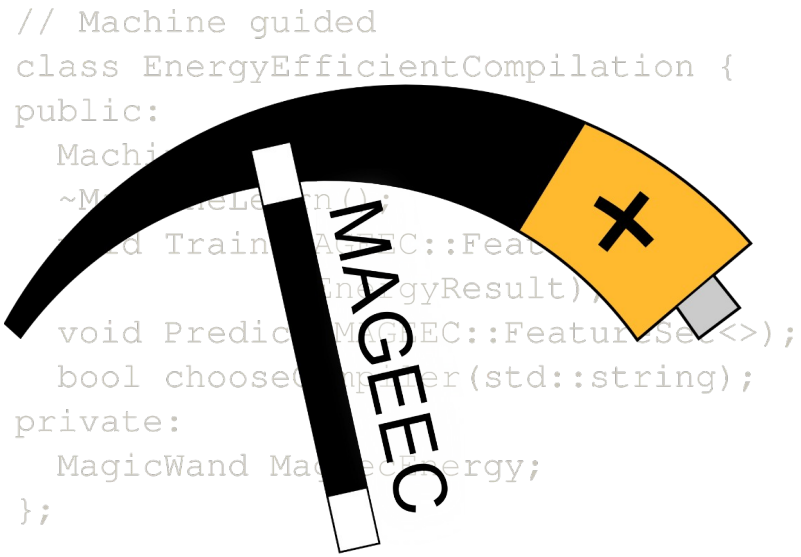
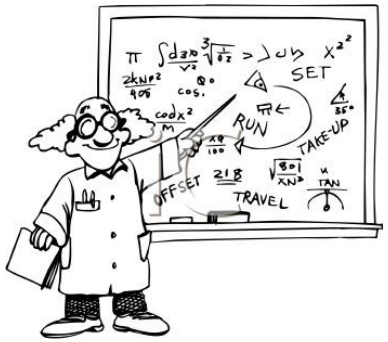


MAchine Guided Energy Efficient Compilation

Simon Hollis, University of Bristol

James Pallister, Embecosm

The MAGEEC Project

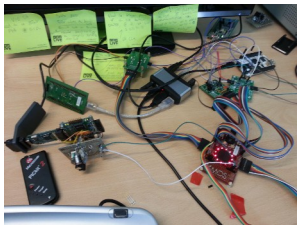


Research into feedback directed optimization

MILEPOST

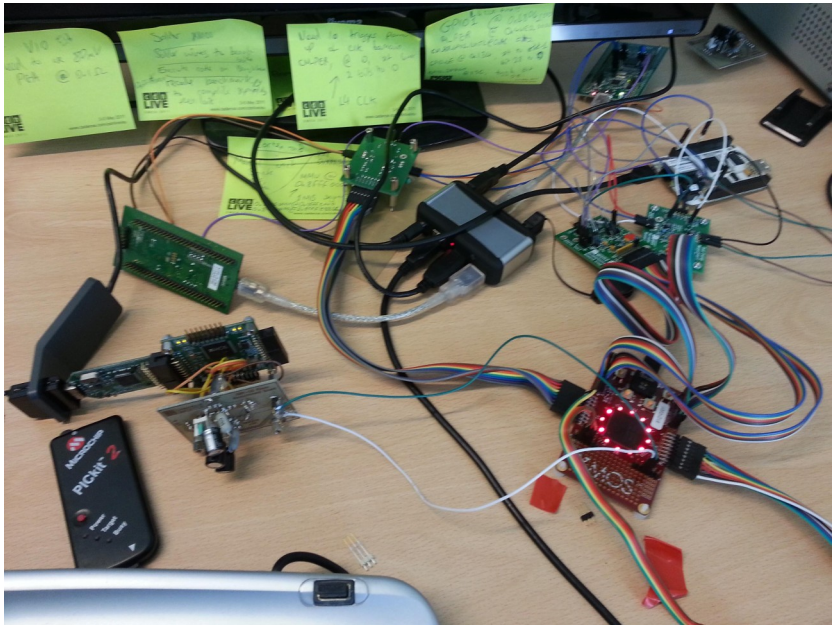
Technology Strategy Board
Driving Innovation

Research into modeling energy usage



Energy measurement

Do Compilers Affect Energy?



- Initial research in 2012 by Embecosm and Bristol University
- The answer is “yes”
- Now published *open access* in a peer-reviewed journal

Identifying Compiler Options to Minimize Energy Consumption for Embedded Platforms
James Pallister; Simon J. Hollis; Jeremy Bennett
The Computer Journal 2013; doi: 10.1093/comjnl/bxt129
http://comjnl.oxfordjournals.org/cgi/reprint/bxt129?ijkey=aA4RYlYQLNVgkE3

How we discovered this

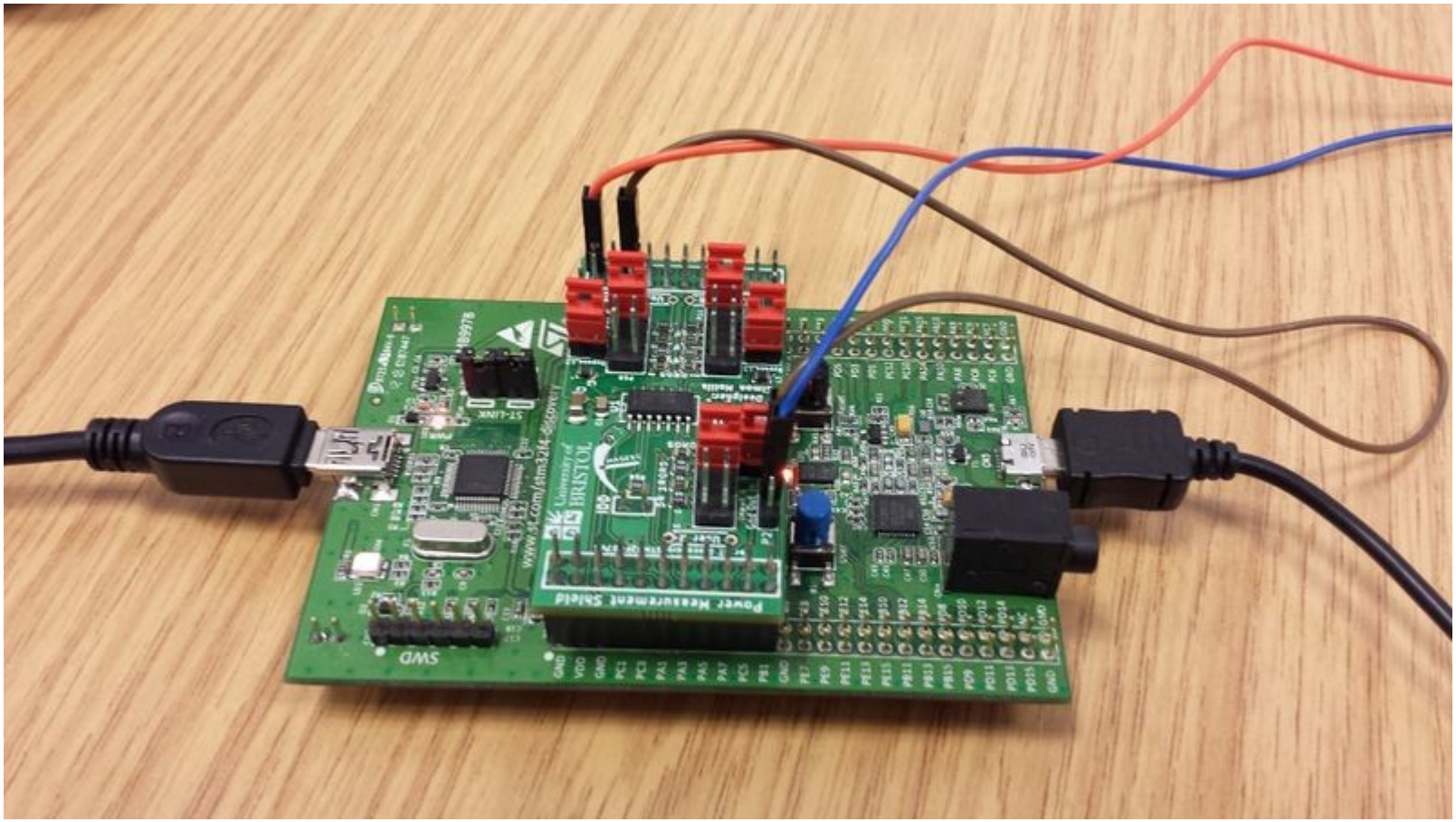
We created:

