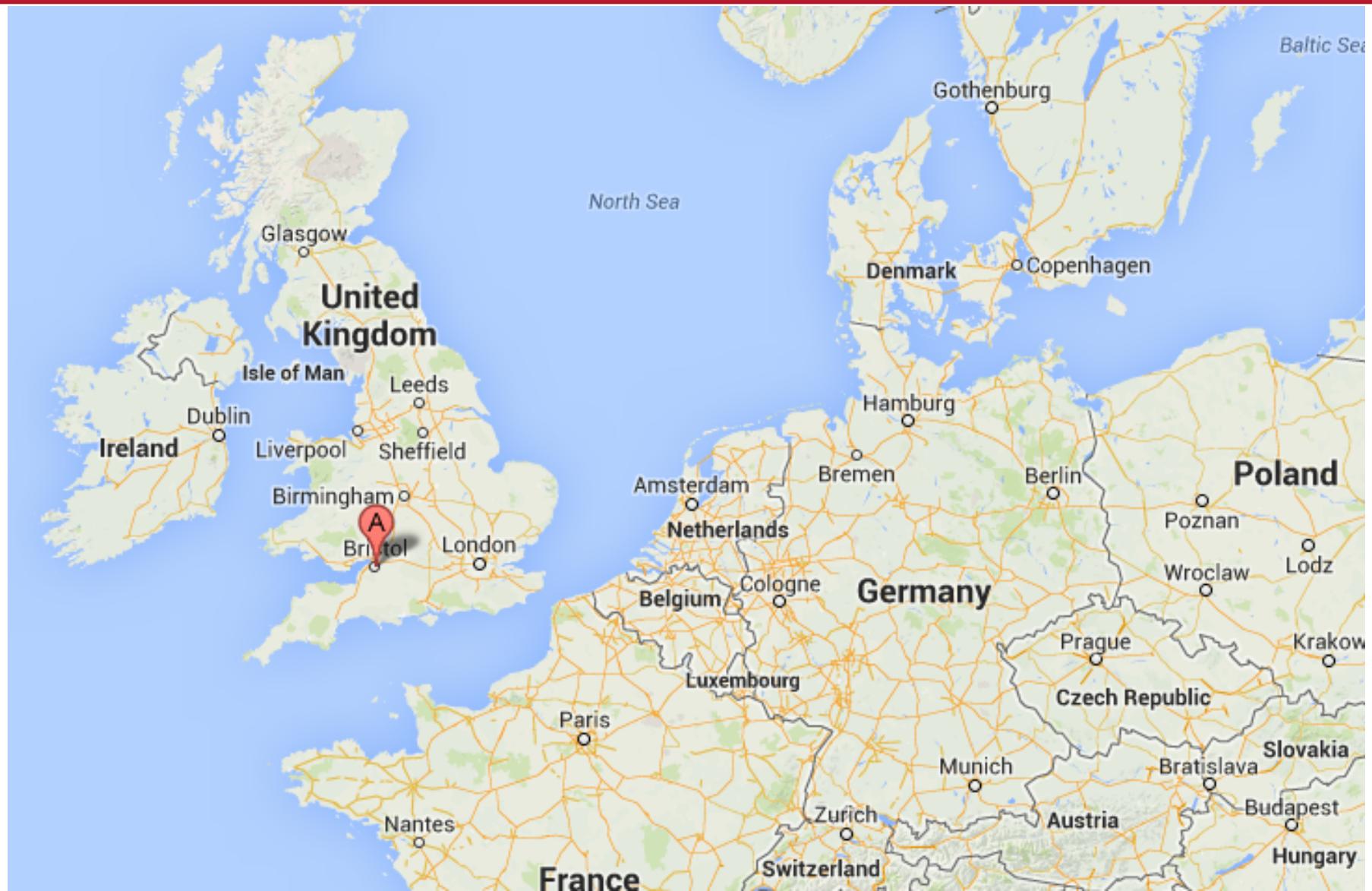


# **Software and energy aware computing**

**Kerstin Eder**

Design Automation and Verification, Microelectronics  
Verification and Validation for Safety in Robots, Bristol Robotics Laboratory

# Bristol



# The University of Bristol

- UoB founded in 1909
  - The first higher education institution in England to admit women on an equal basis to men. ☺
- Top 30 universities globally  
(QS World University Rankings)
- 6 Faculties
- ~14.000 students, 2.000 in FEN
- Computer Science in FEN
- EACO workshops and research to advance the state of the art in Energy Aware Computing



# Software and energy aware computing

---

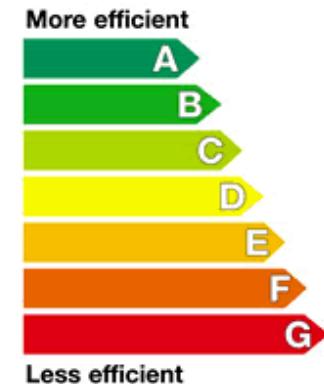
*More power to software developers!*

Kerstin Eder

Design Automation and Verification, Microelectronics  
Verification and Validation for Safety in Robots, Bristol Robotics Laboratory

# Overview

- Introduction and Motivation
  - Energy consumption of Computing
  - Software vs Hardware



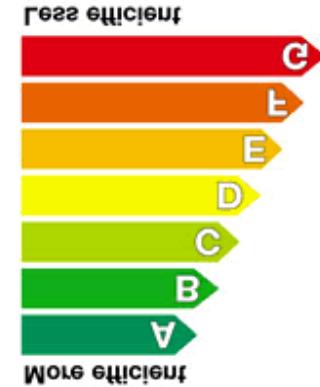
# Overview

- Introduction and Motivation
  - Energy consumption of Computing
  - Software vs Hardware



# Overview

- Introduction and Motivation
  - Energy consumption of Computing
  - Software vs Hardware
  - **Energy Transparency**
- Measuring the energy consumption of software
  - Demonstration with hands-on session by Steve Kerrison
- Energy modeling
  -
- Fundamentals of static analysis of software
- Static analysis and optimization



# Learning Objectives

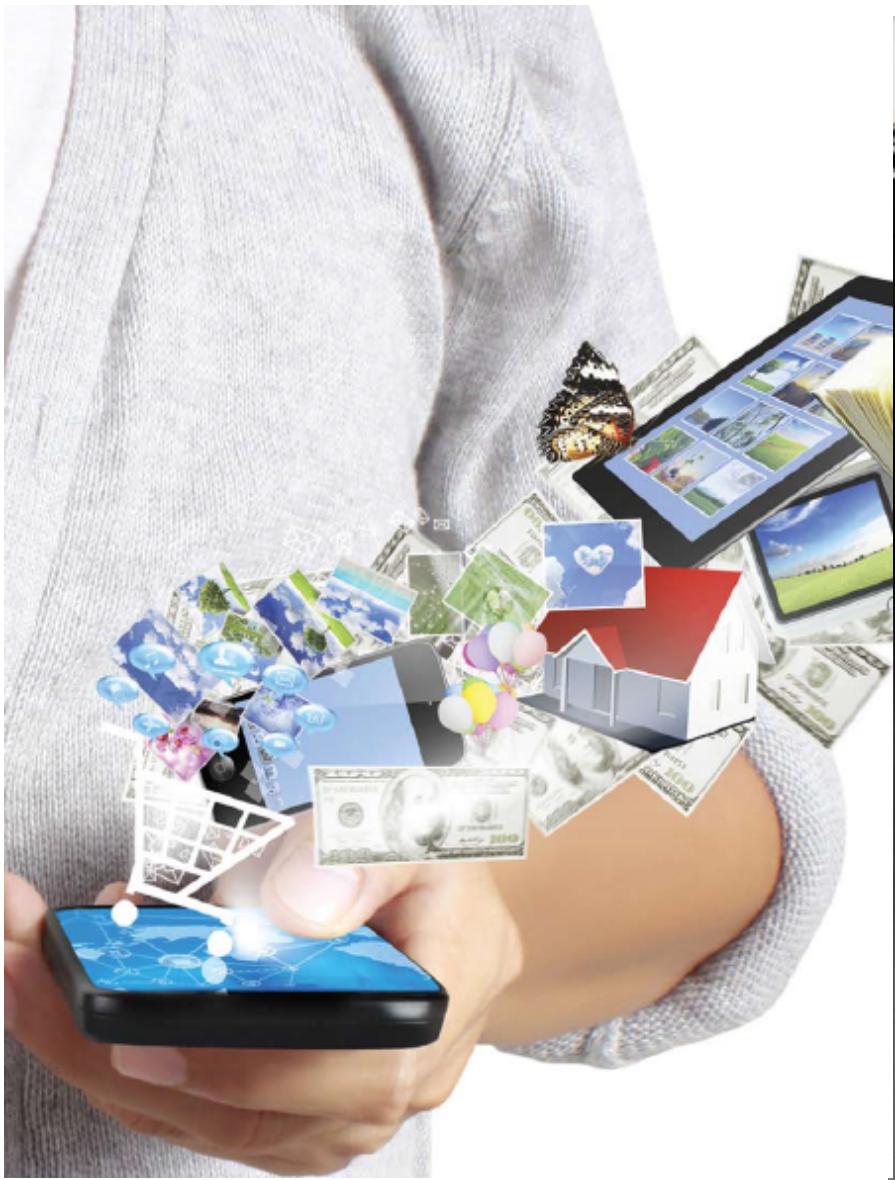
---

- Why software is key to energy efficient computing
- What energy transparency means and why we need energy transparency to achieve energy efficient computing
- How to measure the energy consumed by software
- How to estimate the energy consumed by software *without* measuring
- How to construct energy consumption models

# Introduction and Motivation

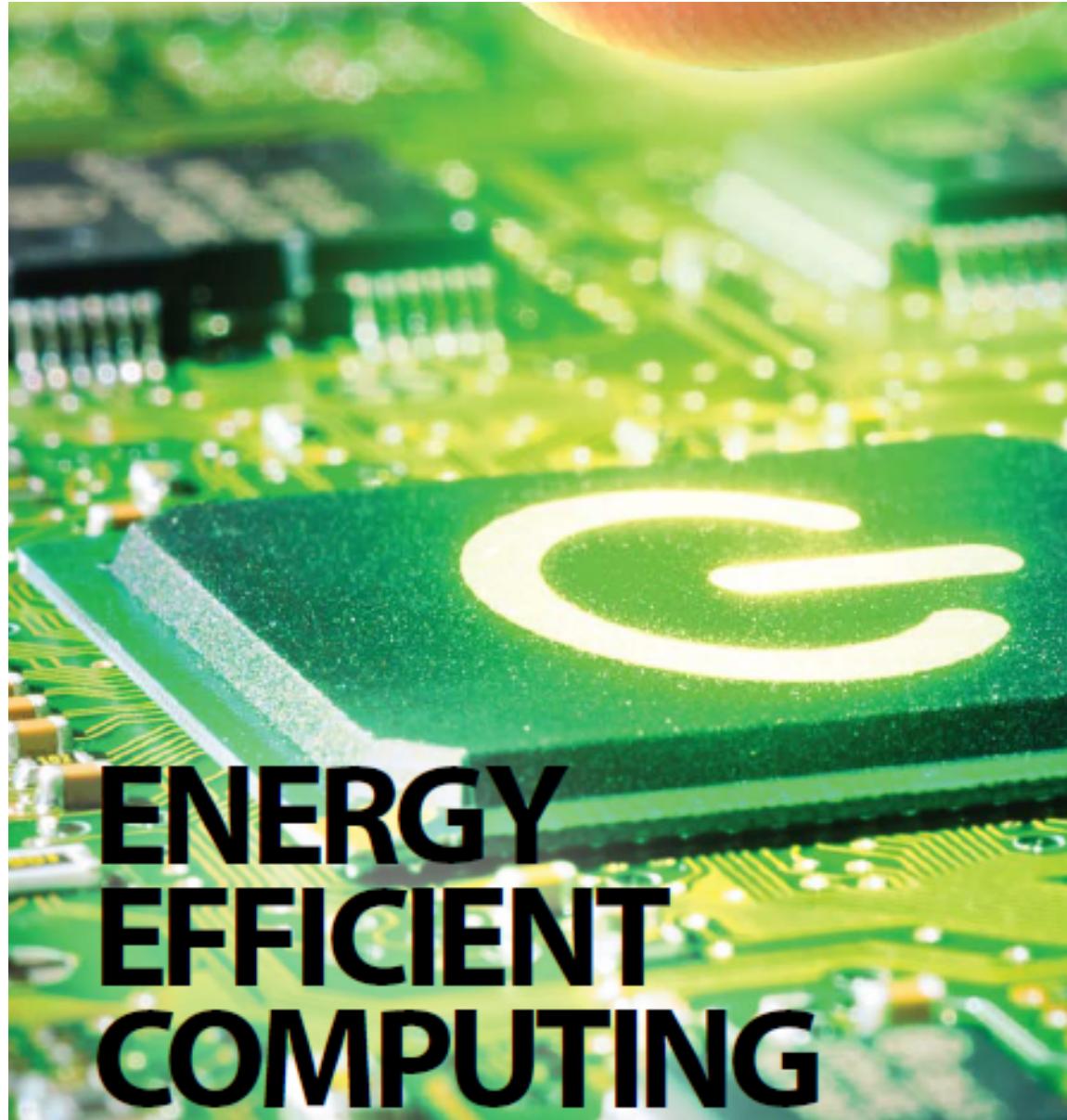


Pictures taken from the Energy Efficient Computing Brochure at:  
[https://connect.innovateuk.org/documents/3158891/9517074/  
Energy%20Efficient%20Computing%20Magazine?version=1.0](https://connect.innovateuk.org/documents/3158891/9517074/Energy%20Efficient%20Computing%20Magazine?version=1.0)



Pictures taken from the Energy Efficient Computing Brochure at:

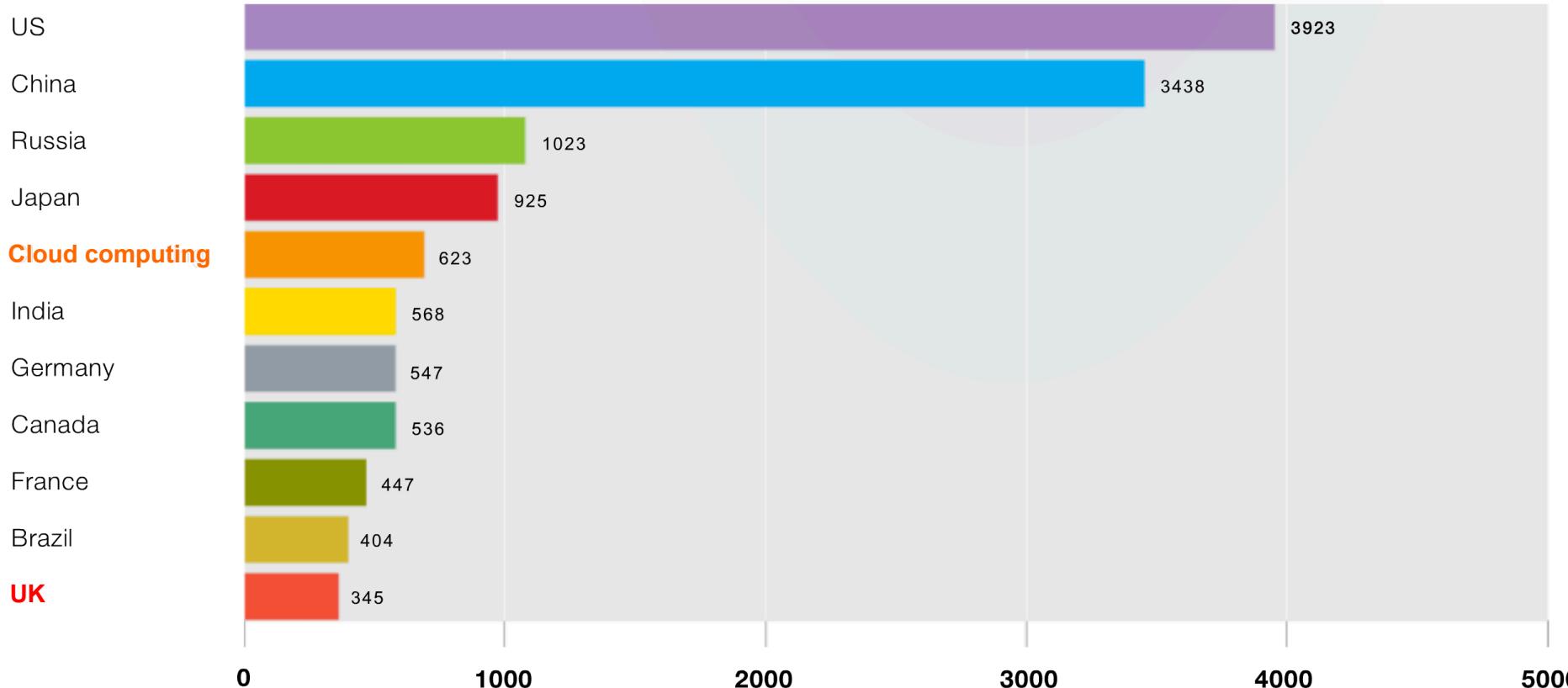
<https://connect.innovateuk.org/documents/3158891/9517074/Energy%20Efficient%20Computing%20Magazine?version=1.0>



Pictures taken from the Energy Efficient Computing Brochure at:

<https://connect.innovateuk.org/documents/3158891/9517074/Energy%20Efficient%20Computing%20Magazine?version=1.0>

# Electricity Consumption (Billion kwH, 2007)



Greenpeace's Make IT Green Report, 2010.

<http://www.greenpeace.org/international/en/publications/Campaign-reports/Climate-Reports/How-Clean-is-Your-Cloud/>

“Despite improved energy efficiency, **energy consumption through electronic devices will triple until 2030** because of a massive rise in overall demand.”



# Crowds in St. Peter's Square

2005



<http://www.spiegel.de/panorama/big-889031-473266.html>

2013



# NEWS TECHNOLOGY

[Home](#) | [World](#) | [UK](#) | [England](#) | [N. Ireland](#) | [Scotland](#) | [Wales](#) | [Business](#) | [Politics](#) | [Health](#) | [Education](#) | [Sci/Enviro](#)

19 March 2012 Last updated at 17:34

1.7K Share

## Free mobile apps 'drain battery faster'

**Free mobile apps which use third-party services to display advertising consume considerably more battery life, a new study suggests.**

Researchers used a special tool to monitor energy use by several apps on Android and Windows Mobile handsets.

**Findings suggested** that in one case 75% of an app's energy consumption was spent on powering advertisements.

Report author Abhinav Pathak said app makers must take energy optimisation more seriously.



Like many games, Angry Birds has a free version supported by targeted advertising

# Energy Aware Computing

# Energy Efficiency of ICT

arChitecture

Blocks

gAtes

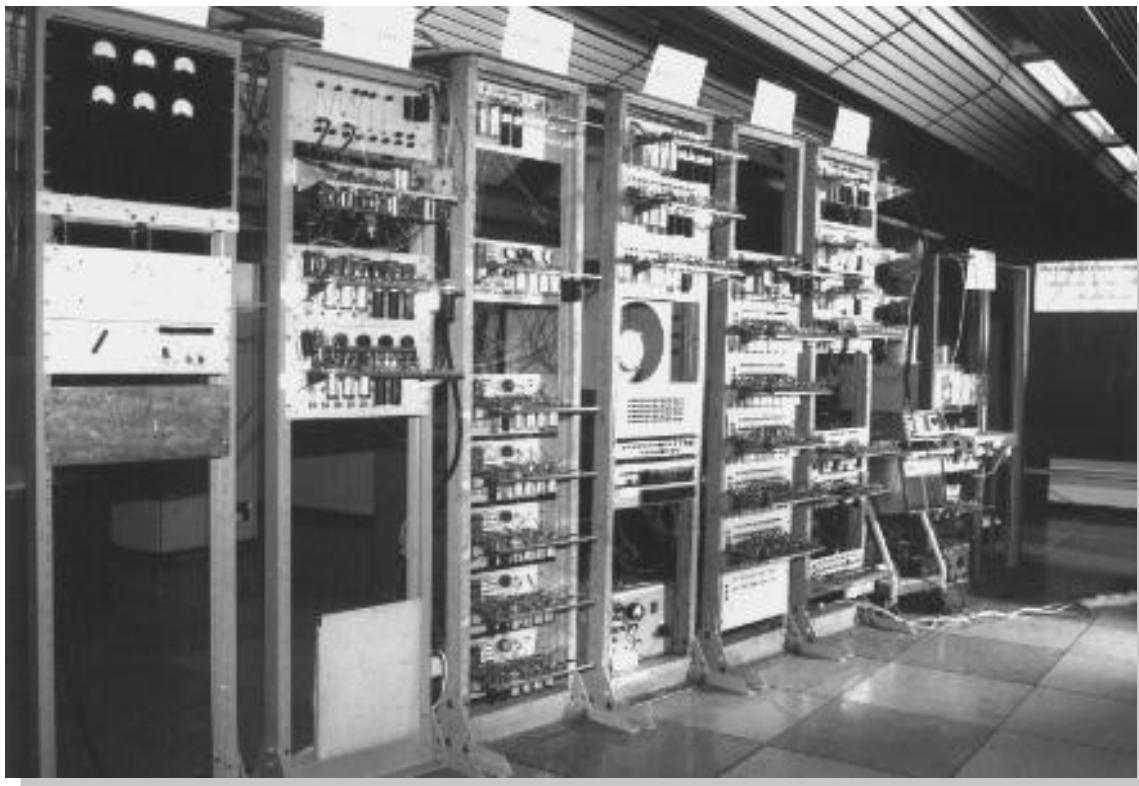


# A historical perspective (based on an inspiring talk by Steve Furber)



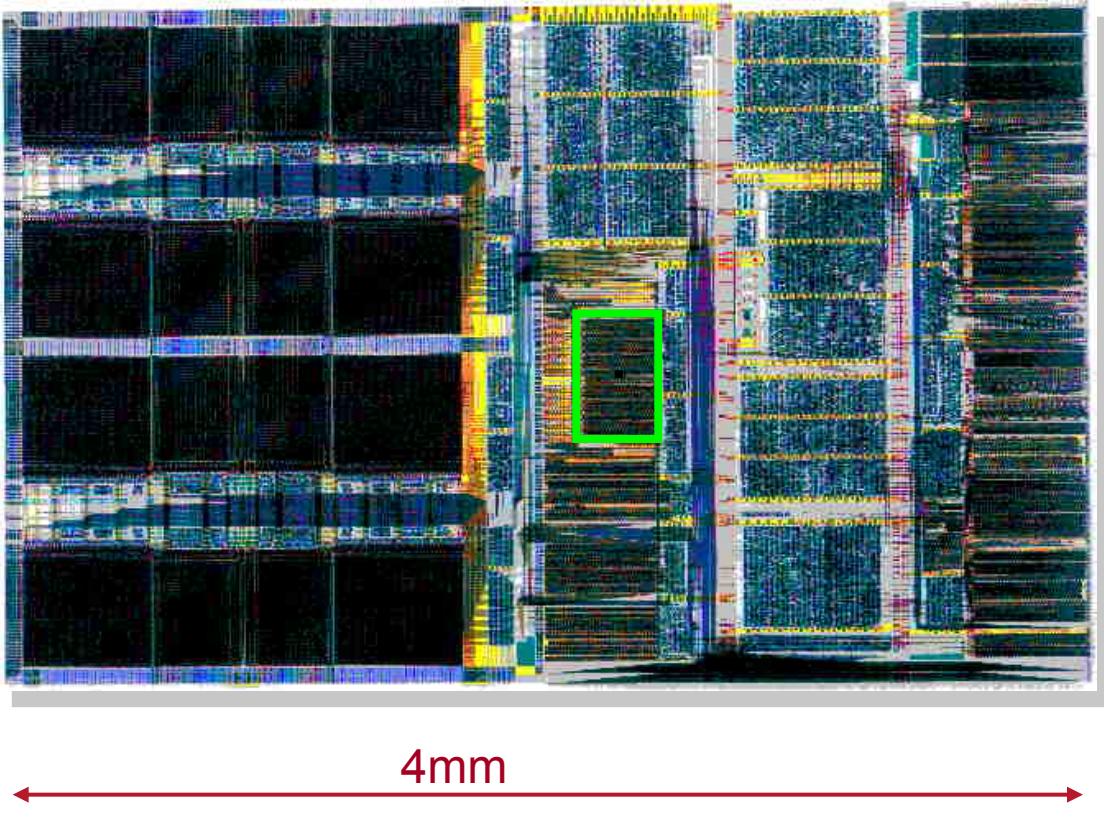
# The Baby (1948)

---



- filled a medium-sized room
- executed **700** instructions per second

# The ARM968 (2005)



- fills  $0.4\text{mm}^2$  of silicon
- executes **200,000,000** instructions per second
- ~300,000 times more than the Baby!

