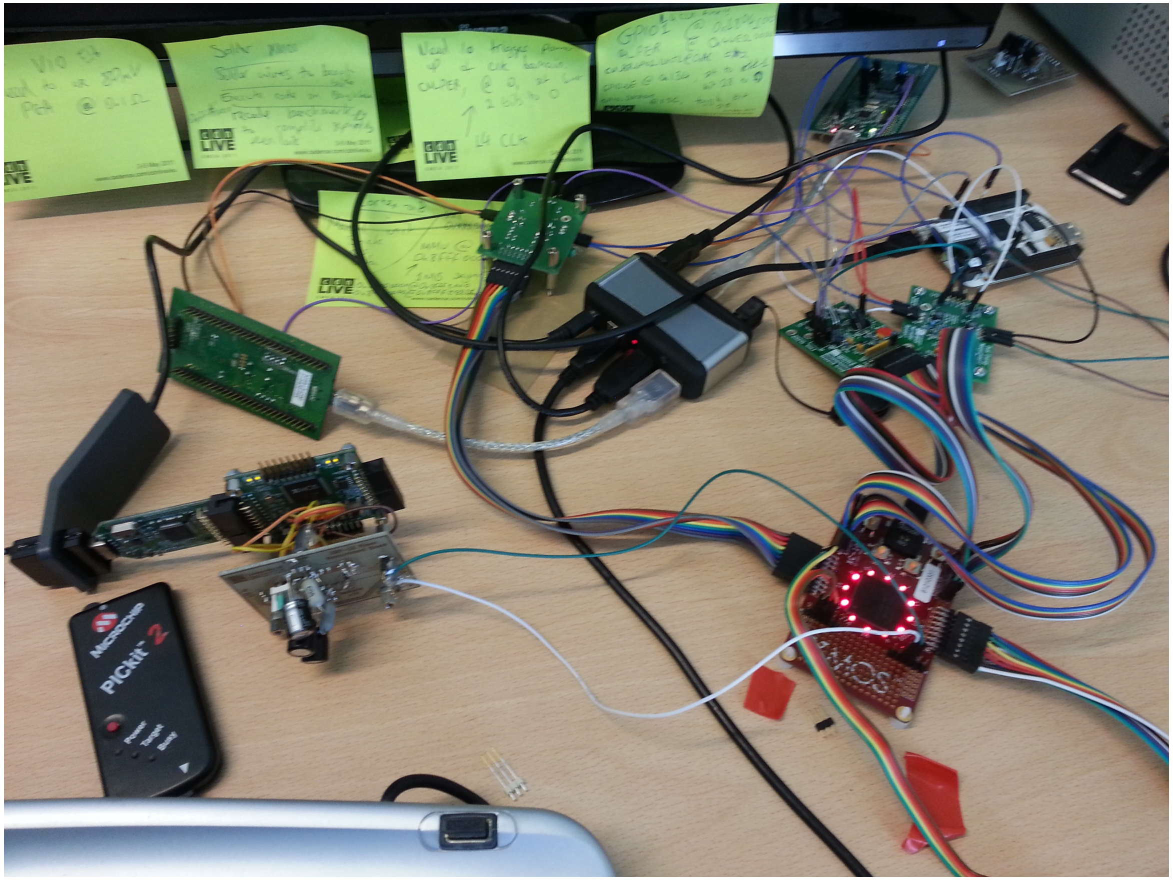
Impact of different compiler options on energy consumption



James Pallister
University of Bristol / Embecosm

Simon Hollis
University of Bristol

Jeremy Bennett
Embecosm

Motivation

- Compiler optimizations are claimed to have a large impact on software:
  - Performance
  - Energy
- No *extensive* study prior to this considering:
  - Different benchmarks
  - Many individual optimizations
  - Different platforms
- This work looks at the effect of many different optimizations across 10 benchmarks and 5 platforms.
- 238 Optimization passes covered by 150 flags
  - Huge amount of combinations

This Talk

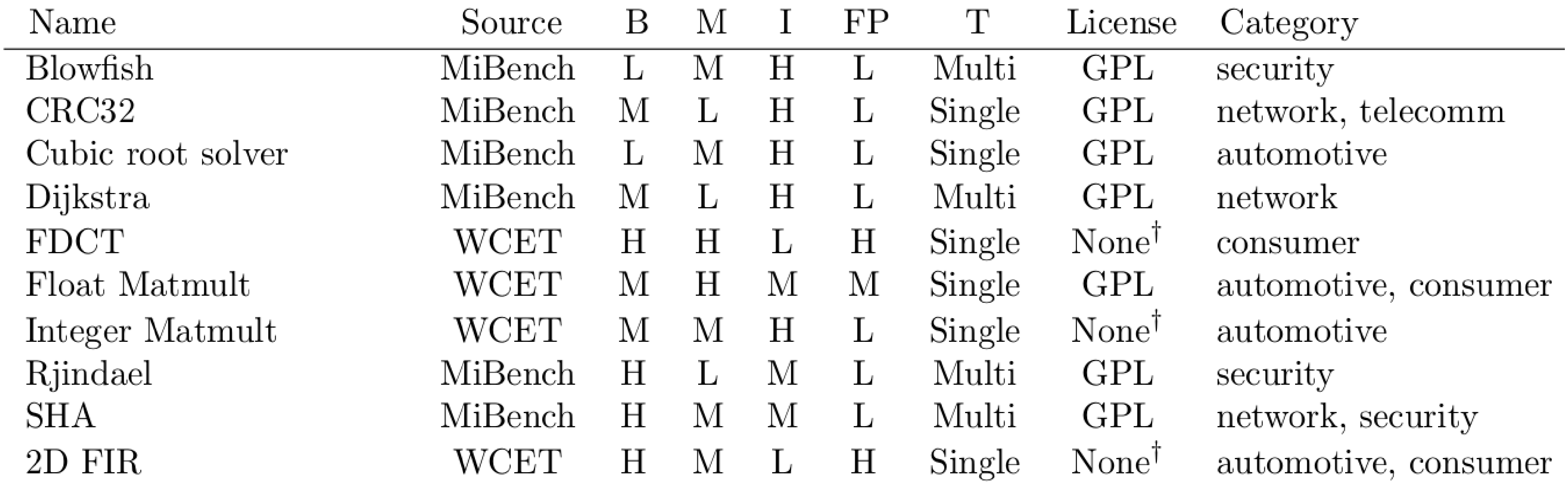
- This talk will cover:
  - Importance of benchmarks
  - How to explore 2^150 combinations of options
  - Demo
  - Correlation between time and energy
  - How to predict the effect of the optimizations
  - The best optimizations

