN. A PDN can be modeled in a number of ways, taking into account the components present on the die, on-chip and off-chip in the circuitry. A simple model that can be used to model the PDN is shown in Figure 2.1, as used in [43]. It is a second-order resistive, inductive and capacitive (parallel RLC) circuit, with the power supply considered as a voltage source. The processor that consumes

current based on the workload activity is represented as a current source. L models the inductance of the connections between the die and the chip and C is the on-die decoupling capacitance. This simple model does not consider the on-chip and off-chip decap capacitors and thus does not show variations at all possible frequencies in the PDN. However, this second-order model does capture resonant behavior and suffices for the purpose of the study.

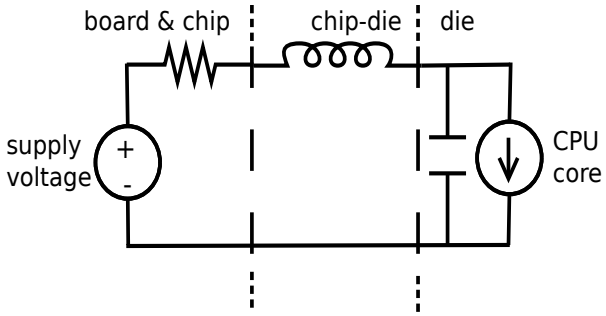


Figure 2.1: Second-order power delivery network model [43]

The values of the components R,L and C can be determined based on the characteristics of the microprocessor, like the clock frequency and the technology. The parameters that can be used to define a PDN are the resonant frequency f, resistance R and quality factor Q. Resonant frequency of a second-order circuit is the frequency at which the current variations in the processor lead to maximum fluctuations in the supply voltage and is computed as :

$$f = \frac{1}{2\Pi\sqrt{LC}}$$

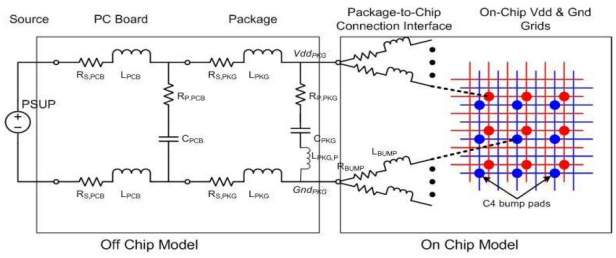
Typical resonant frequency ranges from 50-200MHz in modern microprocessor PDNs. The quality factor of a PDN determines the range of fre-

quencies at which the PDN circuit resonates with more than half the energy than at resonant frequency. Gupta et al [24] provides some analysis on how the different values of R and Q impact the characteristics of a PDN.

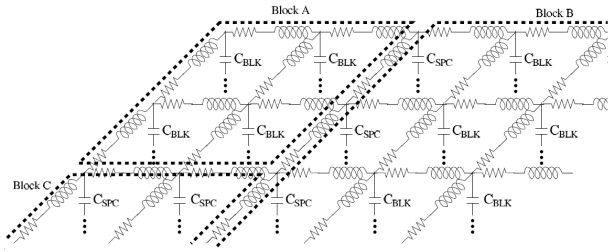
Prior work [43] highlights the impact of technology scaling on the RLC components of the PDN. For instance, the value of R becomes smaller with technology scaling, L remains almost the same, while C increases to constrain inductive noise.

2.2.1 Lumped PDN model vs Distributed model

One of the limitations of the lumped voltage model is that it does not capture local, inter-core voltage variations in a CMP, but instead provides an aggregate view of the voltage variations across the entire chip. With large number of cores on the chip, varying workload activity could lead to large inter-core voltage variations. In order to understand the characteristics and impact of these variations, a distributed voltage model has been proposed [25], which uses a RL network to model the cores/functional units in the core at a much finer granularity. Figure 2.2(a) from [25] shows the detailed power deliver network with a distributed on-chip power-supply grid. The off-chip network models the motherboard, package and off-chip decap capacitors and parasitic inductances. Figure 2.2(b) shows the distributed on-chip grid model. The C4 bumps that connect the grid to the off-chip network are modeled as parallel connections. The on-chip grid is modeled as an RL network as shown. More details on the model can be found in the paper [25].



(a) Package model



(b) On-die grid model

Figure 2.2: Distributed Power Delivery Model [25]

The distributed model provides voltage modeling at a fine granularity and is flexible so that the granularity can be adjusted. However, the lumped model suffices for our purpose as the goal is to study voltage noise characteristics at the package level, especially since the study focuses on understanding the aggregate trends in voltage noise on large CMPs.

Chapter 3

Related Work

Much of the research work addressing voltage noise problem can be categorized into circuit-level, micro-architectural-level and software-level, depending on where in the system the solution has been applied. Below is a brief summary of the different techniques that have been proposed at each of these levels.

3.1 Circuit-level Techniques

One of the techniques applied at the circuit-level to reduce voltage noise is to use a hierarchy of on-die, on-chip, and off-chip decoupling capacitors [3] [9] to reduce the impedance of the PDN. Voltage being a product of current and impedance, smaller impedance would result in smaller peak-to-peak voltage swings. However, even well-designed PDNs have some impedance due to the wires, inductances and capacitances, which leads to voltage noise. Prior work [15] [39] highlighted mechanisms that can be applied at the time of floorplanning for simultaneous power supply planning and voltage noise avoidance. In another work, Ernst et al [20] proposed Razor, a circuit-level technique to dynamically detect and recover from timing failures by augmenting critical

flip-flops in the processor pipeline with shadow latches. The shadow latches provide extra timing margins. So if a timing violation is detected, potentially due to inductive noise, then the data from the shadow latch is used. Razor, however, is expensive to implement, especially in out-of-order processors that have large and complex array structures, with many tight timing paths. DIVA [8] also provides a way to dynamically detect and recover from transient faults, by employing an additional checker processor that runs in parallel of the main out-of-order processor and checks the results before committing the instructions.

3.2 Architectural Techniques

A number of architectural techniques have been proposed to mitigate voltage noise. The main approach adopted is to control the current variations such that the voltage variations are within the tolerable margins, for instance, by managing instruction fetch and issue. Powell et al [44] proposed two techniques - pipeline muffling and resonance tuning, to reduce inductive noise caused by the varying pipeline activity. Pipeline muffling controls the number of functional units switching at any given time by altering the instruction issue. A sudden surge in the current drawn is avoided by a-priori slow current ramp-up of functional units. Gupta et al [24] proposed a checkpoint/rollback framework called DeCoR, that commits instructions only once it is determined that there were no emergencies (using a low voltage sensor), else the pipeline is flushed and rolled back to a previous safe state. Reddi et

al [45] predict voltage emergencies from a history of architectural events and based on the prediction, execution rate is throttled to lower the rate of change of current, thereby reducing voltage emergencies. Gupta et al [26] proposed an event-guided adaptive voltage emergency avoidance scheme by removing recurring emergencies by prefetching, or hardware throttling of events that lead to emergencies or by introducing pseudo-nops in the instruction stream.

3.3 Software Techniques

A number of approaches have been proposed in the software/compiler layer to mitigate voltage emergencies. Toburen [48] proposed compilation techniques to reduce voltage noise. Yun et al [51] presented a power-aware modulo scheduling to reduce step and peak power profiles in VLIW processors. Hazelwood et all [28] further explored the possibility of runtime code optimization techniques to reduce future voltage emergencies. El-Essaway [19] used thread management to manage voltage noise in SMT processors. Gupta et al [27] showed that several microarchitectural events, such as L1/L2/TLB misses, branch mis-predictions, that stall the pipeline are likely to cause voltage emergencies. They further showed that only a few loops in the SPEC benchmarks led to a majority of emergencies. They proposed a few compiler-optimizations that can be used to reduce the occurrence of the above events and thus voltage emergencies.

3.4 Voltage Noise in Multi-core Processors

While a significant amount of this prior work has focused on characterizing voltage noise in single-core processors, very few studies have focused on chip multiprocessors. Reddi et al [46] studied voltage noise in a Core 2 Duo and found that there exist a constructive phase, in which the voltage swing increases with the aggregated activity of the two cores, and a destructive phase, in which different microarchitectural activities on the two cores effectively reduce the occurrences of voltage emergencies. Using an oracle-based droop scheduler, they demonstrated that intelligent scheduling policies could leverage the destructive phases in the combination of workloads to smoothen out supply voltage variations. Gupta et al [25] characterized local, on-die voltage variations using a detailed distributed power model, using a 4-core CMP. Recently, Miller et al [38] studied the impact of synchronization on voltage emergencies in multi-core systems. With all the cores waiting in the barrier, the activity in the system drops down significantly, only to shoot up as soon as the barrier is released, resulting in a surge in current draw. They proposed a barrier release policy optimized to mitigate occurrence of such voltage emergencies.

This thesis presents a detailed characterization of voltage noise in a multi-core system with larger number of cores, and presents a comparative study of voltage variations in high performant out-of-order and small, energy-efficient in-order cores. To the best of my knowledge, this is the first comparative study of voltage noise in big and small cores. The thesis also demonstrates

trends in voltage noise in single-ISA heterogeneous CMP systems.

Chapter 4

pvSim: Simulation Framework for modeling Power and Voltage

Power consumption has become a first-order design constraint of modern microprocessors, as it is becoming a challenge to contain the ever increase demand for power, especially with the rapidly increasing density of transistors on the chip as per Moore's Law. Multi-core architectures were proposed as a technique to lower power consumption, while still achieving maximum performance by utilizing multiple computing resources, each operating at a lower frequency. However, with reducing processor technology, increasing number of cores per package, power management in future multi-core systems will continue to be an area of active research in the field of computer architecture.

Increased rate of change of current due to clock gating and lower supply voltages, have increased the sensitivity of the processor to inductive noise in the PDN. Inductive noise, referred to as di/dt problem, lead to significant variations in the supply voltage, resulting in malfunctioning of the chip. Voltage margins are used to tolerate the worst-case voltage variations. Research has highlighted that the magnitude of these variations will increase with technology scaling, requiring much larger margins to safeguard reliable operation.

This would result in reduced power and performance. There is thus a growing push to move towards resilient architectures, that provide error-recovery circuits that dynamically detect and correct transient errors that might occur due to voltage margin violations. This is an active area of research in the architecture community, both at hardware and software level, to tackle the various aspects.

Power and voltage profiling is thus important to enable the design and evaluation of new schemes to make reliable and energy efficient architectures. While some aspects of this research can be effectively performed on existing hardware, fundamentally new structural designs, enhancements, will increasingly require simulation for design and validation. Accurately modeling power and voltage is thus crucial to generate reliable data. While many functional simulators are available to the research community, there is a need for an integrated simulator that provides for both power and voltage profiling at the granularity of a CPU cycle. This thesis introduces pvSim, an integrated online power and voltage profiler, based on Marssx86 [40], McPAT [36] and lumped voltage model [27].