# ~60 years of progress

---

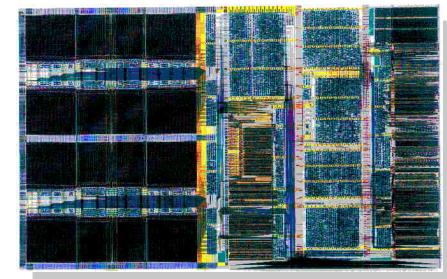
- **Baby, 1948:**

- filled a medium-sized room
- used 3.5 kW of electrical power
- executed 700 instructions per second



- **ARM968, 2005:**

- fills  $0.4\text{mm}^2$  of silicon (130nm)
- uses 20 mW of electrical power
- executes 200,000,000 instructions per second



# Energy efficiency

---

- **Baby:**
  - 5 Joules per instruction
- **ARM968:**
  - 100 pico Joules per instruction



*(James Prescott Joule born  
Salford, 1818)*

# Energy efficiency

- **Baby:**
  - 5 Joules per instruction
- **ARM968:**
  - 0.000 000 000 1 Joules per instruction

50,000,000,000 times  
better than Baby!



(James Prescott Joule born Salford, 1818)

# 10 more years of progress

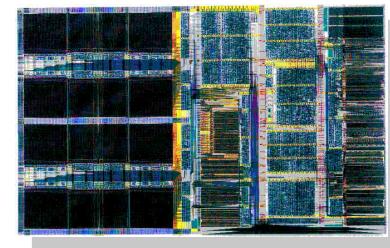
- **Baby, 1948:**

- filled a medium-sized room
- used 3.5 kW of electrical power
- executed 700 instructions per second



- **ARM968, 2005**

- fills  $0.4\text{mm}^2$  of silicon (130nm)
- uses 20 mW of electrical power
- executes 200,000,000 instructions per second



- **ARM Cortex-A35, 2015**

- smallest area configuration  $<0.25\text{mm}^2$
- uses less than 4 mW of electrical power at 100 MHz
- executes  $\sim 210,000,000$  instructions per second



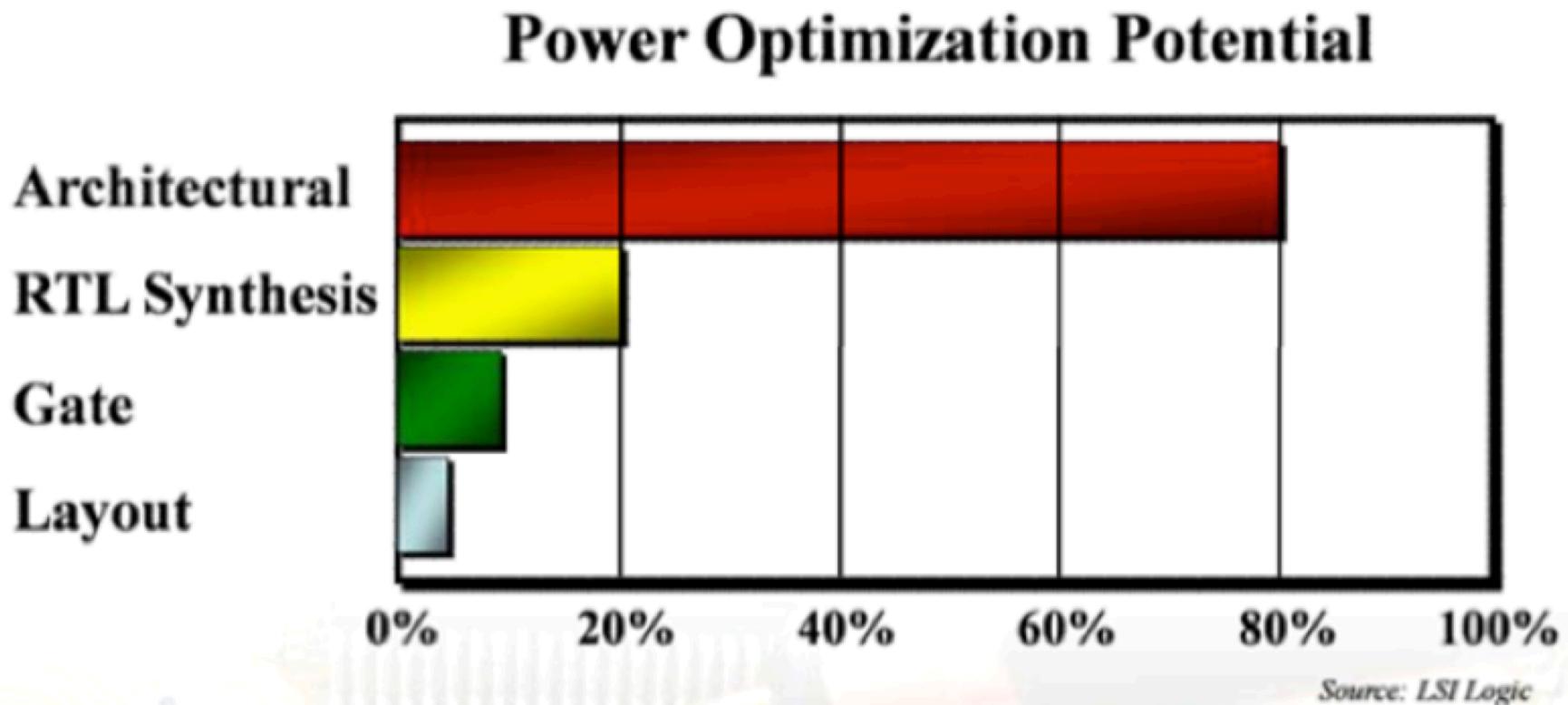
# Hardware Design

---

- Power management largely in domain of Hardware Design
  - Considerations to minimize/optimize
    - Dynamic (switching) and static (leakage) power
  - On-chip power management
    - Modes: on, standby, suspend, sleep, off
- Development of low power electronics

**Where can the greatest savings be made?**

# Greater Savings at Higher Levels



## LOW POWER

# Lack of software support marks the low power scorecard at DAC

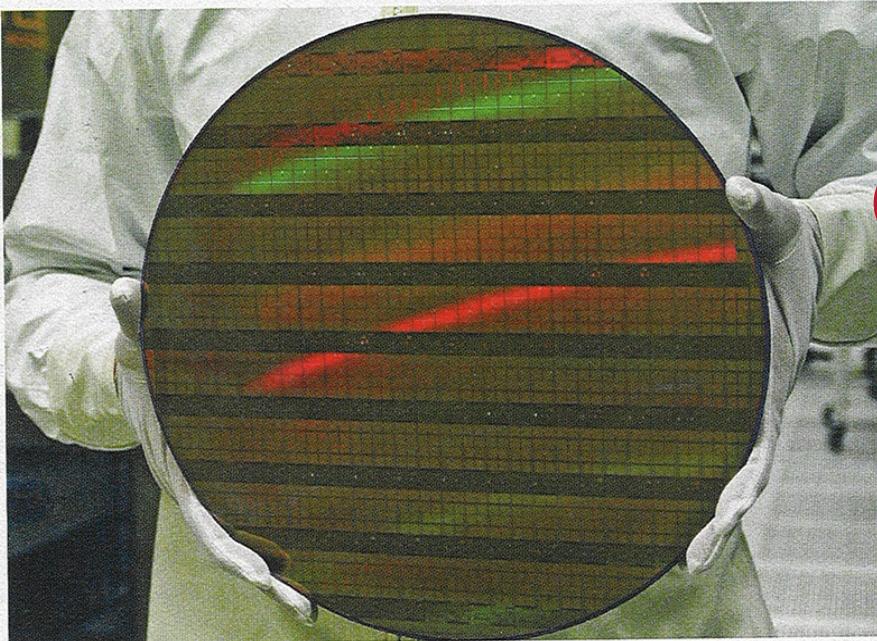
One of the panels at the Design Automation Conference (DAC), which took place in California in early June, set out to get an idea of how well the industry is doing at delivering lower-power systems.

It is becoming clear, writes *Chris Edwards*, that the system level is currently the missing link.

Processes can deliver some gains – and Globalfoundries' Andrew Brotman was able to outline some of the features that the foundry has put into its recently launched low-power high-k, metal gate (HKMG) process.

FinFETs should bring power down as those processes become available, although they are not the only options. But if the software keeps cores active for no good reason, the lower switching power per bit processed won't deliver a realised saving.

In his keynote speech Gadi Singer, vice-president IAG and general manager of the SoC enabling group at Intel Corporation, said that with limited software support, dedicated low-



Intel waits for better low-power software control

power circuitry could save maybe 20% in a typical multimedia-oriented core.

Make the software controlling it

better at controlling the power states and that difference could be three to five times.

During an afternoon panel discuss-

sion Ambrose Low, director of design engineering at Broadcom said: "We have hundreds of knobs in the hardware to turn power down.

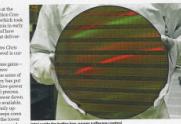
"The question is whether we can take the actual use-cases into consideration and optimise the software to power the logic circuits down. We still have a long way to go."

Ruggero Castagnetti of LSI argued that the desire to do more in software will grow.

"As we see power limits and targets becoming unachievable, customers will be willing to go to that extra step. There is a challenge that needs to be addressed and we have to do more on the systems side," Castagnetti said.

"We should put a challenge to the software designers to see how much power they can save," he added.

Chris Edwards writes the Low-Power Design Blog (enabled by Mentor Graphics) on ElectronicsWeekly.com



Intel's Adrienne Liao, director of design engineering at Temperton Inc., left, and Intel's Brian Duncanson, which took part in the study, look over a photograph of a circuit board under a microscope. They are examining the results of their work to try to prove their theory that software can make the hardware more efficient.

By Michael Kanellos

# Wasted Potential

Huge advances have been made in power-efficient hardware.

**BUT – potential energy savings are wasted by**

- software that does not exploit energy-saving features of hardware;
- poor dynamic management of tasks and resources.

# Energy Efficiency of ICT

alGorithms

soFtware

compilErs

Drivers

arChitecture

Blocks

gAtes



<http://static.datixinc.com/wp-content/uploads/2015/04/7.jpg>



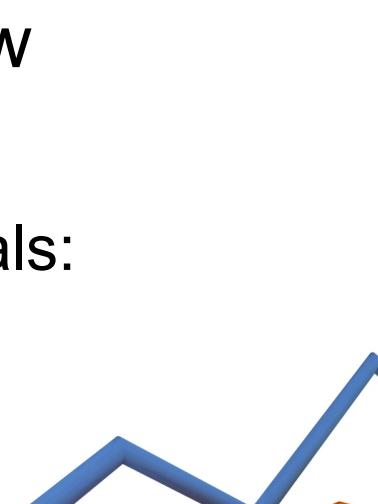
<http://www.clerk.com/cliparts/i/0/5/1/12937499341853355695circuit-board.jpg>

# The Focus is on Software

news

## **Lack of software support marks the low power scorecard at DAC**

**Red dots mark the pollen-gathering center.**

- Software controls the behaviour of the hardware
    - Algorithms and Data Flow
    - Compiler (optimizations)
      - Traditional SW design goals:  
**performance,**  
**performance,**  
**performance**

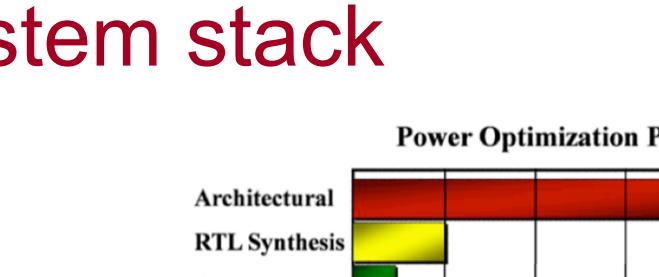


# The Focus is on Software

new

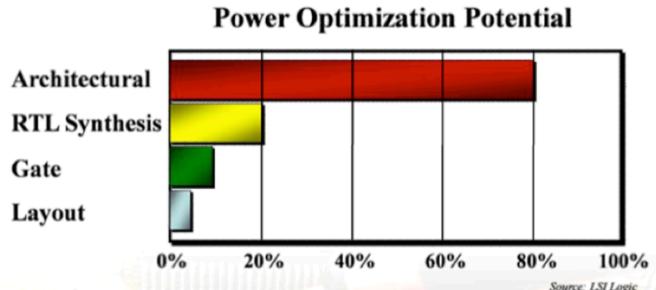
## **Lack of software support marks the low power scorecard at DAC**

Was work for the laser pointer center

- Software engineers often “**blissfully unaware**”
    - Implications of algorithm/code/data on power/energy?
    - Power/Energy considerations
      - at best, secondary design goals
  - BUT the **biggest savings** can be gained from optimizations at the higher levels of abstraction in the system stack
    - Algorithms,
    - Data and
    - SW

The chart is titled "Power Optimization Potential" and shows four categories of optimization levels on the y-axis: Architectural, RTL Synthesis, Gate, and Layout. The x-axis represents percentage from 0% to 100% in increments of 20%. Each category has a colored bar indicating its potential: Architectural is red (approx. 80%), RTL Synthesis is yellow (approx. 20%), Gate is green (approx. 10%), and Layout is light blue (approx. 5%).

Level	Power Optimization Potential (%)
Architectural	~80%
RTL Synthesis	~20%
Gate	~10%
Layout	~5%



## **6.3. SOFTWARE DESIGN FOR LOW POWER**

KAUSHIK ROY AND MARK C. JOHNSON

*School of Electrical and Computer Engineering  
Purdue University  
West Lafayette, Indiana, U.S.A.*

### **1. Introduction**

It is tempting to suppose that only hardware dissipates power, not software. However, that would be analogous to postulating that only automobiles burn gasoline, not people. In microprocessor, micro-controller, and digital signal processor based systems, it is software that directs much of the activity of the hardware. Consequently, the software can have a substantial impact on the power dissipation of a system. Until recently, there were no efficient and accurate methods to estimate the overall effect of a software design on power dissipation. Without a power estimator there was no way to reliably optimize software to minimize power. Since 1993, a few researchers have begun to crack this problem. In this chapter, you will learn

# Aligning SW Design Decisions with Energy Efficiency as Design Goal

Key steps\*:

- “Choose the **best algorithm** for the problem at hand and make sure it fits well with the computational **hardware**. Failure to do this can lead to costs far exceeding the benefit of more localized power optimizations.
- Minimize **memory size** and expensive **memory accesses** through algorithm transformations, efficient mapping of data into memory, and optimal use of memory bandwidth, registers and cache.
- Optimize the **performance** of the application, making **maximum use of available parallelism**.
- Take advantage of **hardware support for power management**.
- Finally, select instructions, sequence them, and order operations in a way that **minimizes switching** in the CPU and datapath.”

\* Kaushik Roy and Mark C. Johnson. 1997. “Software design for low power”. In *Low power design in deep submicron electronics*, Wolfgang Nebel and Jean Mermet (Eds.). Kluwer Nato Advanced Science Institutes Series, Vol. 337. Kluwer Academic Publishers, Norwell, MA, USA, pp 433-460.

# How much?





<http://scottiebales.com/wp-content/uploads/2013/05/transparecny-green.jpg>

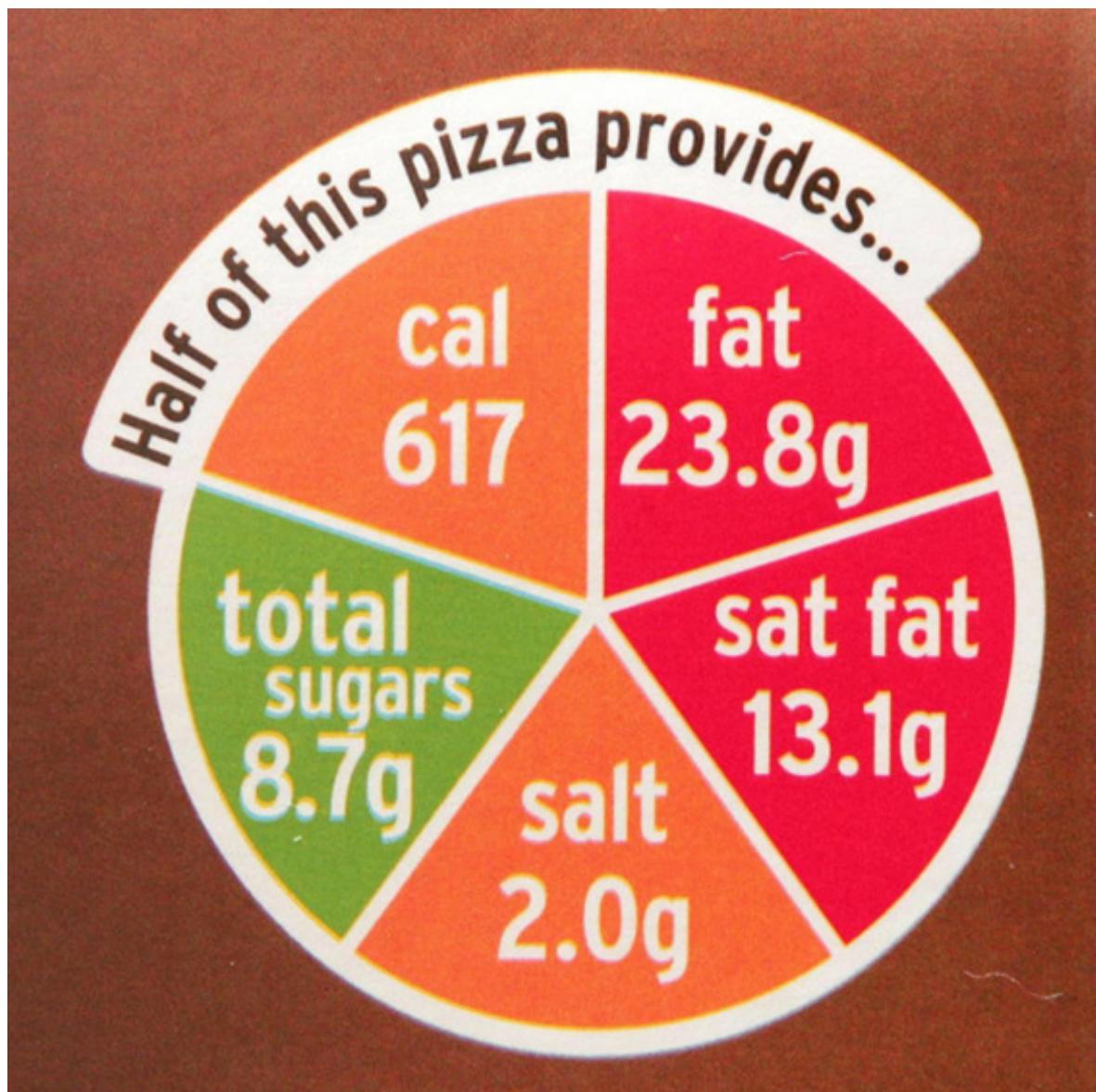
# Energy Transparency

# Energy Transparency

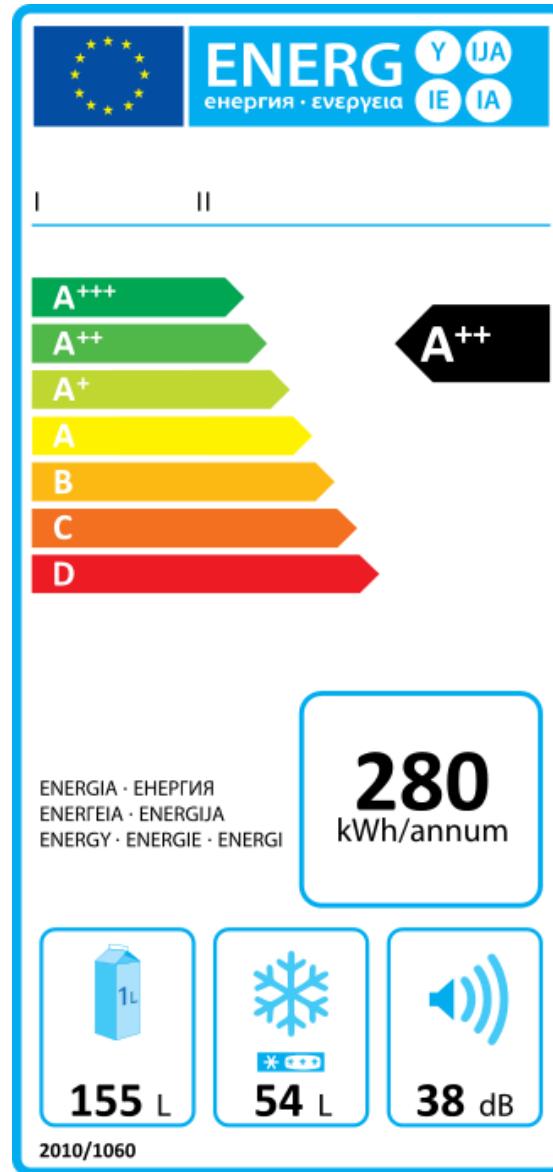
Information on energy usage is available for programs:

- ideally without executing them, and
- at all levels from machine code to high-level application code.