Importance of Benchmarks

- One benchmark can't trigger all optimizations
- Perform differently on different platforms
- Need a range of benchmarks

- Broad categories to be considered for a benchmark:
  - Integer
  - Floating point
  - Branching
  - Memory

Our Benchmark List



Name	Source	B	M	I	FP	T	License	Category
Blowfish	MiBench	L	M	H	L	Multi	GPL	security
CRC32	MiBench	M	L	H	L	Single	GPL	network, telecomm
Cubic root solver	MiBench	L	M	H	L	Single	GPL	automotive
Dijkstra	MiBench	M	L	H	L	Multi	GPL	network
FDCT	WCET	H	H	L	H	Single	None†	consumer
Float Matmult	WCET	M	H	M	M	Single	GPL	automotive, consumer
Integer Matmult	WCET	M	M	H	L	Single	None†	automotive
Rjindael	MiBench	H	L	M	L	Multi	GPL	security
SHA	MiBench	H	M	M	L	Multi	GPL	network, security
2D FIR	WCET	H	M	L	H	Single	None†	automotive, consumer

Choosing the Platforms

- Range of different features in the platforms chosen
  - Pipeline Depth
  - Multi- vs Single- core
  - FPU available?
  - Caching
  - On-chip vs off-chip memory

Platforms Chosen

ARM Cortex-M0	ARM Cortex-M3	ARM Cortex-A8	XMOS L1	Adapteva Epiphany
Small memory	Small memory	Large memory	Small memory	On-chip and off-chip memory
Simple Pipeline	Simple Pipeline, with forwarding logic, etc.	Complex superscalar pipeline	Simple pipeline	Simple superscalar pipeline
		SIMD/FPU		FPU
			Multiple threads	16 cores

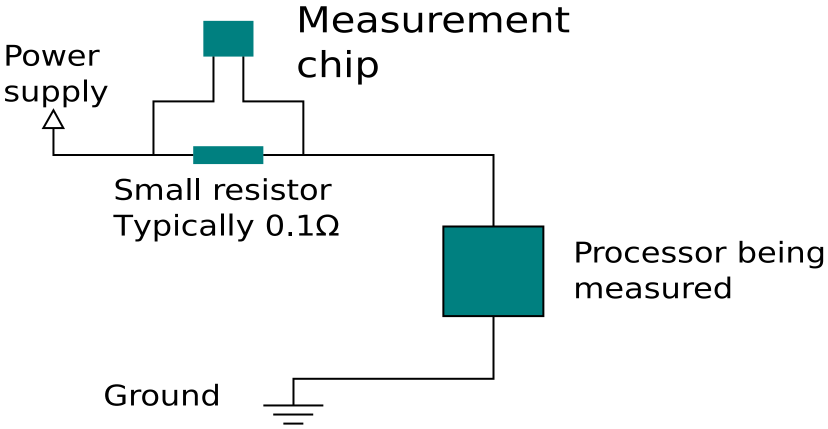
Experimental Methodology

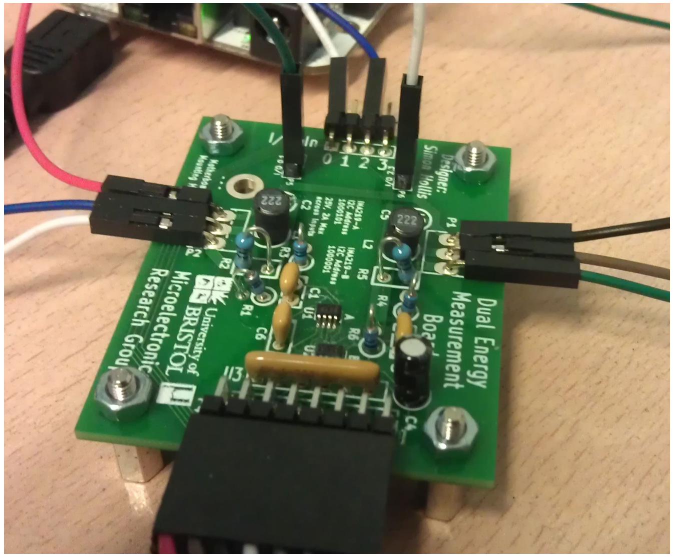
- Compiler optimizations have many non-linear interactions
- 238 optimization passes combined into 150 different options (GCC)
- 82 compiler options enabled by O3
- How to test all of these, while accounting for the interactions between optimizations?

Fractional Factorial Designs

Hardware Measurements

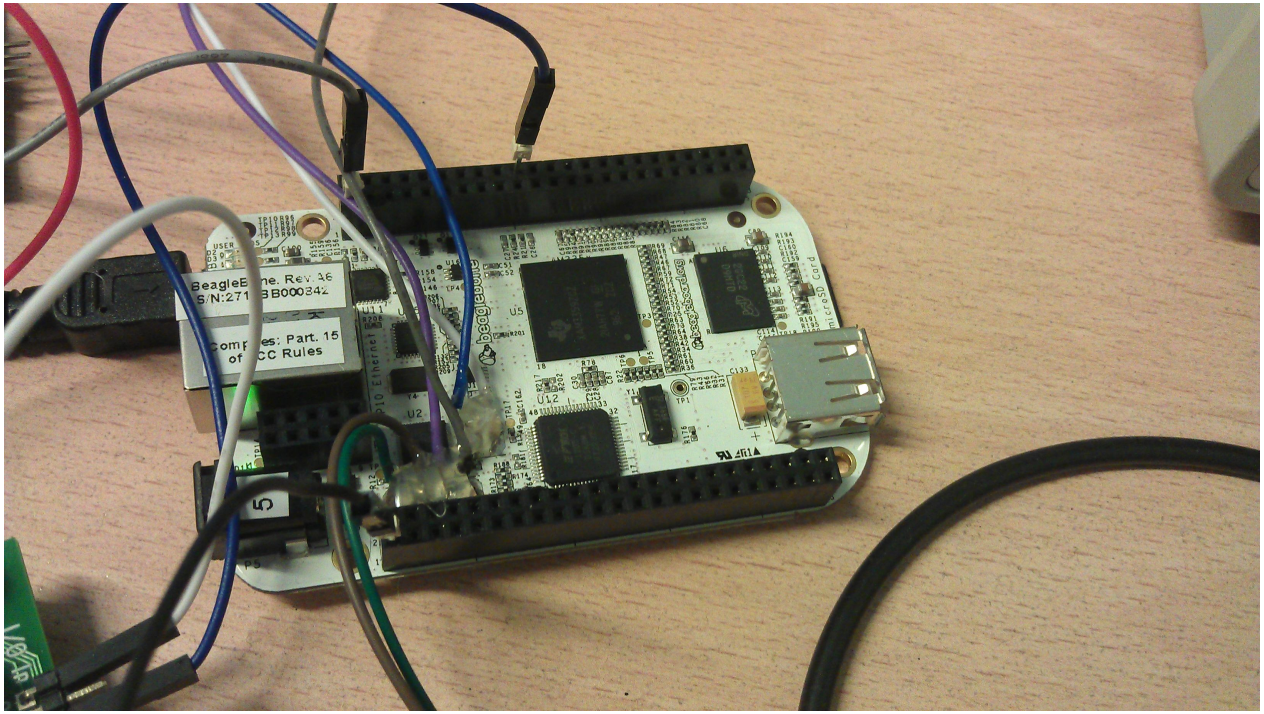
- Current, voltage and power monitor
- 10 kSamples/s
- Low noise
- XMOS board to control and timestamp measurements
- Integrate to get energy consumption