This chapter provides a detailed description of the design and implementation of pvSim and the components that it is built on.

4.1 Marssx86

Marssx86 is a cycle-accurate, full-simulator of the 64-bit x86 architecture, developed by the CAPS research group at State University of New York

at Binghamton. Figure 4.1 shows the block diagram of the simulator. The main features of Marssx86 are:

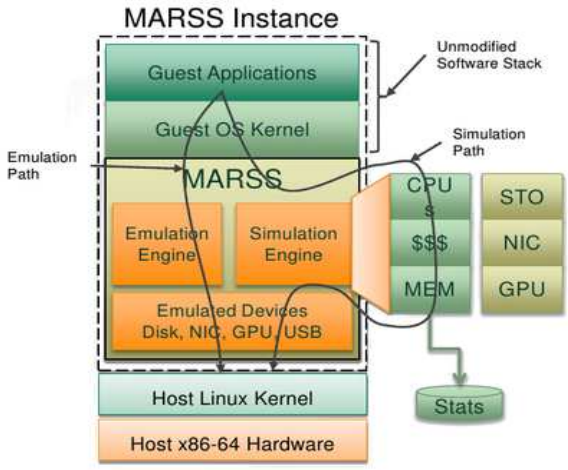


Figure 4.1: Block Diagram of Marssx86 Simulator [40]

(a) *Multi-core:* It supports multi-core simulation of the x86 ISA, with detailed pipeline model. It also supports hardware threads (SMT).

(b) *Full-system:* It uses QEMU base full-system emulation and can boot unmodified operating systems

(c) *Core models:* It is based on PTLsim [50] and leverages the detailed out-of-order core model of PTLsim. The in-order core model it provides is an improvement over the PTLsim simple core model.

(d) *Cache models:* It provides detailed model for coherent caches and on-chip interconnects. Different types of caches can be created, like write-back and write-through. It also supports the MESI and MOESI cache

coherency protocols. A cache hierarchy with multiple levels can be created to simulate different machines.

(e) *Benchmarks:* The benchmarks do not have to be specifically built for execution in the simulation environment. Any pre-compiled binaries and libraries for x86 can be run in the simulator.

(f) *High Performance:* It provides good speed of simulation and yields average instruction commit rate of 200k+ instructions per-second.

(g) *Heterogeneous system:* It can be used to simulate heterogeneous system consisting of both out-of-order and in-order cores.

4.2 McPAT

Several models have been proposed in the research community to enable computer architects to design power-efficient microarchitectures. For example, CACTI [47] was one of the first tools that was developed to estimate the power consumed by RAM-based memory structures. Wattch [14] is a popular tool that is used to model the dynamic power consumption of a processor. McPAT is a fairly new power model, which provides an integrated power, area and timing modeling framework for multi-core and many-core processor configurations, ranging from 90nm to 22nm. pvSim uses the McPAT power model over Wattch for several reasons as described below:

1. Wattch uses a simple linear scaling based on 0.8μ m technology, which are inaccurate to correctly predict the power consumption for recent

technology like 65, 45 or 22nm. McPAT, on the other hand, uses device models based on the ITRS [6] roadmap, to compute the device parameters at different technology levels and hence provides better estimates for power consumption with technology scaling.

2. Wattch estimates only dynamic power dissipation based on the activity counter values obtained from functional simulator. A study of voltage noise, however, requires estimation of static and dynamic power consumption, which McPAT provides.

3. Unlike McPAT, Wattch models power without taking into account timing and area information. This leads to inaccuracy in the power estimates.

4. Some aspects of Wattch are tied to the SimpleScalar simulator [7], like the synthetic RUU model, and thus may not provide accurate estimation when used with other architectural simulators.

5. Rakesh et al [35] indicated that the power estimates produced by Wattch did not match published results on peak and typical power numbers of several processor configurations.

6. McPAT provides a integrated tool that can be used to estimate power of multi-threaded and multi-core processors, unlike Wattch's core power model.

McPAT provides a detailed model for the following components of the chip:

1. in-order and out-of-order cores, with models for subcomponents like the pipelines, load-store unit, execution unit, register-renaming unit, physical register file, memory management units, on-chip caches,

2. networks-on-chip

3. shared caches

4. on-chip memory controller

McPAT also models static leakage power of the different components in accordance with the forecasts in the ITRS road map including bulk CMOS, SOI and double gate transistors, as highlighted in ITRS. McPAT has become a de-facto standard for power modeling in computer architecture research community. It is not tied to a particular microarchitecture and can be used to derive power estimates for systems like the Intel Netburst and the DEC Alpha architecture.

4.3 Lumped Voltage Model

To model the voltage profile of the system, pvSim uses the lumped voltage model, as described in Chapter 2.2. The PDN is modeled based on the parameters of the Pentium 4 package, as detailed in Table 5.2. The impedance plot of the PDN is shown in Figure 5.1. The model takes as input the power trace generated from the functional simulator are convolved with the impulse response from the PDN circuit to obtain the per-cycle voltage trace.

4.4 Integration of Marssx86, McPAT and Lumped Voltage Model - pvSim

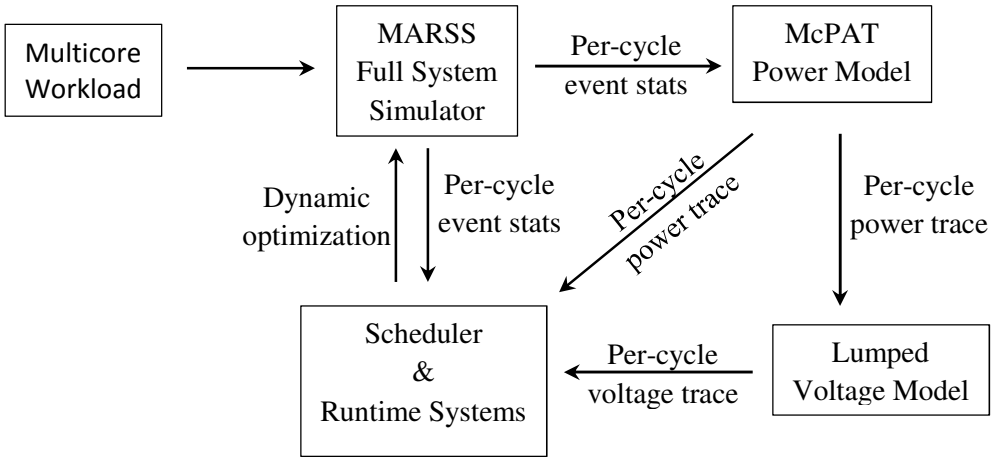


Figure 4.2: Block Diagram of pvSim

This section provides details of how pvSim integrates Marssx86 and McPAT to provide power and voltage data at a CPU-cycle granularity.

4.4.1 Design

Figure 4.2 indicates the block diagram of pvSim. The Marssx86 simulator is used for simulating a benchmark on an x86 machine configuration. The machine configuration and the run-time performance statistics of the benchmark execution are then fed into the McPAT power model to produce power consumption estimates. The power estimates can then be used by subsystems to detect hot-spots and to apply various run-time power optimizations, feeding back into the simulator. To generate voltage profile, the generated power

trace is fed as input to the lumped voltage model as described in Section 4.3. While the lumped model has been used here, the framework is generic and any model of the PDN, like the distributed model [25], could be plugged-in to generate the voltage profile.

4.4.2 Implementation

Below are some of the modifications that were made to Marssx86 and McPAT to build pvSim:

1. *Marssx86:* To generate online power estimates, per-cycle performance statistics are required. Modifications were made to Marssx86 to enable pvSim to compute the activity in the different components of the system in a single cycle. This requires computing the difference in the statistics thus far and till the previous cycle. This computation is performed for every activity counter that is required by the power model, every cycle. However, the framework can be configured to generate power estimates for a bunch of cycles instead of one. Further, there are many events that are not sampled by the simulator. For instance, classification of total number of committed instructions into integer, floating-point, branch, load and store, number of ALU and FPU operations, register file reads, writes, register renames, etc. Marssx86 was modified to gather statistics pertaining to these events to allow proper power computation. The per-cycle statistics are then communicated to the power model, every cycle, to generate the power estimates.

2. *McPAT:* The modifications made to McPAT are as follows:

   - McPAT uses an XML interface as the frontend to take input from any functional simulator. It takes as input an XML file that contains information about the machine configuration and the performance statistics, i.e, value of the event counters. For online power computation, the XML interface was removed and the code was re-structured as a library to provide APIs that can be used by any functional simulator to use the power model. This makes the framework very generic in nature in terms of the functional simulator that can used.
   - McPAT is designed to compute total power consumption given the aggregate activity counter values for the benchmark execution. However, for online power computation, power has to be computed every cycle, given the delta of counters for a cycle. McPAT was modified to ensure that the static leakage component of the power is computed just once at the beginning of the simulation and subsequently, the values of the internal data structures is reset every cycle to compute the run-time dynamic power for that particular cycle.

4.4.2.1 Power modeling of multi-cycle events

There are many microarchitectural events/operations that require multiple cycles to execute, for instance, ALU operations, cache hit and miss events.

Thus, when computing the power consumed per-cycle, it is important to factor the effects of such multi-cycle events. McPAT was modified to take this into account. Every event, like L1 cache miss, is associated with a value of latency, which is the number of cycles it takes for the event to complete. The power consumed by multi-cycle event is evenly distributed across this number of cycles it takes for the event to complete. Every time such an event occurs, the power due to the event is divided by the latency and just one unit of it is added to the power estimate of that cycle. The rest is placed into a per-subsystem queue. In each subsequent cycles, a unit of this power is added to the power estimate of that cycle, till the queue becomes empty, i.e, all the power components have been distributed. The functional simulator internally takes into account the latency of a subsystem to ensure that an event is not triggered until a previous event completes, or the total number of outstanding events is less than the configured value.

4.5 Summary

This chapter introduced pvSim, an integrated simulation framework that models per-cycle power and voltage in a cycle-accurate functional simulation environment for x86 systems. The framework is built using marssx86 as the full-system simulator for x86 systems, McPAT as the power model and lumped voltage model. The rest of this thesis uses pvSim to study voltage noise in large CMP systems.

Chapter 5

Voltage Noise in Large CMPs

In addition to performance and power, reliability is becoming a first-order design criteria for microprocessors. A number of approaches, both in hardware and software, have been proposed to manage the increasing level of voltage noise in chip multi-processors. However, most of the prior work in this area has focused on CMP systems with only a few cores. With the massive penetration of CMPs, it is important to measure and characterize voltage variations in large CMPs. Furthermore, all the prior work have studied voltage noise only in the performance oriented out-of-order (OoO) cores. With the increased adoption of small, power-efficient in-order cores in mobile devices and even in servers, it is critical to understand if there is a difference in the nature of supply voltage noise on the two types of cores. This characterization will enable to better understand the design space of reliability in large CMP systems. This chapter presents a detailed characterization of supply voltage variations in both OoO and in-order core types.