# Transparency



# Transparency



# Transparency

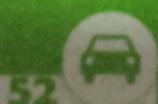
CONVALIDA - VALIDATION

**Complimenti, con la scelta del treno hai contribuito a risparmiare al pianeta emissioni di CO<sub>2</sub>**

Ad esempio, confronta i kg di CO<sub>2</sub> emessi in media\* per un passeggero che viaggia sulle tratte:



Napoli - Milano



Roma - Venezia

\* Dati da elaborazione ENEA (riferimento anno 2008)

\*\* Valore risparmiato per passeggero rispetto alla media tra auto ed aereo



Il contratto di trasporto è disciplinato dalle condizioni generali di trasporto.

## CONDIZIONI DI TRASPORTO - TRENITALIA

Le "Condizioni Generali di trasporto di Trenitalia" sono disponibili presso le Biglietterie di Trenitalia, le agenzie di viaggio e il sito [www.trenitalia.com](http://www.trenitalia.com).

**Attenzione:** Salvo il caso del "biglietto global" (biglietto composto da più titoli di viaggio), i titoli di viaggio non validati o non convalidati incorrono nel pagamento della tariffa minima. I titoli di viaggio con biglietto non convalidato incorrono nel pagamento della tariffa minima. I titoli di viaggio con biglietto per mancanza o guasto delle obbligazioni di viaggio non convalidato non convaliderà il biglietto senza applicare alcuna sgradita.

## MODALITA' DI CONVALIDA DEL BIGLIETTO

I biglietti per treni regionali e gli abbonamenti non sono validati. Per tali titoli di viaggio la validità del viaggio non viene garantita. I titoli di viaggio con biglietto non convalidato incorrono nel pagamento della tariffa minima. I titoli di viaggio con biglietto per mancanza o guasto delle obbligazioni di viaggio non convalidato non convaliderà il biglietto senza applicare alcuna sgradita.

## VALIDATION OF THE TICKET

Tickets not including seat reservation must be validated. For further information please check our website [www.trenitalia.com](http://www.trenitalia.com) or contact our Customer Assistance customer centres.

**Attenzione:** Non tentare di salire al volo o scendere dal treno al di fuori dei marciapiedi.



MISTO

Carta da fonti gestite  
in maniera responsabile

FSC® C002683

# Why Energy Transparency?



Energy transparency enables a deeper understanding of how algorithms and coding impact on the energy consumption of a computation when executed on hardware.

# Learning Objectives

---

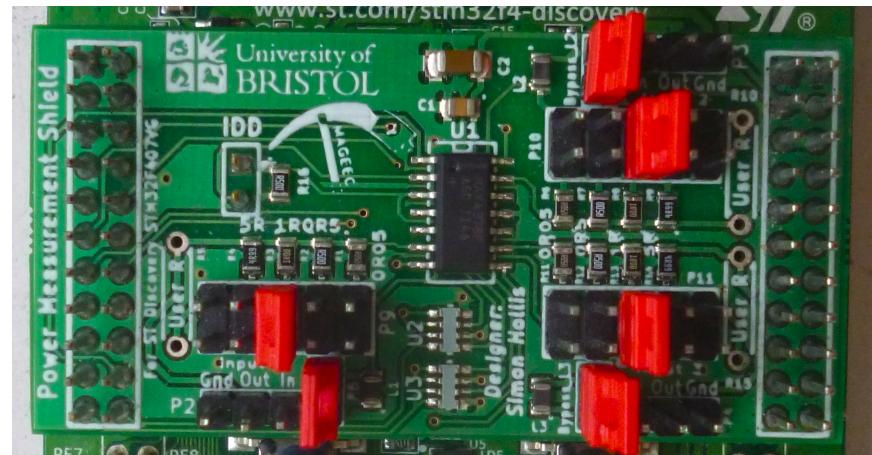
- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- How to measure the energy consumed by software
- How to estimate the energy consumed by software *without measuring*
- How to construct energy consumption models

# Learning Objectives

---

- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- How to measure the energy consumed by software
- How to estimate the energy consumed by software *without measuring*
- How to construct energy consumption models

# Measuring the Energy Consumption of Computation



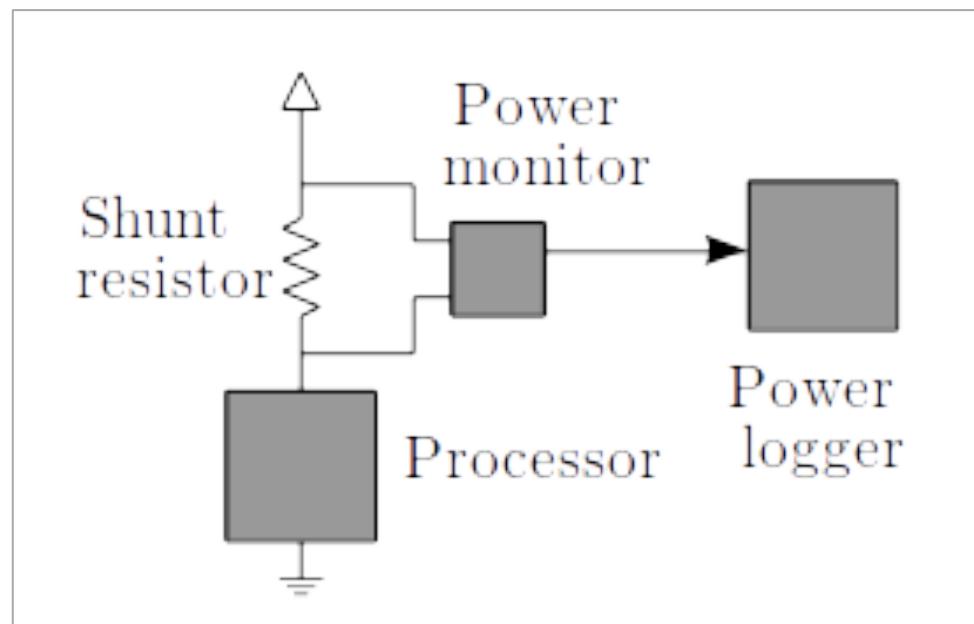
# Measuring Power

Measure voltage drop across the resistor

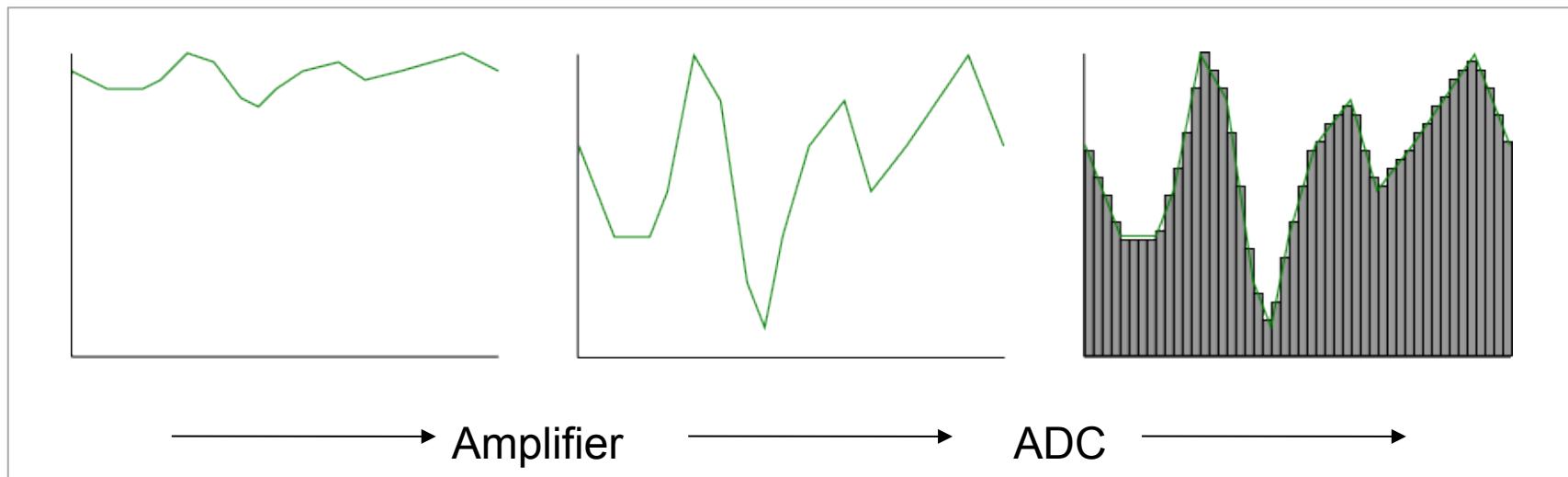
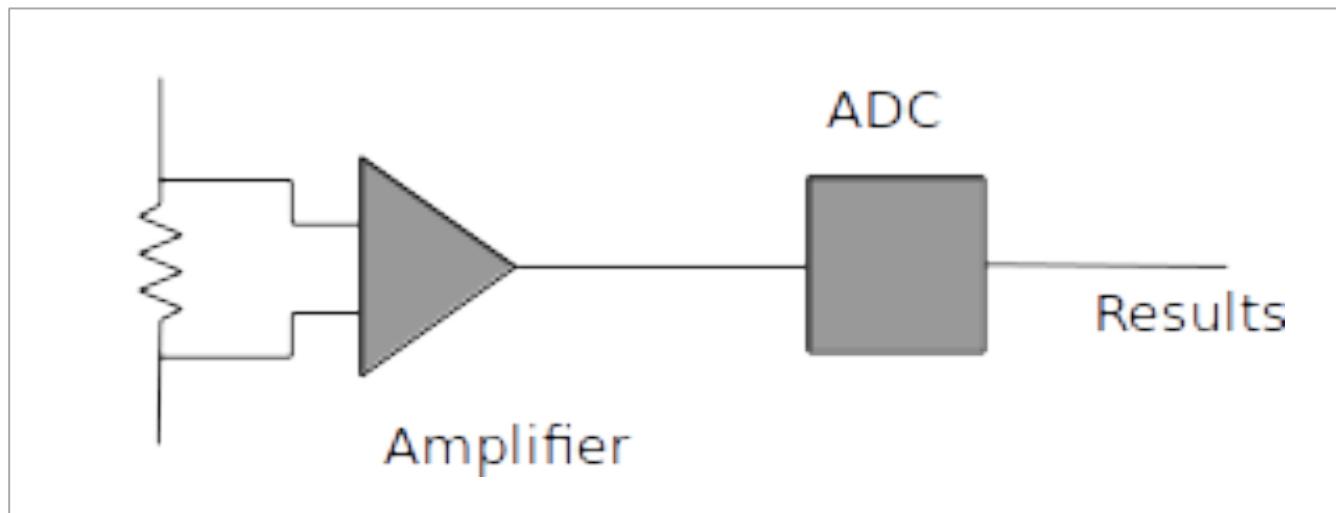
$I = V_{\text{shunt}} / R_{\text{shunt}}$  to find the current.

Measure voltage at one side of the resistor

$P = I \times V$  to calculate the power.



# The Power Monitor



# Measuring Power

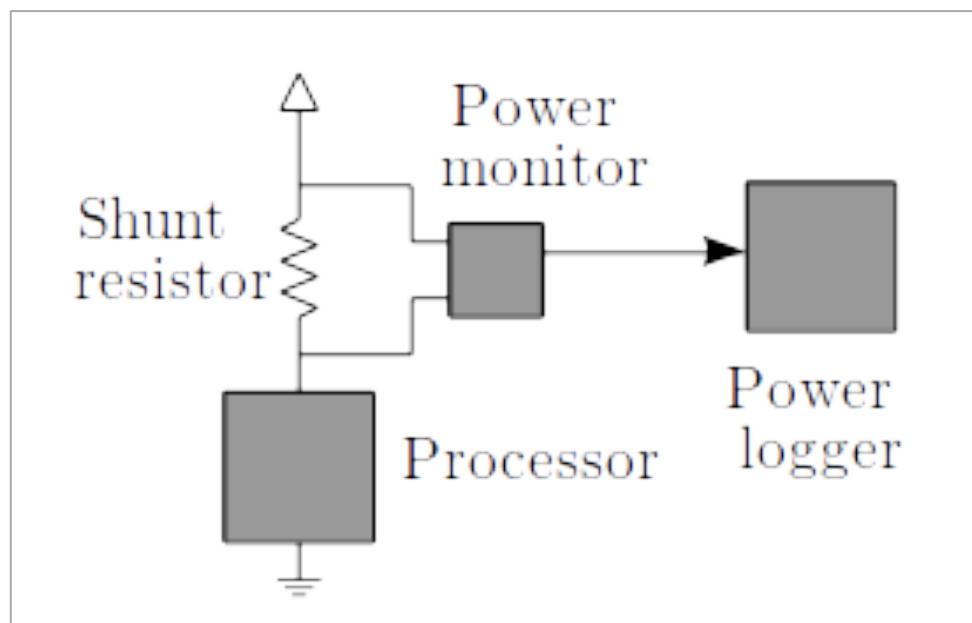
*Repeat frequently, timestamp each sample*

Measure voltage drop across the resistor

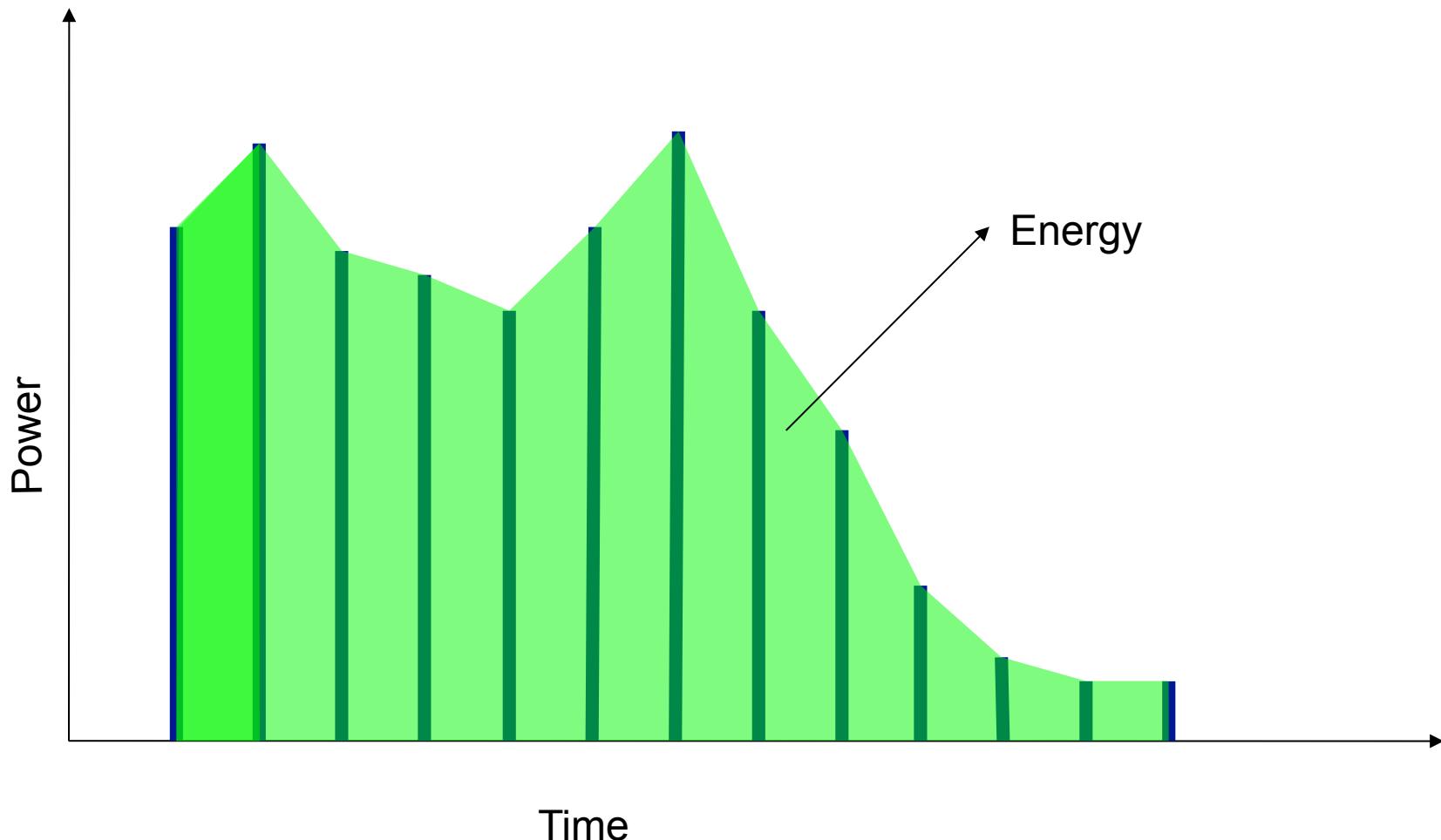
$I = V_{\text{shunt}} / R_{\text{shunt}}$  to find the current

