Power Struggles: Revisiting the RISC vs. CISC Debate on Contemporary ARM and x86 Architectures

by Ernily Blern, laikrishnan Menon, and Karthikeyan Sankaralingarn

Danilo Dominguez Perez
danilo0@iastate.edu

December 8, 2016

Iowa State University

Overview

- Presentation of the Paper
  - Background
  - Infrastructure
  - Methodology
  - Results
  - Conclusion
- Analysis of the Paper

Background

Motivation

- Today's computing landscape is starting to be shaped by smartphones and tablets
- Previously, RISC architectures governed over small devices whereas CISC architectures ruled over servers and desktops
- Nowadays, both architectures are ”trespassing” each other areas

Previous Work

- Previous studies compare performance of RISC vs. CISC ISAs finding simple RISC instructions had better performance (Bhandarkar & Clark, 1991)
- Later studies conclude both ISAs have similar performance if they apply aggressive ILP techniques (Isen, John, & John, 2009)
- **Informal studies** have suggest the power overheads of CISC ISAs are intractable

This Study

- Revisit the RISC vs. CISC debate from a power and energy perspective
- Previous comparisons focused on performance
- Show what is the role of the ISA and make a case for revisiting the RISC vs. CISC question

RISC vs. CISC ISA

	RISC	CISC
Format	Fixed length instructions	Variable length instructions
Operations	Single cycle operations	Multi-cycle operations
Operands	Few addressing modes	Many addressing modes

*** CISC instructions split into RISC-like micro-ops and modern compilers pick mostly RISC-like instructions**

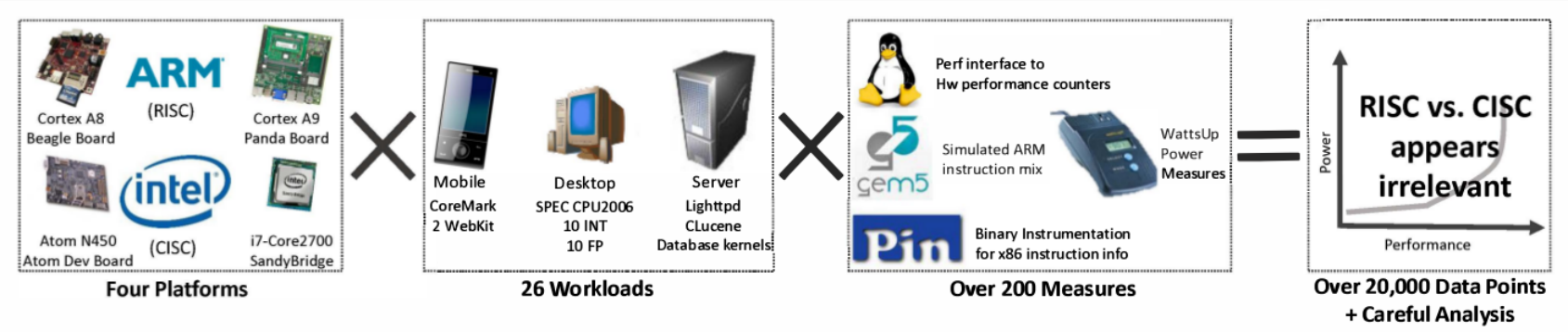
Infrastructure

Infrastructure

- Find implementations with similar microarchitecture and I/O and memory subsystems
- Use workloads for servers, desktops and mobile
- Linux 2.6 LTS kernel
- gcc 4.4 target independent with optimizations enabled (O3)
- *perf* tool used for performance measurements
- Watssup meter connected directed to the board for power measurements
- For instruction mix use DynamoRIO for x86 ISA and gem5 for ARM ISA

Approach

Overall Approach



Platforms

Platform Summary

	32/64b x86 ISA		ARMv7 ISA	
Architecture	Sandybridge	Atom	Cortex-A9	Cortex-A8
Processor	Core 2700	N450	OMAP4430	OMAP3530
Cores	4	1	2	1
Frequency	3.4 GHz	1.66 GHz	1 GHz	0.6 GHz
Width	4-way	2-way	2-way	2-way
Issue	OoO	In Order	OoO	In Order
L1 Data	32 KB	24 KB	32 KB	16 KB
L1 Inst	32 KB	32 KB	32 KB	16 KB
L2	256 KB/core	512 KB	1 MB/chip	256 KB
L3	8 MB/chip	—	—	—
Memory	16 GB	1 GB	1 GB	256 MB
SIMD	AVX	SSE	NEON	NEON
Area	216 mm^2	66 mm^2	70 mm^2	60 mm^2
Tech Node	32 nm	45 nm	45 nm	65 nm
Platform	Desktop	Dev Board	Pandaboard	Beagleboard
Products	Desktop	Netbook	Galaxy S-III	iPhone 4, 3GS
		Lava Xolo	Galaxy S-II	Motorola Droid