5.1 Experimental Setup

This section describes the simulation infrastructure used to study voltage variations in CMP systems.

5.1.1 pvSim Configuration

We use pvSim for the characterization of voltage variations. The configuration parameters for a single out-of-order and in-order core is shown in Table 5.1. The multi-core OoO configurations use a 3-level cache hierarchy, with the size of the shared L3 cache being scaled as the number of cores are increased. On the other hand, the small, in-order core configurations use 2-levels of cache, with the size of L2 scaled with the number of cores.

	Out-of-Order Core	In-Order Core
Clock Rate	3.0 GHz	1.6GHz
Fetch Width	4	2
Decode Width	4	2
Inst Window	128 ROB, 64-LSQ	-
BTB	1K Entries	1K Entries
RAS	1024 Entries	1024 Entries
L1 I/D cache	32KB each, 4-way, 2 cycles	32KB, 4-way, 2cycles
L2 cache	128KB, 8-way, 12 cycles	128KB, 8-way, 10 cycles
L3 cache	1MB, shared, 40 cycles	-
Int ALU & mult/div	2 per-core, 1 cycle	2 per core, 4 cycles
FP ALU	2 per core, 6 cycles	1 per core, 6 cycles

Table 5.1: Core Configurations

For this study, pvSim is configured such that only the power consumed by the core, private and shared caches was modeled and do not include power

consumed by other components, like the memory controller and interconnects, as previous studies [27] have shown that voltage variations are most sensitive to load variations in the core and caches. The package characteristics of the second-order lumped model used are similar to that of Pentium 4 microprocessor and are summarized in Table 5.2. The impedance plot of the PDN is shown in Figure 5.1. It can be seen that the resonance frequency of the PDN occurs at 100MHz with a peak impedance of 10mΩ. The supply voltage is configured at 1V. The PDN is kept the same as the number of cores are varied, to demonstrate the impact of increase in core count on the magnitude and frequency of voltage variations.

Resonant Frequency	Resistance	Quality Factor	Peak Impedance (Z)
100 MHz	1mΩ	3	10mΩ

Table 5.2: PDN Parameters

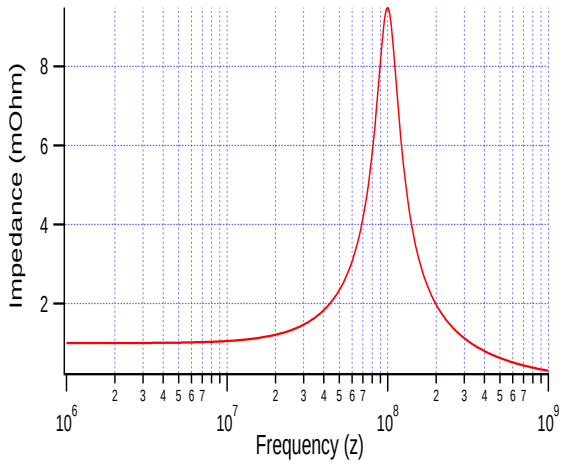


Figure 5.1: Impedance Plot of the PDN used for modeling Voltage

5.1.2 Benchmarks

The multi-threaded PARSEC benchmarks [11] were used for the study. The PARSEC benchmark suite has been designed to allow researchers to explore the design space of chip-multiprocessors. The suite consists of emerging workloads in the areas of data mining, synthesis (RMS) and visual recognition, as well as applications that are representative of large-scale real-world multi-threaded programs. The benchmark suite consists of programs with varied characteristics like working set size, locality, data sharing, I/O traffic, synchronization, among others, as highlighted in [11].

Each benchmark in the suite is simulated for 200 million instructions from the region of interest. The number of threads of execution equal the number of simulated cores and are affined to a particular core. The *simlarge* input set was used to run each of the benchmarks.

5.1.3 Limitations

The absolute values of power and voltage are subject to errors in simulation and modeling at different levels in pvSim. For instance, the level of voltage variations are dependent on the per-cycle power consumption of the system, which in pvSim depends both on the power model and the functional simulator marssx86. However, the relative trends are consistent with the data that has been published in the literature.

5.2 Characterization of Voltage Noise in Out-of-Order Cores

Out-of-order core is a de-facto core type used to build all types of servers, ranging from low-end to high-end. An OoO core achieves high performance by extracting parallelism in the dynamic instruction stream of a program, at the cost of high power consumption. This section presents a characterization of voltage noise in the PARSEC benchmarks when executing on OoO cores.

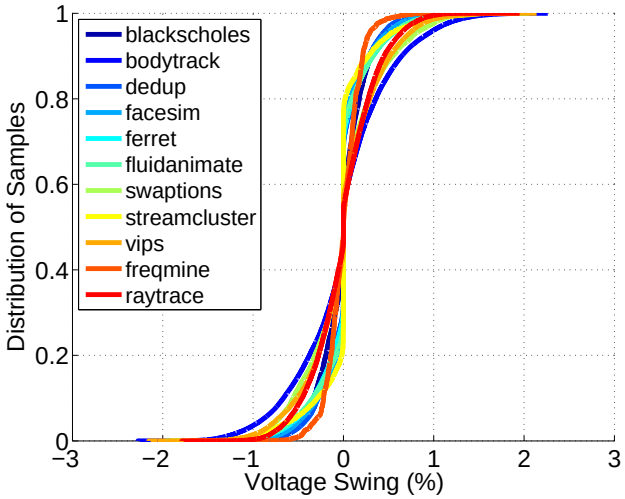


Figure 5.2: Cumulative Distribution of voltage swings for the PARSEC benchmarks on a single OoO core

Figure 5.2 indicates the distribution of samples of voltage swings for the PARSEC benchmarks. It can be seen that majority of the samples are distributed close to the nominal supply voltage and a very small percentage of all the samples are below 1% of the nominal. Only *bodytrack* and *vips* experience a maximum voltage drop of greater than 2%. Thus, for all the

future experiments, we assume an aggressive voltage margin of 2% for the OoO cores, purely for characterization purpose.

5.2.1 Correlation with Current Variations

From Figure 5.2, we see that different benchmarks result in different levels of maximum voltage swing. During execution, due to the different microarchitectural activity during program execution, the rate of change of current, varies from program to program. To quantify the relationship between the magnitude of voltage swing and the magnitude of the swing in current, we conducted the following experiment - compute the maximum swing in current during the benchmark execution by using a sliding window of about 30 cycles, to compute the maximum swing in current in that period and the maximum of all these values during the entire simulation is used as the maximum swing in current for that benchmark. Figure 5.3 shows that the magnitude of the peak voltage drop is strongly correlated to the maximum fluctuation in current, observed during the benchmark execution. Inspite of the lower variation in current in *bodytrack* when compared to *vips*, *bodytrack* experiences a larger droop. This can be attributed to the nature of voltage fluctuations in *bodytrack*, as explained below in Section 5.2.2. The voltage noise in the *bodytrack* benchmark are predmoninantly due to the variations in current matching the frequency of the PDN, leading to much larger amplitude of voltage fluctuations.

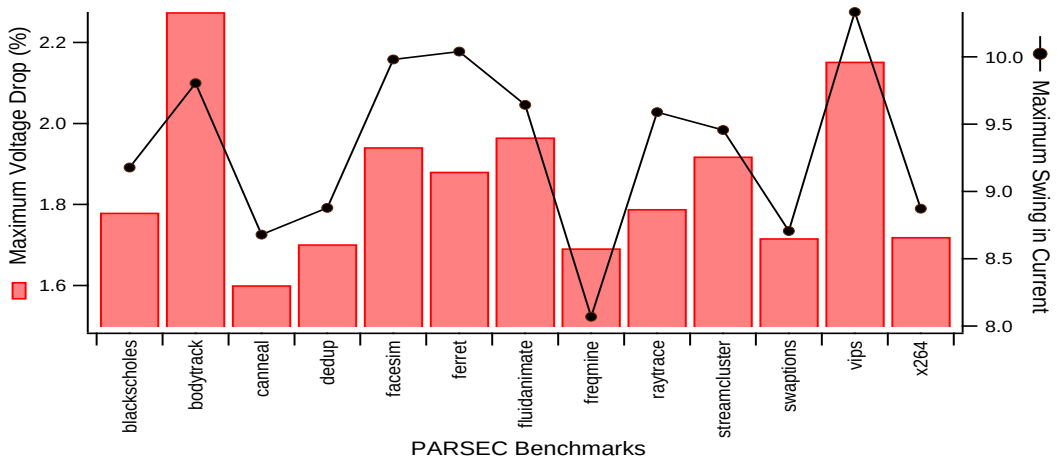


Figure 5.3: Maximum voltage undershoots in PARSEC benchmarks on a single OoO core, with correlation to maximum swing in current

5.2.2 Characteristics of Current Variations

As highlighted in Figure 5.3, each of the PARSEC benchmarks experiences a different magnitude of maximum voltage drop. This section provides a closer look at the reasons behind the difference in magnitude, using the benchmarks *bodytrack* and *freqmine* as examples.

bodytrack is a computer vision program which tracks a 3D pose of a markerless human body using mutliple cameras through a sequence of images [11]. Consider the voltage and current trace in Figure 5.4 of the *bodytrack* benchmark. The voltage, as annotated, corresponds to the maximum drop observed during the entire simulation of the benchmark. As can be observed, the current and voltage trace leading up to the maximum swing, resonate with a period of about 30 cycles, which is also the period of the PDN. Current variations at the

resonant frequency of the PDN, build up energy and result in a larger voltage swing when compared to an isolated maximum current event. This resonance is the reason why the magnitude of the worst case swing is maximum in *bodytrack*. The kernel of the benchmark that causes this resonance, as indicated in Table 5.3, is the computation of particle density which is performed at every time step.