- A new benchmark suite: BEEBS
- An energy measurement system to monitor the energy of the systems under test
- Lots of scripts to run GCC with different optimisation settings

BEEBS

- The **B**ristol/**E**mbecosm **E**mbedded **B**enchmark **S**uite
  - a free and open source benchmark suite for embedded use
- Underlying principles
  - GPL licensed
  - no I/O
  - avoid library calls
- BEEBS 2.0 released 5 September 2014
  - 80 benchmarks
  - some data variants of the same benchmarks
  - Get it @ http://beebs.eu

A Free and Open Source Energy Measurement System



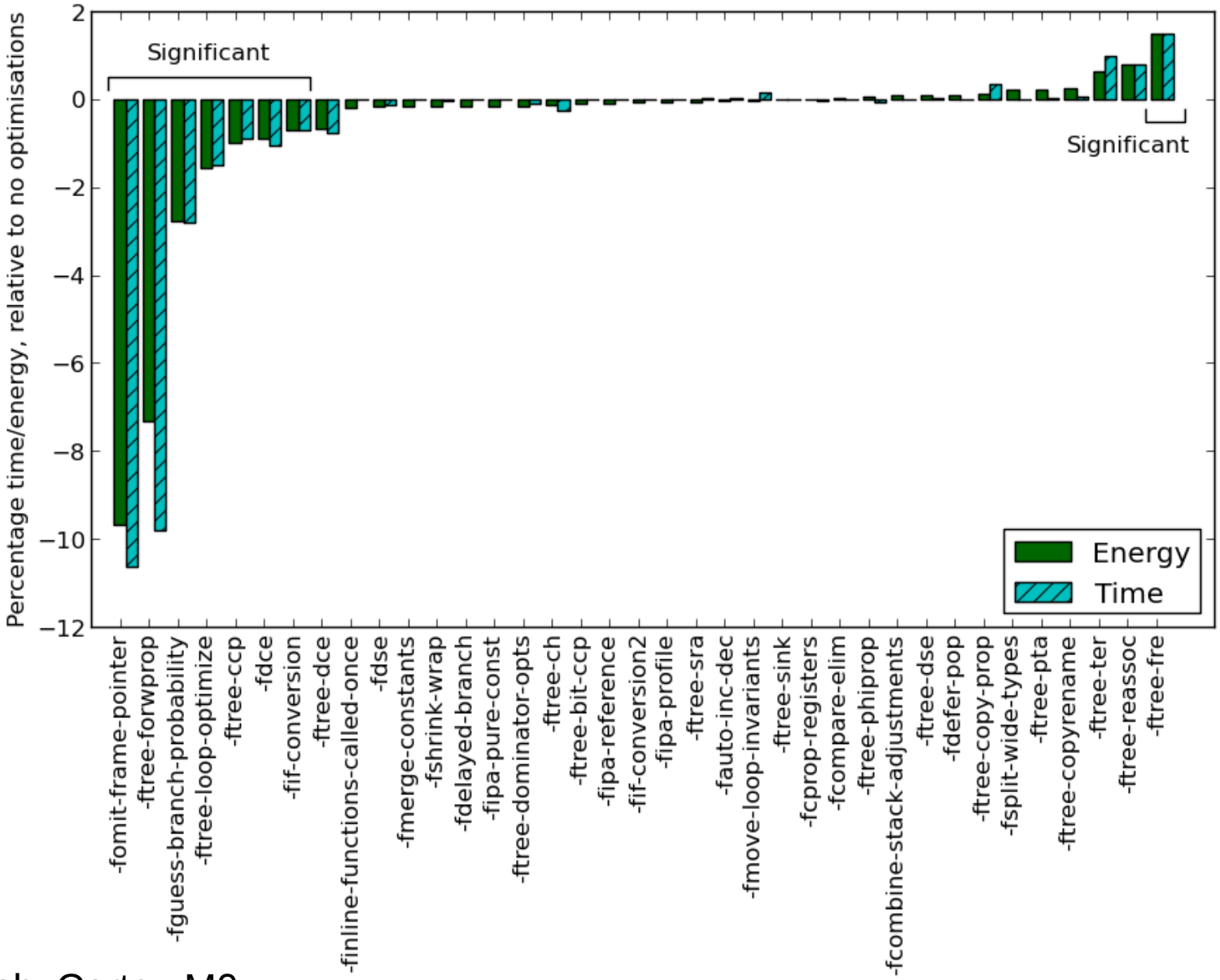
mageec.org/wiki/Power_Sensing_Board

What did we learn?

- Energy consumption ≈ Execution time
  - Generalization, not true in every case
- Optimization unpredictability

