

INVESTIGATING THE EFFECTS OF DECOUPLING CACHE AND CORE SPEED ON
POWER, THROUGHPUT, AND ENERGY CONSUMPTION

DAVID DANIEL PRUITT

Doctoral Program in Computer Science

APPROVED:

Eric Freudenthal, Ph.D., Chair

Shirley Moore, Ph.D.

Art Duval, Ph.D.

Luc Longpre, Ph.D.

Stephen L. Crites, Jr., Ph.D.
Dean of the Graduate School

Copyright ©

by

David Pruitt

2022

Dedication

To the friends that made this process possible

“It is possible to commit no errors and still lose. That is not a weakness. That is life.”

PREVIEW

INVESTIGATING THE EFFECTS OF DECOUPLING CACHE AND CORE SPEED ON
POWER, THROUGHPUT, AND ENERGY CONSUMPTION
OF CPUS

by

David Daniel Pruitt BS CS MS CS

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at El Paso
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

Department of Computer Science
THE UNIVERSITY OF TEXAS AT EL PASO
August 2022

Acknowledgements

I would like to thank everyone that assisted during my long and arduous journey in completing this project. I'm sure I'll miss some names here. I would like to thank the various members of the Robust Autonomic systems group for their support over the years. Gabe Arellano for sharing writing tips. Edward Dragone and Edward Hudgins for their help in various technical areas. Adrian Veliz for forging the path and advising of the landmines that appear. Daniel Cervantes for allowing me to bounce ideas off him and his support when I was trying to teach. I would also like to thank the unofficial members, David Reyes and Oscar Veliz for answering the stupid questions I had with the appropriate vigor.

I would also like to thank Eric Freudenthal and Salamah Salamah for providing assistance, useful advice, and direction when needed that made this journey possible. I want to thank Dr Shirley Moore for working through ideas, spotting problems, and providing helpful suggestions. I want to thank Dr Art Duval and Luc Longpre for taking the time out of their busy schedules to read over drafts and provide feedback.

“You ever figure procrastination is your brain’s way of stopping you from making a terrible mistake? Yeah... Me too.” Cayde-6

Abstract

A variety of computer systems from HPC to mobile systems are power limited and performance sensitive. These systems use very similar components at different scales. Dynamic Voltage and Frequency Scaling (DVFS) features enable modulation of CPU performance and efficiency characteristics to power, energy and timing requirements.

Programs have a variety of computational characteristics. If a CPU subsystem substantially limits a particular program's execution progress, that program's throughput will vary proportionally with the subsystem's clock frequency. In contrast, if a CPU subsystem does not substantially limit throughput, the impact of a change in its clock frequency will result in a diminimus change in a program's execution time.

Dynamic Voltage and Frequency Scaling (DVFS) power domains commonly encompass entire cores and their associated caches. This work indicates that moderate energy efficiency gains may be attainable for some programs if limiting and non-limiting subsystems' (D)VFS domains are decoupled. This decoupling enables tuning of their relative performance to application characteristics. Widely used simulation and modeling tools were extended to support this exploratory research.

Table of Contents

Dedication	iii
Acknowledgements	v
Abstract	vi
Table of Contents	vii
List of Tables	ix
List of Figures	x
List of Equations	xi
Chapter 1: Introduction	1
1.1 Motivation	1
1.2 Strategy	2
1.3 Intuition	2
1.4 Results	3
1.5 Contributions	4
Chapter 2: CPU Power and Energy Consumption	5
2.1 Introduction	5
2.2 Relationship of frequency, Throughput, and Efficiency	5
2.3 Device Power Consumption	6
2.3.1 Active Power	6
2.3.2 Background Power	7
2.5 Active Power Proportion	7
2.6 Power Domains and Decoupling	8
2.7 Other Approaches	8
Chapter 3: Methodology	10
3.1 Overview	10
3.2 Benchmark Kernels	12
3.3 Hardware Event Measurement	13
3.4 Power Modelling	14
3.4.1 Walker's Model	15