Data from TI OMAP3530, TI OMAP4430, Intel Atom N450, and Intel i7-2700 datasheets, `www.beagleboard.org` & `www.pandaboard.org`

Benchmarks

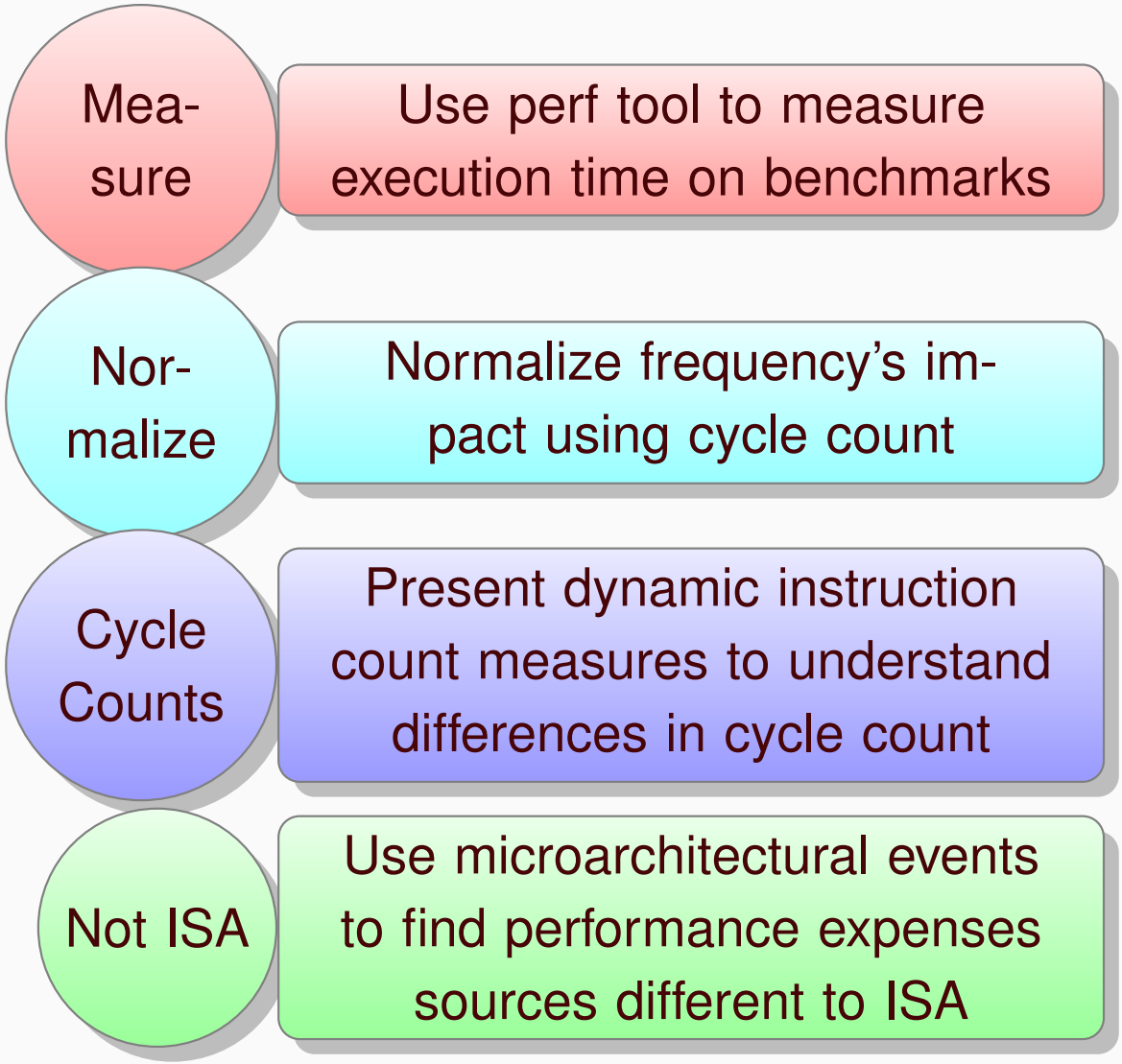
Domain	Benchmarks	Notes
Mobile client	CoreMark	Set to 4000 iterations
	WebKit	Similar to BBench
Desktop	SPECCPU2006	10 INT, 10 FP, test inputs
Server	lighttpd	Represents web-serving
	CLucene	Represents web-indexing
	Database kernels	Represents data-streaming and data-analytics