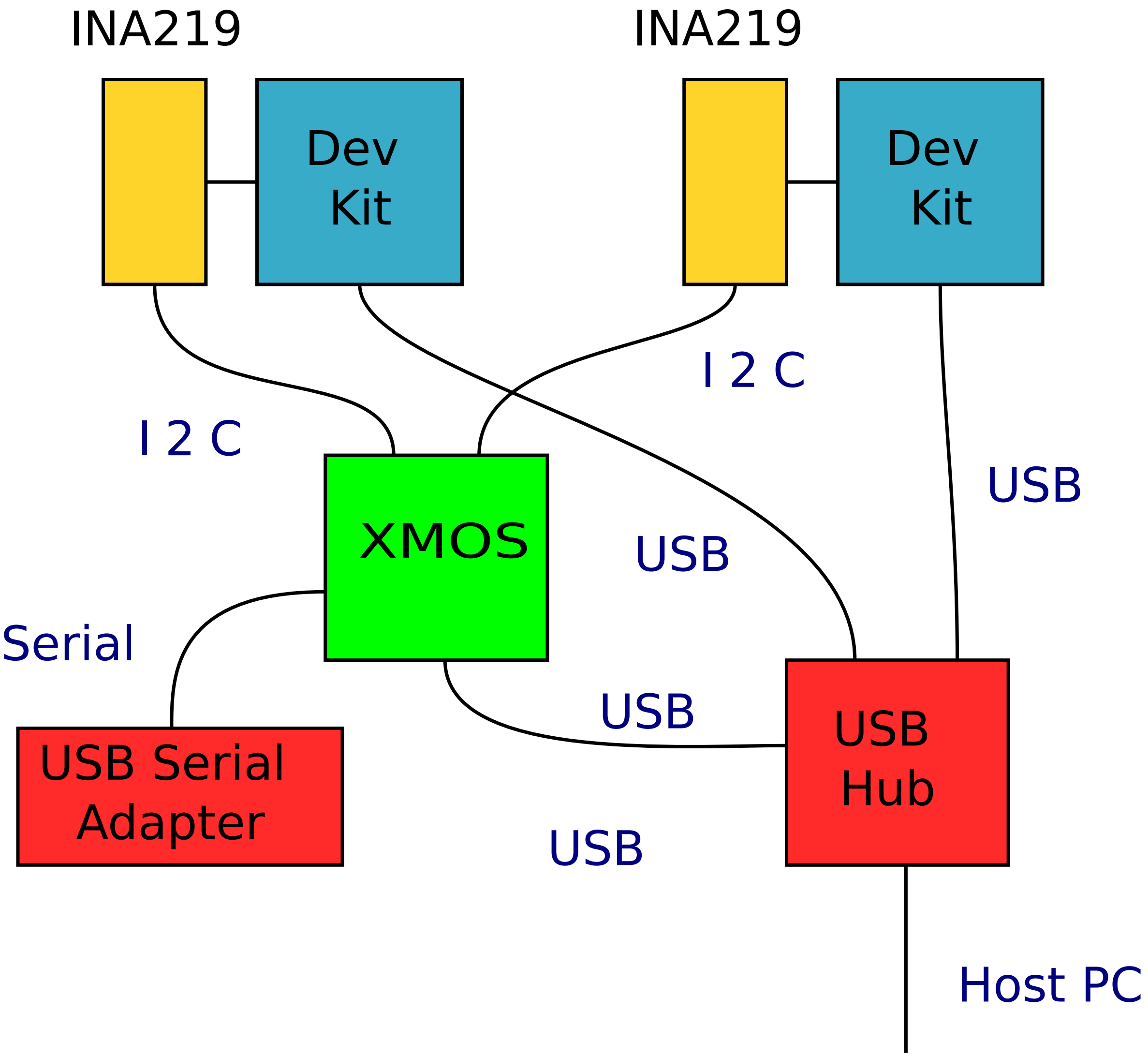


Instrumenting the Hardware

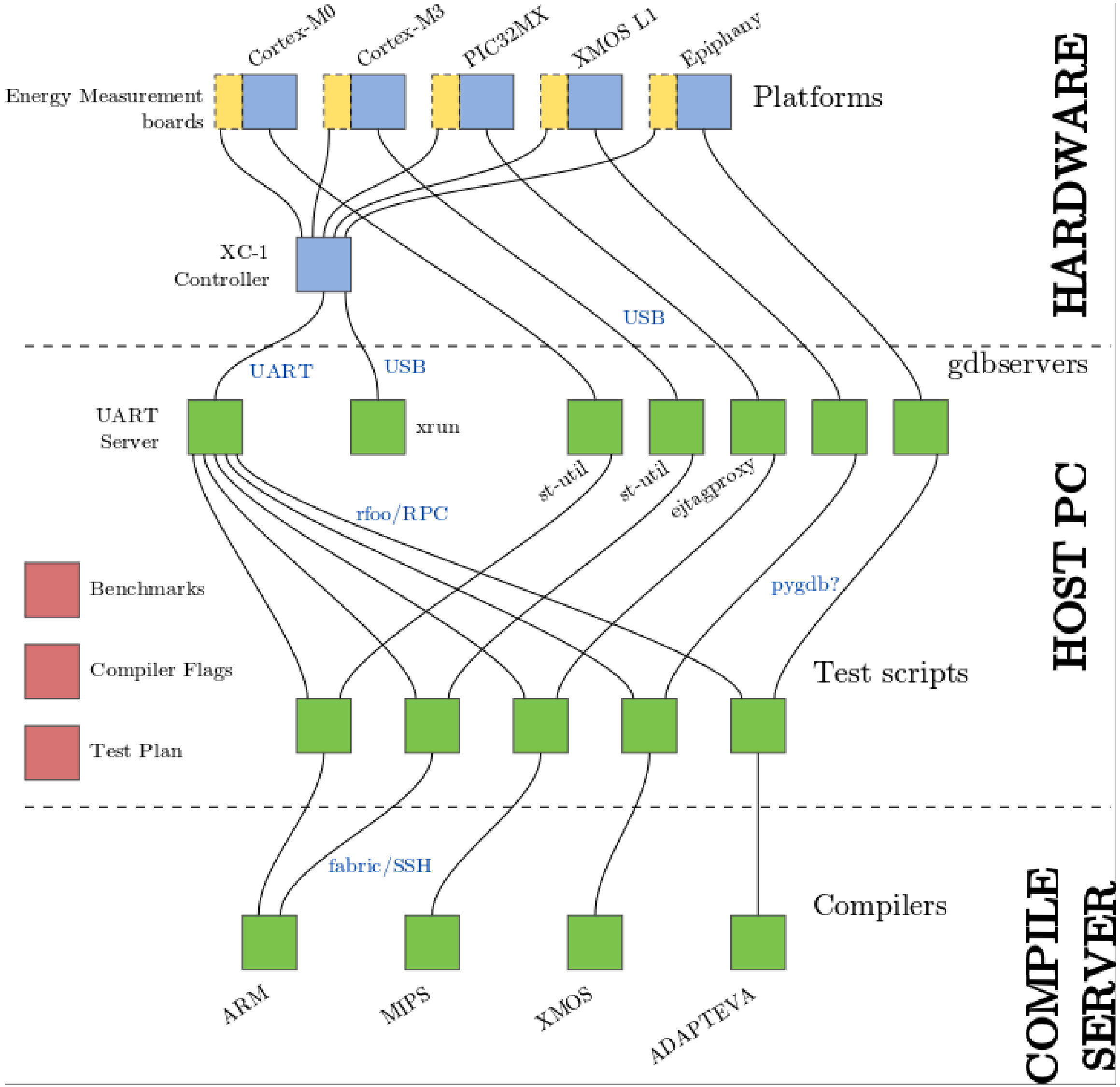
- How to attach the power measurement circuit to the hardware?
- Invasive...



Hardware


INA219
INA219
Dev Kit
Dev Kit
I2C
I2C
USB
XMOS
USB
Serial
USB
USB Serial Adapter
USB Hub
USB
Host PC

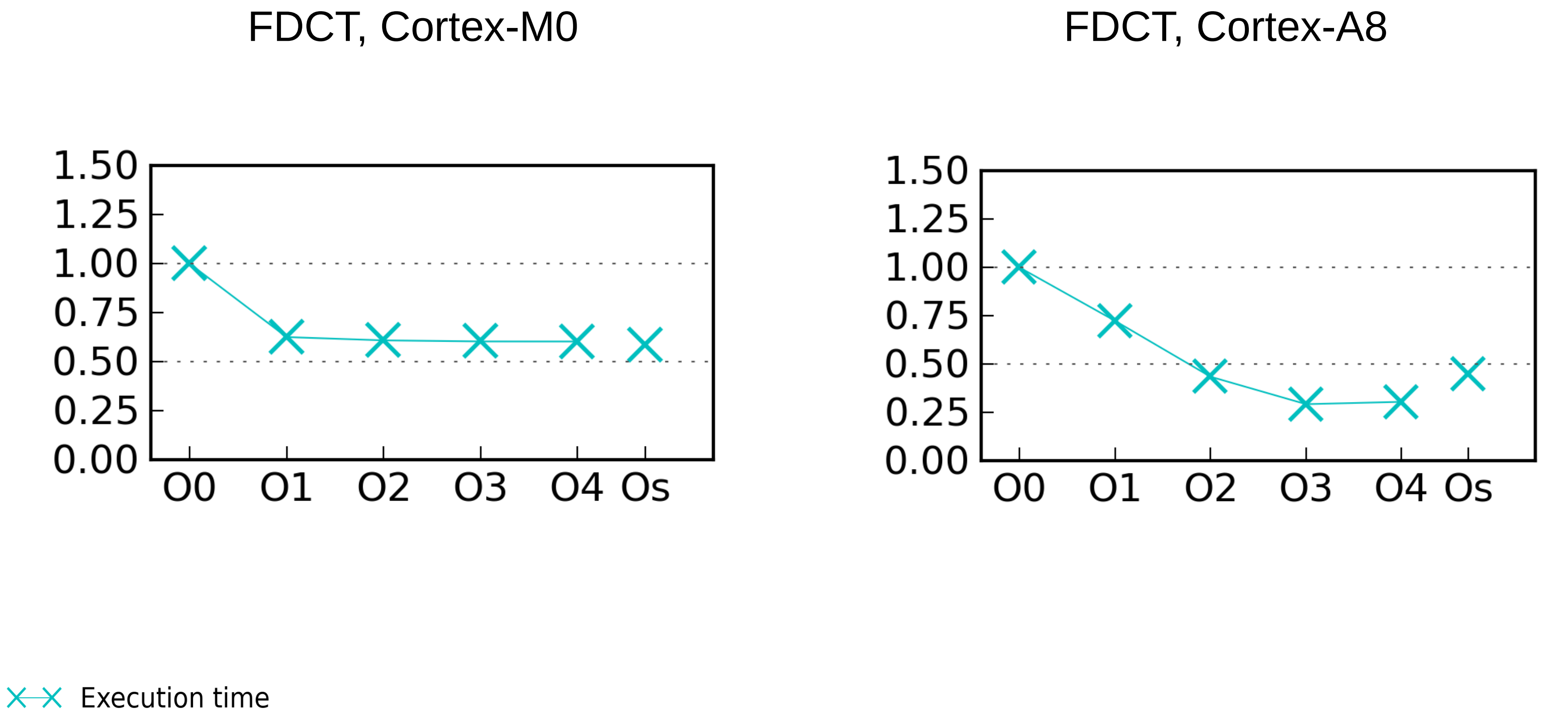
Software


Cortex-M0
Cortex-M3
PIC32MX
XMOS L1
Epiphany
Energy Measurement boards
Platforms
HARDWARE
XC-1 Controller
USB
UART
USB
gdbservers
UART Server
xrun
st-util
st-util
ejtagproxy
rfoo/RPC
pygdb?
HOST PC
Benchmarks
Compiler Flags
Test Plan
Test scripts
fabric/SSH
Compilers
ARM
MIPS
XMOS
ADAPTEVA
COMPILE SERVER