3.4.2 Event Selection	15
3.4.4 Validation Study	17
3.5 Gem5	17
Gem5 Validation	19
Gem5 Modifications to Support Decoupling	20
Chapter 4 Experimental Results and Analysis	22
4.1 Introduction	22
4.2 L2 Limited Study	25
4.3 Core Limited Study	29
4.4 Balanced Study	33
4.5 Synopsis and Potential Extensions of this Work	35
4.5.1 Open Questions	36
4.5.1.1 Slower clock frequencies	36
4.5.1.2 Decoupling Additional Subsystems	36
4.5.1.3 Dynamic Optimization	36
4.5.1.4 Thread Interactions	36
4.5.1.5 Speculative Execution	37
References	38
Appendix A Benchmarks	41
Appendix B Gem5 Modifications	42
Appendix C Power Measurement	43
Vita	44

List of Tables

Table 3.1 Benchmark Kernels.....	13
Table 3.2 Power Model Events and Coefficients.....	17
Table 3.3 Power Model Validation Results	17
Table 3.4 Gem5 CPU Model Differences.....	18
Table 3.5 Gem5 Memory Model Differences.....	19
Table 3.6 L2 Cache Performance Comparison, TX1-A model vs Actual	19
Table 3.7 L2 Cache Performance Comparison, TX1-Final model vs Actual.....	20
Table 4.1 Most Efficient Configuration and Energy Summary	22
Table 4.2 Frequency Notation.....	23
Table 4.3 Benchmark Speedup	25
Table A.1 Benchmark Sources	41
Table A.2 Benchmark configurations	41

List of Figures

Figure 3.1 Methodology.....	11
Figure 4.1 Power and Throughput of L2 Applications.....	26
Figure 4.2 Energy Breakdown of L2 Applications.....	27
Figure 4.3 L2 Energy Contour Plot L2 Applications Unicore.....	28
Figure 4.4 L2 Energy Contour Plot L2 Applications Multicore.....	29
Figure 4.5 Power and Throughput of Core Applications.....	31
Figure 4.6 Energy Breakdown of Core Applications.....	31
Figure 4.7 Energy Contour Plot Core Applications Unicore.....	32
Figure 4.8 Energy Contour Plot Core Applications Multicore.....	32
Figure 4.9 Power and Throughput of Balances Applications.....	34
Figure 4.10 Energy Breakdown of Balanced Applications.....	34
Figure 4.11 Energy Contour Plot Balanced Applications Unicore.....	34
Figure 4.12 Energy Contour Plot Balance Applications Multicore.....	35

List of Equations

Equation 3.1 Power Consumption Model	15
--	----

PREVIEW

Chapter 1: Introduction

1.1 MOTIVATION

Modern computer systems from HPC to mobile systems are power limited and performance sensitive. These systems use very similar components at different scales [1].

Furthermore, application-specific CPUs may be optimized for energy efficiency. For example, CPUs and their subsystems can be selected and clocked to provide required performance. Even instruction sets are optimized to satisfy embedded systems' power, performance, and energy requirements.

(D)VFS features enable modulation of CPU performance and efficiency characteristics to better match system power and energy limitations, and application constraints. Higher frequency provides greater throughput at the cost of increased power consumption. When sufficient parallelism is exposed by the application and available from a system, optimization of efficiency via DVFS can also maximize throughput. Energy consumed by computing components is classified as either active or background. Active energy consumption is due to gate state changes on the computational data or control path. Background energy is consumed by leakage current and activity required to keep the system “alive.”

The throughput of a CPU-limited computation is limited by the rate of gate transitions. The energy required for a gate transition activity is monotonic with voltage. Voltage is monotonic with frequency. Therefore, energy required to actively compute a result is monotonic with frequency. And therefore, if all energy went to active gate transitions, a computation would be most efficient at lowest available frequency.

Background energy consumption is the integral of background power consumption over the time required to complete a computation. Background power varies only 20% over the range