Measure voltage at one side of the resistor

$P = I \times V$  to calculate the power

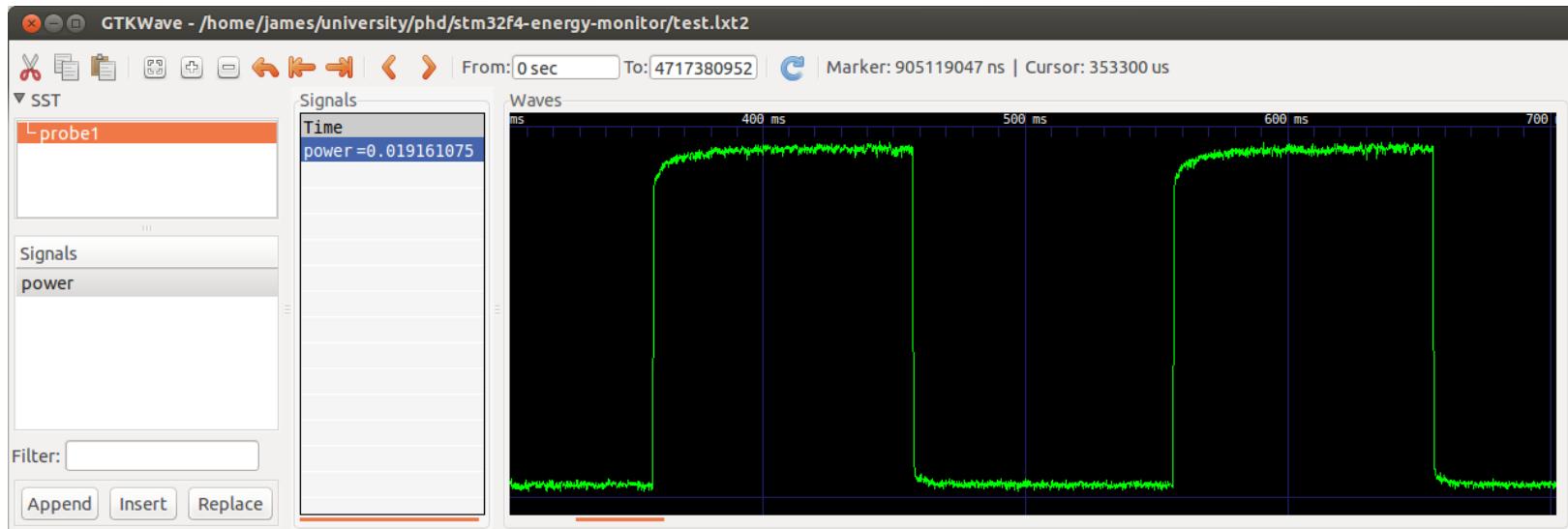


# Measuring Energy

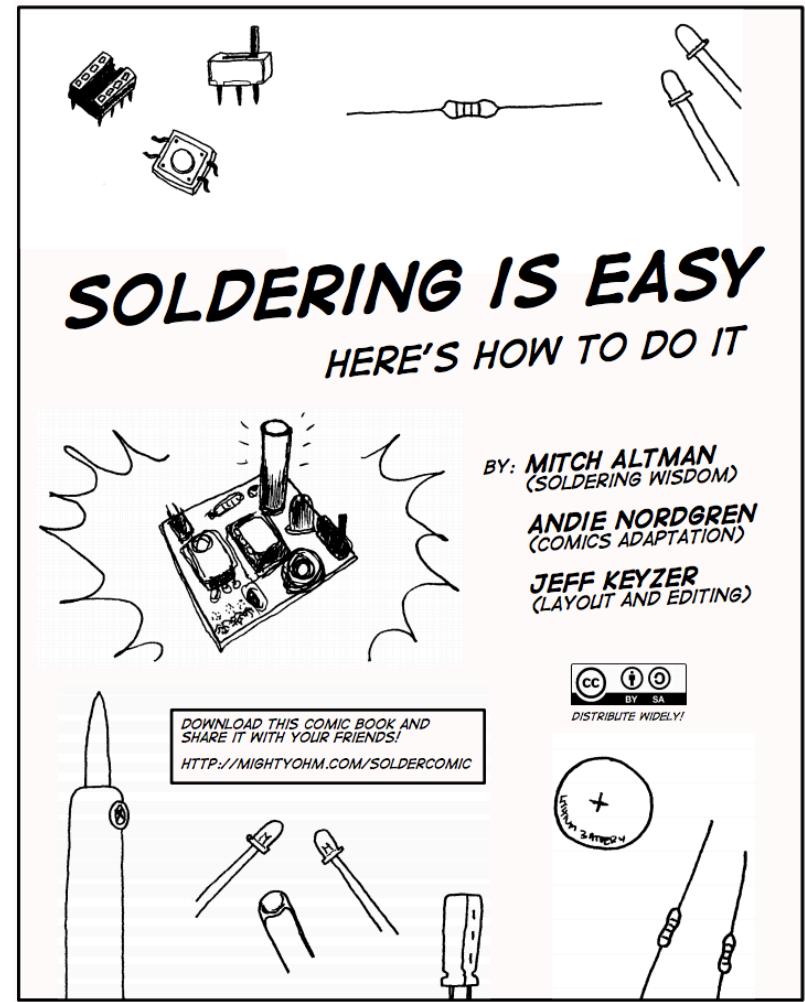
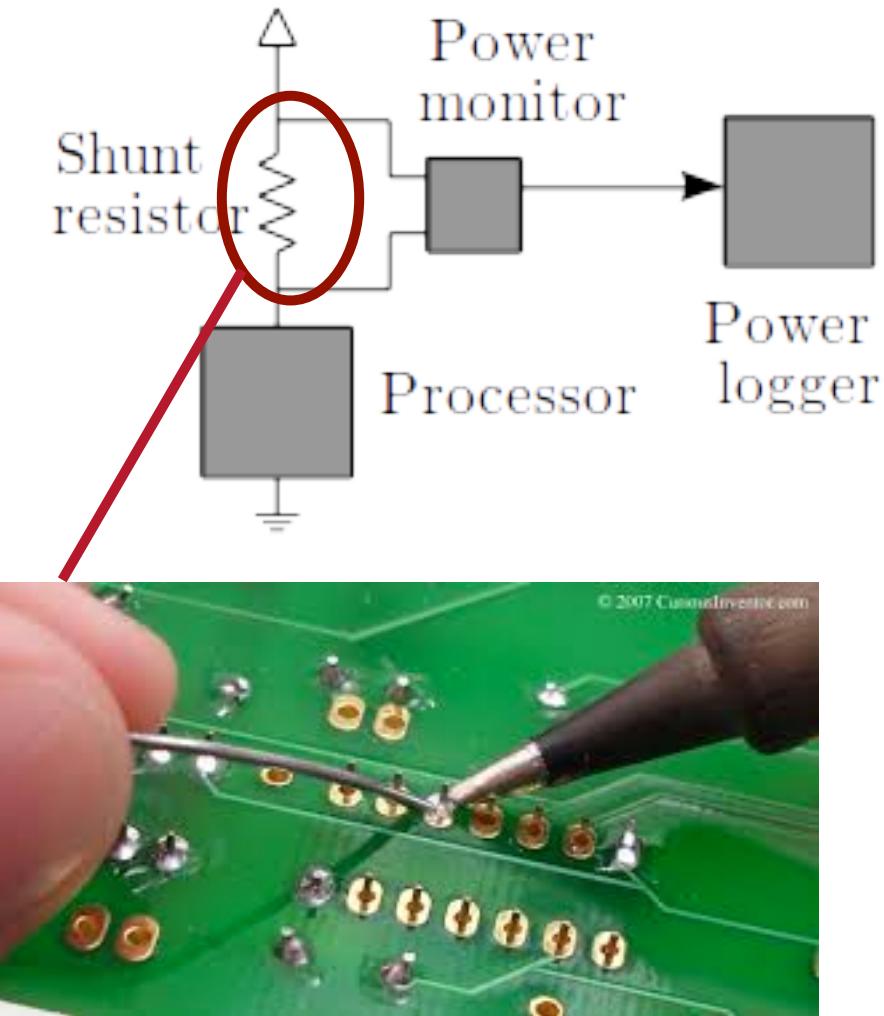


# How much data?

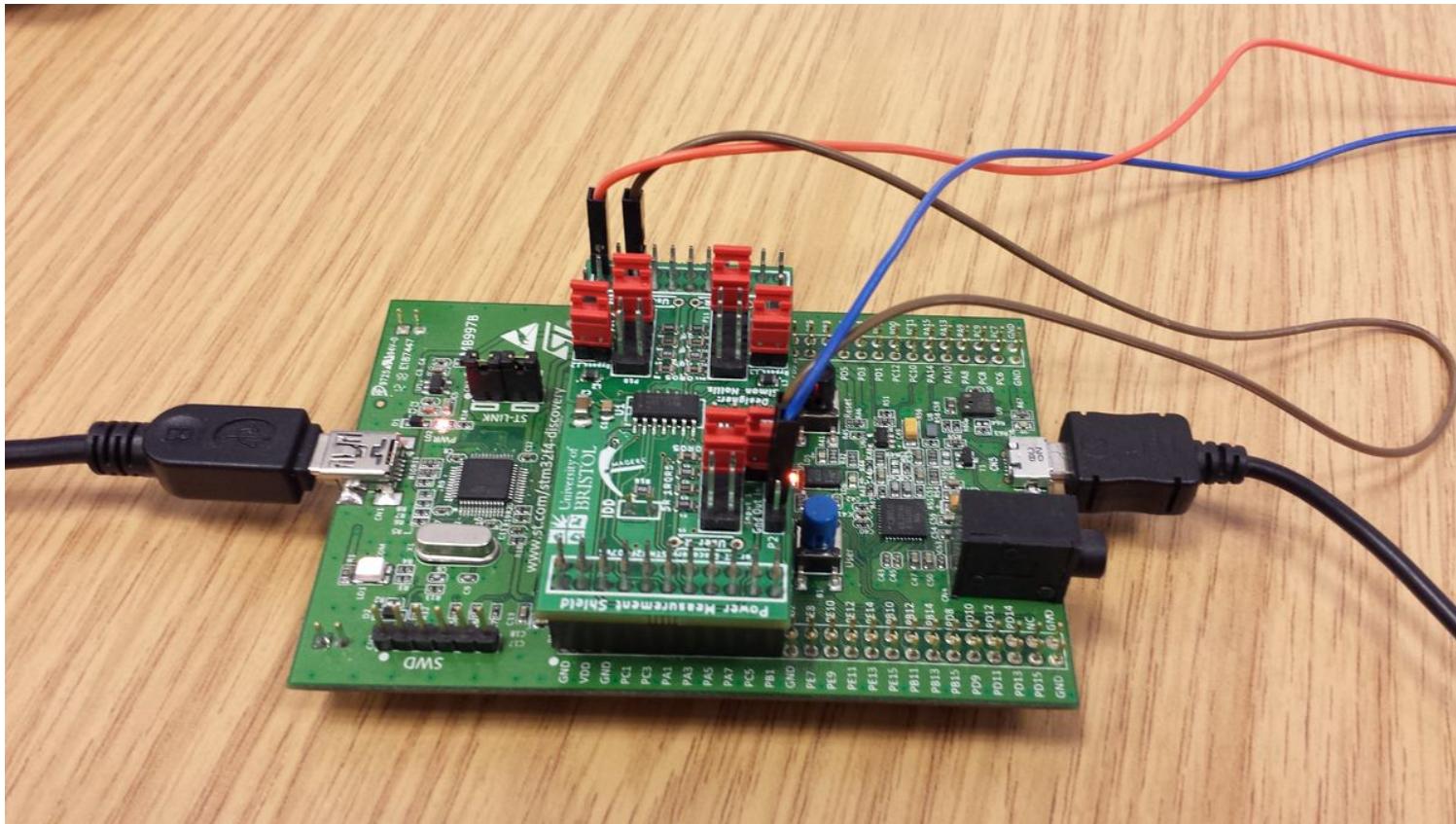
Currently 500,000 Samples/second  
6,000,000 S/s possible in bursts



# The Showstopper 😞

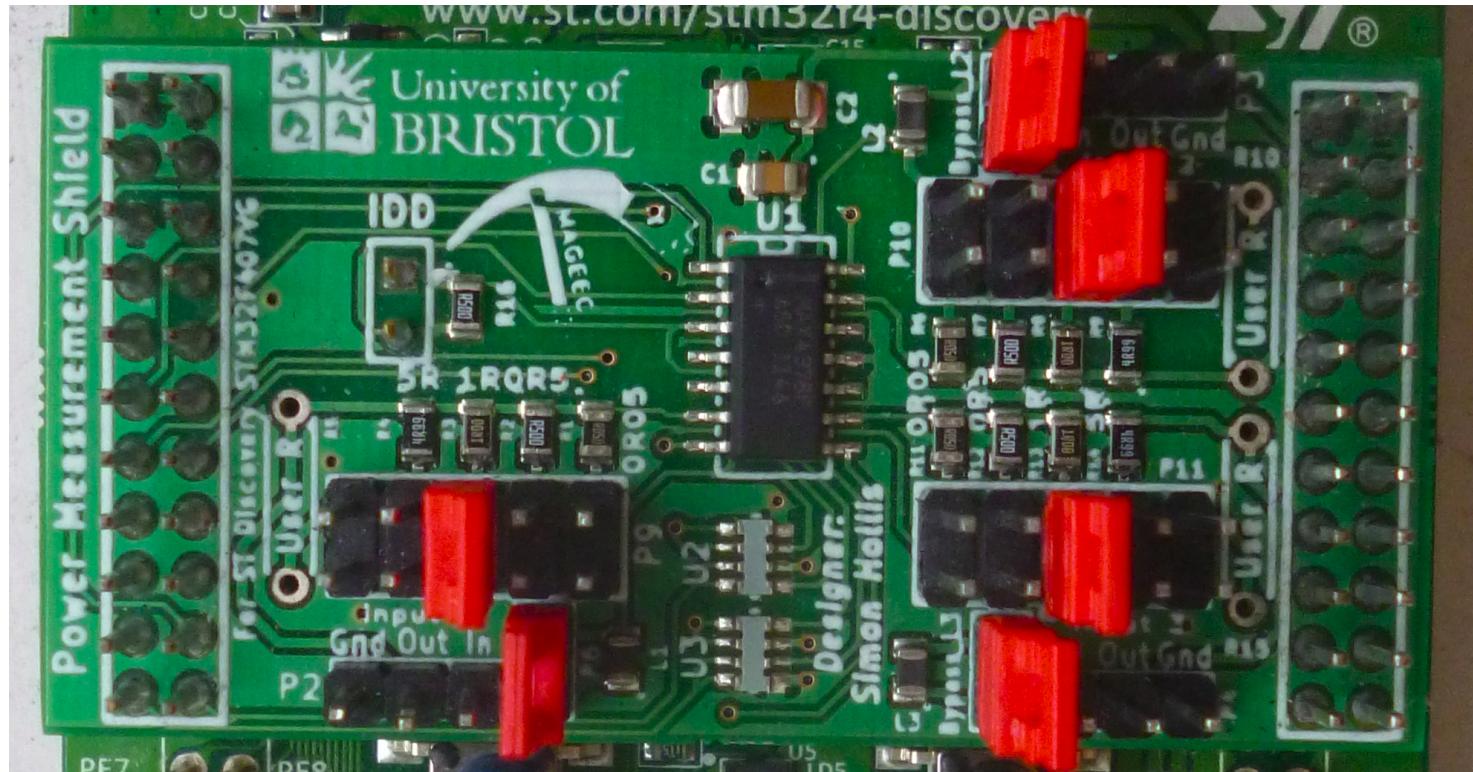


# Open Energy Measurement Board



<http://mageec.org/>

# Open Energy Measurement Board



<http://mageec.org/>

# Open Energy Measurement Board

 Ground Electronics

---

[Home](#)   [Blog](#)   [Catalog](#)   [About Us](#)

---

[Home](#) > [Products](#) > **MAGEEC Energy Measurement Kit**



**MAGEEC Energy Measurement Kit**  
£43.50  
[Add to cart](#)

The MAGEEC WAND is capable of measuring energy consumption at 3 independent points and with simultaneous measurement of targets at 2,000,000 samples/second.

The platform is comprised of an ARM Cortex M4-based STM32F4DISCOVERY board plus a custom shield, which is connected via USB to a host computer.

The shield, STM32F4DISCOVERY firmware, and a Python framework and applications, were developed as part of the [MAGEEC project](#).

Hardware has been made available to members of the MAGEEC project, other research groups and as part of a [workshop at FOSDEM 2014](#). Embecons have funded the production of a limited number of kits which are now being made generally available at cost. There are no plans to produce any more once these are sold.

For further details, including a bill of materials, see the [WAND Kit GitHub repository](#).

[!\[\]\(bbfaaa077afd377ebc1cfdf7c60be592\_img.jpg\)](#) [!\[\]\(36caca24ac4e5eb2b56386f61bb38c49\_img.jpg\)](#) [!\[\]\(50fc73e5b16f42e0566e8b1c2afd9457\_img.jpg\)](#) [!\[\]\(403fe9015d6fd0758c9d8c67d3c88968\_img.jpg\)](#)

<http://groundelectronics.com/products/mageec-energy-measurement-kit>

# Energy Measurement

## A hands-on session by Steve Kerrison

# Summary: Energy Measurement

---

- We can directly measure the energy consumed during the execution of a program.
- In most cases, specialized hardware and modifications to hardware are required to enable measurement.
- The accuracy of the measurements depends on the sampling frequency, on the measuring hardware and on the characteristics of the target you want to measure.

# Software and energy aware computing

*More power to software developers!*

Kerstin Eder

Design Automation and Verification, Microelectronics  
Verification and Validation for Safety in Robots, Bristol Robotics Laboratory



University of  
BRISTOL



Department of  
COMPUTER SCIENCE



The Royal Academy  
of Engineering



# BREAK

(with the next two slides serving as screen cover  
during the break)

Energy Aware Computing (EACO) research at the UNIVERSITY OF BRISTOL includes both Computer Science and Electronic Engineering, with significant cross-departmental expertise and collaboration in energy monitoring and modelling, static analysis and compilers, processor architectures and embedded multi-core system design.

The EACO Workshop series at the University of Bristol brings together academia and industry to identify and address intellectual challenges in Energy Aware Computing with the aim to reduce the energy consumption of computation. Topics of EACO Workshops span the entire system stack from application software and algorithms, via programming languages, compilers, operating systems, instruction sets and micro architectures to the design of hardware.

University of Bristol contact: Kerstin Eder



The UNIVERSITY OF GLASGOW's James Watt Nanofabrication Centre use micro- and nano-technology research and manufacturing facilities to develop technology including Terahertz optics and Silicon nano-wires, healthcare applications and energy harvesting. The Centre coordinates the Generate Renewable Energy Efficiently using NanoFabricated Silicon (GREEN Silicon) project, where the Seebeck effect is used to produce thermoelectric generators using Si/SiGe heterolayer technology, resulting in more efficient energy harvesting.

University of Glasgow contact: Douglas Paul



TYNDALL NATIONAL INSTITUTE is one of Europe's leading centres in ICT research and development. Applying an "atoms to systems" philosophy, energy research in Tyndall includes advanced concepts for low-power computing and efficient power supplies, energy storage and harvesting solutions, and technologies for wireless sensor networks applied to energy and resource optimisation in buildings and factories.

Tyndall coordinates a number of projects in the ICT-Energy field including the MANPOWER, SINAPS, SQWIRE, PowerSWIPE and DEEPEN projects.

Tyndall National Institute contact: Giorgos Fagias

coordinator: Prof. Luca Gammaitoni

NIPS Laboratory, Dipartimento di Fisica  
Università di Perugia  
Via A. Pascoli, 1 - 06123 Perugia, Italy

telephone: +39-0755852733

fax: +39-0755848458

email: luca.gammaitoni@nipslab.org

## LOW ENERGY ICT

The goal of the ICT-Energy project is to create a coordination activity among researchers working on energy reduction in ICT from Nanoscale Devices to Exascale Computing.

By bringing together the Toward Zero-Power ICT community with the MINECC (MINimizing Energy Consumption of Computing) community this project enables a concerted effort to lower energy consumption across the ICT sector.

Our aim is to assess the impact of existing research efforts and propose measures to increase the visibility of ICT-Energy related initiatives to the scientific community, targeted industries and to the public at large through the exchange of information, dedicated networking events, education and media campaigns.

[www.ict-energy.eu](http://www.ict-energy.eu)



Coordinating research efforts towards

The UNIVERSITY OF PERUGIA's Noise In Physical Systems (NIPS) Lab studies the effects of fluctuations in electrical fields, heat, sound and other mediums. This has led to the development of novel energy harvesting and noise sensing devices.

The NIPS Laboratory coordinates the LANDAUER project where the operation of basic physical switches below the Landauer limit is studied to investigate conceptually new devices and novel computing paradigms with radically improved power efficiency.

University of Perugia contact: Luca Gammaitoni



ROSKILDE UNIVERSITY's Programming, Logic and Intelligent Systems (PLIS) group focus on the theoretical aspects of programming languages and their applications. PLIS has significant expertise in software verification, program analysis and transformation.

The PLIS group coordinates the Whole Systems Energy Transparency (ENTRA) project where advanced program analysis and energy modelling techniques are used to predict the energy consumption of programs early on during software development. This enables energy-aware software engineering.

contact: John Gallagher



The UNIVERSITY OF HEIDELBERG's Engineering Mathematics and Computing Lab (EMCL) applies numerical analysis to optimise the performance and energy consumption of High Performance Computing (HPC) as used in leading edge scientific programming. The EMCL coordinates the EXA2GREEN project which aims to drastically reduce the energy consumed in HPC by developing advanced power consumption monitoring and profiling, and designing a smart, power-aware scheduling technology for HPC.

University of Heidelberg contact: Vincent Heuveline



At the HITACHI CAMBRIDGE LABORATORY (HCL) researchers investigate new designs of micro and optoelectronic devices, based on entirely new concepts, such as single electron logic circuits. Revolutionising the electronic devices used to power Information technology has the potential to cut energy consumption by orders of magnitude.

HCL coordinates the Towards Low Power ICT (TOLOP) project which aims at the realization of novel low power devices (single electron transistors and single atom transistors), including implementation theory and the corresponding design architectures.

HCL contact: David Wilkins

BARCELONA SUPERCOMPUTING CENTER (BSC) uses HPC expertise to develop entirely new system-architecture models for low-energy HPC.

The BSC coordinates the Parallel Distributed Infrastructure for Minimization of Energy (ParadINE) project where radical software-hardware co-design techniques are being developed that are driven by future device characteristics on one side, and by a programming model based on message passing on the other side. This approach is expected to yield dramatic energy savings in heterogeneous distributed systems.

BSC contact: Adrià Ortíz Berlanga



AALBORG UNIVERSITY  
DENMARK

AALBORG UNIVERSITY's Center for Embedded Software Systems (CISS) improve embedded systems development through the use of model-driven design tools. These allow design to be written in a verifiable way, and analyzed for energy consumption and performance.

The CISS coordinates the Self Energy-Supporting Autonomous Computation (SEASATION) project which aims at increasing the scale of systems that are self-supporting by balancing energy harvesting and consumption. The research addresses the challenge of programming systems that reconfigure themselves in response to changing tasks, resources, errors and available energy.

Aalborg University contact: Kim Goldstrand Larsen



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE (EPFL)

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE (EPFL) specialises in embedded and low-power systems, efficiently designed software algorithms and system level optimisations.

EPFL coordinates the PHIDIAS project which proposes the development of an ultra-low power smart bio-sensing wireless body sensor network, making use of new signal processing models and methods for efficient data handling. This enables long term low energy monitoring of bio-signals.

EPFL contact: Pierre Vandergheynst

[www.ict-energy.eu](http://www.ict-energy.eu)



**If you want an ultimate low-power system, then you have to worry about *energy usage at every level in the system design*, and you have to get it right from top to bottom, because any level at which you get it wrong is going to lose you perhaps an order of magnitude in terms of power efficiency.**

The hardware technology has a first-order impact on the power efficiency of the system, but you've also got to have software at the top that avoids waste wherever it can. You need to avoid, for instance, anything that resembles a polling loop because that's just burning power to do nothing.

I think one of the hard questions is whether you can pass the responsibility for the software efficiency right back to the programmer.

**Do programmers really have any understanding of how much energy their algorithms consume?**

I work in a computer science department, and it's not clear to me that we teach the students much about how long their algorithms take to execute, let alone how much energy they consume in the course of executing and how you go about optimizing an algorithm for its energy consumption.

Some of the responsibility for that will probably get pushed down into compilers, but I still think that fundamentally, at the top level, **programmers will not be able to afford to be ignorant about the energy cost of the programs they write.**

What you need in order to be able to work in this way at all is instrumentation that tells you that running this algorithm has this kind of energy cost and running that algorithm has that kind of energy cost.