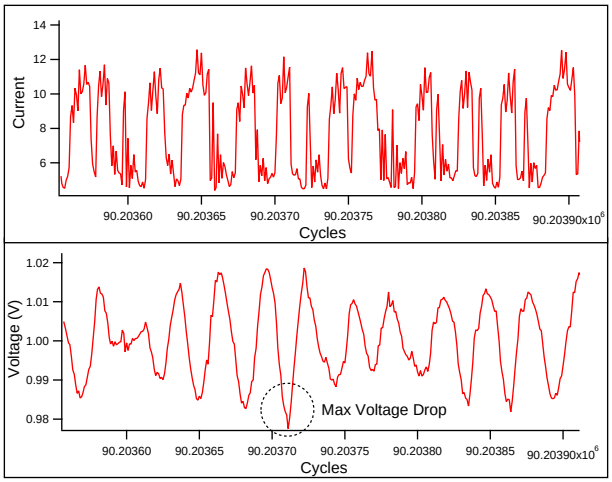


Figure 5.4: Current and Voltage trace snippet for *bodytrack*

freqmine application uses an array-based adaptation of the Frequent-Pattern growth method for Frequent Itemset Mining, which is typically applied in problems like protien sequencing and log analysis. Figure 5.5 shows the current and voltage trace observed around the worst-case voltage fluctuation during benchmark execution. Unlike in the case of *bodytrack*, the current variations do not resonate with the first droop frequency of the PDN. However, from our analysis of the entire current trace, this event corresponds to a large spike in

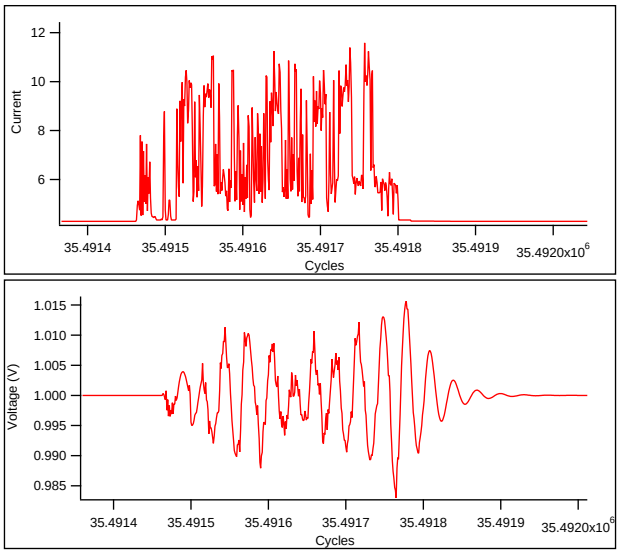


Figure 5.5: Current and Voltage trace snippet for *freqmine*

current, leading to an isolated fluctuation in voltage.

5.2.3 PARSEC Kernels Causing Maximum Droop

Each of the PARSEC benchmarks consist of a set of kernels that execute in parallel, as described in [11]. Table 5.3 shows the names of the kernels that lead to the maximum voltage drop during simulation for a few benchmarks. Exploring particular aspects of these kernels, like the underlying microarchitectural events that lead to the voltage noise, is beyond the scope of this thesis and is left as future work.

Benchmark	Kernel	Description
bodytrack	Calculate particle weights	This kernel is executed during every time step during every time step, and is thus computationally most intensive
fluidanimate	Compute Densities	This kernel estimates the density of the fluid at each particle
raytrace	Traverse BVH	BVHs are used to ray trace a dynamic scene. This kernel traverses this structure and is executed often during execution
dedup	Compression	This kernel performs parallel compression
freqmine	Build FP-tree header	This kernel scans the transaction database and counts number of occurrence of each item

Table 5.3: PARSEC kernels that lead to maximum voltage droop

5.2.4 Impact of Core Count on Voltage Noise

Prior studies [46] have shown that supply voltage variations in a single core are correlated to the microarchitectural events like cache misses, TLB misses, branch mis-predictions, among others. As the number of cores increase, the different level of workload activity on the different cores lead to constructive and destructive interference in voltage noise. Further, Miller et al [38] highlighted that as the number of cores increase beyond 32, large voltage variations are primarily caused by increased synchronization among the threads of a program. This section aims at studying the general trend in the magnitude of voltage variations as the number of OoO cores are increased.

Figure 5.6 shows the change in maximum voltage drop for each benchmark, as the number of cores is increased from 1 to 16. For benchmarks like *bodytrack, canneal, dedup, fluidanimate, streamcluster,etc*, the maximum worst case drop increases as the number of cores increase. However, for many others

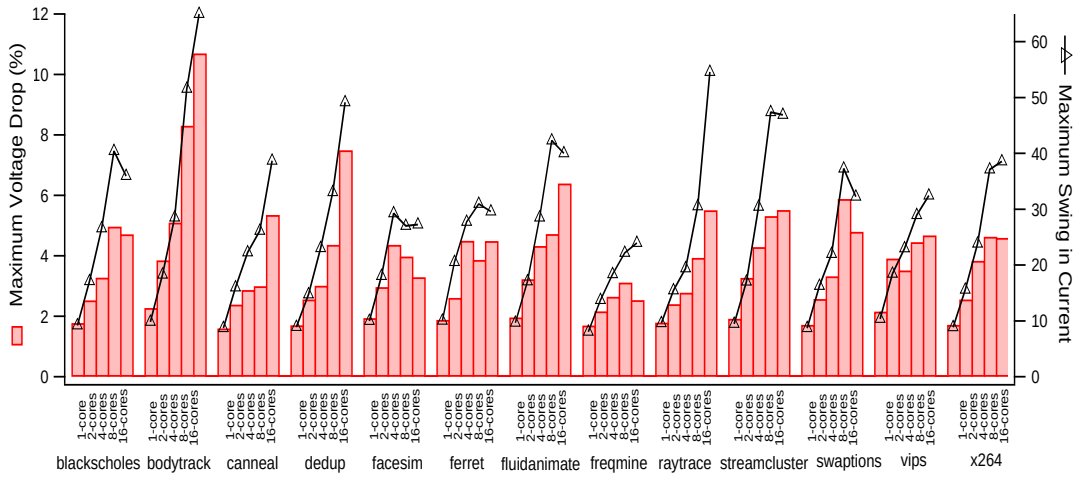


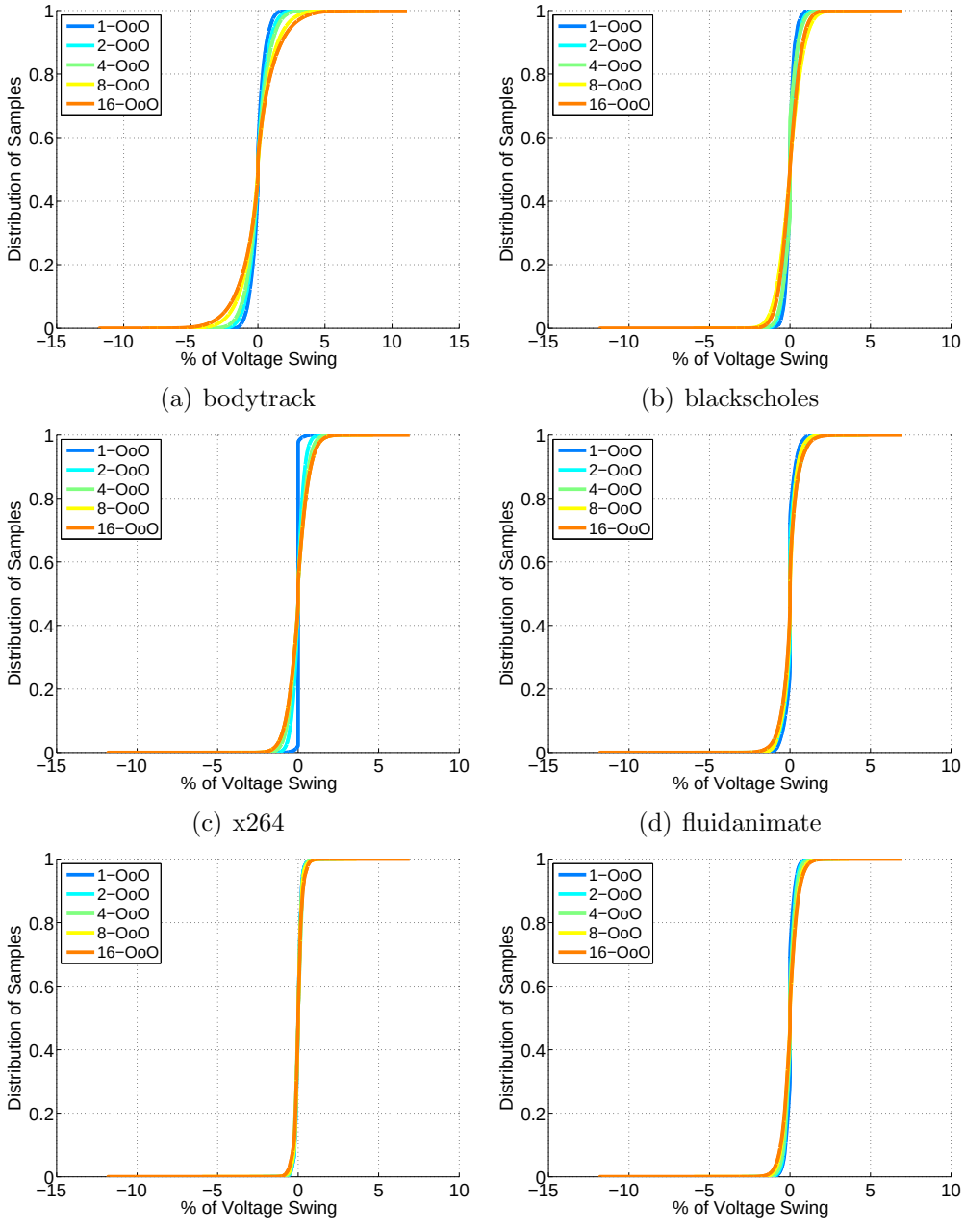
Figure 5.6: Impact of increase in core count on maximum voltage undershoot in OoO core

facesim, ferret, freqmine, swaptions, vips, the maximum drop either decreases slightly or remains stable as the number of cores increase beyond 4. This is an interesting trend, as it points to a potential destructive interference that takes place across the cores, that effectively smoothens the voltage noise. This is a useful data point, as it further highlights that destructive interference takes place even on systems with large cores and that a *droop scheduler*, as proposed by [46], could be employed to mitigate voltage noise.

Figure 5.6 also highlights that the strong correlation between the magnitude of the maximum voltage drop and rate of change of current continues to hold even as the number of cores increase, and the increase in the magnitude of the voltage swing can be attributed to a proportional change in current.

Figure 5.7 shows the cumulative distribution of voltage samples with increasing number of cores (OoO), for a few representative PARSEC bench-

marks. It can be seen that while the maximum voltage swing increases, for most benchmarks like *fluidanimate, freqmine, facesim*, the distribution is not impacted with increase in the core count. As in with a single-core, only about 0.05% of all the samples exceed the voltage margin of 2%. The *bodytrack* benchmark, shows the maximum spread of voltage swings from the nominal supply voltage, which increases with the number of cores.