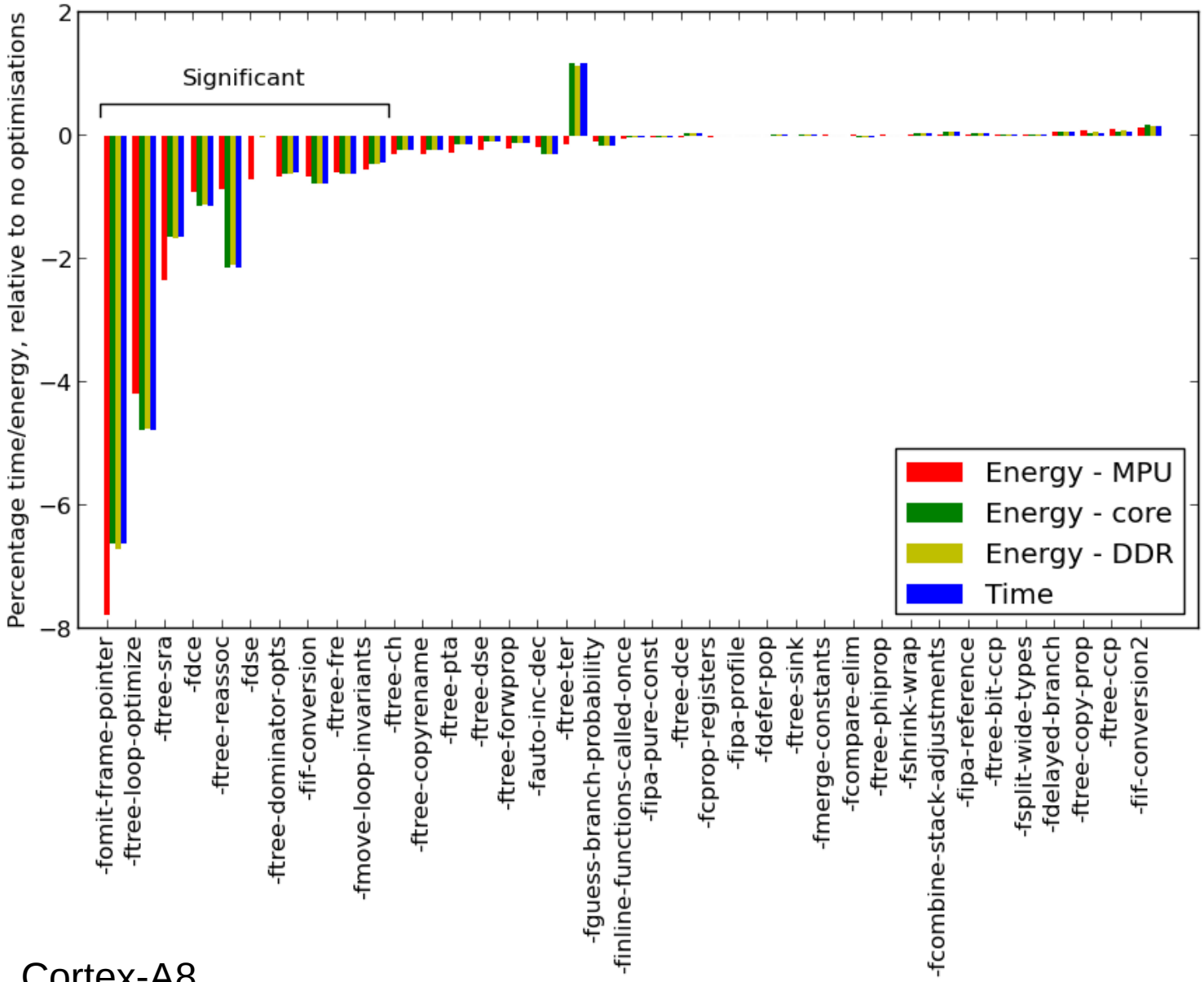
- No compiler optimization is universally good across benchmarks and platforms