**You need tools that give you feedback and tell you how good your decisions are.**

Currently the tools don't give you that kind of feedback.

# Dynamic Energy Monitoring for desktop applications

# The EACOF

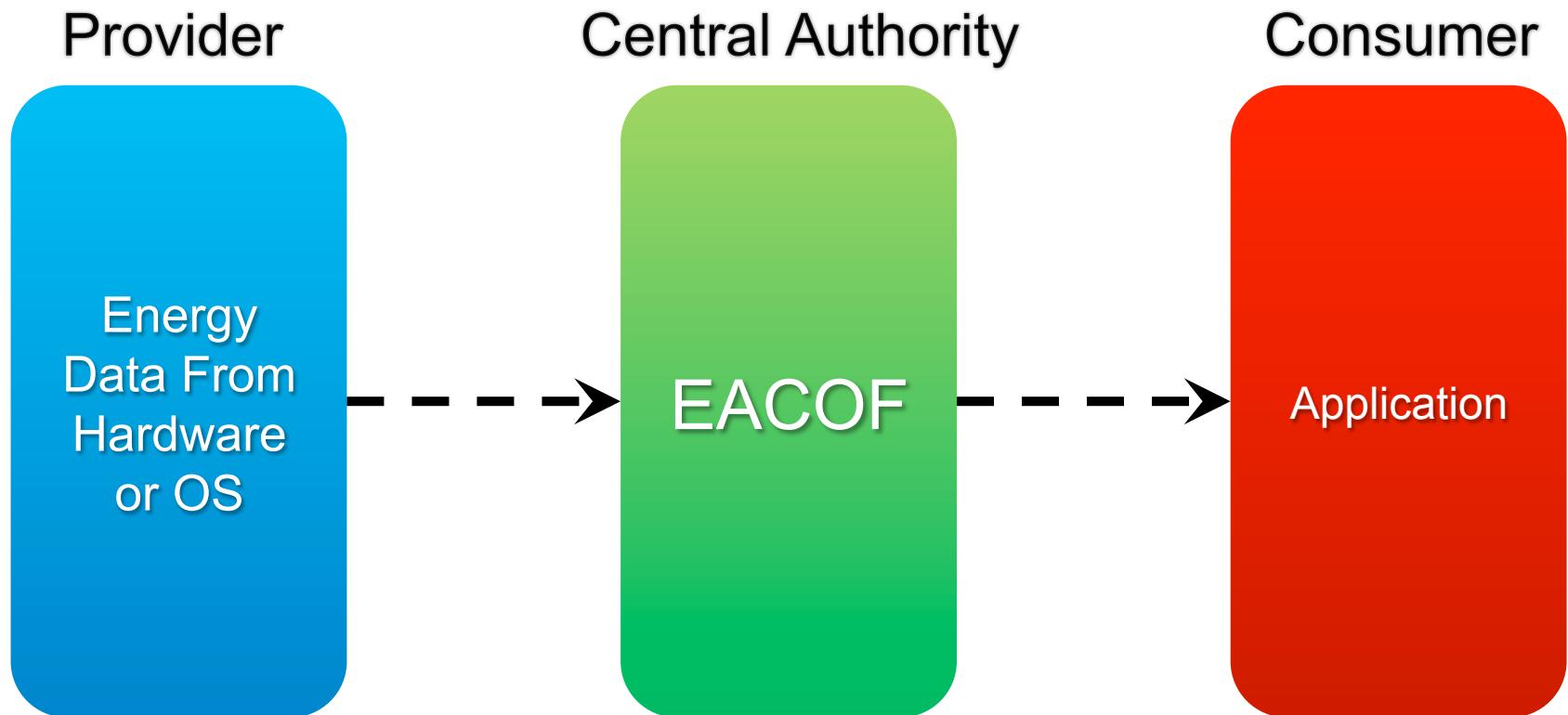
A simple Energy-Aware  
COmputing Framework

<https://github.com/eacof>

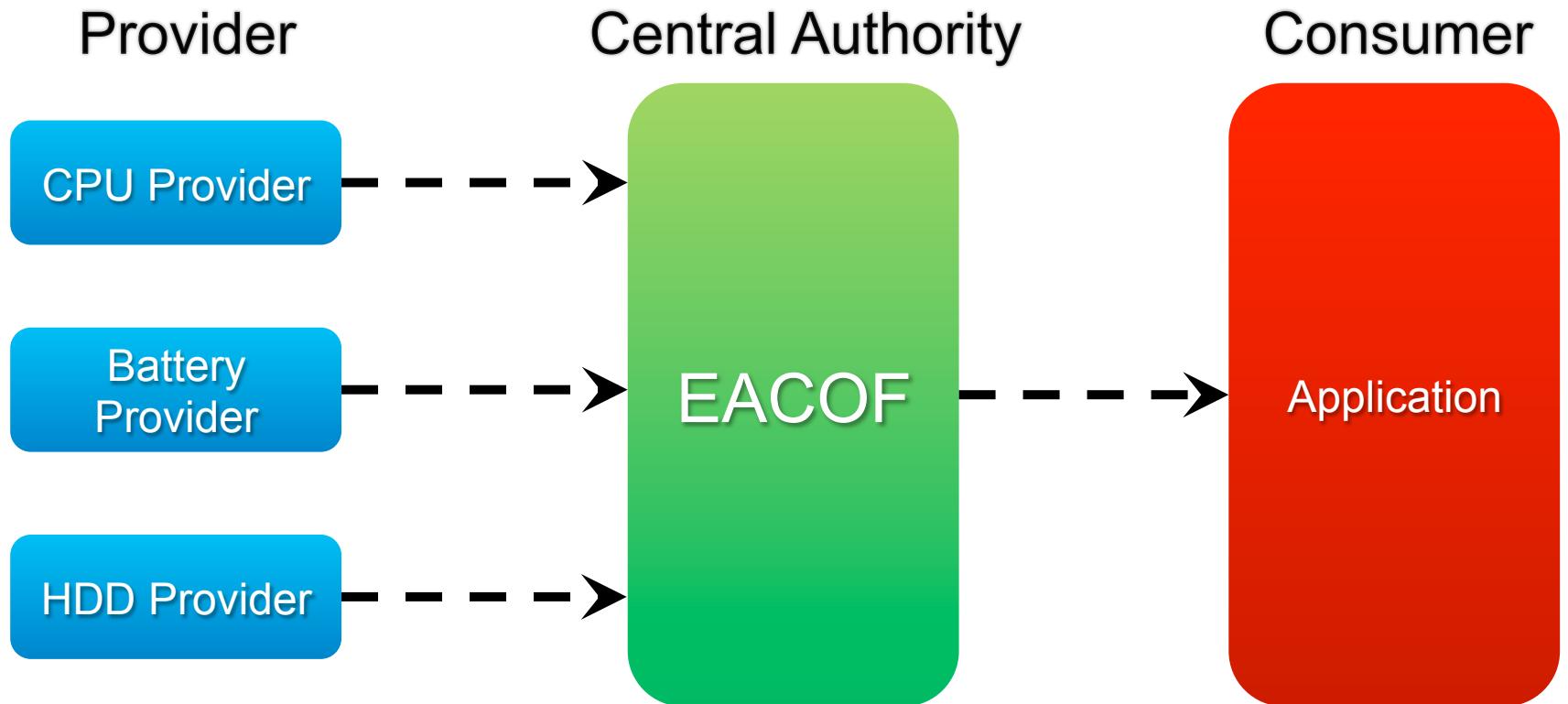


# High Level

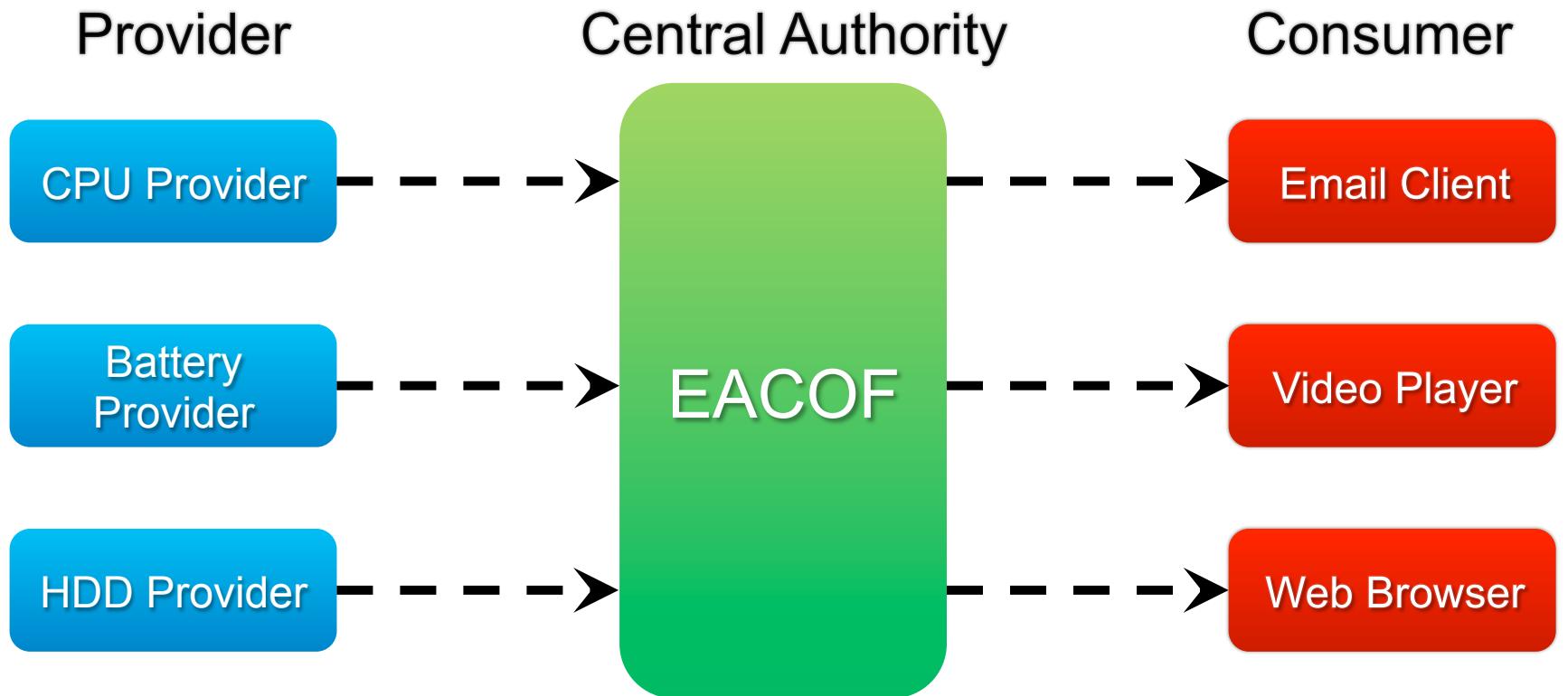
---



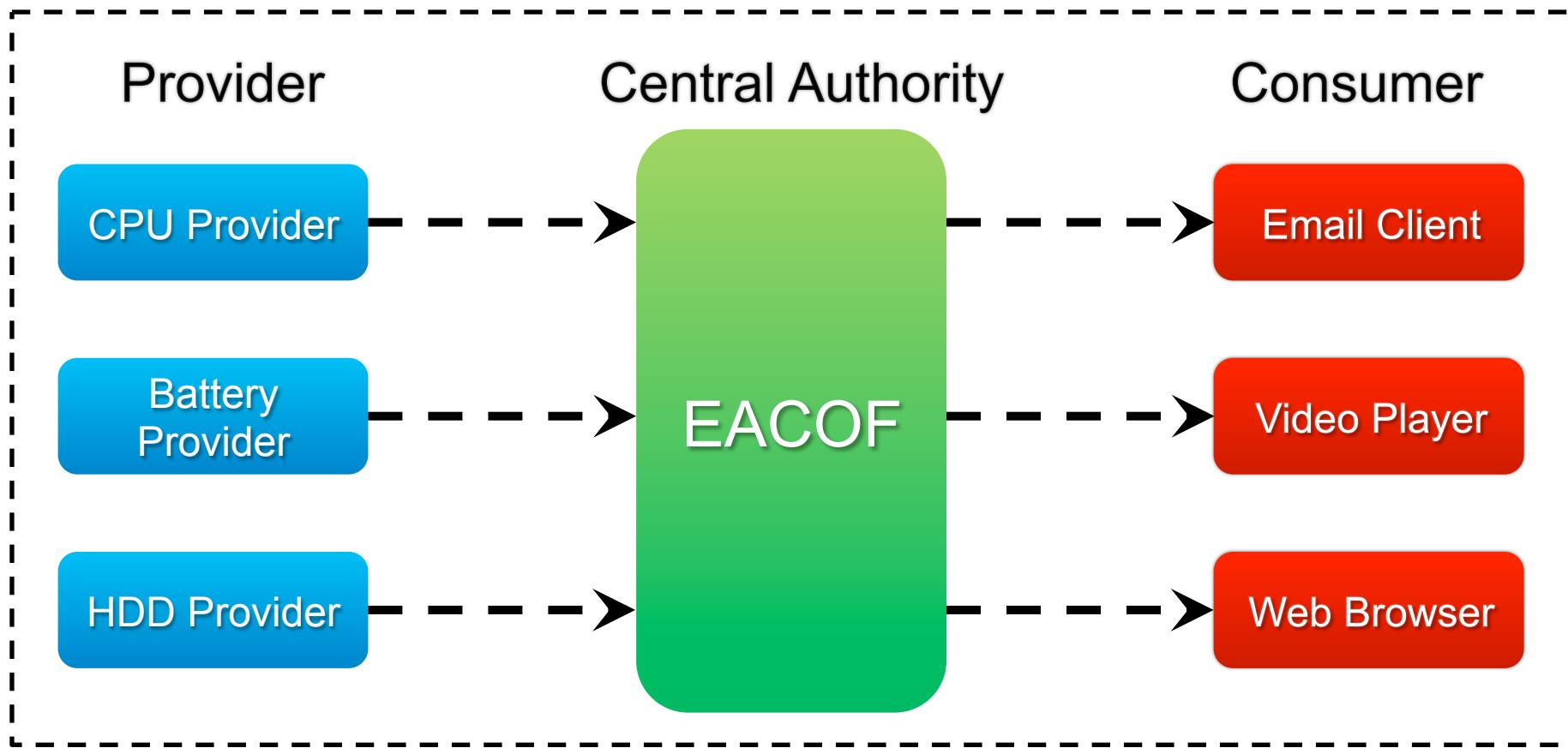
# Providers



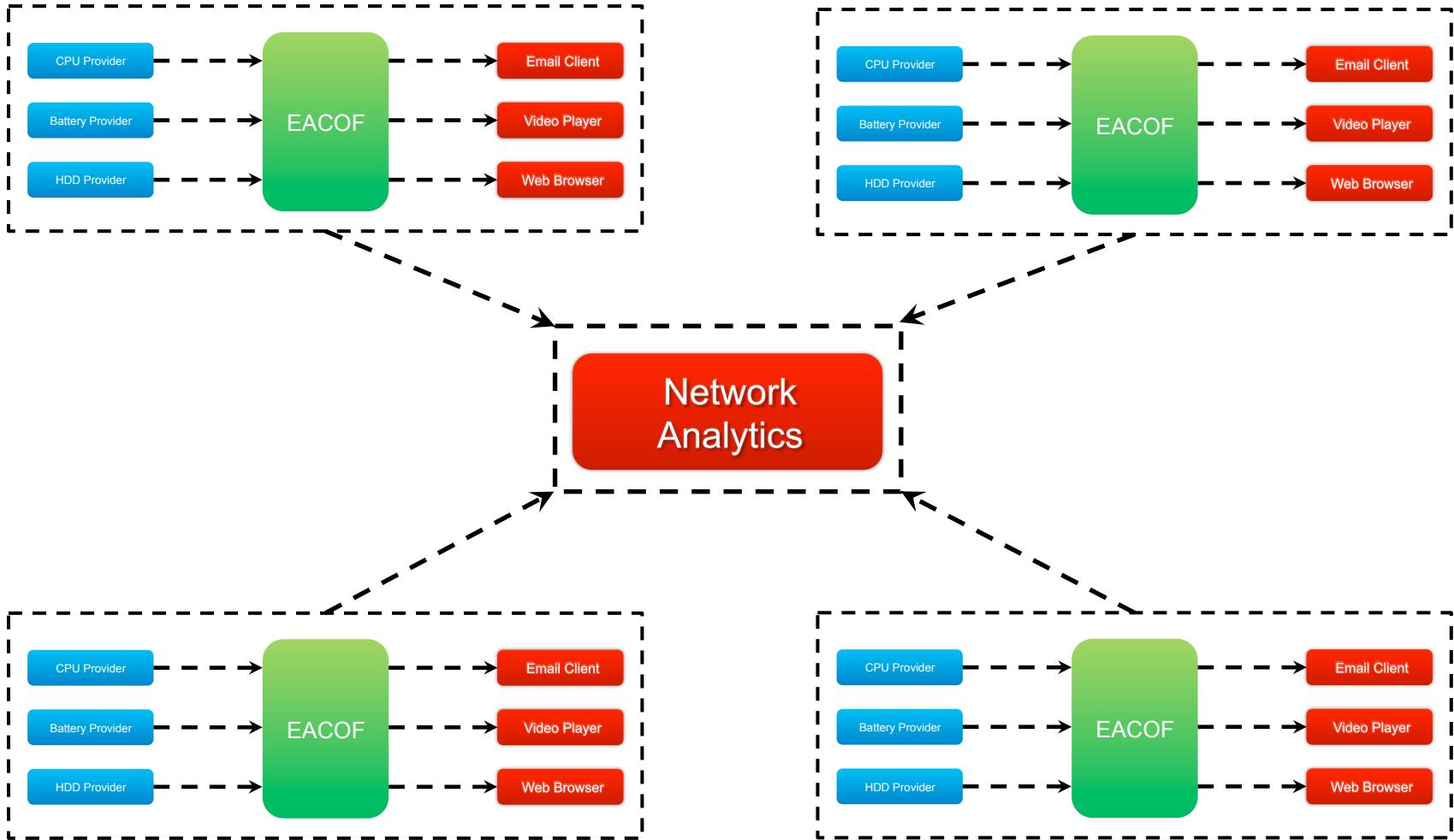
# Consumers



# One Machine



# Networked



# How to use EACOF

# Simple Provider Example

---

```
while(1) {  
    collectEnergyData();  
    waitABit();  
}
```

# Simple Provider Example + EACOF

---

```
#include <eacof.h>
eacof_Probe *probe;
eacof_Sample sample;
initEACOF();
createProbe(&probe, 1, EACOF_DEVICE_BATTERY_ALL);
while(1) {
    sample = collectEnergyData();
    addSample(probe, sample);
    waitABit();
}
deleteProbe(&probe);
```

# Simple Consumer Example

---

```
for (int i = 0; i < 10000; i++) {  
    printf("Hello EACOF!");  
}
```

# Simple Consumer Example + EACOF

---

```
#include <eacof.h>
eacof_Checkpoint *checkpoint;
eacof_Sample sample;
initEACOF();
setCheckpoint(&checkpoint, EACOF_PSPEC_ALL, 1,
    EACOF_DEVICE_BATTERY_ALL);
for (int i = 0; i < 10000; i++) {
    printf("Hello EACOF! \n");
    sampleCheckpoint(checkpoint, &sample);
}
deleteCheckpoint(&checkpoint);
```

# The EACOF API

---

```
#include <eacof.h>
initEACOF();
createProbe(); deleteProbe();

activateProbe(); deactivateProbe();

addSample();

setCheckpoint(); deleteCheckpoint();
sampleCheckpoint();
```

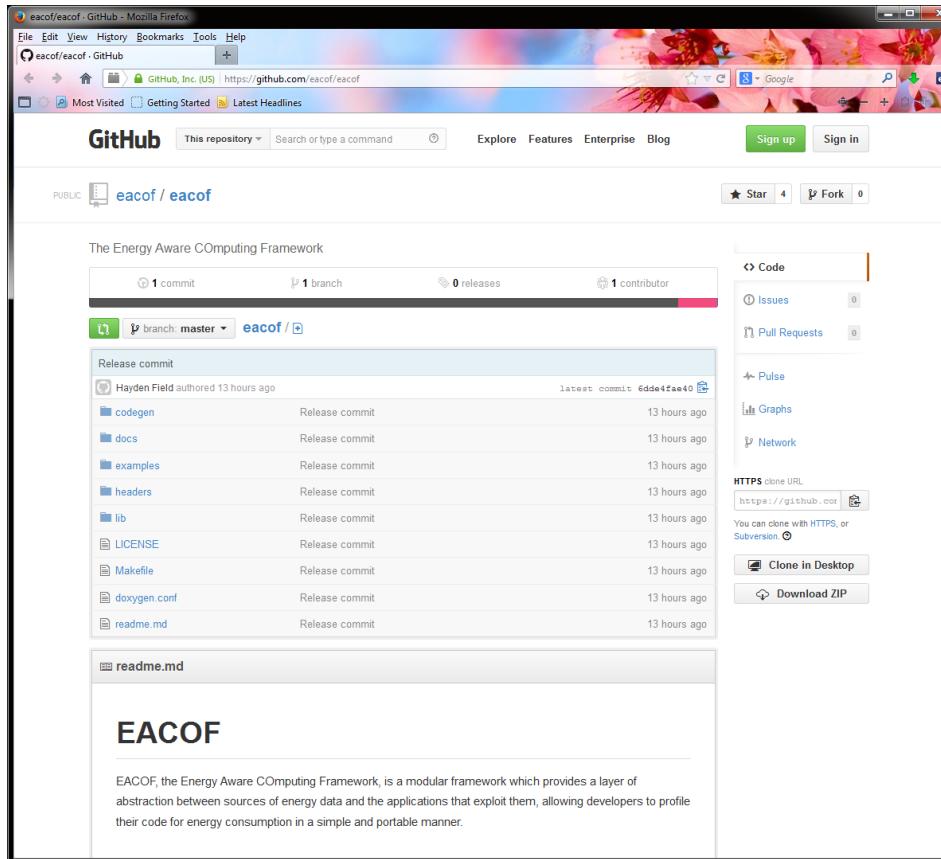
# Comparing Sorting Algorithms

- Sorting of integers in [0,255]