Results

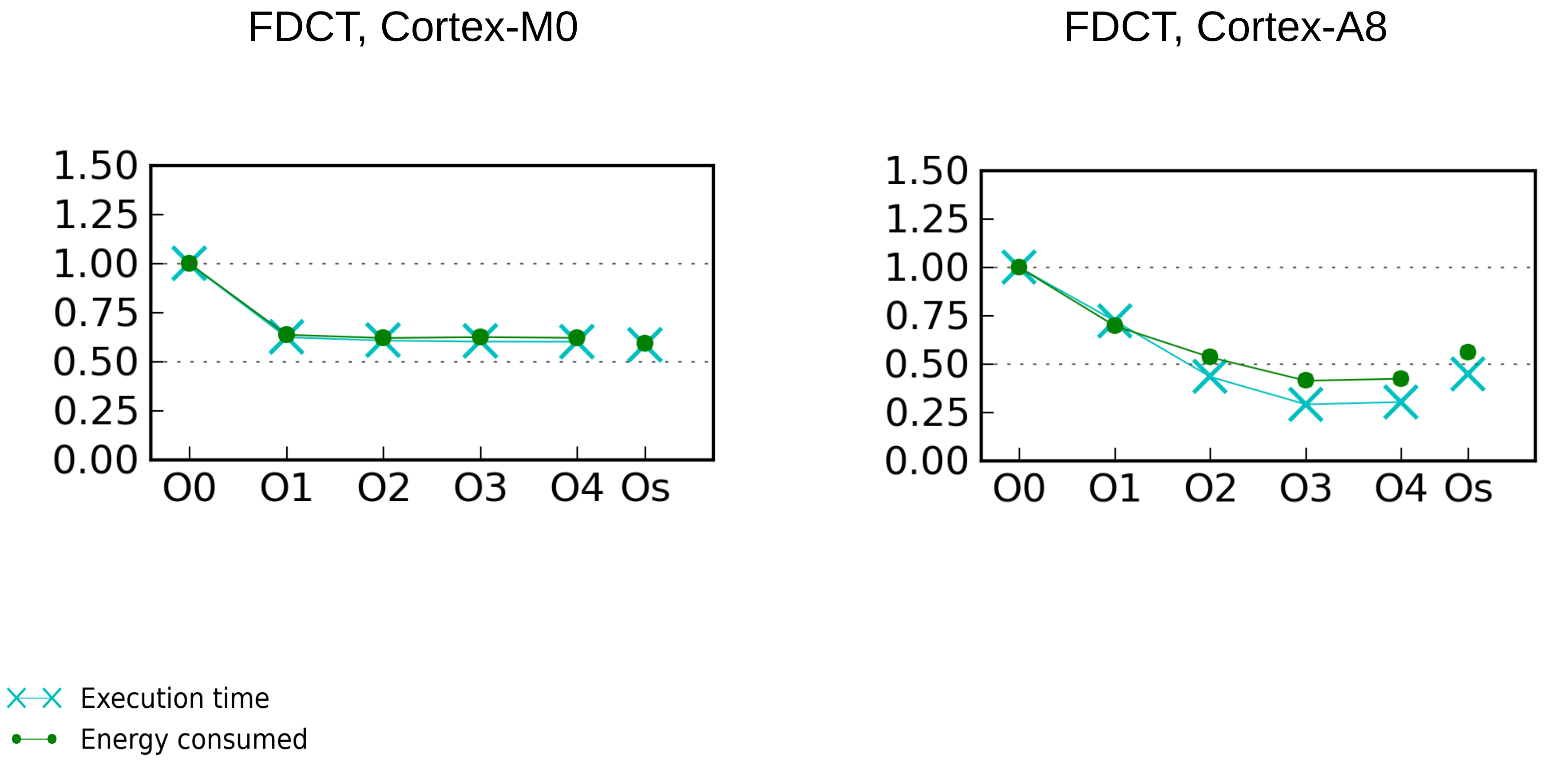
- Energy consumption ≈ Execution time
  - Generalization, not true in every case
- Optimization unpredictability
- No optimization is universally good across benchmarks and platforms