Time ≈ Energy – Cortex M0 & -O1



O1 Flags, Blowfish, Cortex-M0

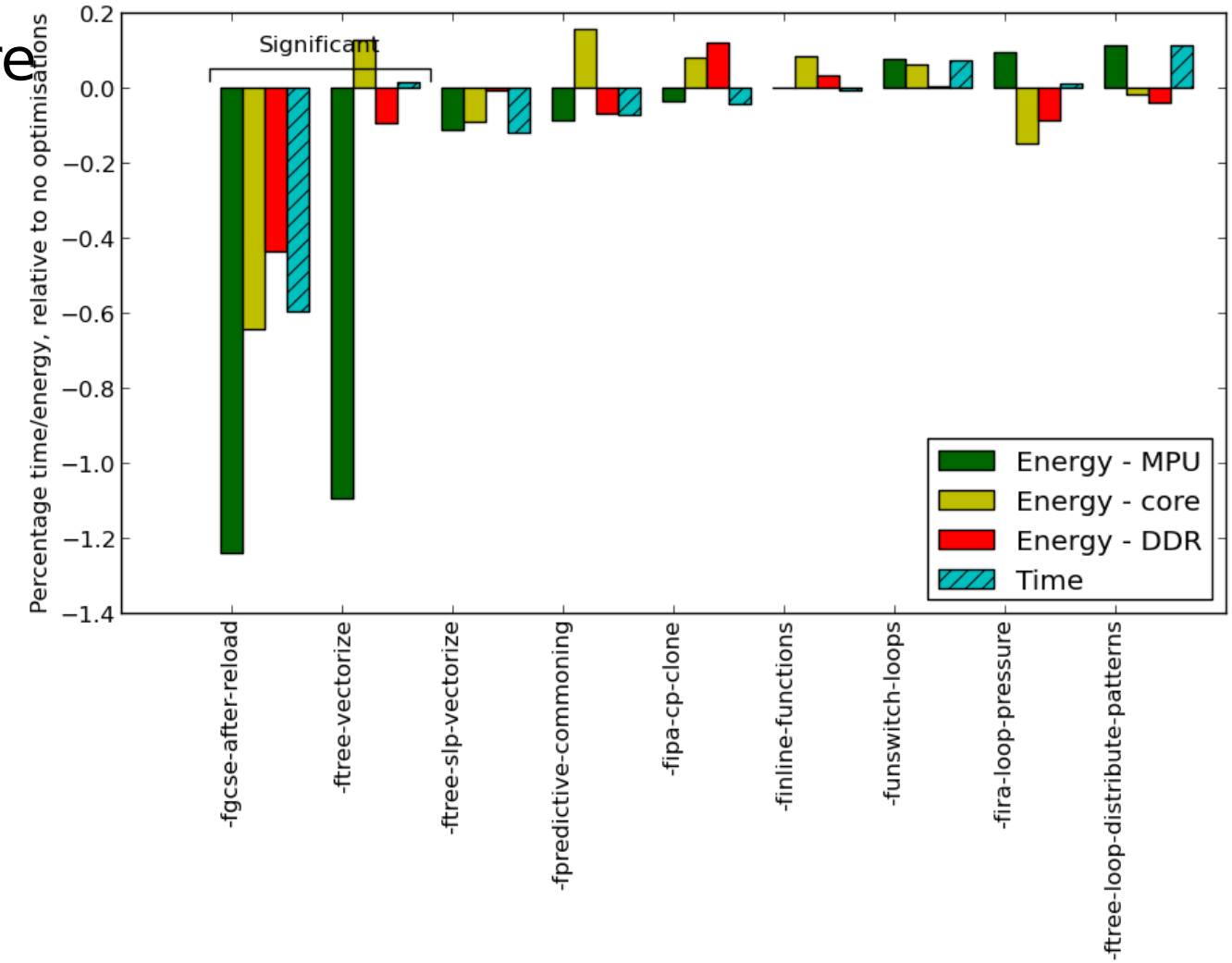
Less Correlation – Cortex A8 &-O1



O1 Flags, Rijndael, Cortex-A8

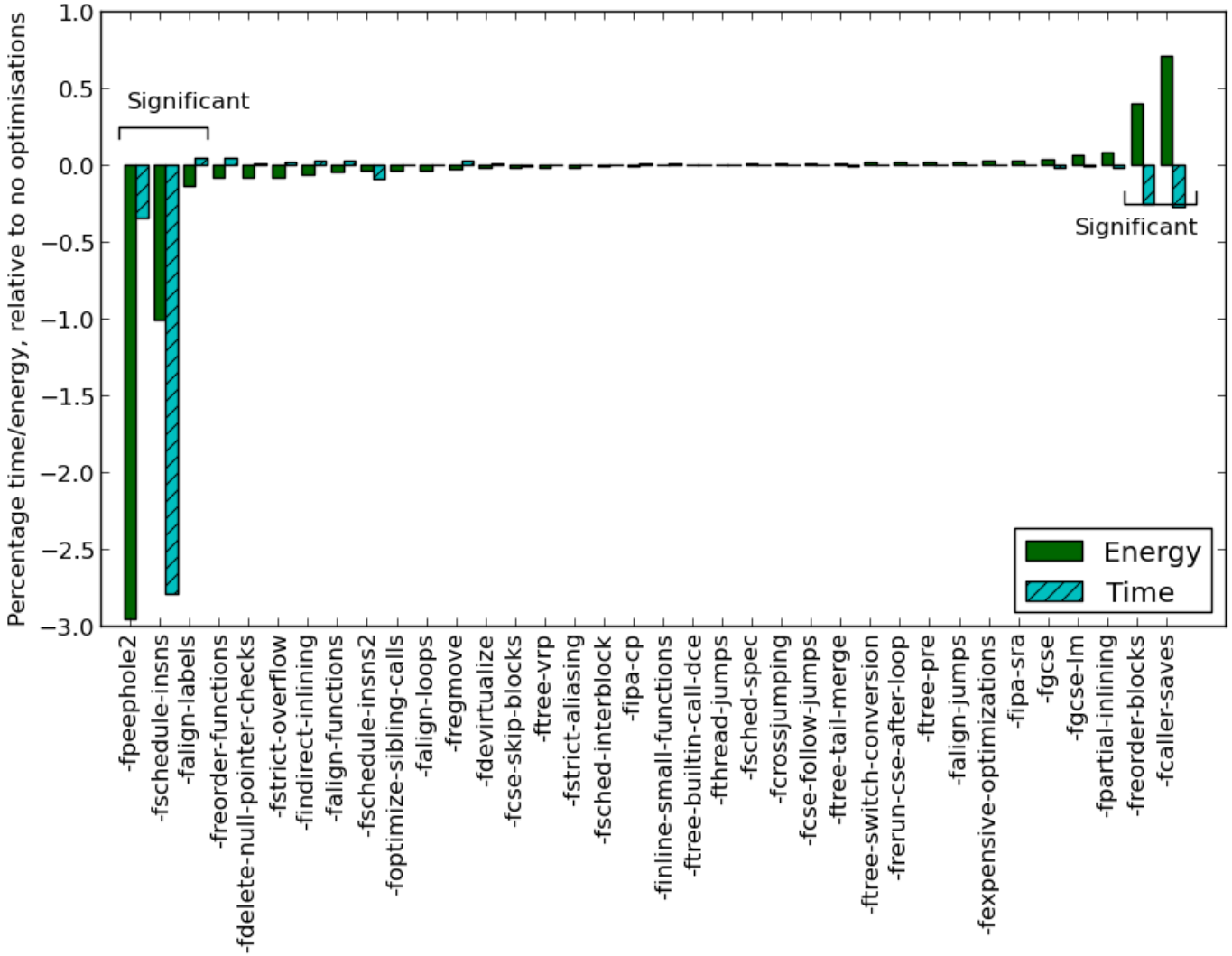
Time ≠ Energy – Cortex A8 & -O3

- Cortex-A8 has more complex pipeline
- -ftree-vectorize
  - NEON SIMD unit
  - Much lower power



O3 Flags, 2DFIR, Cortex-A8

Time ≠ Energy – Cortex M3 & -O2



O2 Flags, Blowfish, Cortex-M3

-fpeephole2

Constant folding, strength reduction, algebraic simplification

-fschedule-insns

Reorder instructions to reduce execution stalls

-fcaller-saves

“Enable values to be allocated in registers that will be clobbered by function calls, by emitting extra instructions to save and restore the registers around such calls.”

Conclusion: Mostly, Time ≈ Energy

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

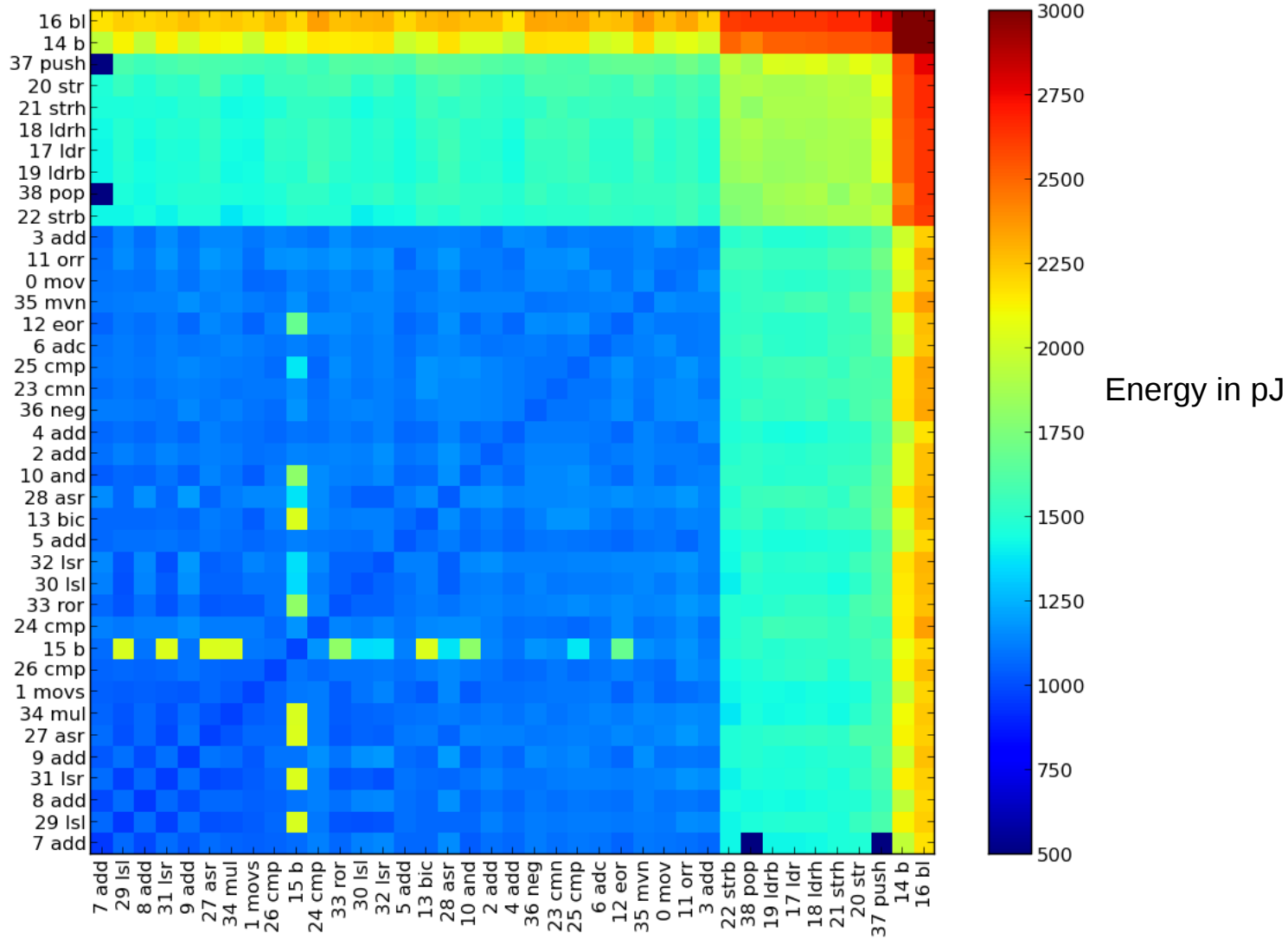
- Complex pipelines:
  - Still a correlation
  - But more variability
  - SIMD, superscalar execution
- To get the most optimal energy consumption we need better than "go fast"