Algorithm	Num Elements	Data Type											
		uint8_t			uint16_t			uint32_t			uint64_t		
		Total Time (s)	Total Energy (J)	Average Power (W)	Total Time (s)	Total Energy (J)	Average Power (W)	Total Time (s)	Total Energy (J)	Average Power (W)	Total Time (s)	Total Energy (J)	Average Power (W)
Bubble Sort	50,000	5.53	66.66	12.03	5.39	65.29	12.09	5.66	69.05	12.19	5.78	71.83	12.41
Insertion Sort	200,000	7.98	■102.18	12.75	7.98	■103.00	12.85	7.46	■98.81	13.21	7.54	■105.03	13.89
Quicksort	2,000,000	5.51	61.73	11.20	5.53	61.90	11.19	5.52	61.60	11.15	5.51	62.90	★11.42
Merge Sort	60,000,000	•6.06	•72.33	11.93	6.07	72.46	11.93	6.12	75.65	12.36	•5.93	•76.98	★12.98
qsort	100,000,000	•5.84	•72.39	12.37	6.15	76.90	12.48	6.79	86.29	12.69	•5.69	•73.25	12.86
Counting Sort	200,000,000	0.23	◆2.92	12.75	0.24	◆3.16	13.23	0.25	◆3.58	14.15	0.35	◆5.12	14.44

- Insertion Sort: 32 bit version more optimized
- ◆ Counting Sort:
  - 75% more energy for 64 bit compared to 8 bit values
  - Sorting 64 bit values takes less time than sorting 8 bit values, but consumed more energy
- ★ Average power variations between algorithms

# Invitation: EACOF is open source!



[github.com/eacof](https://github.com/eacof)

# Learning Objectives

---

- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- ✓ How to measure the energy consumed by software
- How to estimate the energy consumed by software *without measuring*
- How to construct energy consumption models

# Learning Objectives

---

- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- ✓ How to measure the energy consumed by software
- How to estimate the energy consumed by software *without* measuring
- How to construct energy consumption models

# Static Analysis of Energy Consumption

# ENTRA

Whole-Systems  
Energy Transparency





# The ENTRA Project



- Whole Systems ENergy TRAnsparency

EC FP7 FET MINECC:

***“Software models and programming methodologies supporting the strive for the energetic limit (e.g. energy cost awareness or exploiting the trade-off between energy and performance/precision).”***



University of  
BRISTOL



# Acknowledgements

---

## The partners in the EU ENTRA project



John Gallagher and team

Pedro López García and team

Henk Muller and team

Steve Kerrison, Kyriakos Gerogiou, James Pallister, Jeremy Morse and Neville Grech

# Static Energy Usage Analysis

Original Program:

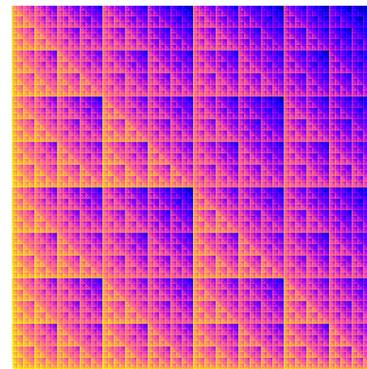
```
int fact (int x) {  
    if (x<=0)a  
        return 1b;  
    return (x *d fact(x-1))c;  
}
```

Extracted Cost Relations:

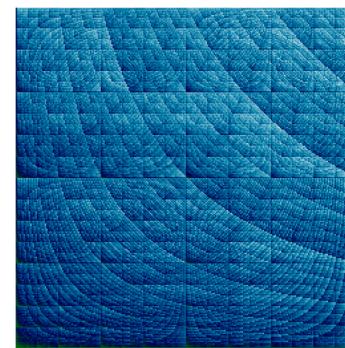
$$\begin{aligned}C_{\text{fact}}(x) &= C_a + C_b && \text{if } x \leq 0 \\C_{\text{fact}}(x) &= C_a + C_c(x) && \text{if } x > 0 \\C_c(x) &= C_d + C_{\text{fact}}(x-1)\end{aligned}$$

- Substitute  $C_a$ ,  $C_b$ ,  $C_d$  with  
the **actual energy required to execute the  
corresponding lower-level (machine) instructions.**

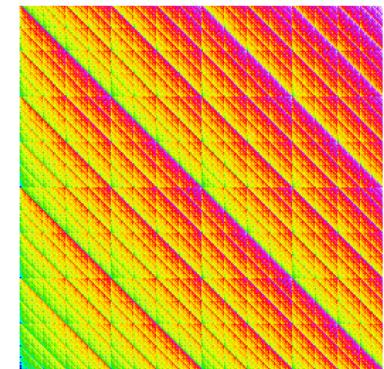
# Energy Modelling captures energy consumption



$\wedge$



$\times$



$+$

# Modelling Considerations

---

- At what level should we model?
  - instruction level, i.e. machine code
  - intermediate representation of compiler
  - source code
- Models require measurements
  - need to associate entities at a given level with costs, i.e. energy consumption
    - accuracy
    - usefulness

# Modelling Considerations

---

- At what level should we model?
  - instruction level, i.e. machine code
  - intermediate representation of compiler
  - source code
- Models require measurements
  - need to associate entities at a given level with costs, i.e. energy consumption
    - accuracy – the lower the better
    - usefulness – the higher the better



# ISA-Level Energy Modelling

Energy Cost ( $E$ ) of a program ( $P$ ):

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

Instruction  
Base Cost,  
 $B_i$ , of each  
instruction  $i$

Circuit State  
Overhead,  
 $O_{i,j}$ , for each  
instruction  
pair

# ISA-Level Energy Modelling

Components of an Energy Model:

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

- $B_i$  and  $O_{i,j}$  are energy costs.
- Characterization of a model through measurement produces these values for a given processor.

Based on V. Tiwari, S. Malik and A. Wolfe. "Instruction Level Power Analysis and Optimization of Software", Journal of VLSI Signal Processing Systems, 13, pp 223-238, 1996.

# ISA-Level Energy Modelling

Components of an Energy Model:

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

- $N_i$  is the number of times that instruction  $i$  is executed, and
- $N_{i,j}$  is the number of times that the execution of instruction  $i$  is followed by the execution of instruction  $j$ .

Based on V. Tiwari, S. Malik and A. Wolfe. "Instruction Level Power Analysis and Optimization of Software", Journal of VLSI Signal Processing Systems, 13, pp 223-238, 1996.

# Exercise: E(fact(3))?

```
int fact (int x) {  
    int ret = x;  
    while (--x)  
    {  
        ret *= x;  
    }  
    return ret;  
}
```

```
fact:  
    sub    r3, r0, #1  
    cmp    r3, #0  
    beq    .L2  
.L3:  
    mul    r0, r3  
    sub    r3, r3, #1  
    cmp    r3, #0  
    bne    .L3  
.L2:  
    bx    lr
```

**How much energy  
does a call to  
fact(3) consume?**

# Base Cost Characterization

Instruction	Base Cost [pJ]
sub	600
cmp	300
beq	500
mul	900
bne	500
bx	700

fact:

sub r3, r0, #1

cmp r3, #0

beq .L2

.L3:

mul r0, r3

sub r3, r3, #1

cmp r3, #0

bne .L3

.L2:

bx lr

# Overhead Characterization

fact:

sub r3, r0, #1

cmp r3, #0

beq .L2

.L3:

mul r0, r3

sub r3, r3, #1

cmp r3, #0

.L2:

bne .L3

bx lr

$O_{i,j}$ [pJ]	beq	bne	bx	cmp	mul	sub
beq	0	10	10	30	30	30
bne	10	0	10	30	30	30
bx	10	10	0	60	60	60
cmp	10	10	10	0	20	20
mul	10	10	10	30	0	30
sub	10	10	10	20	30	0

# Instruction Characterization

Instruction	Base Cost [pJ]
beq	500
bne	500
bx	700
cmp	300
mul	900
sub	600

$O_{i,j}$ [pJ]	beq	bne	bx	cmp	mul	sub
beq	0	10	10	30	30	30
bne	10	0	10	30	30	30
bx	10	10	0	60	60	60
cmp	10	10	10	0	20	20
mul	10	10	10	30	0	30
sub	10	10	10	20	30	0

# ISA-Level Energy Modelling

## Components of an Energy Model:

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

Instruction	Base Cost [pJ]
beq	500
bne	500
bx	700
cmp	300
mul	900
sub	600

O <sub>i,j</sub> [pJ]	beq	bne	bx	cmp	mul	sub
beq	0	10	10	30	30	30
bne	10	0	10	30	30	30
bx	10	10	0	60	60	60
cmp	10	10	10	0	20	20
mul	10	10	10	30	0	30
sub	10	10	10	20	30	0

Based on V. Tiwari, S. Malik and A. Wolfe. "Instruction Level Power Analysis and Optimization of Software", Journal of VLSI Signal Processing Systems, 13, pp 223-238, 1996.

# ISA-Level Energy Modelling

Components of an Energy Model:

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

- $N_i$  and  $N_{i,j}$  represent the number of times specific instructions and instruction pairs are executed.
- How can we determine these?

# Exercise

```
@ Argument is in r0
fact:
    sub      r3, r0, #1
    cmp      r3, #0
    beq      .L2          @ Never iterate loop if num == 1
.L3:
    mul      r0, r3        @ Accumulate factorial value in r0
    sub      r3, r3, #1    @ r3 is decrementing counter
    cmp      r3, #0
    bne      .L3          @ Loop if we haven't reached 0
.L2:
    bx       lr            @ Return, answer is in r0
```

Which instruction sequence is being executed for a call to fact(3)?

# Exercise

```
@ Argument is in r0
fact:
    sub      r3, r0, #1
    cmp      r3, #0
    beq      .L2          @ Never iterate loop if num == 1
.L3:
    mul      r0, r3        @ Accumulate factorial value in r0
    sub      r3, r3, #1    @ r3 is decrementing counter
    cmp      r3, #0
    bne      .L3          @ Loop if we haven't reached 0
.L2:
    bx       lr            @ Return, answer is in r0
```

A call to `fact(3)` would invoke the following instructions in this order:

- `sub, cmp, beq` (not taken),
- `mul, sub, cmp, bne` (taken),
- `mul, sub, cmp, bne` (not taken),
- `bx`

# Exercise

Instruction	Base Cost [pJ]
beq	500
bne	500
bx	700
cmp	300
mul	900
sub	600

$O_{i,j}$ [pJ]	beq	bne	bx	cmp	mul	sub
beq	0	10	10	30	30	30
bne	10	0	10	30	30	30
bx	10	10	0	60	60	60
cmp	10	10	10	0	20	20
mul	10	10	10	30	0	30
sub	10	10	10	20	30	0

A call to fact(3) would invoke the following instructions in this order:

- sub, cmp, beq (not taken),
- mul, sub, cmp, bne (taken),
- mul, sub, cmp, bne (not taken),
- bx

# Exercise

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

*sub, cmp, beq (not taken), mul, sub, cmp, bne (taken),  
mul, sub, cmp, bne (not taken), bx*

$$E_{fact(3)} =$$

# Exercise

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

*sub, cmp, beq (not taken), mul, sub, cmp, bne (taken),  
mul, sub, cmp, bne (not taken), bx*

$$\begin{aligned} E_{fact(3)} &= 3*600pJ + 3*300pJ + 500pJ + 2*900 + 2*500pJ + 700pJ \\ &+ 3*20pJ + 10pJ + 30pJ + 2*30pJ + 2*10pJ + 30pJ + 10pJ \\ &= 6920pJ = \underline{\underline{6.92nJ}} \end{aligned}$$

# Is it really this easy?

Energy Cost ( $E$ ) of a program ( $P$ ):

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j})$$

Instruction  
Base Cost,  
 $B_i$ , of each  
instruction  $i$

Circuit State  
Overhead,  
 $O_{i,j}$ , for each  
instruction  
pair

# Is it really this easy?

Energy Cost ( $E$ ) of a program ( $P$ ):

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j}) + \sum_k E_k$$

Instruction Base Cost,  $B_i$ , of each instruction  $i$

Circuit State Overhead,  $O_{i,j}$ , for each instruction pair

Other Instruction Effects

# Energy Modelling

Energy Cost ( $E$ ) of a program ( $P$ ):

$$E_P = \sum_i (B_i \times N_i) + \sum_{i,j} (O_{i,j} \times N_{i,j}) + \sum_k E_k$$

Instruction  
Base Cost,  
 $B_i$ , of each  
instruction  $i$

Circuit State  
Overhead,  
 $O_{i,j}$ , for each  
instruction  
pair

Other  
Instruction  
Effects  
(stalls,  
cache  
misses,  
etc)

# XCore Energy Modelling

Energy Cost ( $E$ ) of a **multi-threaded** program ( $P$ ):

$$E_p = P_{\text{base}} N_{\text{idle}} T_{\text{clk}} + \sum_{t=1}^{N_t} \sum_{i \in \text{ISA}} ((M_t P_i O + P_{\text{base}}) N_{i,t} T_{\text{clk}})$$

Idle base power and duration

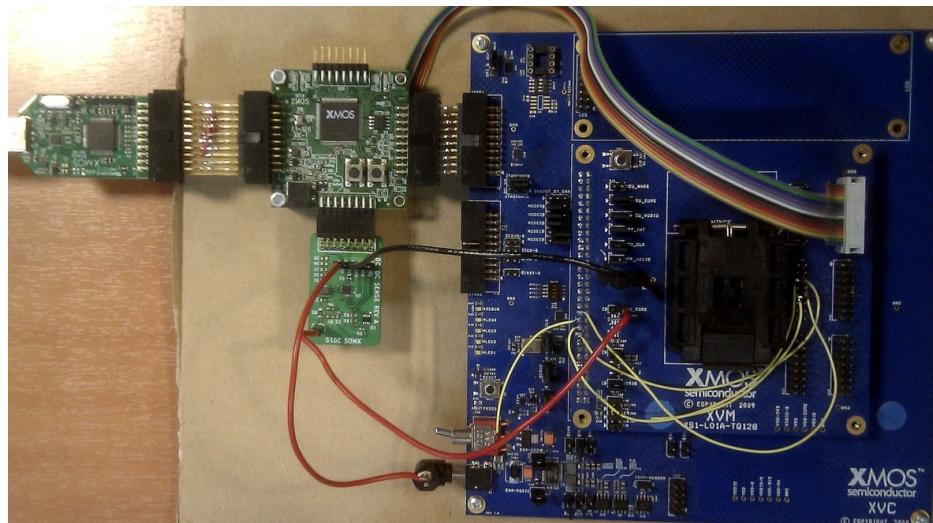
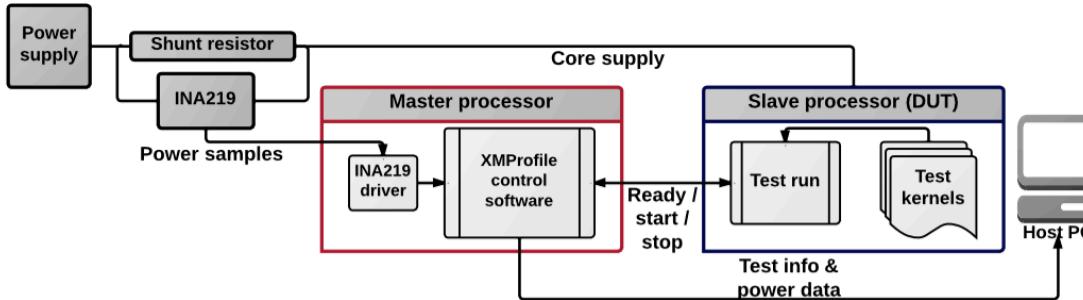
Concurrency cost, instruction cost, generalised overhead, base power and duration

- Use of execution statistics rather than execution trace.
- Fast running model with an average error margin of less than 7%.

S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages.

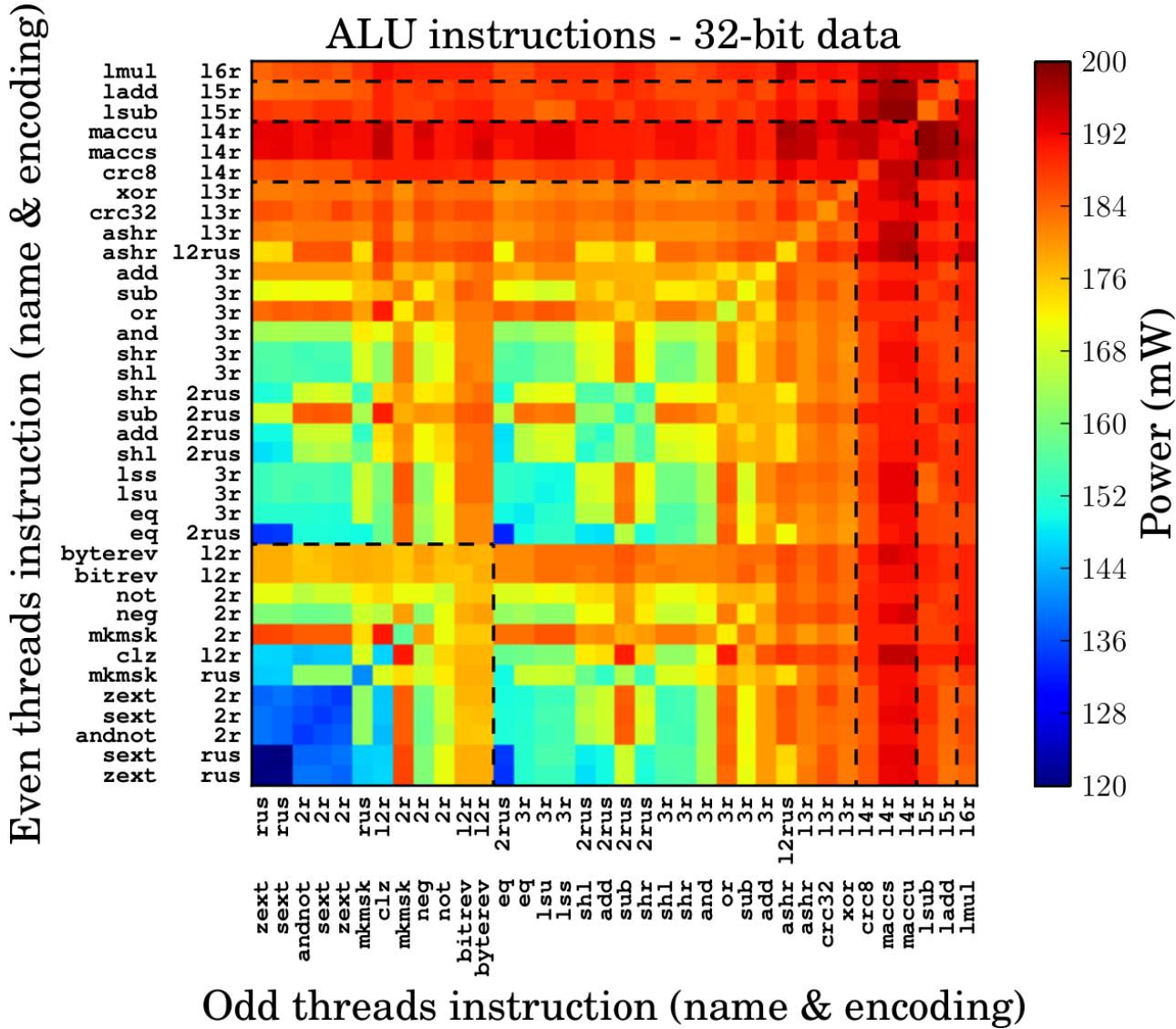
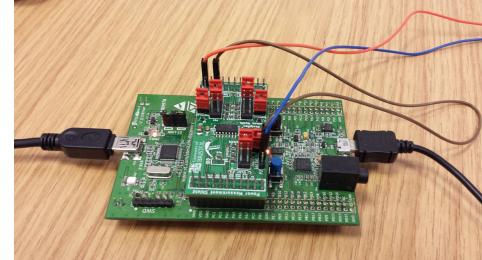
DOI=10.1145/2700104 <http://doi.acm.org/10.1145/2700104>

# The set up...

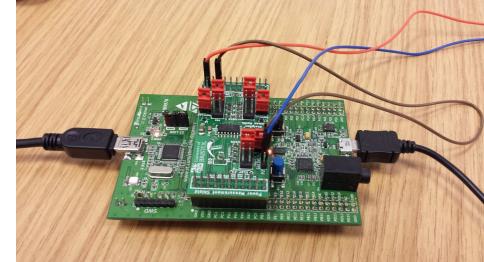


S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages.  
DOI=10.1145/2700104 <http://doi.acm.org/10.1145/2700104>