(a) bodytrack (b) blackscholes

(c) x264 (d) fluidanimate

(e) freqmine (f) dedup

Figure 5.7: Cumulative distribution of voltage samples for different PARSEC benchmarks with increasing core count (OoO)

To quantify this observation, we computed the percentage of droops experienced during the benchmark execution, where, a voltage droop corresponds to a voltage drop beyond the set voltage margins. Assuming an aggressive voltage margin of 2%, the voltage trace is scanned to count the number of instances when it exceeds the margin. The total number of droops divided by the total number of cycles of execution yields the percentage of droops. Table 5.4 provides the data for the PARSEC benchmarks, for different number of cores. The number of voltage droops on a single or dual core are very negligible and hence are not shown in the table. The two main observations from this data are that (1) the absolute values of the droops indicate that the frequency of occurrence of such voltage swings is very low, (2) while for a majority of benchmarks, the percentage of droops increase with increase in the number of cores, for some they remain the same or decrease, indicating both constructive and destructive interference across the threads of the benchmarks.

Benchmark	Percentage of Droops		
	4-cores	8-cores	16-cores
blackscholes	0.0001062	0.081617	0.017661
bodytrack	1.0283	14.423	30.955
canneal	0	0	0.00014005
dedup	0.000016884	0.00056403	0.012634
facesim	0.0048493	0.0012403	0.00001534
fluidanimate	0.020794	0.0046054	0.16463
streamcluster	0.0043466	0.038968	0.054671
swaptions	0.00024206	1.4905	0.059073
x264	0.00019395	0.045323	0.053422

Table 5.4: Impact of increase in core count on droops for OoO cores

5.3 Characterization of Voltage Noise in In-Order Cores

In-order cores are increasingly being used in embedded devices due to their power efficiency and small area. For workloads that have very high instruction level parallelism, the in-order cores provide good performance. This section presents a characterization of voltage noise in the PARSEC benchmarks when executing on the in-order cores and highlights the impact of increase in the number of in-order cores on voltage noise.

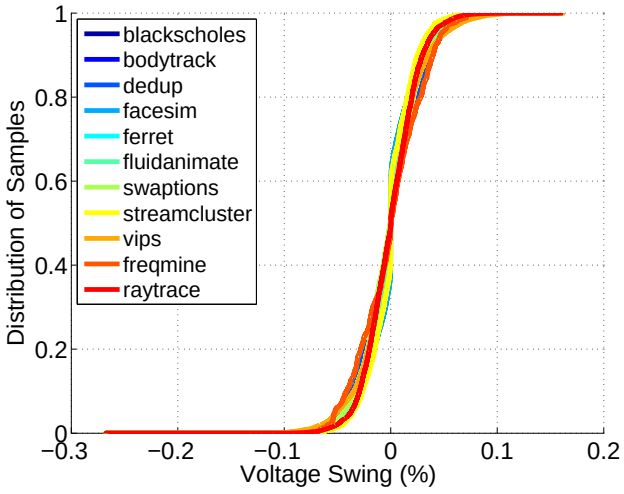


Figure 5.8: Cumulative Distribution of voltage swings for the PARSEC benchmarks on a single in-order core

Figure 5.8 shows the cumulative distribution of the voltage variations observed when running the PARSEC benchmarks. It can be seen that the samples for the different benchmarks are packed more tightly around the nominal and also the lines are more tightly bound together, with little variation. A very small percentage of all the samples are just below 0.2% of the nominal. Thus,

for all the future experiments with in-order cores, we assume an aggressive voltage margin of 0.2%, again, purely for characterization purpose only.

5.3.1 Correlation with Current Variations

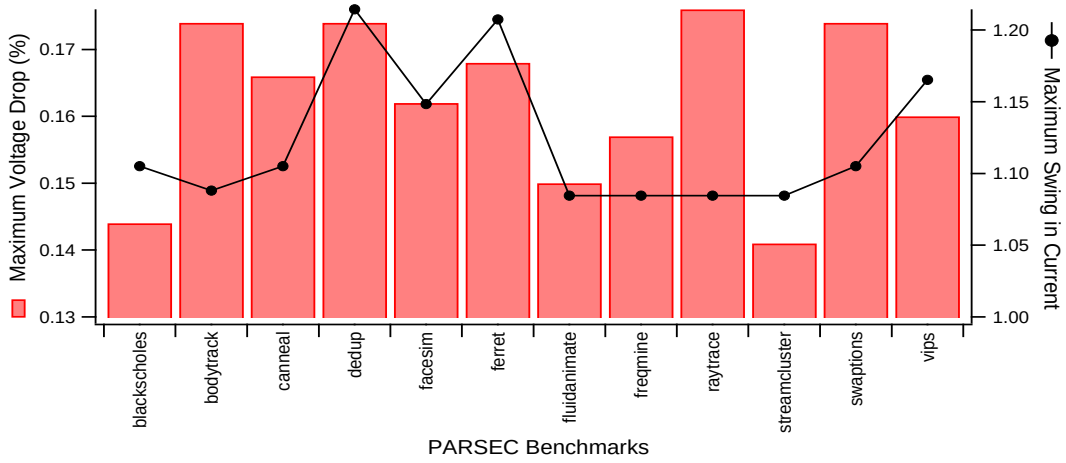


Figure 5.9: Maximum voltage undershoots in PARSEC benchmarks on a single in-order core, with correlation to maximum swing in current

An analysis similar to that in Section 5.2.1 was performed to evaluate the correlation between the magnitude of the maximum voltage drop and maximum swing in current in the in-order cores. The data in Figure 5.9 indicates that the two do not seem correlated in the case of an in-order core and is contrary to expectation. To investigate the cause behind this observation, we examined the current and voltage traces for all the benchmarks for cycles in the vicinity of maximum voltage drop. Figure 5.10 and Figure 5.11 shows two sample traces for the *bodytrack* and *streamcluster* benchmarks. In both the traces, the voltage drop corresponds to an isolated high current event, and

the current variations do not resonate with the PDN. This is similar to the observation on the OoO cores for some benchmarks, but does not substantially explain the cause of the observation in Figure 5.9. We leave this investigation as part of our future work.

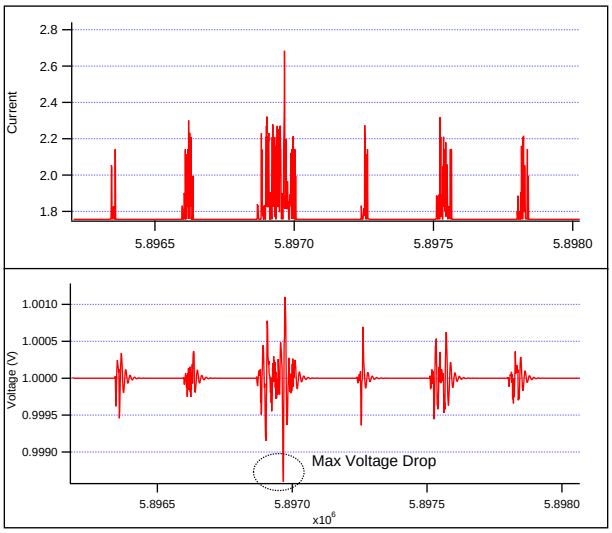


Figure 5.10: Current and Voltage trace snippet for *bodytrack*

5.3.2 Impact of Core Count on Voltage Noise

Figure 5.12 indicates how the maximum voltage droop varies as the number of in-order cores are increased from 1 to 16. Unlike in the case of OoO cores, the maximum voltage droop increases linearly with increase in the core count for all the benchmarks.

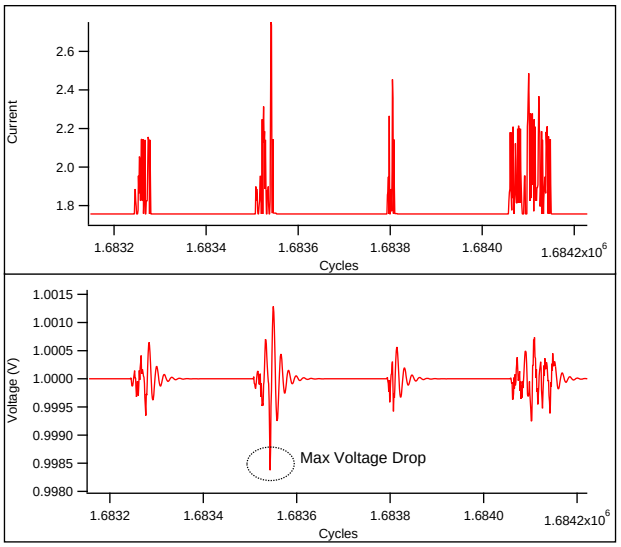


Figure 5.11: Current and Voltage trace snippet for *streamcluster*

5.4 Big Core vs Small Core

The big out-of-order cores and small in-order cores differ in the way the dynamic instruction stream is executed in the microprocessor. The big cores exploit the possibility of executing independent instructions out-of-order, to utilize the CPU cycles that would otherwise go unused due to stalls in the pipeline. For example, while the data operands required to execute an instruction are being brought into the memory, instead of stalling, the processor starts executing next instructions in the stream that are not dependent on the stalled instruction. This reduces the wastage of CPU cycles. The OoO cores have additional hardware structures like the Re-order Buffer (ROB), load-store queues (LSQ), register renaming unit (RNU), to name a few, to support out-of-order execution without affecting the correctness of the programs. Due

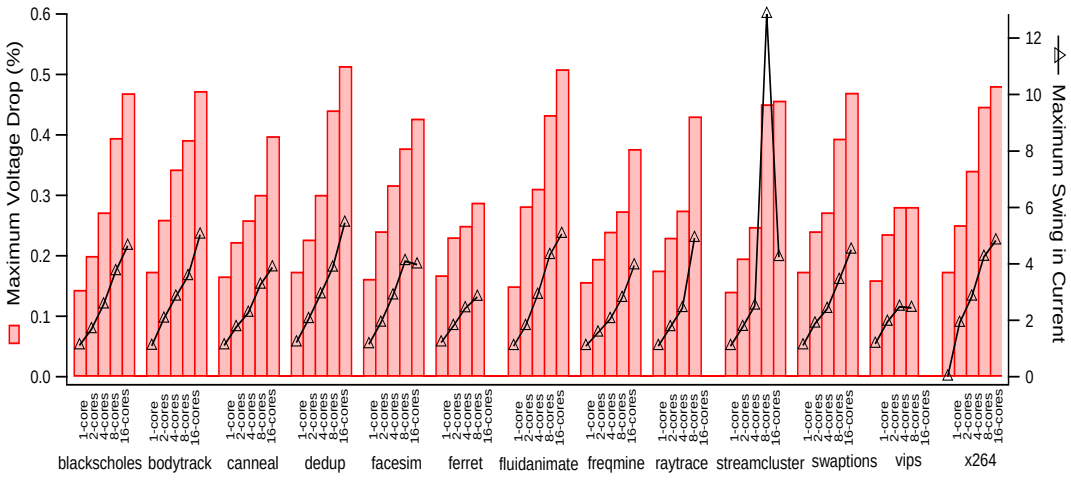


Figure 5.12: Impact of increase in core count on maximum voltage droop in in-order core

to these additional structures and logic, the power consumption of the OoO cores is much larger than the in-order cores. However, the added complexity significantly improves the performance of programs that do not have inherent instruction level parallelism (ILP) and also for programs that have a very high memory level parallelism (MLP), at the cost of higher power consumption. The in-order cores, on the other hand, provide good performance for compute-intensive workloads whose subsequent instructions in the stream are mostly independent (i.e, programs with high ILP) and can be executed in parallel in the in-order cores. This has been demonstrated by a number of prior research [17] [22] [33] [37] [49] addressing the issue of appropriately scheduling workloads on a heterogeneous system consisting of both big and small cores.