Limitations

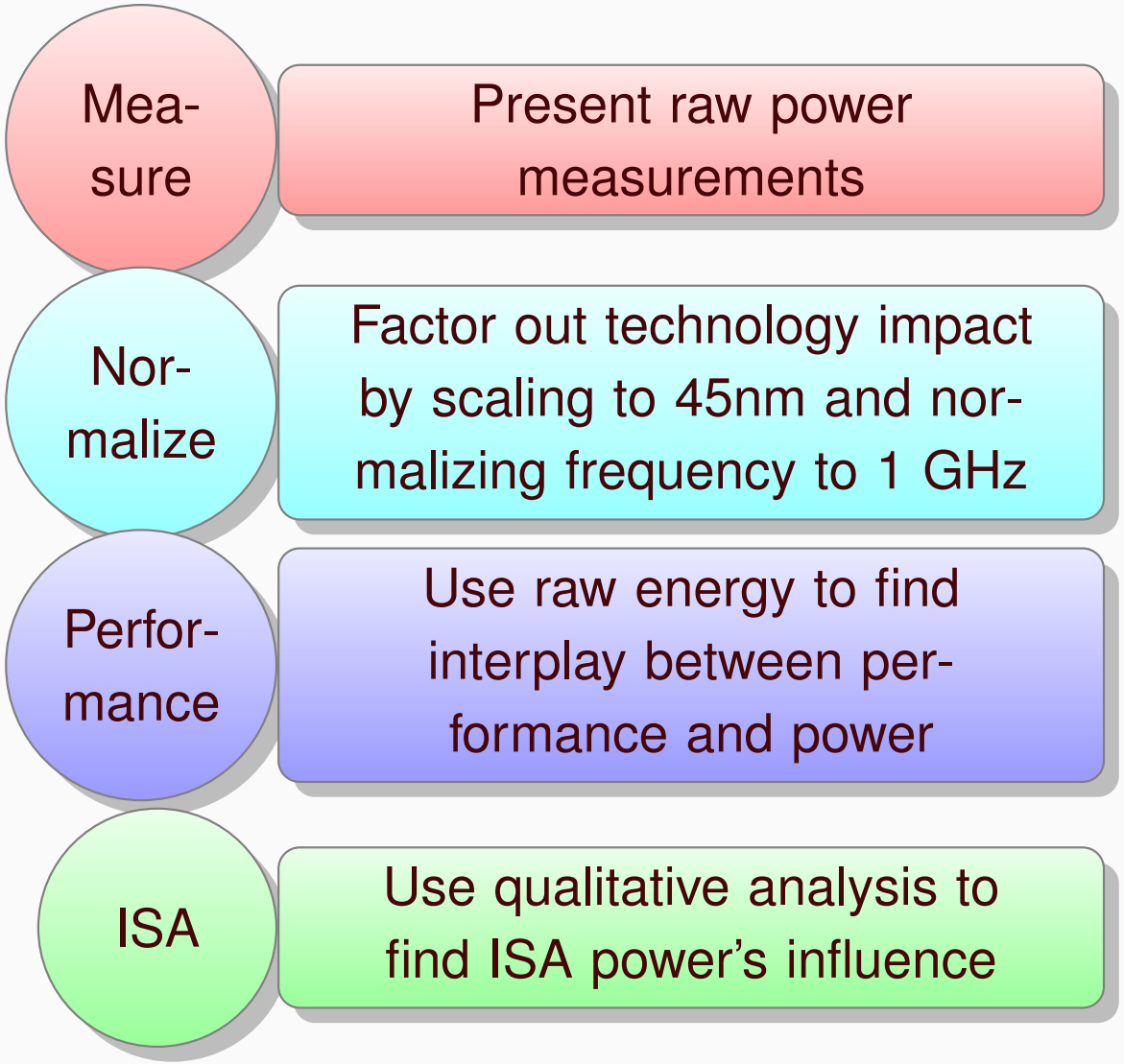
- Core: No platform uniformity across ISAs
- Tools: use gcc uniform optimizations. Specific architecture optimizations less than 10% effect