Overview

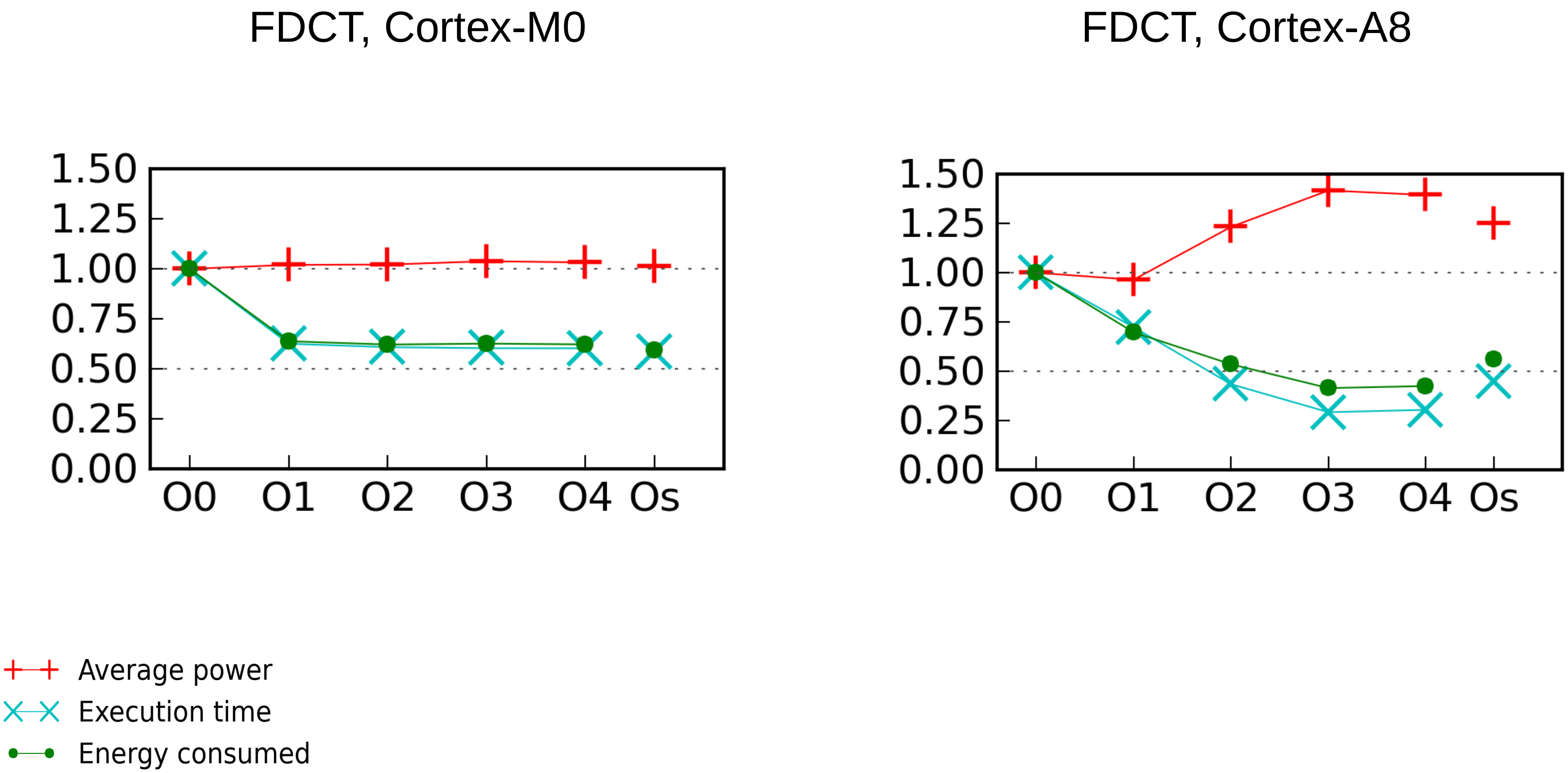

FDCT, Cortex-M0

FDCT, Cortex-A8

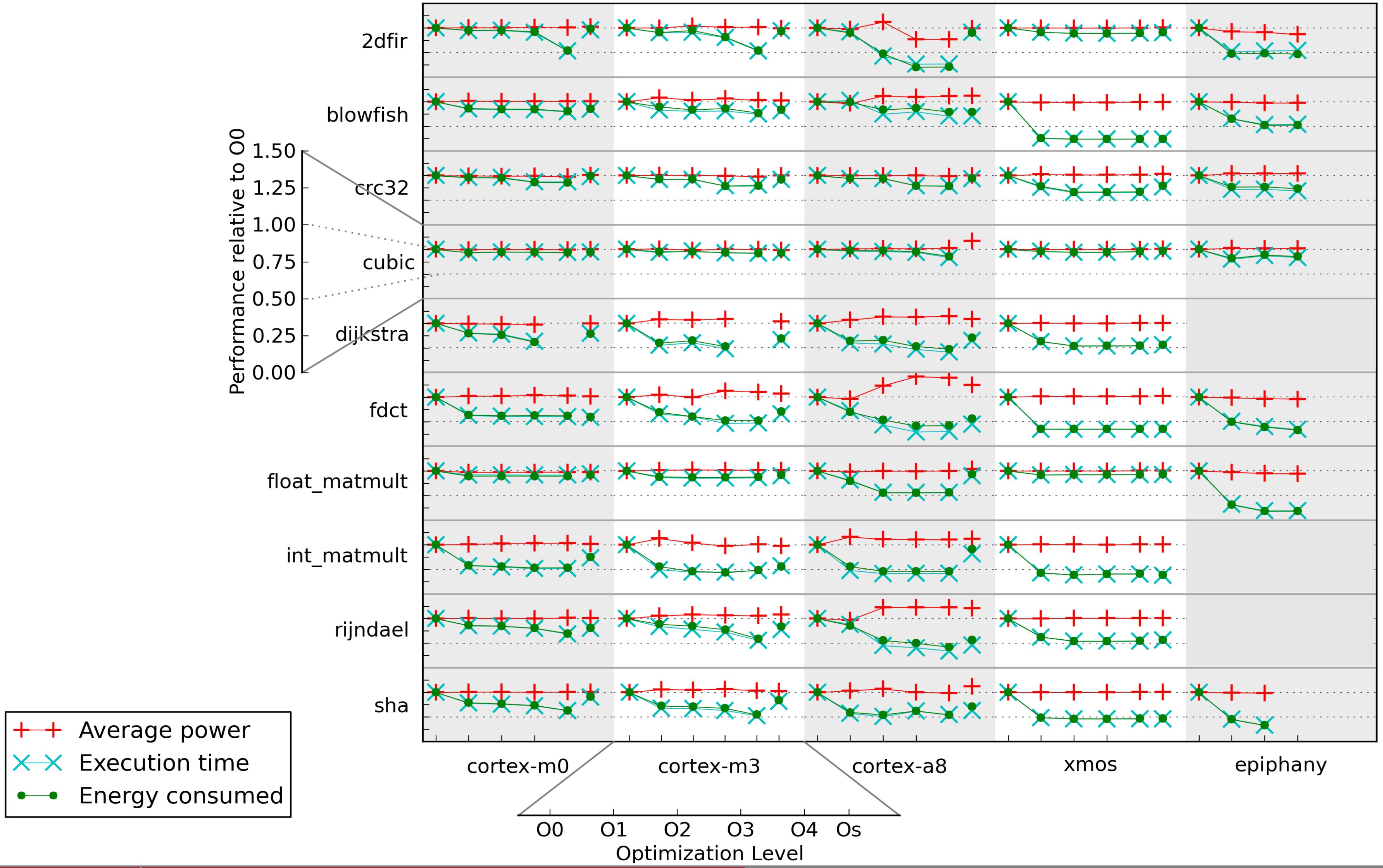
Overview


FDCT, Cortex-M0
FDCT, Cortex-A8
1.50
1.25
1.00
0.75
0.50
0.25
0.00
O0
O1
O2
O3
O4
Os
Execution time
Energy consumed