The difference in the workload execution pattern and the power profile of the two types of cores, results in different levels of voltage noise.

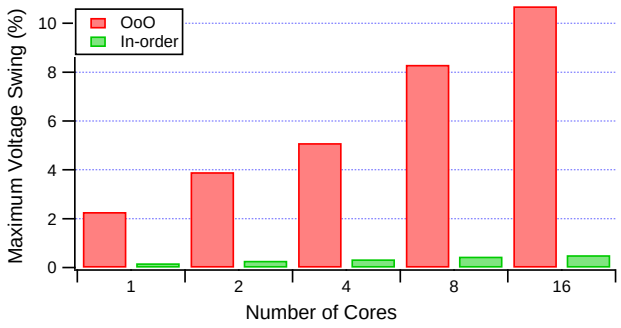


Figure 5.13: Maximum voltage droop comparison between OoO and in-order cores

As seen in Figure 5.13, the maximum voltage droop in in-order cores is an order of magnitude smaller when compared to that in OoO cores. Furthermore, Figure 5.14 indicates a very interesting trend in the rate of increase of the magnitude of the worst case voltage swing for both types of cores. As can be seen, the magnitude of voltage swing in an in-order core increases linearly from one to eight cores, like the OoO cores, but increases only gradually beyond that (this is based on the data for 20 and 27 in-order cores). This trend has positive implications on the design of future servers composed of only the in-order cores, from the perspective of reliability. However, due to conflicting requirement of good performance, the reliability benefits of in-order core configurations might not be achievable.

Consider Figure 5.15 which indicates the performance equivalence between the two types of cores. It appears that about 4 times the number of OoO cores are required to achieve nearly the same performance as the OoO cores. For instance, with 4 in-order cores, about 50% of the PARSEC benchmarks

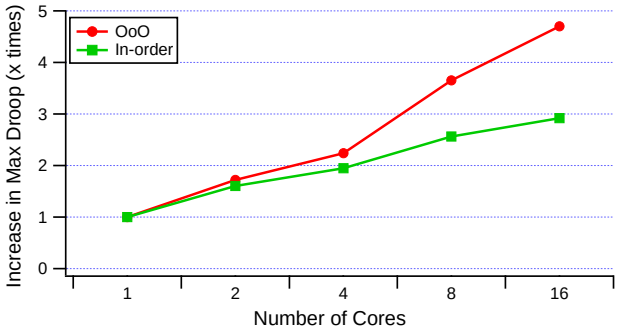


Figure 5.14: Rate of increase in maximum voltage swing with increase in the number of cores, normalized to 1-core

yield better performance when compared to that of a single OoO core, and similarly for other configurations. From the reliability perspective, replacing out-of-order cores with more number of in-orders might appear acceptable, however, Figure 5.16 shows that the homogeneous in-order configurations result in a larger energy-delay product (EDP), implying poor energy efficiency. This nullifies the benefits of using the in-order cores for better reliability and low power. Thus, the number of in-order cores in a system cannot be arbitrarily increased to match the performance of the corresponding OoO cores.

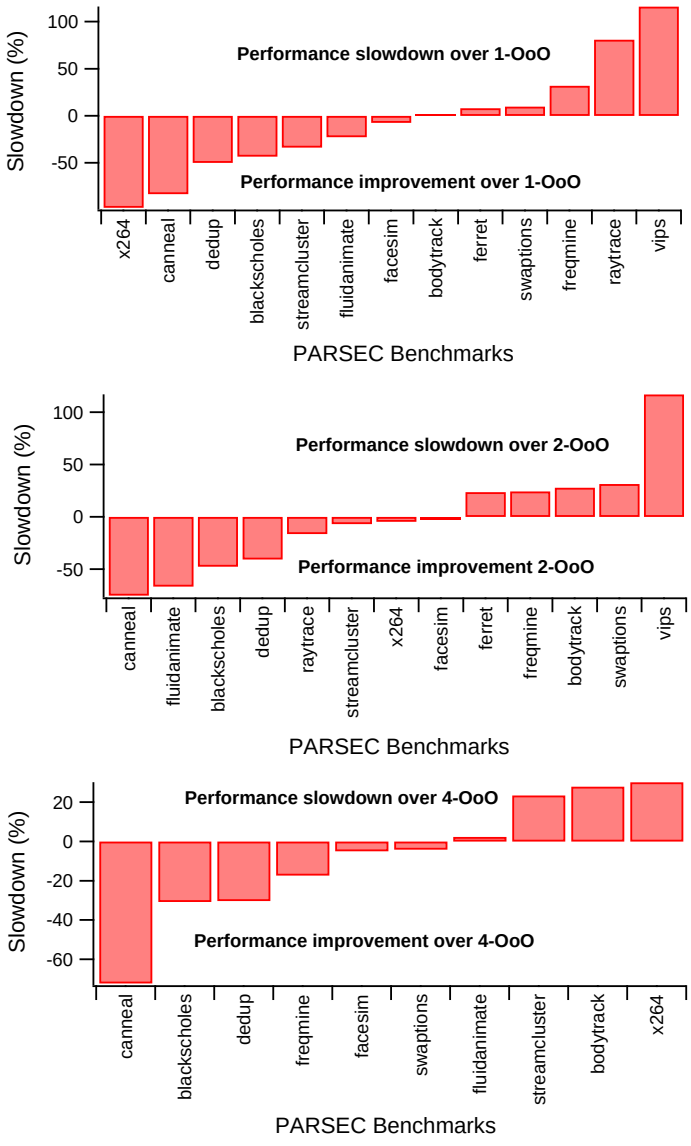


Figure 5.15: Performance Equivalence between OoO and in-order cores

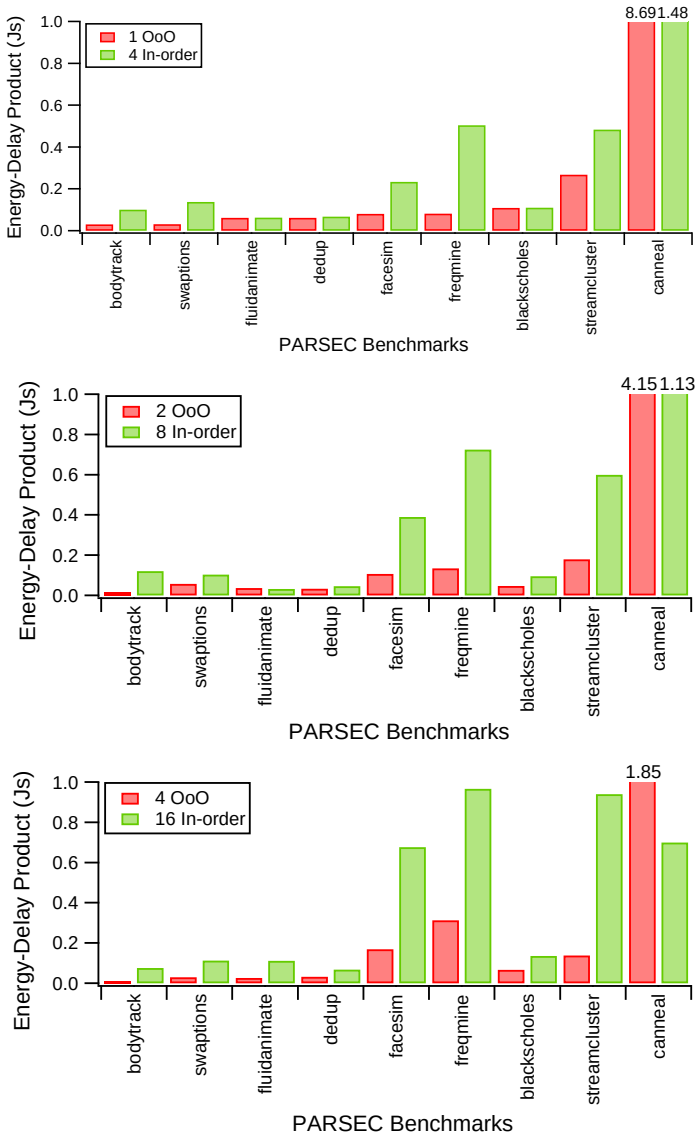


Figure 5.16: Energy-Delay Product comparison between performance equivalent OoO and in-order cores

5.5 Summary

This chapters presents a characterization of voltage noise in large CMP systems consisting of upto 16 cores, which are either out-of-order or in-order. The maximum voltage drop in an out-or-oder core is an order of magnitude larger when compared to the in-order core. The data indicates that the magnitude of maximum voltage drop increases linearly as the number of cores increase, for both types of cores. However, for the in-order cores the maximum drop increases only very slowly beyond 16 cores. Furthermore, the magnitude of this drop in increasing number of out-of-order cores is found to be strongly correlated to the maximum swing in the current. However, the correlation is not strong for the in-order cores. Current variations that resonate with the PDN lead to much larger voltage variations, as indicated by the noise experienced by the *bodytrack* benchmark. In an in-order core, most variation observed were due to isolated high current events.

Experimental results also indicate that the level of voltage noise, for instance, the percentage of voltage droops, do not increase significantly as the number of cores are increased, potentially due to destructive interference across the cores leading to voltage smoothing.

This chapter also considered performance equivalent out-of-order and in-order system configurations, to evaluate the performance, power and reliability design constraints. The results indicate that as the number of in-order cores are increased to match the performance of an out-of-order configuration, the energy-efficiency reduces, even though the configurations offer better

reliability.

How can we build a systems that provides a good trade-off between the performance of the OoO cores and the power-efficiency and reliability of the in-order cores? The next chapter evaluates single-ISA heterogeneous CMPs based on these design constraints.