The unpredictability of optimisations makes the process chaotic for prediction

Can we statically model the energy consumption?

- The previous slides were based on hardware measurement.
- ?? Can we model the same?
  - Execution time is relatively easy to model.
- How about energy?

Case Study: static energy model for the Cortex-M0



Instruction Pairings

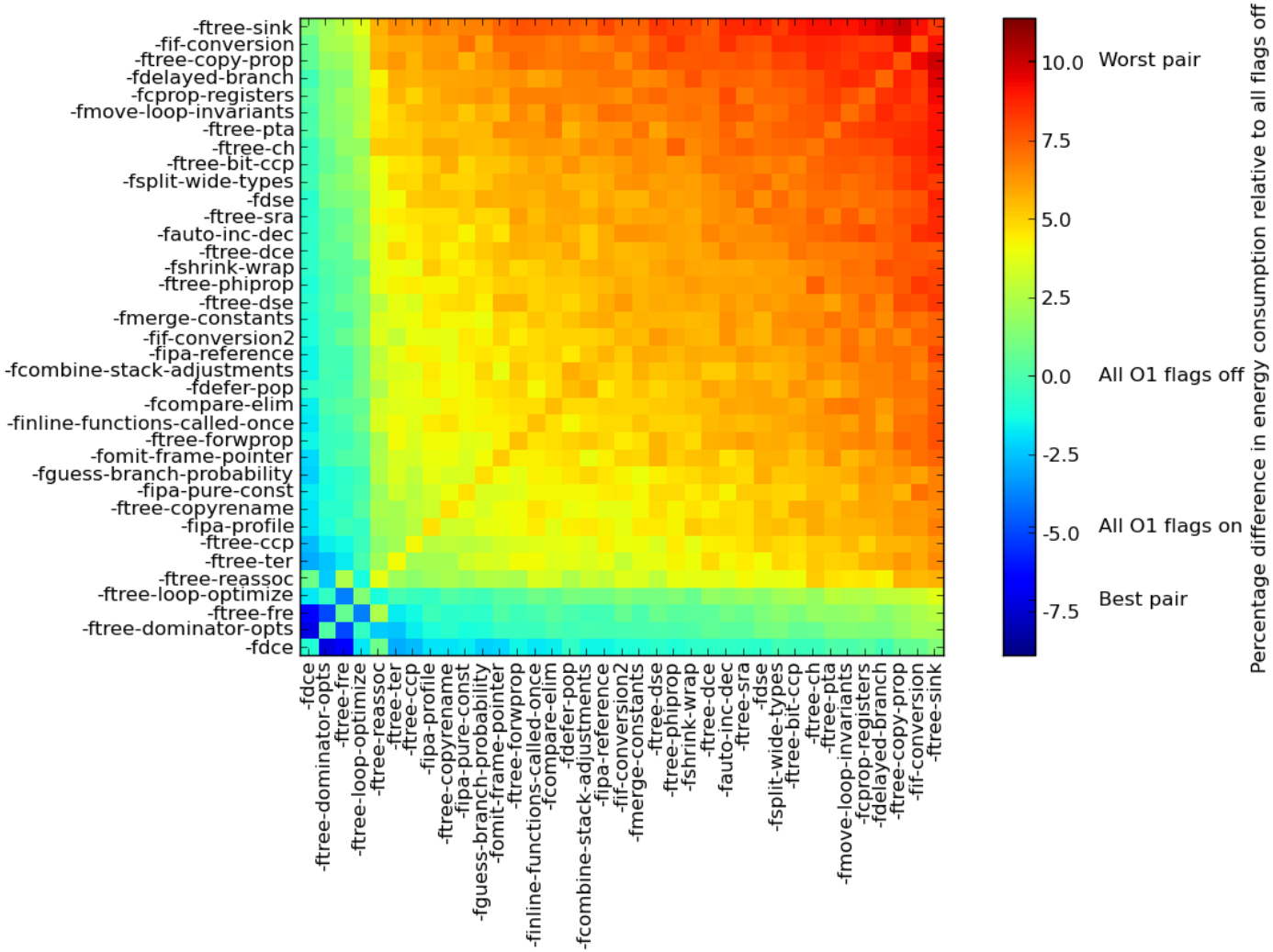
Can we statically model the energy consumption?

- OK, so we can model the energy consumption of instruction pairs in a simple processor.

- How about the effect of compiler optimisations on this?