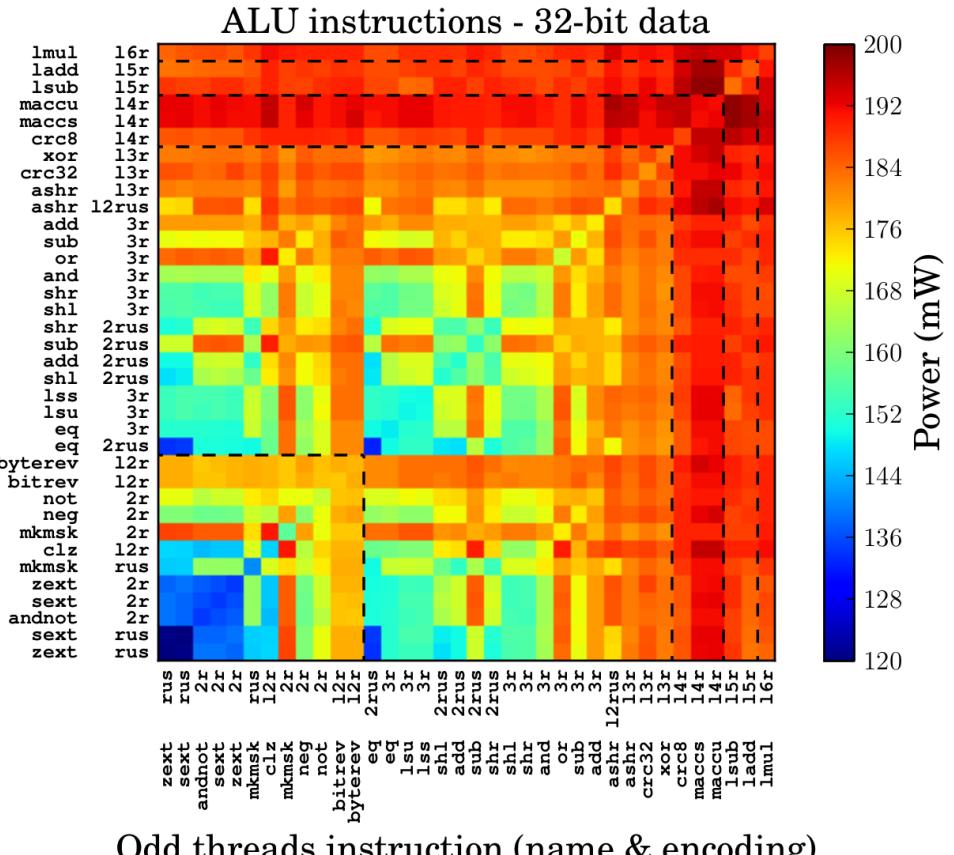
# ISA Characterization



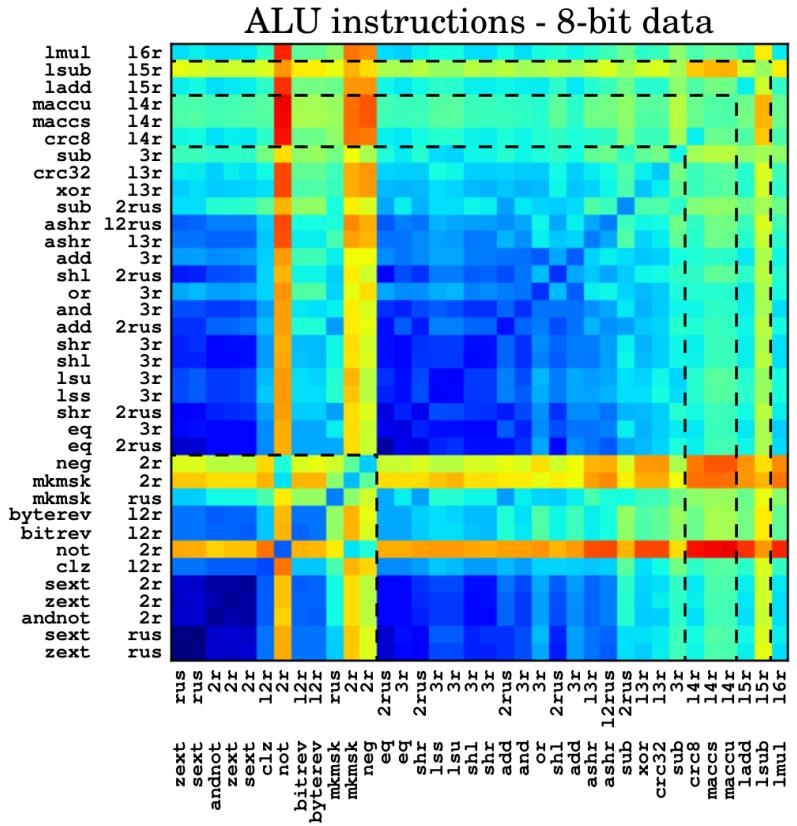
# ISA Characterization



Even threads instruction (name & encoding)



Odd threads instruction (name & encoding)

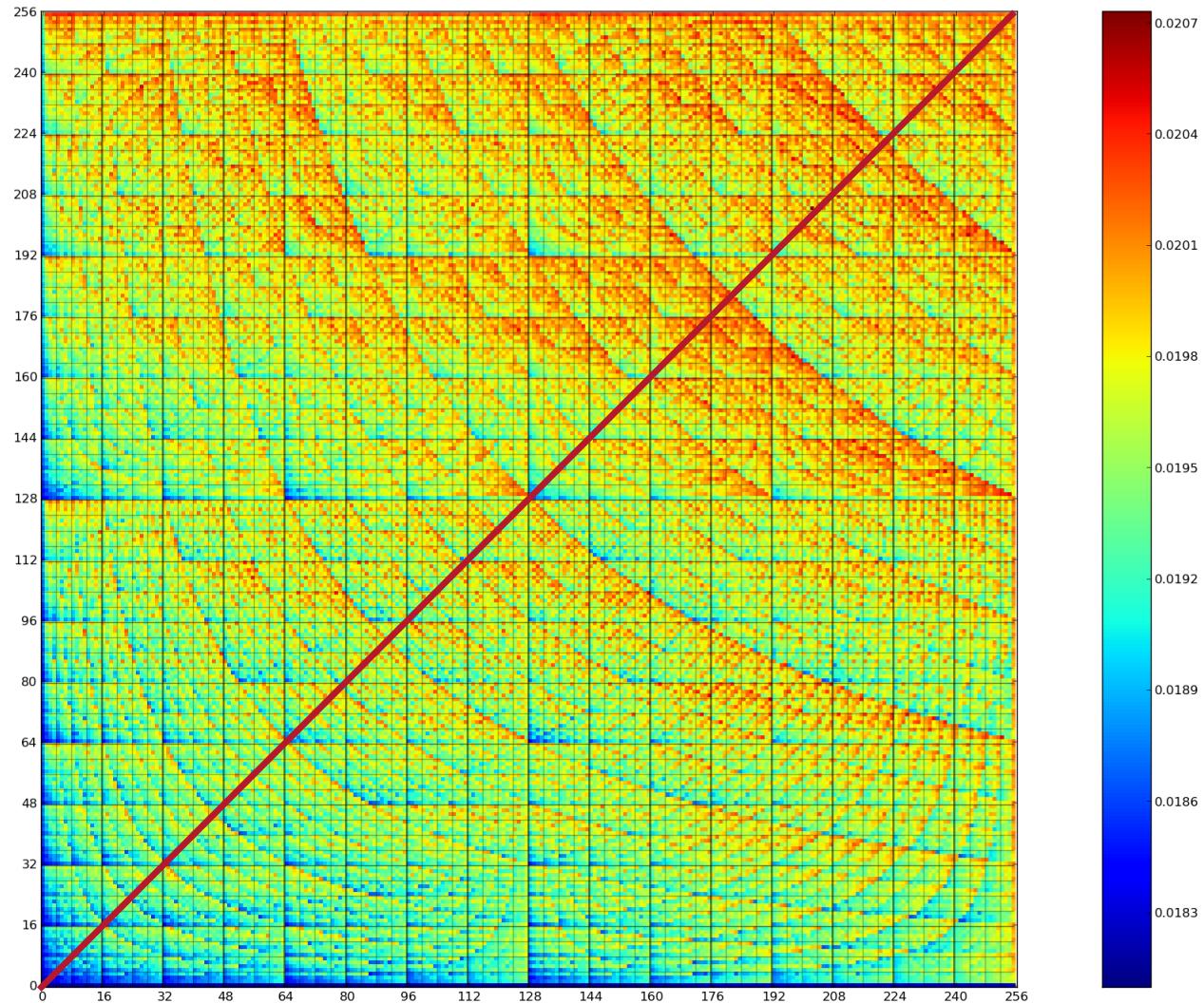


Odd threads instruction (name & encoding)

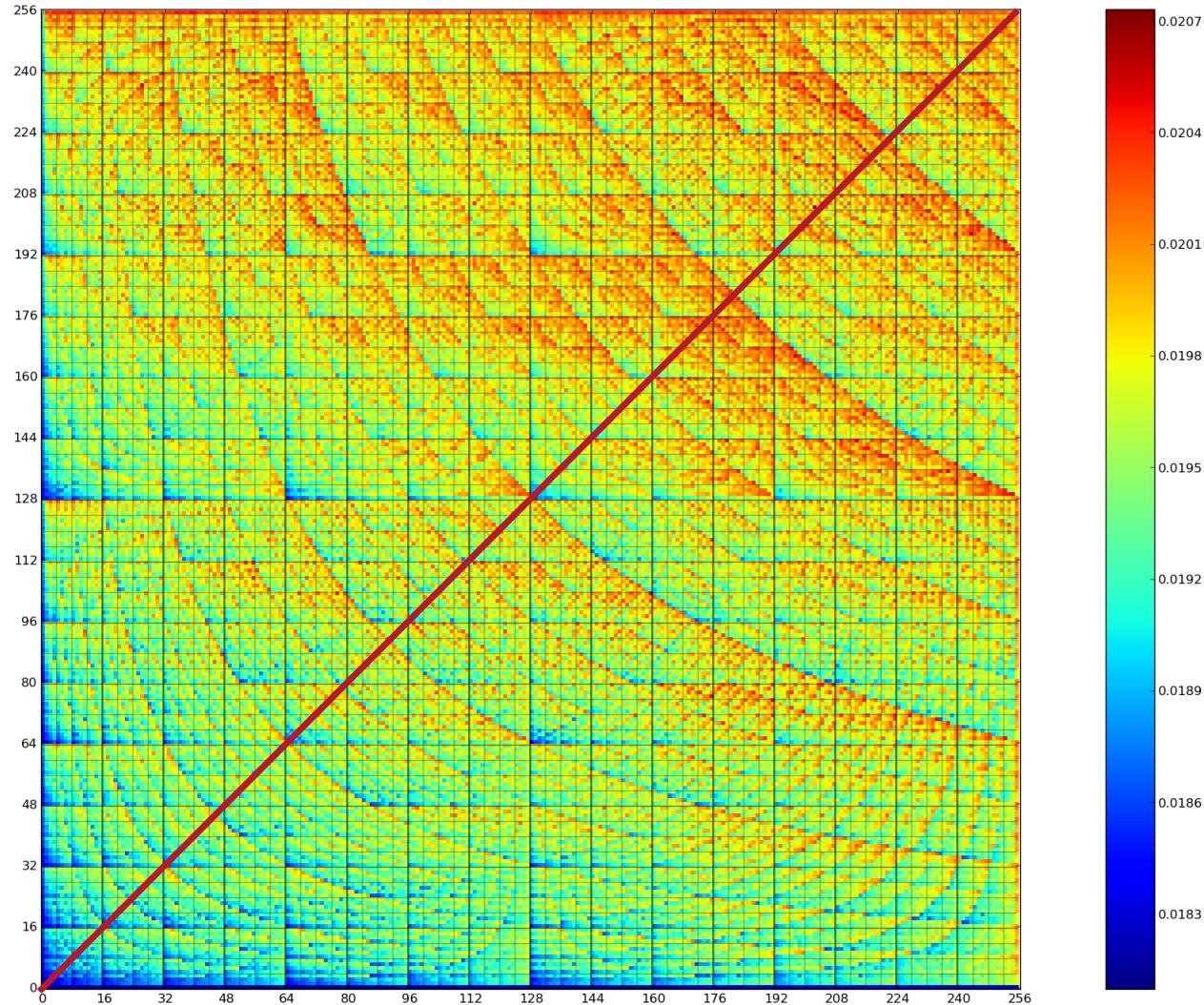
S. Kerrison and K. Eder. 2015. "Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor". ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages.

DOI=10.1145/2700104 <http://doi.acm.org/10.1145/2700104>

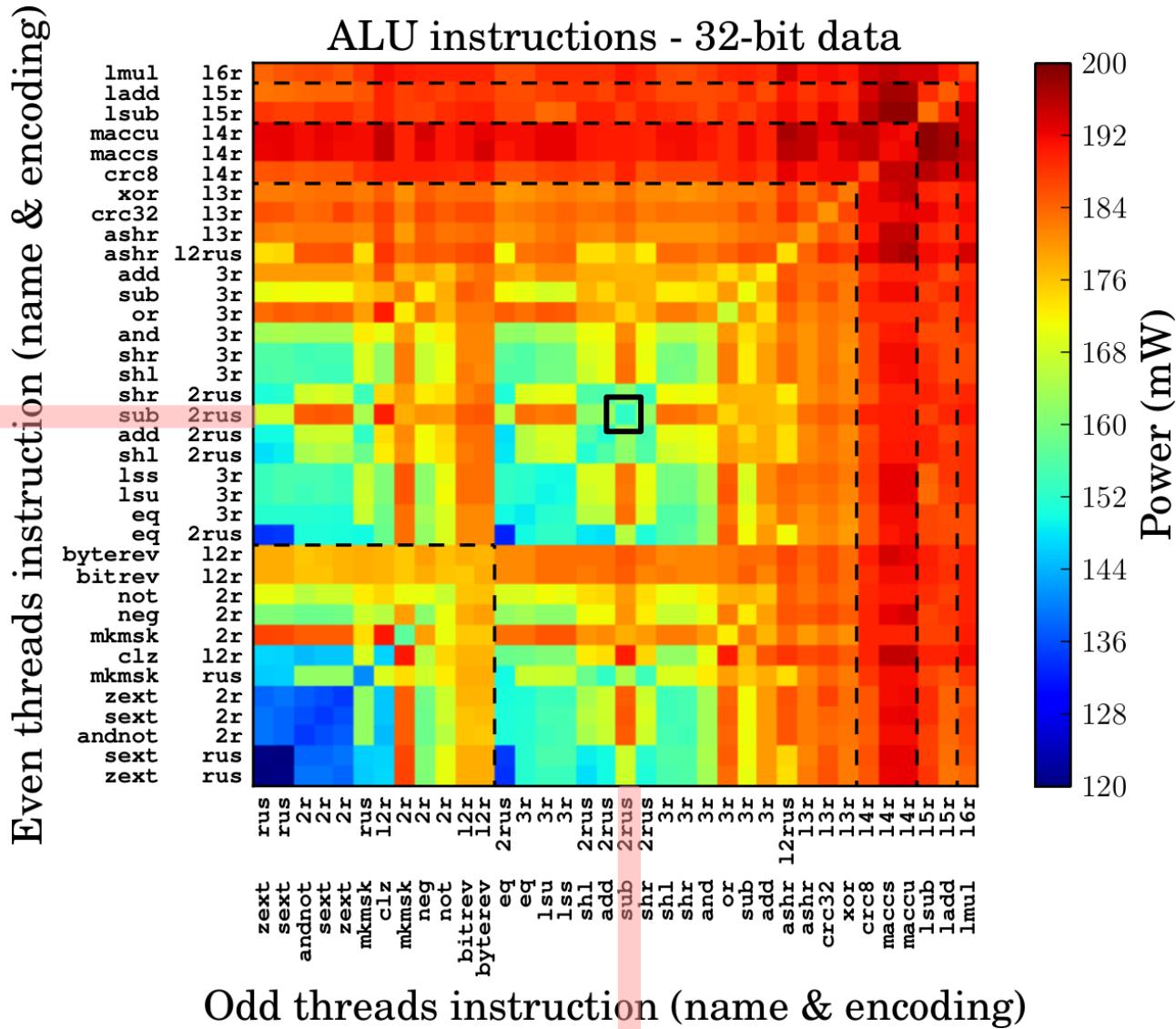
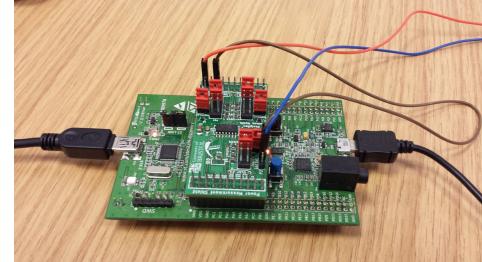
$$a^*b = b^*a$$



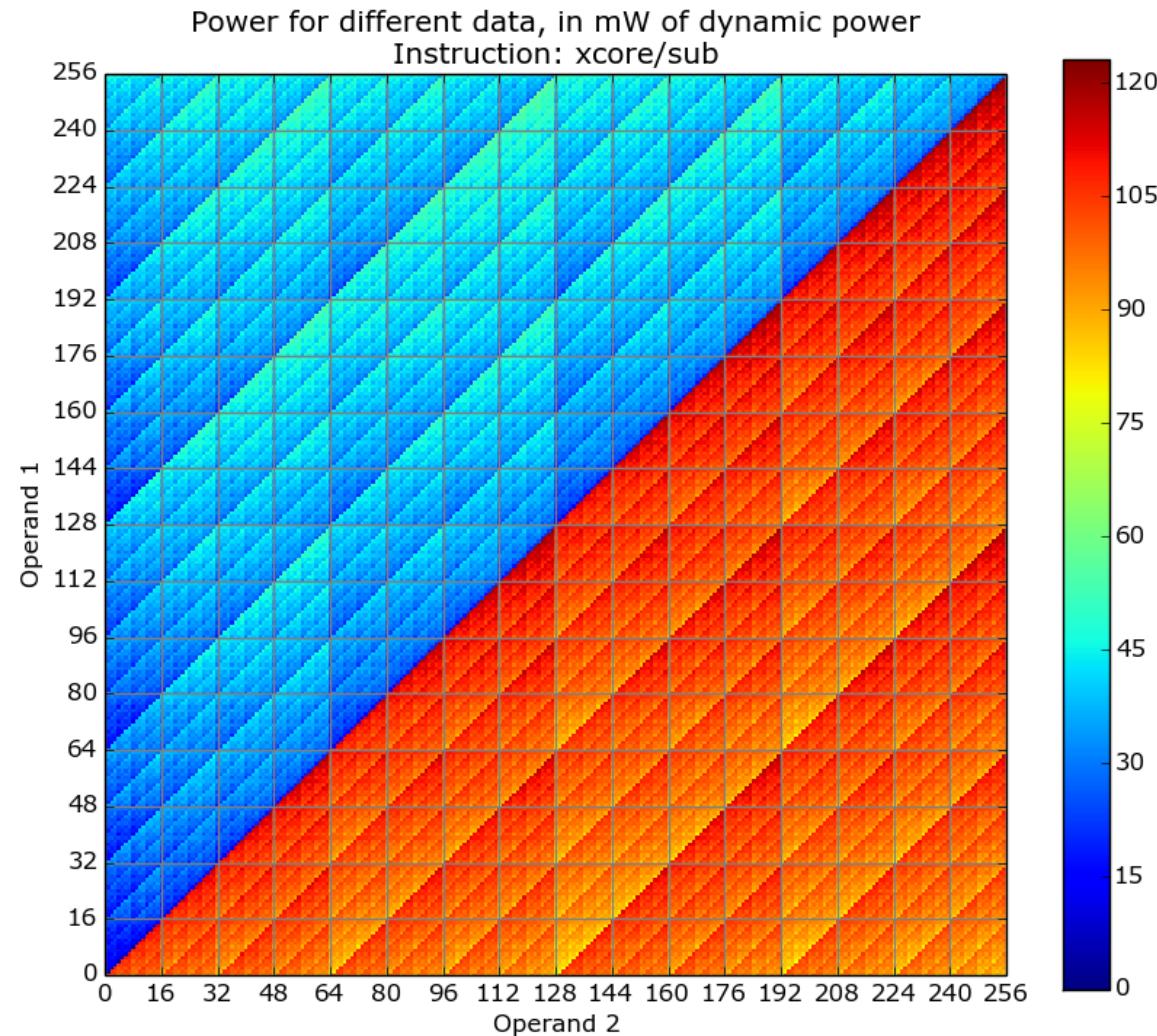
# Energy( $a^*b$ ) $\neq$ Energy( $b^*a$ )



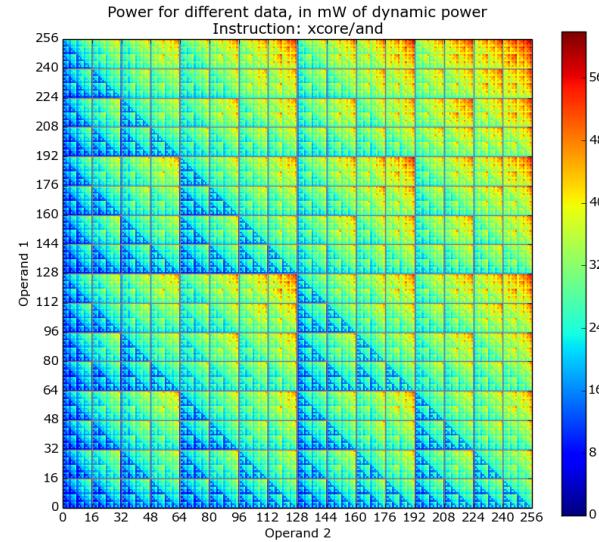
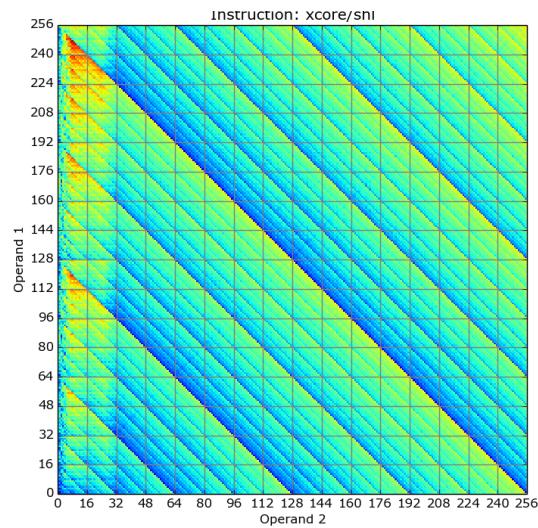
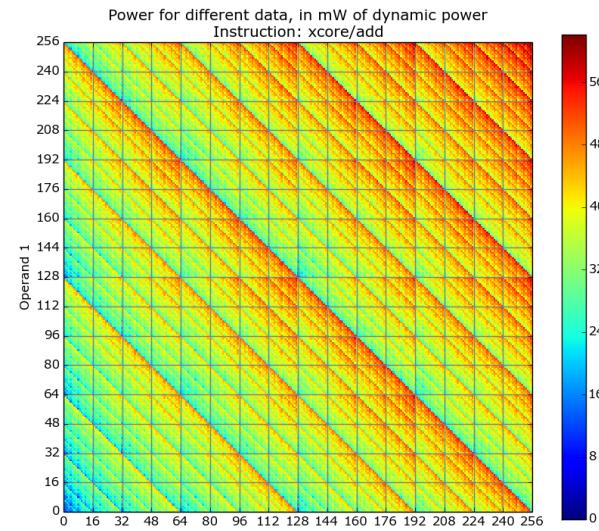
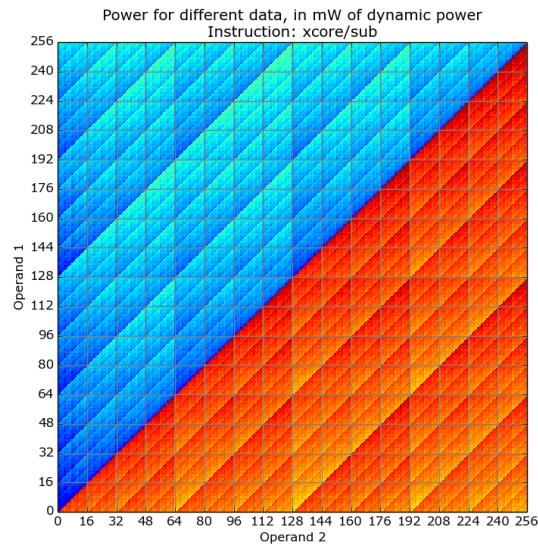
# ISA Characterization