Overview

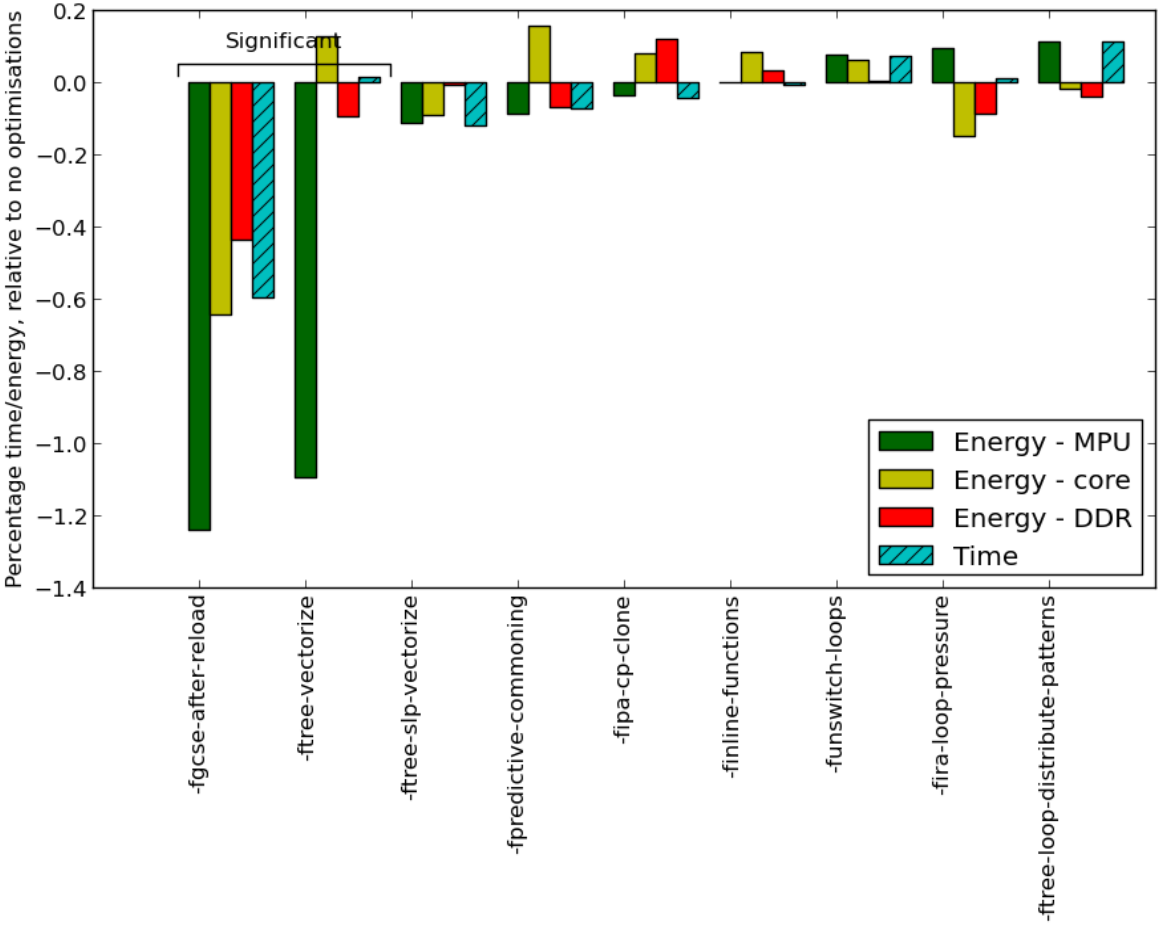

FDCT, Cortex-M0
1.50
1.25
1.00
0.75
0.50
0.25
0.00
O0
O1
O2
O3
O4
Os
FDCT, Cortex-A8
1.50
1.25
1.00
0.75
0.50
0.25
0.00
O0
O1
O2
O3
O4
Os
Average power
Execution time
Energy consumed

Overview


2dfir
blowfish
crc32
cubic
dijkstra
fdct
float_matmult
int_matmult
rijndael
sha
Performance relative to O0
1.50
1.25
1.00
0.75
0.50
0.25
0.00
cortex-m0
cortex-m3
cortex-a8
xmos
epiphany
O0
O1
O2
O3
O4
Os
Optimization Level
Average power
Execution time
Energy consumed