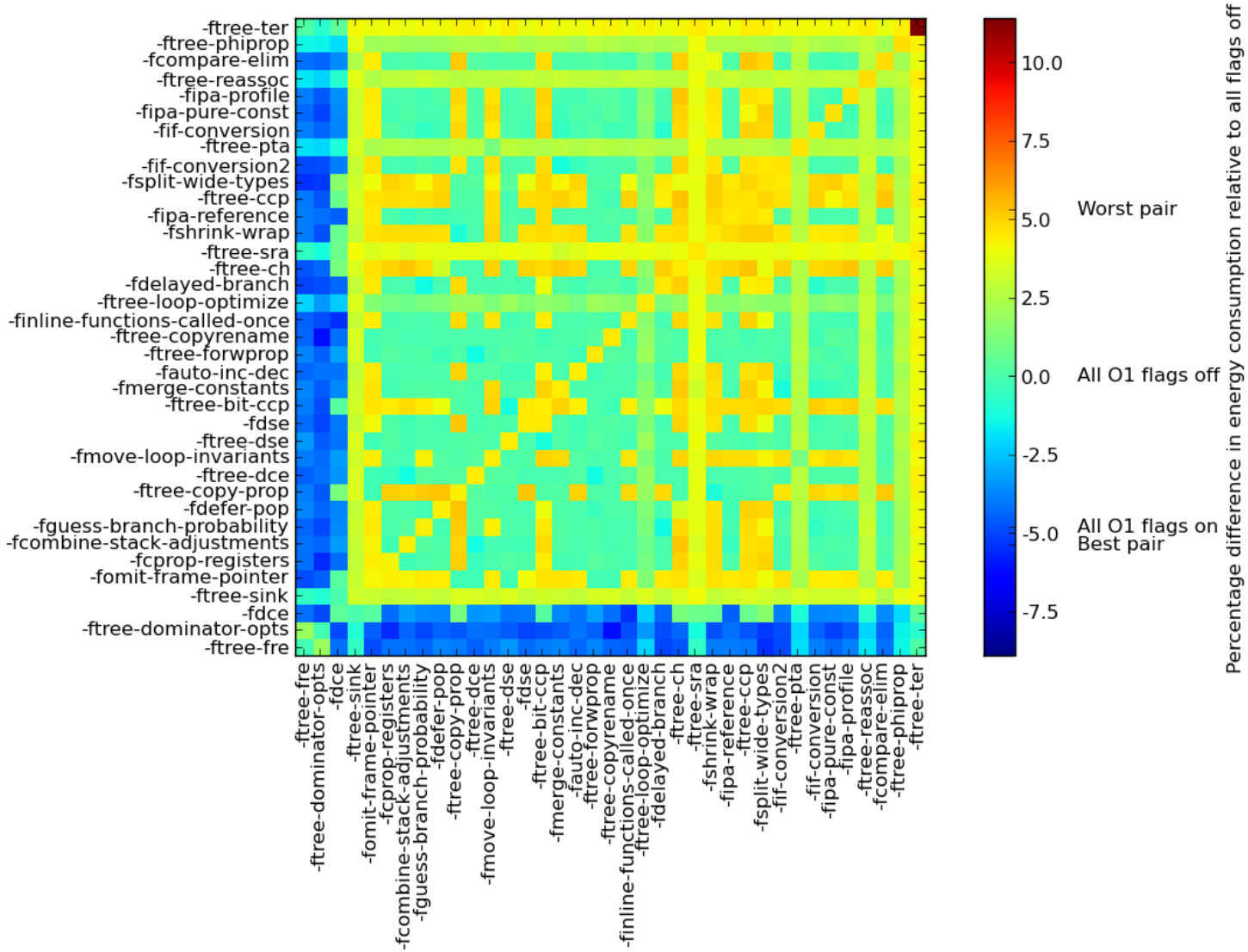
Energy model for flags interactions

- Model constructed from 1 and 2 -way interactions
- Doesn't predict very well



Measured energy for flag interactions

- Pairs of optimizations on top of O0
- Little correlation to our model
- Possibly higher order interactions occurring?



O1 Flags, Cubic, Cortex-M0

Conclusion: Which optimization to choose?

For the general case, this question can't be answered

- Unpredictable interactions
- Many non-linear effects
- Not enough data recorded in the fractional factorial design to model
- Evidence of higher order interactions between optimizations?

Conclusion: Optimizations are common across architectures…

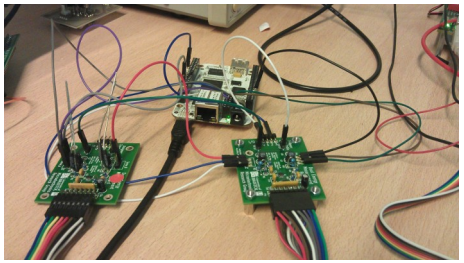
… Sometimes

- Common options across all the ARM platforms for a particular benchmark
- A few consistently good options for Epiphany
  - Simpler instruction set
  - Newer compiler
  - Many more registers than ARM

Recap: How MAGEEC can help

Objective is energy optimization

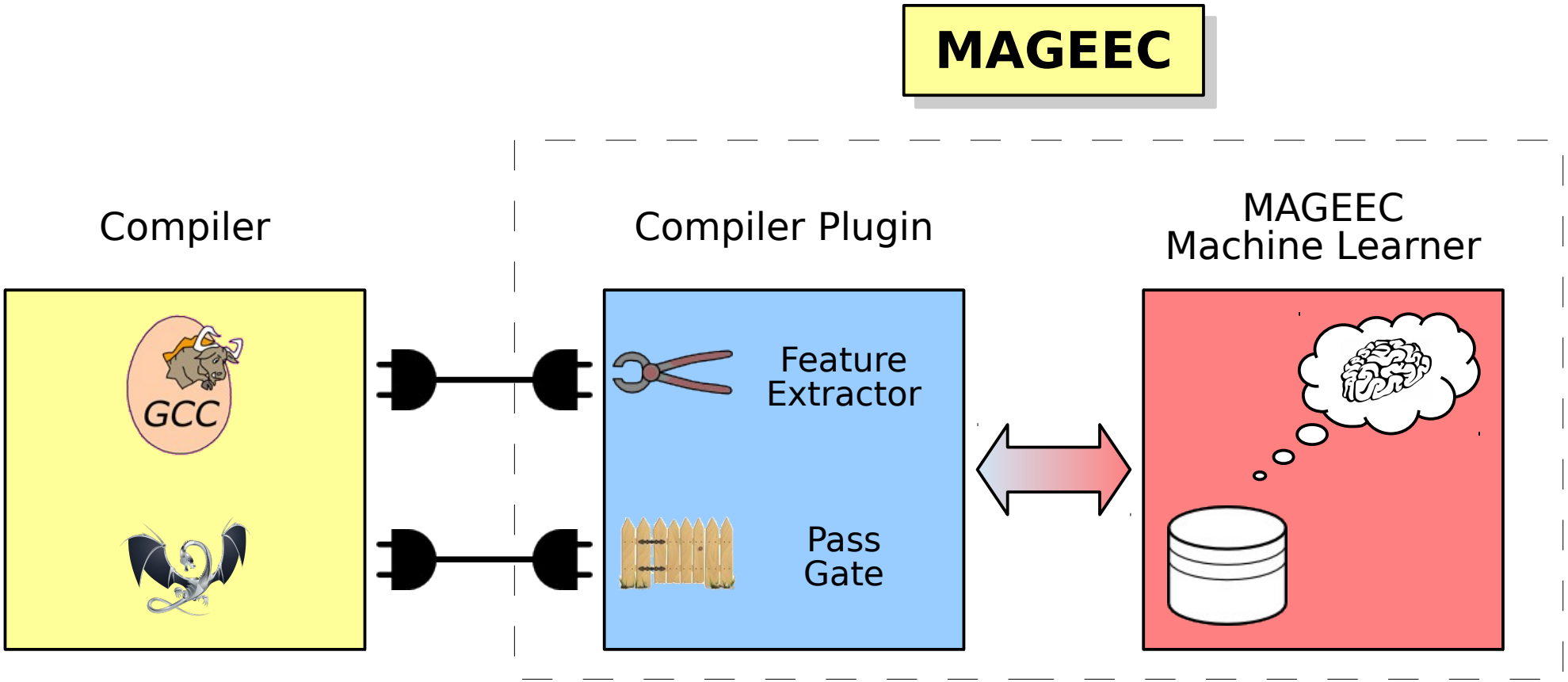
Generic framework: GCC *and* LLVM initially