Chapter 6

Voltage Noise in Single-ISA Heterogeneous Multi-Core Systems

A class of multi-cores, called heterogeneous multi-cores is gaining popularity, both in the industry and research community, due to their better energy-efficiency. For example, Intel's Sandy Bridge [2], AMD's Fusion [1], NVidia's Tegra [4], include CPU and GPU or IBM's Cell [30] that has accelerators. These, however, execute a different instruction set and are tailored to perform a particular functionality, like GPUs for graphics processing. While they provide performance and power efficiency, they cannot be used to execute general purpose workloads and require specialized programming in some cases like the GPU. Single-ISA heterogeneous multi-core architectures like ARM's big.LITTLE [5], on the other hand, use a combination of different types of cores, like the big OoO and small in-order cores, that both execute the same instruction set, but have different capabilities, performance and power levels. Depending on the characteristics of the workload, the runtime can select the most appropriate core that can provide best performance while minimizing energy consumption. The key insight in such a design is that different workloads have different amounts of ILP and MLP, making them suitable for execution on a particular type of core [17] [22] [33] [37] [49]. For instance, a workload

	Homogeneous OoO	Homogeneous In-order	Heterogeneous	TDP
Config-I	2	7	1 OoO, 2 in-order	42-45W
Config-II	3	10	1 OoO, 3 in-order	57-58W
Config-III	4	13	2 OoO, 3 in-order	74-77W
Config-IV	6	20	2 OoO, 6 in-order	110-113W
Config-V	8	27	3 OoO, 8 in-order	148-150W

Table 6.1: TDP Equivalence across different CMP configurations

with large amounts of ILP would be most suitable for a wide-issue, in-order core with no additional logic needed for extracting parallelism. The single-ISA offers the benefit of not having to write specialized programs or re-compilation, making dynamic scheduling decisions possible.

In this light, this section analyzes voltage noise in such single-ISA heterogeneous multi-core systems. Since the PDN of a microprocessor is designed taking into account the designated peak power of the processor, for a fair comparison of the level of voltage noise across different multi-core configurations comprising of different types of cores, this study relies on configurations with same designated peak power as reported by McPAT. The TDP equivalent configurations considered are summarized in Table 6.1. The mapping of OoO to in-order cores is not linear due to the different sizes of the last-level caches, adjusted to obtain reasonable performance with larger configurations.

6.1 Experimental Setup

Heterogeneous configurations were simulated using the different core models provided by *marssx86*, using frequency scaling in the in-order cores to run them at a lower frequency. The different cores in the heterogeneous configuration have a private L1 and L2 caches and all the cores share a last level L3 cache. The configuration of each of the two types of cores is the same as in Table 5.1.

The PDN used is also the same as that in Table 5.2. The supply voltage is also kept same at 1V. Further, PARSEC benchmarks are used for the study. Three PARSEC benchmarks, *fluidanimate, facesim* and *canneal* are excluded from the results presented in this chapter, as *fluidanimate* and *facesim* require an even number of benchmark threads to be executed but there were odd-numbered core configurations in some of our experiments, and the run-time of the *canneal* benchmark on the in-order and heterogeneous configurations was not feasible for conducting repeated experiments.

6.2 Voltage noise in a heterogeneous CMP

Figure 6.1 shows the maximum voltage swing in a heterogeneous configuration, compared to their TDP-equivalent OoO and in-order configurations. In Config-V, the maximum voltage swing in the heterogeneous configuration is roughly 53% lesser than in the corresponding OoO configuration, but an order of magnitude larger than the in-order configuration. However, consider Figure 6.2 that shows the EDP of the heterogeneous configurations. It can be

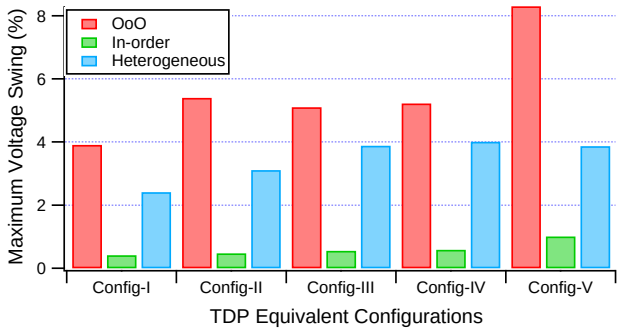
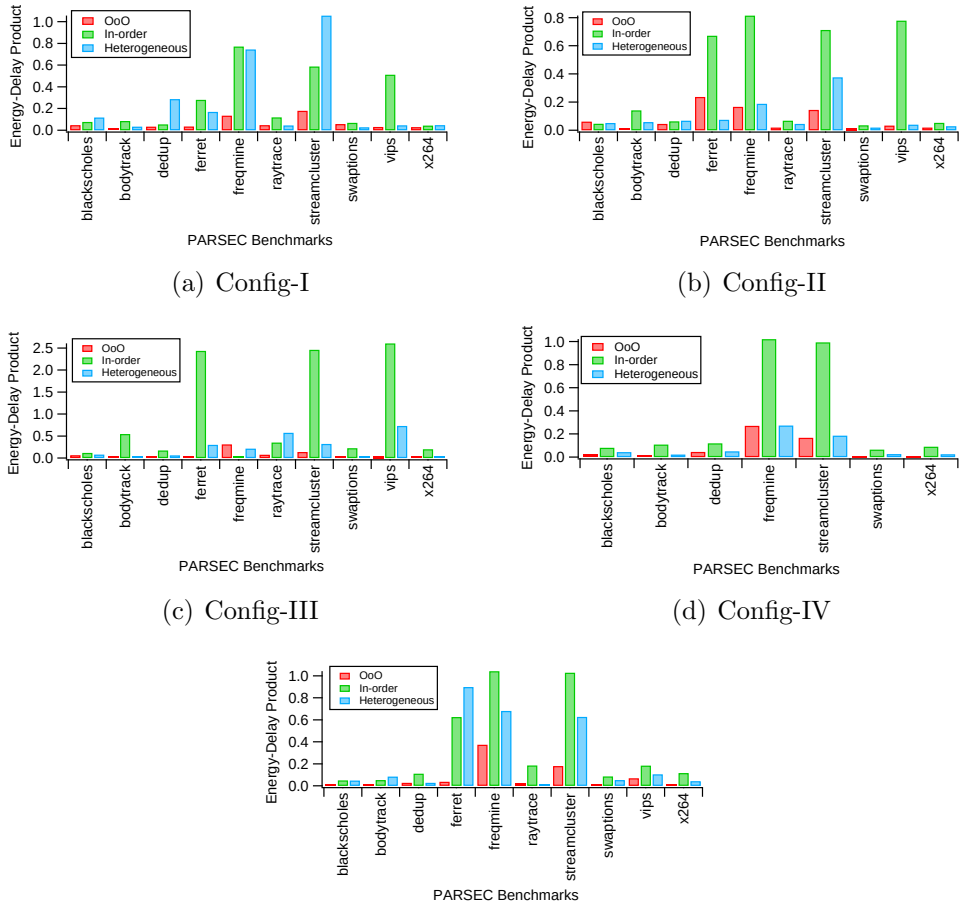


Figure 6.1: Maximum Voltage Droop in different TDP Equivalent Configurations

seen that for larger core configurations, from Config-II and above, the EDP in heterogeneous configurations is comparable to that of the corresponding OoO configurations, but is much lower when compared to the in-order configurations. In Config-I, for larger number of benchmarks like *dedup, freqmine, streamcluster, blackscholes, x264*, the EDP of the heterogeneous configuration is higher or equal to the in-order configuration. Config-I has the lowest ratio of OoO in the heterogeneous configuration, resulting in poor EDP.

When compared to the OoO configurations, the voltage guardbands on a corresponding heterogeneous system can be reduced on an average by about 35%. The benefits of reducing the voltage guardbands are twofolds, *a*) reduction in supply voltage, thus saving power; *b*) increase in operating frequency, increasing performance.A reduction in the voltage guardband can be translated to a proportional reduction in the supply voltage. Since $P = cV^2f$, the savings in power are proportional to the square of the reduction in the voltage guardbands. This would further lower the EDP, making heterogeneous

configurations power-efficient.



(a) Config-I

(b) Config-II

(c) Config-III

(d) Config-IV

(e) Config-V

Figure 6.2: Comparison of Energy-Delay Product across TDP Equivalent Configurations

6.2.1 Limitations of the study

Single-ISA Heterogeneous systems studied in this chapter are a subject of active research in the community. As highlighted earlier, workload schedul-

ing is a fundamental problem in the design space of single-ISA heterogeneous CMPs. Incorrect mapping of workloads to the most appropriate core can lead to suboptimal performance and even higher power consumption. Since some of the PARSEC benchmarks have threads executing different kernels, it is possible that some of these threads are more suited to run on a particular type of core. However, in our experiment, we did not consider the most appropriate mapping and hence the absolute values of EDP and voltage noise are not optimal. Further, since the two types of cores run at different frequencies, the precise characteristics of the PDN have to be determined as well.

6.3 Summary

A class of single-ISA heterogeneous multi-core systems consist of a combination of out-of-order and in-order cores, to provide good performance and energy-efficiency. This chapter provides some preliminary results on the nature of voltage noise observed on such heterogeneous systems. The results indicate that the maximum voltage swing experienced on a heterogeneous system lies in between the corresponding out-of-order and in-order configurations, which could allow reduction in voltage guardbands, potentially leading to improvements in both performance and power. However, the design space for heterogeneous systems is huge and requires more rigorous characterization.

Chapter 7

Conclusion

This thesis presents a detailed characterization of voltage noise in large multi-core systems, comparing the differences in voltage noise in big and small cores. The results indicate that as the number of out-of-order cores increase, the magnitude of the worst-case voltage droop increases, while in the case of an in-order core, the worst-case swings seems to plateau beyond 16 cores. Thus, microarchitectures designed for worst-case voltage noise will require very large voltage guardbands on larger systems, resulting in wastage of power and reduced peak operating frequency. The data, however, shows that the frequency of the worst-case swings is very low, less than 0.05%, and is not impacted as the number of cores increase, indicating the feasibility of microarchitecture designs that are optimized for typical case behavior.

The thesis also presents a preliminary characterization of voltage noise on single-ISA heterogeneous systems. The experiments indicate that the worst-case voltage droops in a heterogeneous system are larger when compared to a TDP-equivalent in-order configuration, but smaller than the corresponding OoO configurations. Since reducing the voltage margin benefits both power and performance, it enhances the EDP of the heterogeneous systems.

Bibliography

[1] Amd fusion. http://sites.amd.com/us/game/downloads/fusion-for-desktops/Pages/overview.aspx.

[2] Intel sandy bridge. http://www.intel.com/content/www/us/en/processors/core/core-i5-processor.html?cid=sem117p3989.

[3] *Microelectronics Packaging Handbook*, volume I. Chapman & Hall, 2nd edition, 1997.

[4] Nvidia. the benefits of multiple cpu cores in mobile devices. http://www.nvidia.com/content/PDF/tegrawhitepapers/Benets-of-Multi-core-CPUs-in-MobileDevices_Ver1.2.pdf, 2010.