Methodology

Methodology: Performance Analysis



Methodology: Power and Energy Analysis

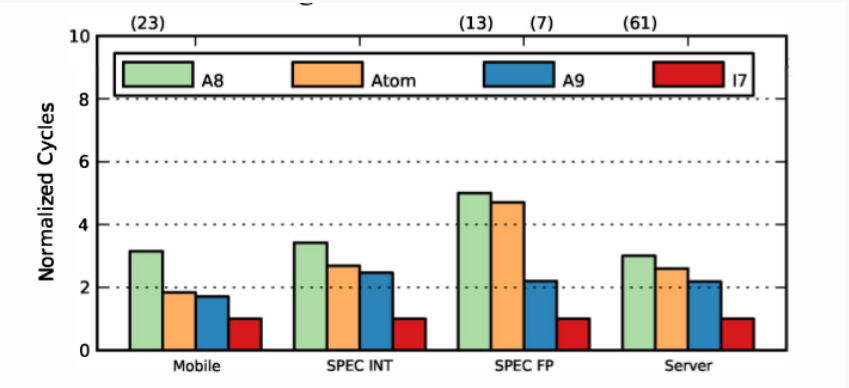


Measured Data Analysis and Findings

Results: Cycle Count

- Performance gaps of up to 1.5x from A8 to Atom
- Performance gaps of up to 2.5x from A9 to i7

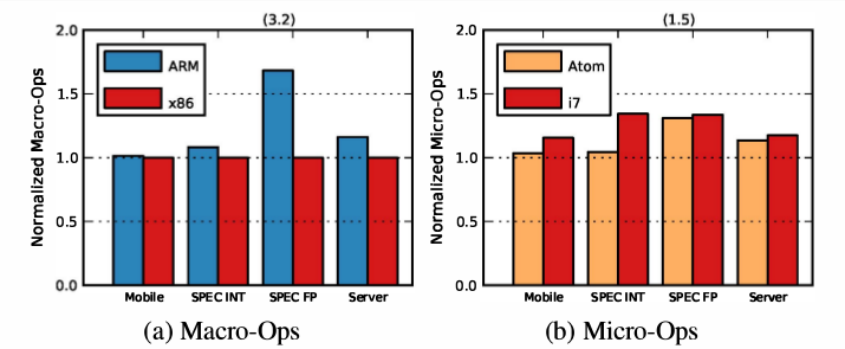
Cycle Count Normalized to i7



Results: Instruction Count

- Instruction count similar across ISAs
- gcc tends to pick similar RISC-like instructions
- CPI was less in x86 architectures: geometric mean of 3.4 for A8, 2.2 for A9, 2.1 for Atom and 0.7 for i7