Energy measured *not* modeled

Working system, not research prototype

Overall Design



Output from MAGEEC (1)

```
$ avr-gcc -O2 -mmcu=atmega328p 2dfir.c boardsupport.o -o O2e.out \
          -fplugin=/opt/mageec/lib/libmageec_gcc.so \
          -fplugin-arg-libmageec_gcc-dumppasses
MAGEEC:  Targetting 'GCC-4.9.0' for 'SOMETARGET'
LEARNER: Hello!
MAGEEC:  New source file example.c
LEARNER: New file
GCC:     Start File
          gcc_data:  (nil)
          user_data: (nil)
Pass: '*warn_unused_result',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: '*diagnose_omp_blocks',  Type: GIMPLE,  Function: 'main',  Gate: 0
…
Pass: 'ssa',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: 'mageec-extractor',  Type: GIMPLE,  Function: 'main',  Gate: 1
Current Function: main
  ( 1) Basic Block Count     : 35
  ( 2) BB with 1 successor   : 23
  ( 3) BB with 2 successor   : 11
  ( 4) BB with > 2 successor : 0
  ( 5) BB with 1 predecessor : 23
  ( 6) BB with 2 predecessor : 11
  ( 7) BB with > 2 predecesso: 0
  ( 8) BB with 1 pred 1 succ : 21
  ( 9) BB with 1 pred 2 succ : 1
…
```

Enhancement request

Standard GCC pass

Feature extractor

Features

Output from MAGEEC (2)

```
Pass: 'ubsan',  Type: GIMPLE,  Function: 'main',  Gate: 0
Pass: '*early_warn_uninitialized',  Type: GIMPLE,  Function: 'main',  Gate: 0
Pass: '*rebuild_cgraph_edges',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: 'inline_param',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: 'einline',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: 'early_optimizations',  Type: GIMPLE,  Function: 'main',  Gate: 1
  New gate: 0
Pass: 'release_ssa',  Type: GIMPLE,  Function: 'main',  Gate: 1
  New gate: 0
Pass: '*rebuild_cgraph_edges',  Type: GIMPLE,  Function: 'main',  Gate: 1
…
Pass: 'tailr',  Type: GIMPLE,  Function: 'main',  Gate: 1
Pass: 'ch',  Type: GIMPLE,  Function: 'main',  Gate: 1
  New gate: 0
Pass: 'stdarg',  Type: GIMPLE,  Function: 'main',  Gate: 0
  New gate: 1
Pass: 'cplxlower',  Type: GIMPLE,  Function: 'main',  Gate: 1
…
MAGEEC:  End of source file
LEARNER: End file
GCC:     End File
          gcc_data:  (nil)
          user_data: (nil)
GCC:     Finish
MAGEEC:  Finish
LEARNER: Goodbye!
```

Overridden to disable

Overridden to enable

Energy Measurement

```
$ avrdude -carduino -patmega328p -P/dev/ttyUSB0 \
          -D -U flash:w:O2e.out ; energytool -m 1 read EE00 PA0

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.00s

avrdude: Device signature = 0x1e950f
avrdude: reading input file "O2e.out"
avrdude: input file O2e.out auto detected as ELF

...

avrdude: verifying ...
avrdude: 1760 bytes of flash verified

avrdude: safemode: Fuses OK (H:00, E:00, L:00)

avrdude done.  Thank you.

Measurement point 1
Energy:              25.293 mJ
Time:               310.452 ms
Power:               81.471 mW
Average current:     16.370 mA
Average voltage:      4.977 V
```

Flashing device complete

Energy measurement results

Video

- https://www.youtube.com/watch?v=W0hqoCdDQYA (Approx min 30)

Results for AVR ATMega328PU

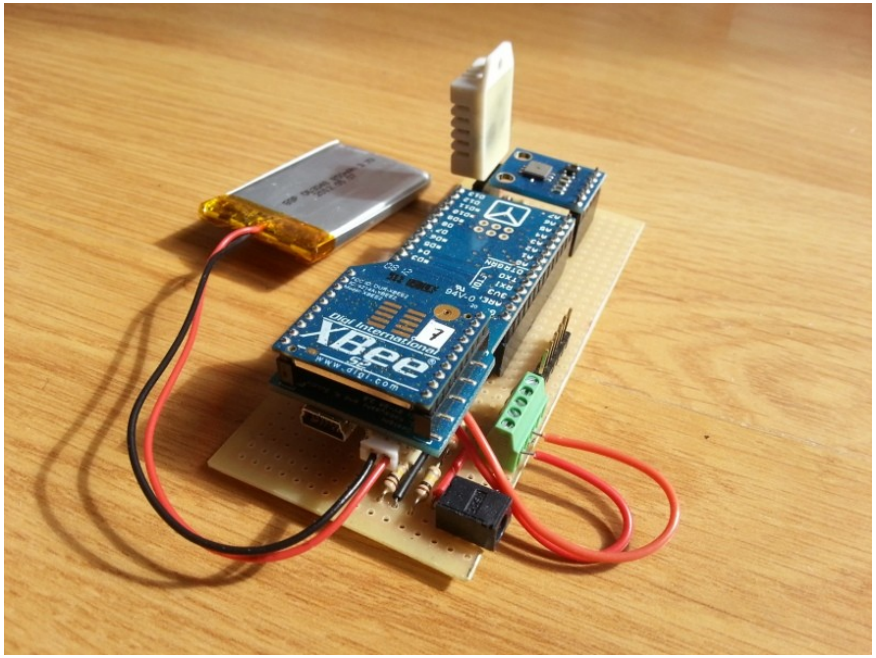
Standard GCC -O0		
Energy	29.8	mJ
Time	329.1	ms
Power	90.6	mW
Average current	17.9	mA
Average voltage	5.1	V

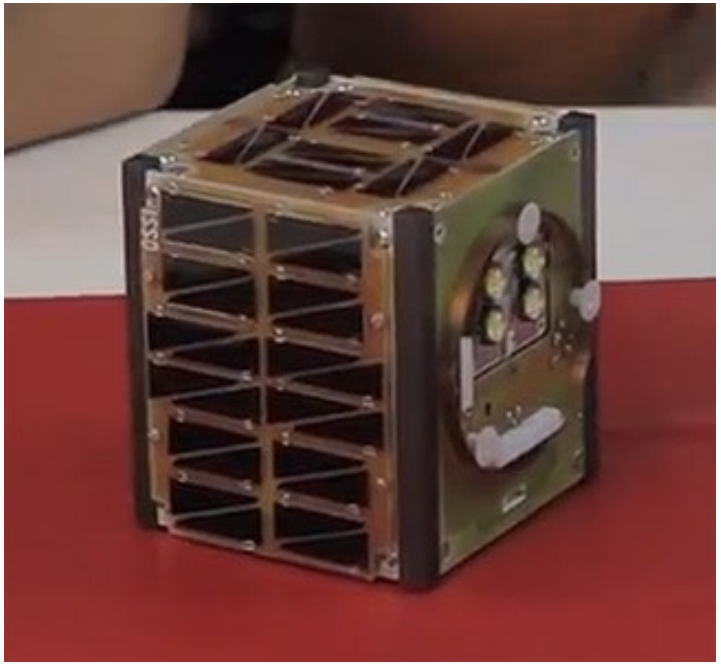
MAGEEC GCC		
Energy	27.6	mJ
Time	309.8	ms
Power	89.1	mW
Average current	17.6	mA
Average voltage	5.1	V