When Time ≠ Energy

- Complex pipeline
- -ftree-vectorize
  - NEON SIMD unit
  - Much lower power



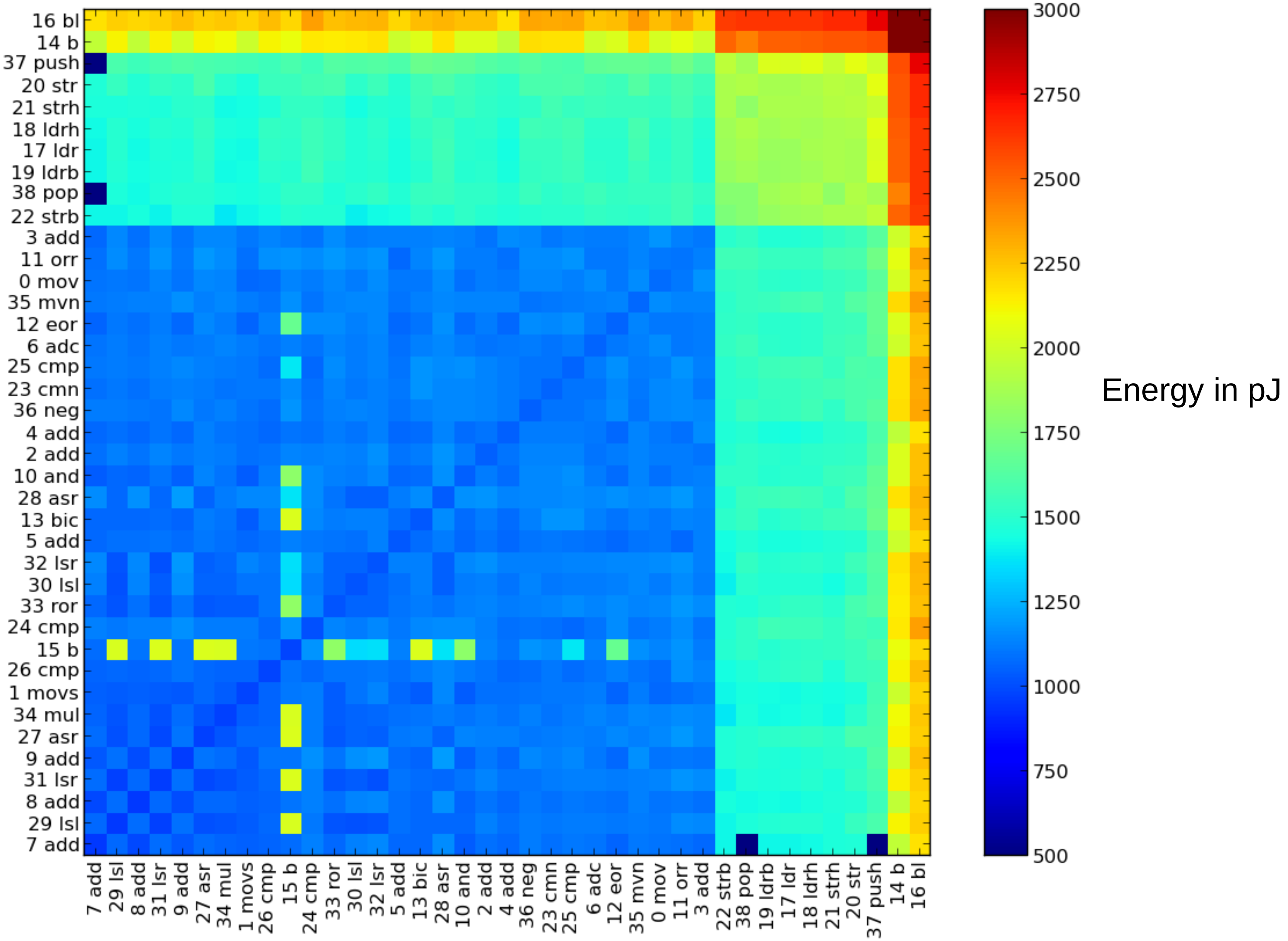
O3 Flags, 2DFIR, Cortex-A8

Conclusion: Mostly, Time ≈ Energy

- Highly correlated
- Especially so for 'simple' pipelines
- Little scope for stalling or superscalar execution

- Complex pipelines:
  - Still a correlation
  - But more variability
  - SIMD, superscalar execution
- To get the most optimal energy consumption we need better than “go fast”

Case Study: Cortex-M0



What does this mean?

For the Compiler Writer

- Current optimization levels (O1, O2, etc.) are a good balance between compile time and performance/energy.
- Never completely optimal
- Machine learning
  - MILEPOST
  - Genetic algorithms

- Current optimizations targeted for performances
- Few (if any) optimizations in current compilers designed to reduce energy consumption

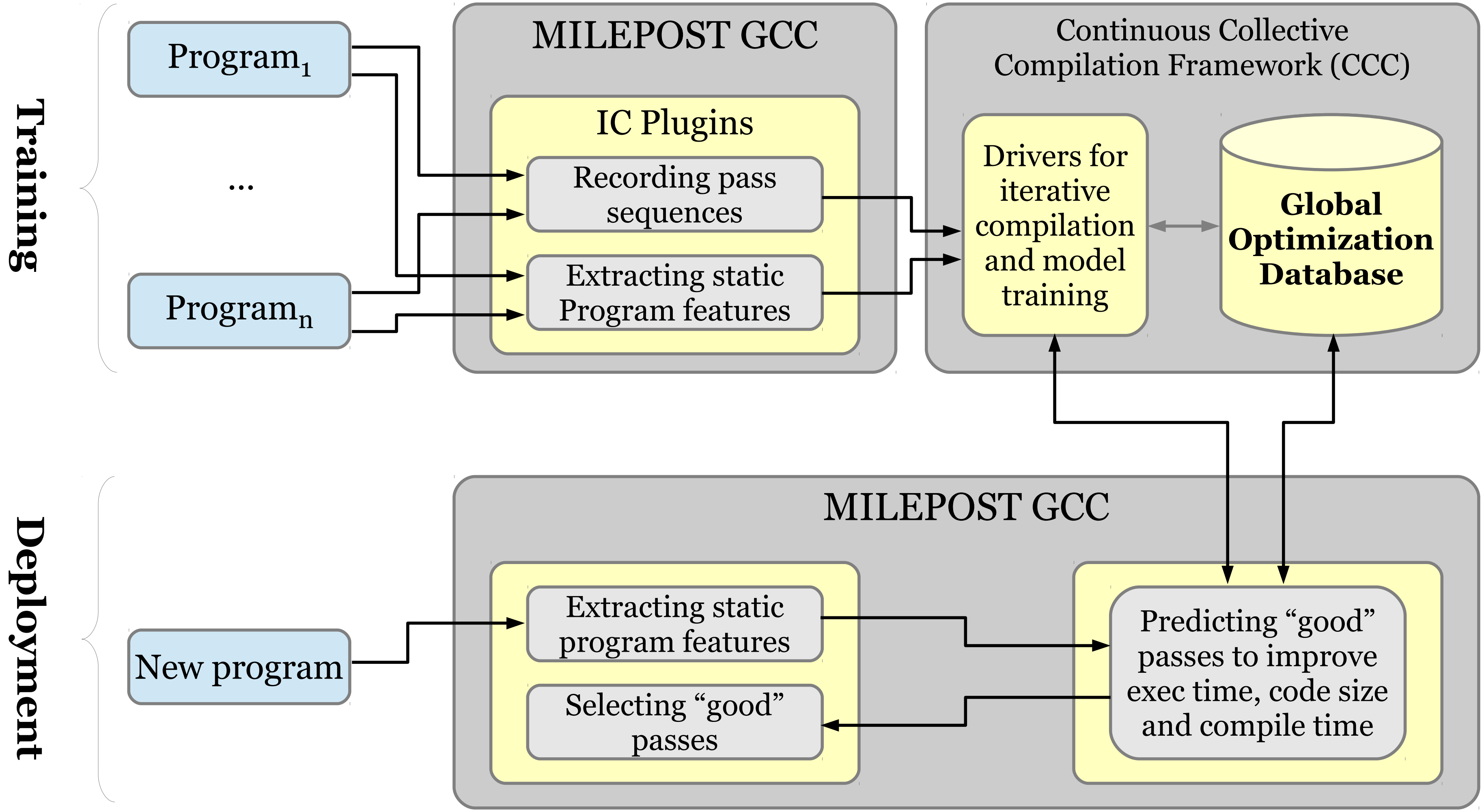
What does this mean?

For the embedded developer

- Try the optimization levels – O3 is a good bet
- Use hardware peripherals
- SIMD

- Power Modes
  - Sleep
- Memory
  - Closer to the processor the better
  - Exploit RAM

MILEPOST GCC


Training
Program$_1$
…
Program$_n$
MILEPOST GCC
IC Plugins
Recording pass sequences
Extracting static Program features
Continuous Collective Compilation Framework (CCC)
Drivers for iterative compilation and model training
Global Optimization Database
Deployment
New program
MILEPOST GCC
Extracting static program features
Selecting “good” passes
Predicting “good” passes to improve exec time, code size and compile time

From Fursin et al, 2008

Conclusion

- Time ≈ Energy
  - True for simple pipelines
  - Mostly true for complex pipelines
  - Good approximation
- Optimization unpredictability
  - Difficult to model the interactions between optimizations
- Commonality across platforms
  - Instruction set plays a role
  - Common options for the ARM platforms, but not Epiphany

Questions and Demonstration

james.pallister@bristol.ac.uk

simon@cs.bris.ac.uk

jeremy.bennett@embecosm.com

All data at: www.jpallister.com/wiki