[5] Greenhalgh, p. big.little processing with arm cortex-a15 & cortex-a7: Improving energy efciency in high-performance mobile platforms. http://www.arm.com/les/downloads/big_LITTLE_Final.pdf, Sept. 2011.

[6] Semiconductor Inustries Association. Model for assessment of cmos technologies and roadmaps (mastar). http://www.itrs.net/models.html, 2007.

[7] T. Austin, E. Larson, and D. Ernst. Simplescalar: An infrastructure for computer system modeling. *Computer*, 35(2):59–67, February 2002.

[8] Todd M. Austin. Diva: A reliable substrate for deep submicron microarchitecture design. In *Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture*, MICRO 32, pages 196–207, 1999.

[9] H. B. Bakoglu. *Circuits, Interconnections and Packaging for VLSI.* Addison-Wesley.

[10] M. Becchi and P. Crowley. Dynamic thread assignment on heterogeneous multiprocessor architectures. In *Journal of Instruction-Level Parallelism (JILP)*, June, 2008.

[11] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The parsec benchmark suite: characterization and architectural implications. In *Proceedings of the 17th international conference on Parallel architectures and compilation techniques*, PACT '08, pages 72–81, 2008.

[12] K. A. Bowman, J. W. Tschanz, Nam S. Kim, J. C. Lee, C. B. Wilkerson, S. L. L. Lu, T. Karnik, and V. K. De. Energy-efficient and metastability-immune timing-error detection and instruction-replay-based recovery circuits for dynamic-variation tolerance. In *Solid-State Circuits Conference. ISSCC 2008. Digest of Technical Papers. IEEE International In Solid-State Circuits Conference.*

[13] Keith A. Bowman, James W. Tschanz, Shih-Lien Lu, Paolo A. Aseron, Muhammad M. Khellah, Arijit Raychowdhury, Bibiche M. Geuskens, Carlos Tokunaga, Chris Wilkerson, Tanay Karnik, and Vivek K. De. A 45 nm

resilient microprocessor core for dynamic variation tolerance. *J. Solid-State Circuits*, 46(1), 2011.

[14] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th annual international symposium on Computer architecture*, ISCA '00, pages 83–94, 2000.

[15] Hung-Ming Chen, Li-Da Huang, I-Min Liu, and M. D.F. Wong. Simultaneous power supply planning and noise avoidance in floorplan design. *Trans. Comp.-Aided Des. Integ. Cir. Sys.*, 24(4):578–587, November 2006.

[16] J. Chen and L.K. John. Efcient program scheduling for heterogeneous multi-core processors. In *Proc. of 46th Design Automation Conference (DAC)*, July, 2009.

[17] Y. Chou, B. Fahs, and S. Abraham. Microarchitecture optimizations for exploiting memory-level parallelism. In *Proceedings of the 31st annual international symposium on Computer architecture*, ISCA '04, 2004.

[18] K. V. Craeynest, A. Jaleel, L. Eeckhout, .P Narvaez, and J.S Emer. Scheduling heterogeneous multi-cores through performance impact estimation (pie). In *Proc. of ISCA*, 2013.

[19] W. El-Essawy and D. Albonesi. Mitigating inductive noise in smt processors. In *Proceedings of the 2004 International Symposium on Low Power*

Electronics and Design, ISLPED '04, 2004.

[20] D. Ernst, Nam Sung Kim, S. Das, S. Pant, R. Rao, Toan Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Proc. of Micro*, 2003.

[21] S. Ghiasi, T. Keller, and F. Rawson. Scheduling for heterogeneous processors in server systems. In *Proc. of the Second Conference on Computing Frontiers (CF)*, May, 2005.

[22] Soraya Ghiasi, Tom Keller, and Freeman Rawson. Scheduling for heterogeneous processors in server systems. In *Proceedings of the 2nd conference on Computing frontiers*, CF '05, pages 199–210, 2005.

[23] B. Greskamp, L. Wan, U. R. Karpuzcu, J. J. Cook, J. Torrellas, D. Chen, and C. B. Zilles. Blueshift: Designing processors for timing speculation from the ground up. In *HPCA*, pages 213–224, 2009.

[24] M. S. Gupta, K. K. Rangan, M. D. Smith, G. Y. Wei, and D. Brooks. Decor: A delayed commit and rollback mechanism for handling inductive noise in processors. In *HPCA'08*, pages 381–392, 2008.

[25] Meeta S. Gupta, Jarod L. Oatley, Russ Joseph, Gu-Yeon Wei, and David M. Brooks. Understanding voltage variations in chip multiprocessors using a distributed power-delivery network. In *Proceedings of the conference on Design, automation and test in Europe*, DATE '07, pages 624–629, 2007.

[26] Meeta S. Gupta, Vijay Janapa Reddi, Glenn Holloway, Gu-Yeon Wei, and David M. Brooks. An event-guided approach to reducing voltage noise in processors. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '09, pages 160–165, 2009.

[27] M.S. Gupta, K.K. Rangan, M.D. Smith, G. Y. Wei, and D. Brooks. Towards a software approach to mitigate voltage emergencies. In *Proc. of ISLPED*, 2007.

[28] Kim Hazelwood and David Brooks. Eliminating voltage emergencies via microarchitectural voltage control feedback and dynamic optimization. In *Proceedings of the 2004 International symposium on Low power electronics and design*, ISLPED '04, pages 326–331, 2004.

[29] N. James, P. Restle, J. Friedrich, B. Huott, and B. McCredie. Comparison of split-versus connected-core supplies in the power6 microprocessor. In *Solid-State Circuits Conference, 2007. ISSCC 2007. Digest of Technical Papers. IEEE International*, pages 298–604, 2007.

[30] J. A. Kahle, M. N. Day, H. P. Hofstee, C. R. Johns, T. R. Maeurer, and D. Shippy. Introduction to the cell multiprocessor. *IBM J. Res. Dev.*, 49(4/5):589–604, July 2005.

[31] Y. Kim, L. K. John, S. Pant, S. Manne, M. J. Schulte, W. L. Bircher, and M. S. S. Govindan. Audit: Stress testing the automatic way. In *MICRO*, pages 212–223, 2012.

[32] D. Koufaty, D. Reddy, and S. Hahn. Bias scheduling in heterogeneous multi-core architectures. In *Proc. of the European Conference on Computer Systems (EuroSys)*, April, 2010.

[33] David Koufaty, Dheeraj Reddy, and Scott Hahn. Bias scheduling in heterogeneous multi-core architectures. In *Proceedings of the 5th European conference on Computer systems*, EuroSys '10, pages 125–138, 2010.

[34] M. Kruijf, S. Nomura, and K. Sankaralingam. Relax: An architectural framework for software recovery of hardware faults. In *Proceedings of the 37th International Symposium on Computer Architecture (ISCA)*, 2010.

[35] R. Kumar, K. I. Farkas, N. P. Jouppi, P. Ranganathan, and D. M. Tullsen. Single-isa heterogeneous multi-core architectures: The potential for processor power reduction. In *Proceedings of the 36th annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 36, pages 81–, 2003.

[36] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *42st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO-42 2009), December 12-16, 2009, New York, New York, USA*, pages 469–480, 2009.

[37] Tong Li, Paul Brett, Rob C. Knauerhase, David A. Koufaty, Dheeraj

Reddy, and Scott Hahn. Operating system support for overlapping-isa heterogeneous multi-core architectures. In *HPCA*, pages 1–12, 2010.

[38] T.N. Miller, R. Thomas, P. Xiang, and R. Teodorescu. Vrsync: Characterizing and eliminating synchronization-induced voltage emergencies in many-core processors. In *Proc. of ISCA*, 2012.

[39] F. Mohamood, M. B. Healy, Sung Kyu Lim, and H.-H. S. Lee. Noise-direct: A technique for power supply noise aware floorplanning using microarchitecture profiling. In *Proceedings of the 2007 Asia and South Pacific Design Automation Conference*, ASP-DAC '07, pages 786–791, 2007.

[40] A. Patel, F. Afram, S. Chen, and K. Ghose. MARSSx86: A Full System Simulator for x86 CPUs. In *Design Automation Conference 2011 (DAC'11)*, 2011.

[41] F. J. Pollack. New microarchitecture challenges in the coming generations of cmos process technologies. In *MICRO*, pages 2–, 1999.

[42] M. D. Powell, A. Biswas, S. Gupta, and S. S. Mukherjee. Architectural core salvaging in a multi-core processor for hard-error tolerance. In *ISCA*, pages 93–104, 2009.

[43] M. D. Powell and T. N. Vijaykumar. Exploiting resonant behavior to reduce inductive noise. In *in Intl Symp. on Computer Architecture*, pages 288–299. Press, 2004.

[44] M.D. Powell and T.N. Vijaykumar. Pipeline muffling and a priori current ramping: architectural techniques to reduce high-frequency inductive noise. In *Proc. of ISPLED*, 2003.

[45] V. J. Reddi, M. S. Gupta, G. H. Holloway, G. Y. Wei, M. D. Smith, and D. Brooks. Voltage emergency prediction: Using signatures to reduce operating margins. In *HPCA'09*, pages 18–29, 2009.

[46] V.J. Reddi, S. Kanev, W. Kim, S. Campanoni, M.D. Smith, G. Y. Wei, and D. Brooks. Voltage smoothing: Characterizing and mitigating voltage noise in production processors via software-guided thread scheduling. In *MICRO*, 2011.

[47] S. Thoziyoor, J. Ho Ahn, M. Monchiero, J. B. Brockman, and N. P. Jouppi. A comprehensive memory modeling tool and its application to the design and analysis of future memory hierarchies. In *Proceedings of the 35th Annual International Symposium on Computer Architecture*, ISCA '08, pages 51–62, 2008.

[48] M. Toburen. Power analysis and instruction scheduling for reduced di/dt in the execution core of high-performance microprocessors. *Master's Thesis, NC State University, USA*.

[49] Kenzo Van Craeynest, Aamer Jaleel, Lieven Eeckhout, Paolo Narvaez, and Joel Emer. Scheduling heterogeneous multi-cores through performance impact estimation (pie). In *Proceedings of the 39th Annual Inter-*

national Symposium on Computer Architecture, ISCA '12, pages 213–224, 2012.

[50] M. T. Yourst. Ptlsim: A cycle accurate full system x86-64 microarchitectural simulator. In *2007 IEEE International Symposium on Performance Analysis of Systems and Software, April 25-27, 2007, an Jose, California, USA, Proceedings*, pages 23–34. IEEE Computer Society, 2007.

[51] Han-Saem Yun and Jihong Kim. Power-aware modulo scheduling for high-performance vliw processors. In *Proceedings of the 2001 International symposium on Low power electronics and design*, ISLPED '01, pages 40–45, 2001.