- Based on minimal training:
  - just 10 single function programs
  - 700 training runs

Next steps

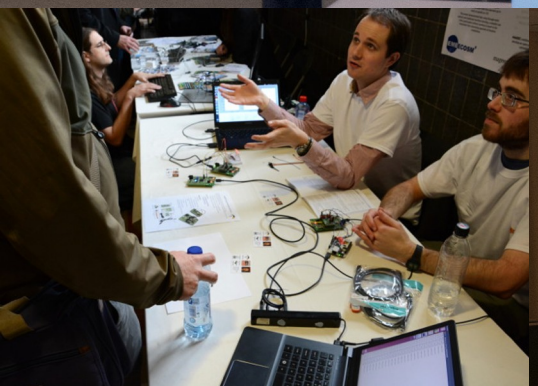
- We now have BEEBS V2
  - A much larger training set
- We have decided on J48/C5 as the machine learning algorithm
- We have case studies





Exploitation


MAGEC

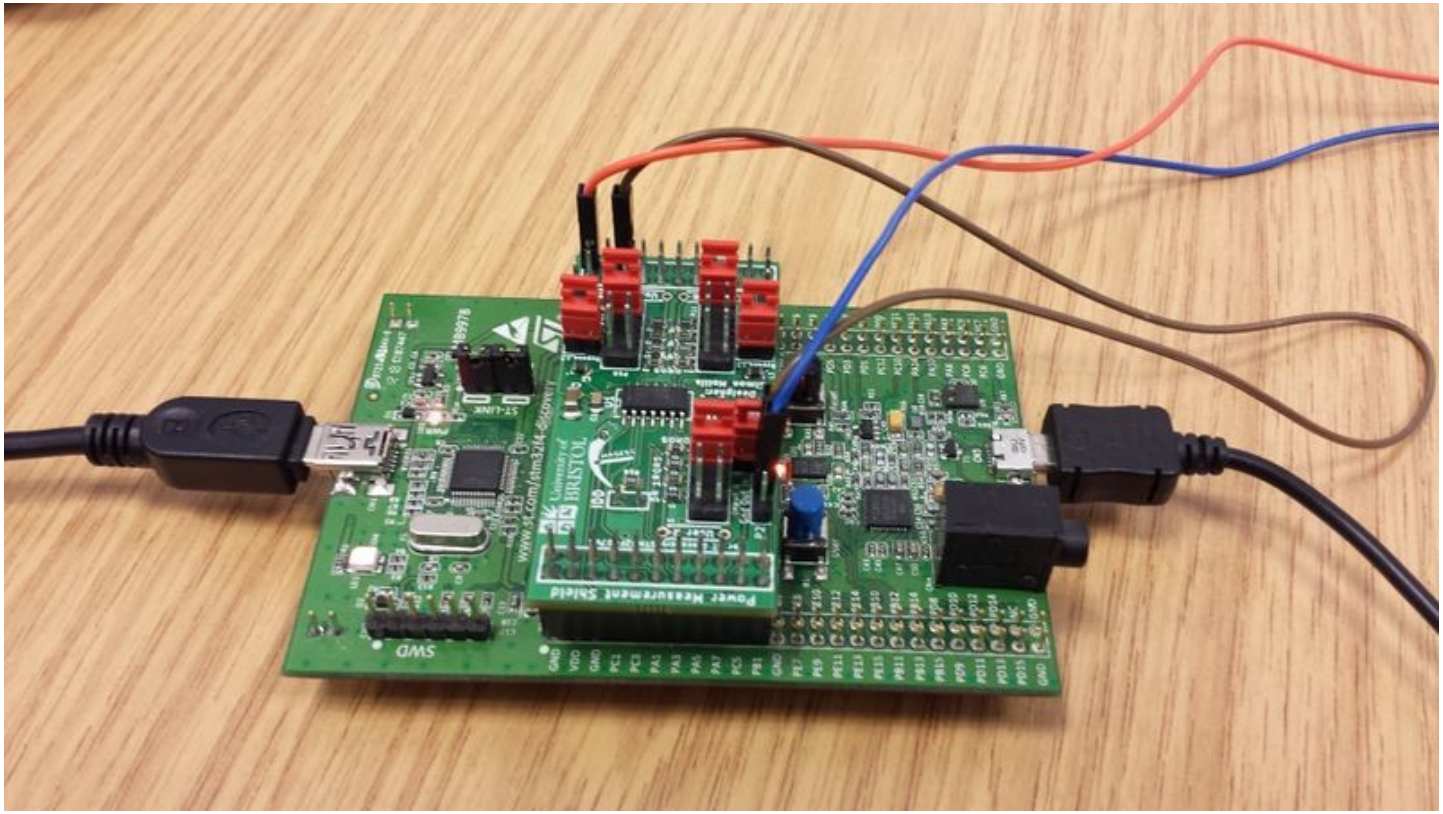


Where Can I Get It?

- Project Website: http://mageec.org
- MAGEEC Source: http://github.com/mageec/mageec
- BEEBS: http://beebs.eu/
- Mailing List: mageec@mageec.org
- IRC: #mageec on Freenode
- Orders now being taken for at-cost measurement boards
  http://mageec.org/2014/08/21/now-taking-orders-for-energy-measurement-hardware/

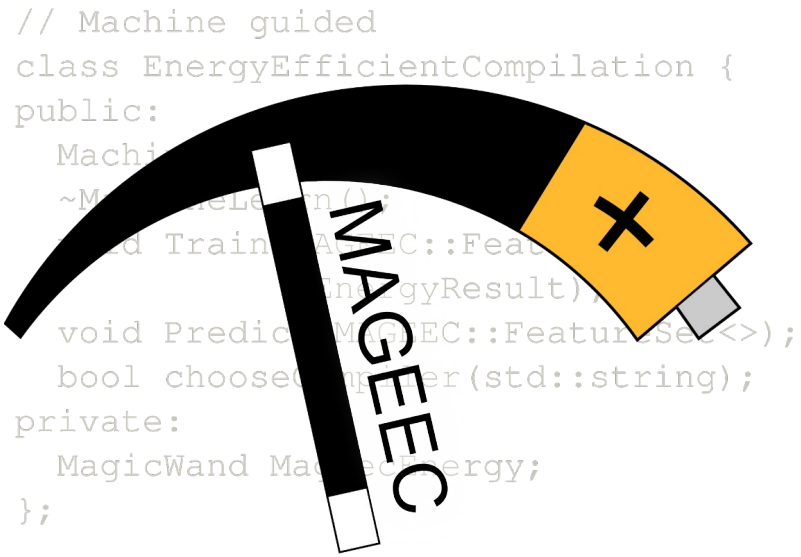
Orders taken for a Free and Open Source Energy Measurement System

£46+VAT



mageec.org/wiki/Power_Sensing_Board

http://mageec.org/2014/08/21/now-taking-orders-for-energy-measurement-hardware/



Thank you

mageec.org
www.embecosm.com
cs.bris.ac.uk

EMBECOSM®

University of BRISTOL