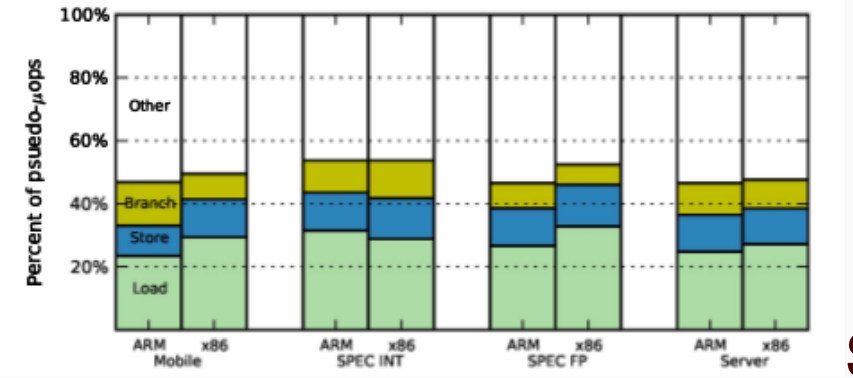
Instruction Count Normalized to i7



Results: Instruction Format and Mix

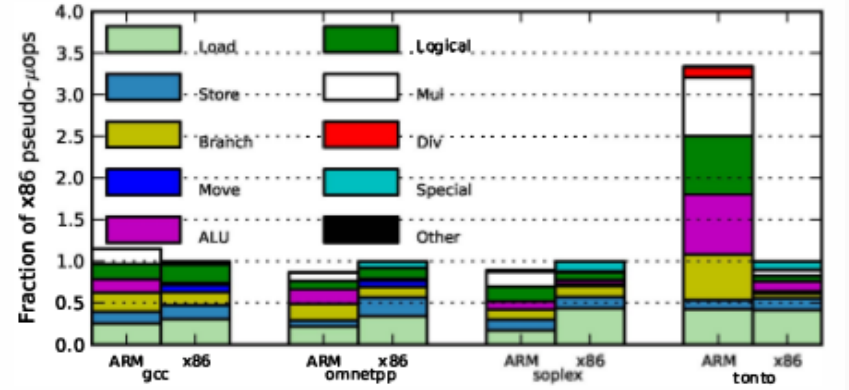
- Similar code densities in both architectures
- Fraction of loads and stores similar across ISA for all suites
- Large instruction counts for ARM are due to absence of FP instructions like `fsincon`, `fy12xpl`
- ISA effects are indistinguishable between x86 and ARM implementations

Instruction Mix



Sele

Instruction Counts



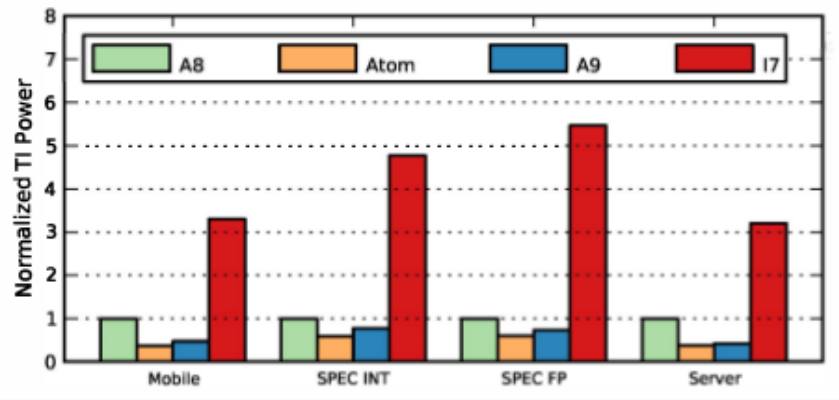
Results: Microarchitecture

- Microarchitecture has the most impact on performance
- The large cache, accurate branch prediction and bigger issue width on OoO processors (A9 and i7) help x86 to have better performance
- ISA has not considerable effect on performance

Results: Power and Energy

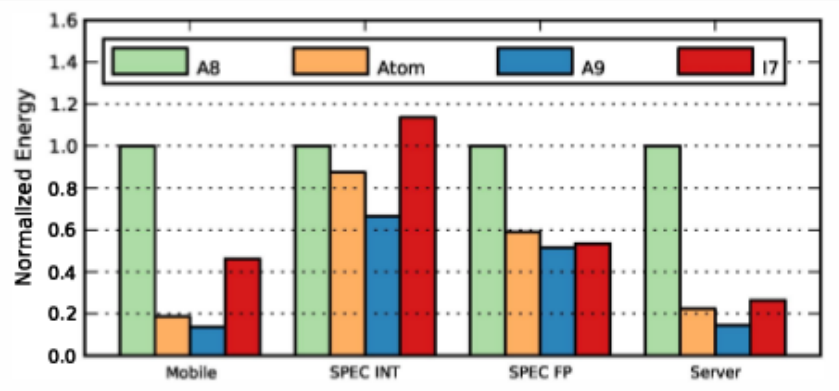
- With scaled frequency and technology, ISA is irrelevant
- i7 processors are designed to have high performance but they are not power-optimized
- Atom, on the other hand, have power consumption on par with ARM A8 and A9

Tech. Independent Avg. Power Normalized to A8



Raw

Average Energy Normalized to A8



Conclusions

- x86 CPI $<$ ARM CPI
- ISA performance effects indistinguishable between x86 and ARM
- Beyond micro-op translation, x86 ISA introduces no overheads over ARM ISA
- Choice of performance optimizations impacts power consumption more than ISA
- ISA's impact on energy is insignificant

Analysis of the Paper

- The paper was introduced as an analysis on the effect of ISAs on power and energy but end up presenting more data on performance
- A more interesting study would be to find what optimizations for performance make i7 consume more power
- Also, what factors make Atom to have performance on par of ARM A9 and be power-friendly

References

Bhandarkar, D., & Clark, D. W. (1991). Performance from architecture: comparing a risc and a cisc with similar hardware organization. In *Acm sigarch computer architecture news* (Vol. 19, pp. 310–319).

Isen, C., John, L. K., & John, E. (2009). A tale of two processors: Revisiting the risc-cisc debate. In *Spec benchmark workshop* (pp. 57–76).