# The Impact of Data on Energy Consumption



# W/A/B-Case Energy Consumption



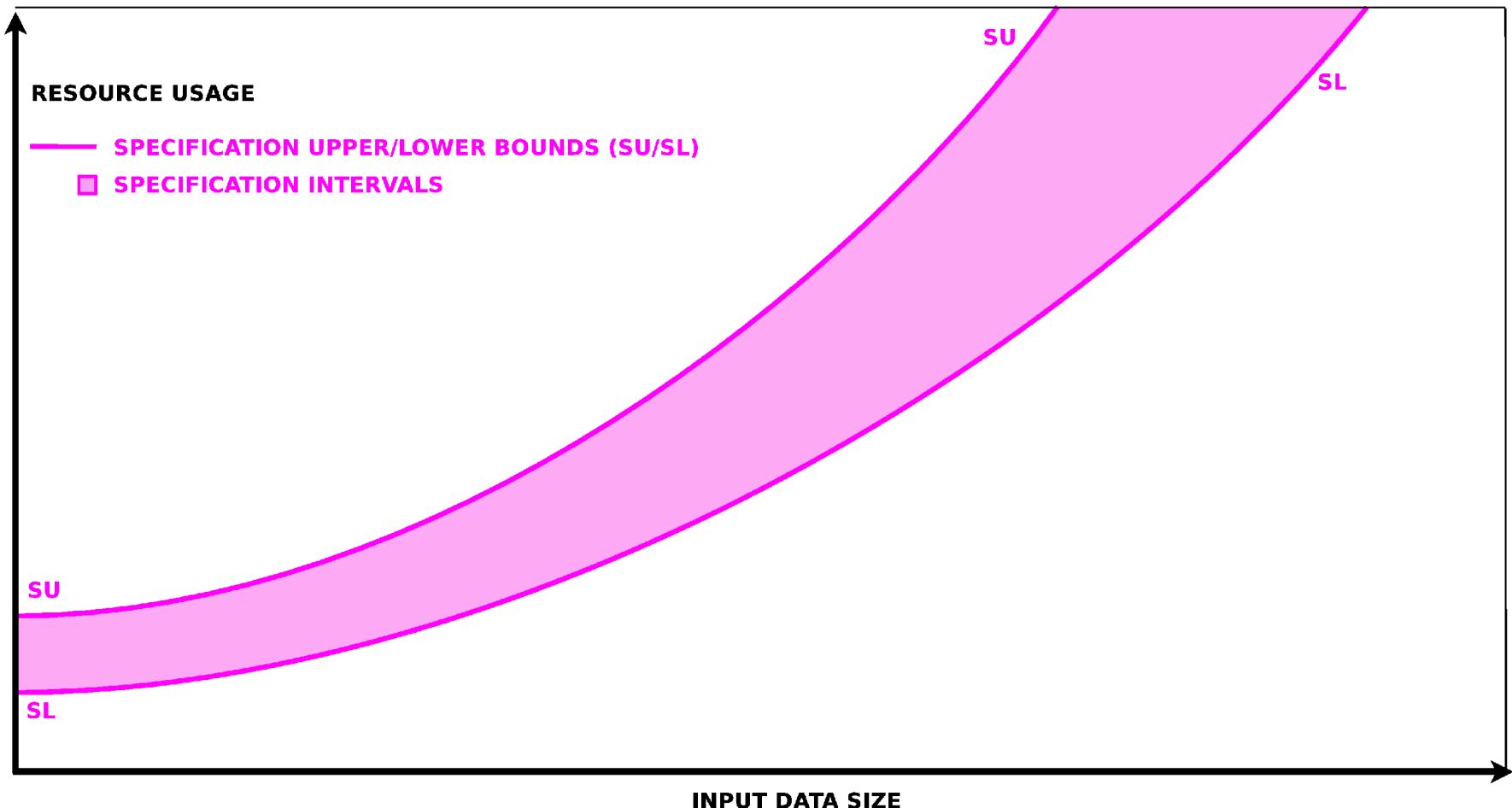
*A quick jump forward to*  
Static Resource  
consumption Analysis

# Static Resource Analysis

---

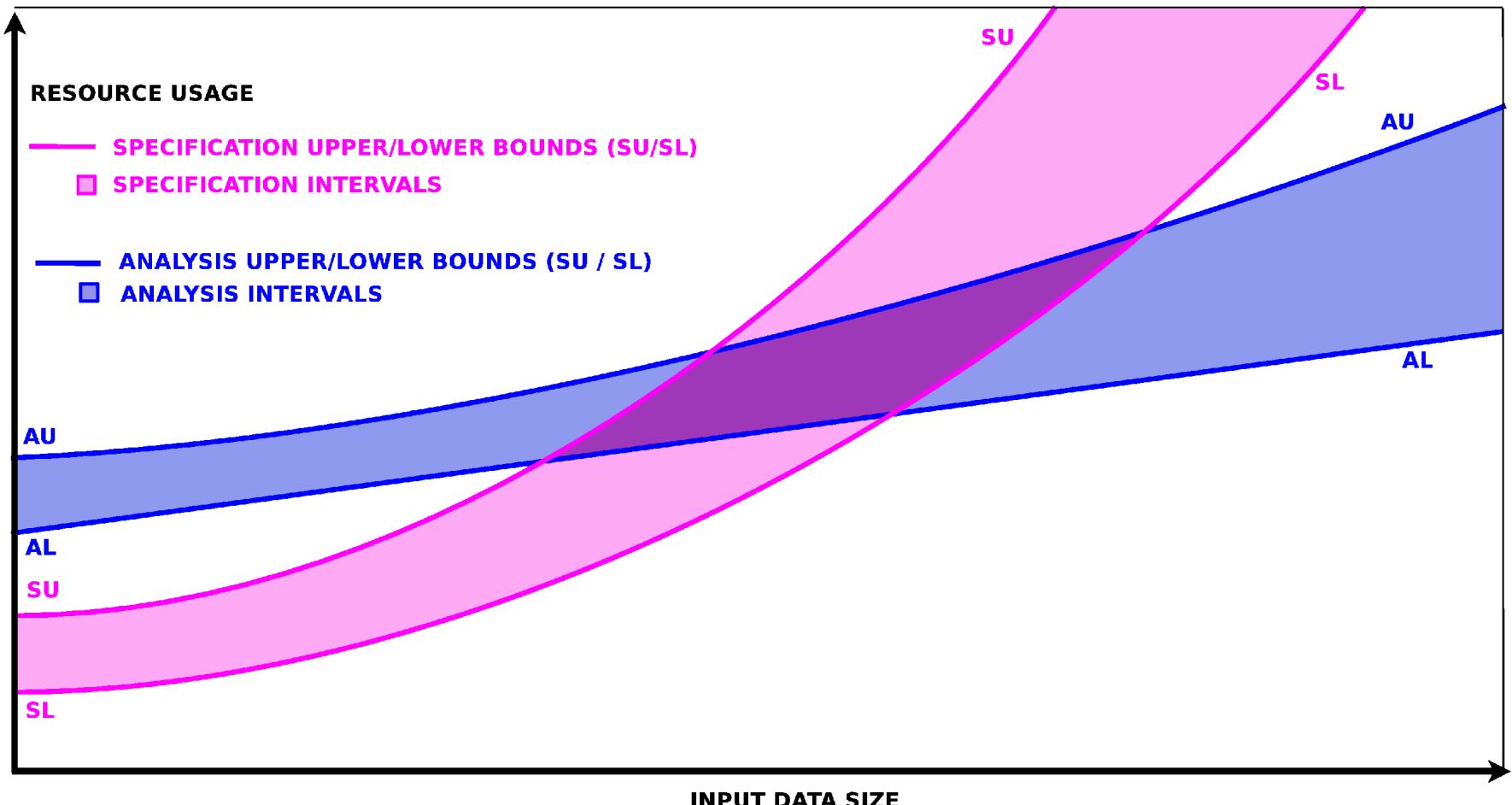
- Techniques automatically infer **upper and lower bounds** on resource usage of a program.
- Bounds expressed using **monotonic arithmetic functions per procedure** parameterized by program's input size.
- **Verification** can be done statically by checking that the upper and lower bounds on resource usage defined in the specifications hold.

# Specified Resource Usage



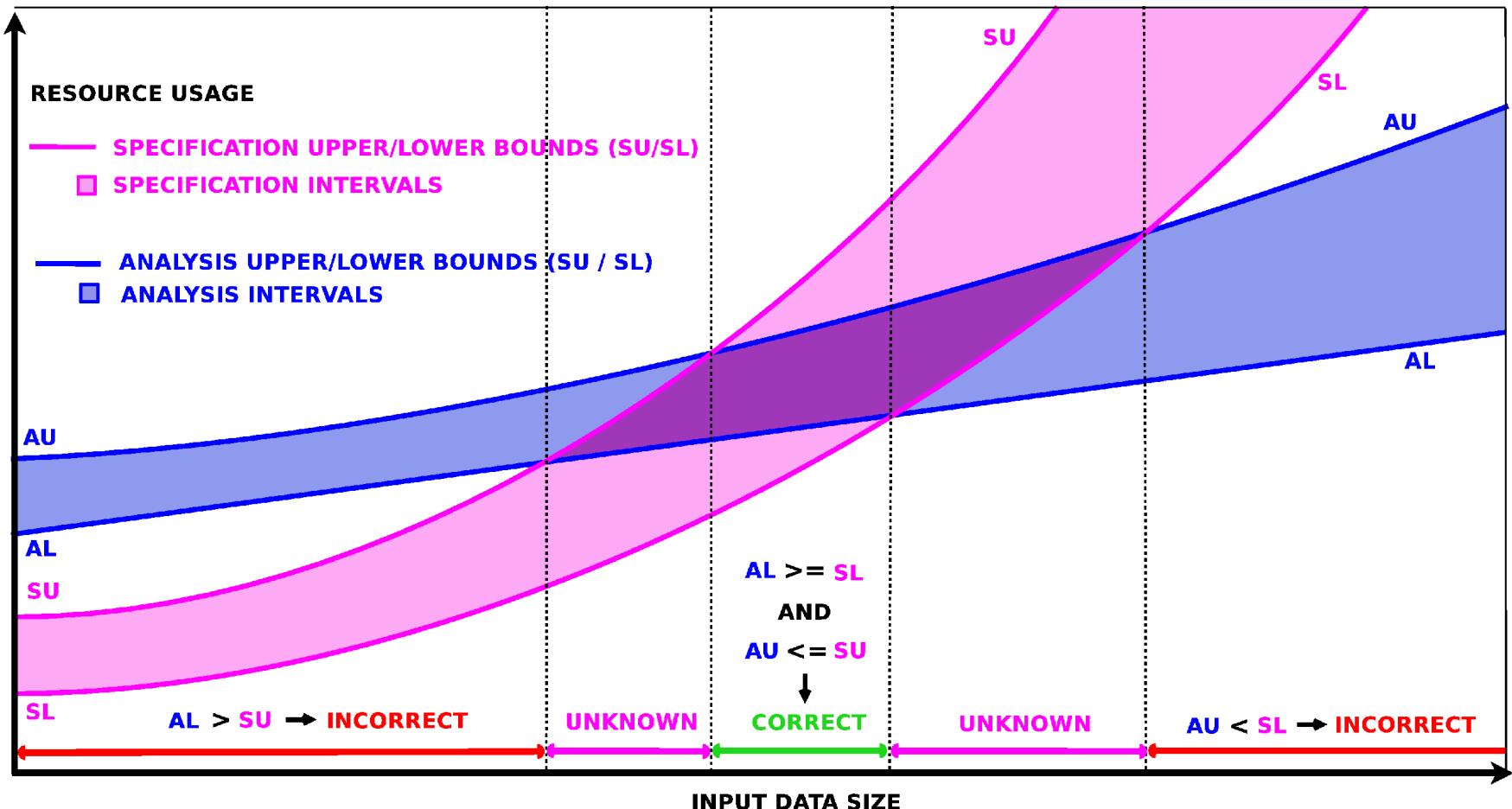
Source: Pedro Lopez Garcia, IMDEA Software Research Institute

# Analysis Result



Source: Pedro Lopez Garcia, IMDEA Software Research Institute

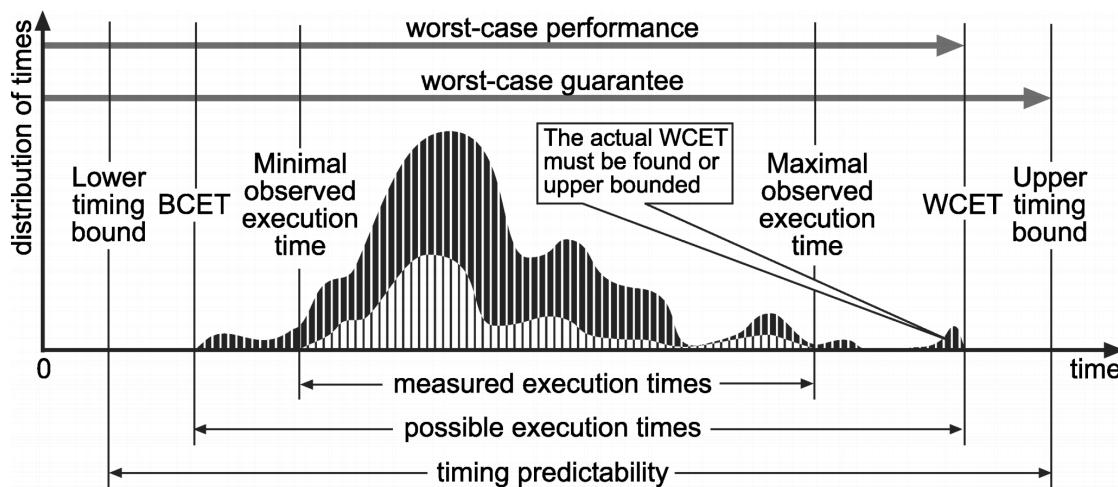
# Verification



Source: Pedro Lopez Garcia, IMDEA Software Research Institute

# Worst Case Execution Time

- Worst Case Execution Time (WCET) Analysis:
  - WCET model
  - WCET bounds (are often safety critical)
    - safe, i.e. no underestimation
    - tight, i.e. ideally very little overestimation



From “The Worst-Case Execution-Time Problem — Overview of Methods and Survey of Tools” by WILHELM et al. (2008)

Does this work for energy consumption analysis?

# Worst Case Energy Consumption

---

- WCEC analysis goes well beyond WCET analysis.
  - data independence of execution time through the use of synchronous logic
  - embedded real-time systems that are timing predictable execute instructions in a fixed number of clock cycles
  - WCET then depends only on the WC execution path
- Energy consumption is data dependent.
  - Data dependent energy modelling

# Data Dependent Energy Modeling for Worst Case Energy Consumption Analysis

James Pallister, Steve Kerrison, Jeremy Morse and Kerstin Eder

Dept. Computer Science, Merchant Venturers Building,  
Bristol, BS8 1UB. Email: [firstname.lastname@bristol.ac.uk](mailto:firstname.lastname@bristol.ac.uk)

**Abstract**—This paper examines the impact of operand values upon instruction level energy models of embedded processors, to explore whether the requirements for safe worst case energy consumption (WCEC) analysis can be met. WCEC is similar to worst case execution time (WCET) analysis, but seeks to determine whether a task can be completed within an energy budget rather than within a deadline. Existing energy models that underpin such analysis typically use energy measurements from random input data, providing average or otherwise unbounded estimates not necessarily suitable for worst case analysis.

We examine energy consumption distributions of two benchmarks under a range of input data on two cache-less embedded architectures, AVR and XS1-L. We find that the worst case can be predicted with a distribution created from random data. We propose a model to obtain energy distributions for instruction sequences that can be composed, enabling WCEC analysis on program basic blocks. Data dependency between instructions is also examined, giving a case where dependencies create a bimodal energy distribution. The worst case energy prediction remains safe. We conclude that worst-case energy models based on a probabilistic approach are suitable for safe WCEC analysis.

## I. INTRODUCTION

In real-time embedded systems, execution time of a program must be bounded. This can provide guarantees that tasks will meet hard deadlines and the system will function without failure. Recently, efforts have been made to give upper bounds on program energy consumption to determine if a task will complete within an available energy budget. However, such analysis often uses energy models that do not explicitly consider the dynamic power drawn by switching of data, instead producing an upper-bound using averaged random or scaled instruction models [1], [2].

A safe and tightly bound model for WCEC analysis must be close to the hardware's actual behavior, but also give confidence that it never under-estimates. Current models have not been analyzed in this context to provide sufficient confidence, and power figures from manufacturer datasheets are not sufficiently detailed to provide tight bounds.

Energy modeling allows the energy consumption of software to be estimated without taking physical measurements. Models may assign an energy value to each instruction [3], to a predefined set of processor modes [4], or use a detailed approach that considers wider processor state, such as the data for each instruction [5]. Although measurements are typically more accurate, models require no hardware instrumentation, are more versatile and can be used in many situations, such as

arXiv:1505.03374v2 [cs.PF] 3 Nov 2015

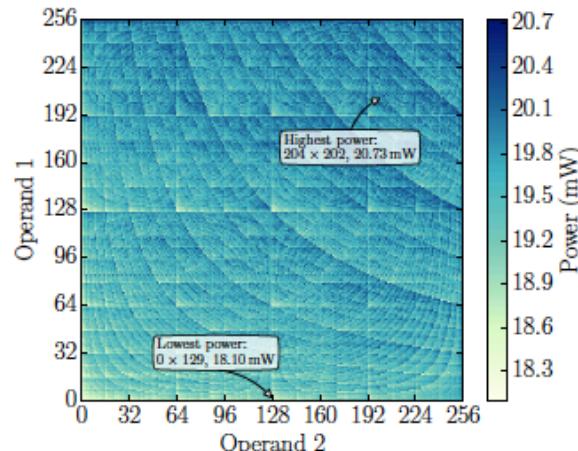


Fig. 1. Power map of mul instruction, total range is 15 % of SoC power.

In this paper, we find 15 % difference in a simple 8-bit AVR processor. This device has no caches, no OS and no high power peripherals. This difference can be seen in Figure 1, which shows the power for a single cycle, 8-bit multiply instruction in this processor. The diagram was constructed by taking hardware measurements for every possible eight bit input.

Accounting for data dependent effects in an energy model is a challenging task, which we split into two parts. Firstly, the energy effect of an instruction's manipulation of processor state needs to be modeled. This is an infeasible amount of data to exhaustively collect. A 32-bit three-operand instruction has  $2^{96}$  possible data value combinations.

Secondly, a technique is required to derive the energy consumption for a sequence of instructions from such a model. The composition of data dependent instruction energy models is a particularly difficult task. The data causing maximum energy consumption for one instruction may minimize the cost in a subsequent, dependent instruction. Finding the greatest cost for such sequences requires searching for inputs that maximize a property after an arbitrary computation, which is again an infeasibly large task. Over-approximating by summing the worst possible data dependent energy consumption of each instruction in a sequence, regardless of whether such a computation can occur, would lead to a significant overestimation of

# Worst Case Energy Consumption

---

- WCEC analysis goes well beyond WCET analysis.
  - data independence of execution time through the use of synchronous logic
  - embedded real-time systems that are timing predictable execute instructions in a fixed number of clock cycles
  - WCET then depends only on the WC execution path
- Energy consumption is data dependent.
  - Data dependent energy modelling
  - Critical questions:
    - *Which data should be used to characterize a WCEC model?*
    - *Which data causes the WCEC for a given program?*
    - *Which data triggers the most switching during the execution of the program?*

# On the infeasibility of analysing worst-case dynamic energy

Jeremy Morse, Steve Kerrison and Kerstin Eder  
University of Bristol

March 9, 2016

## Abstract

In this paper we study the sources of dynamic energy during the execution of software on microprocessors suited for the Internet of Things (IoT) domain. Estimating the energy consumed by executing software is typically achieved by determining the most costly path through the program according to some energy model of the processor. Few models, however, adequately tackle the matter of dynamic energy caused by operand data. We find that the contribution of operand data to overall energy can be significant, prove that finding the worst-case input data is NP-hard, and further, that it cannot be estimated to any useful factor. Our work shows that accurate worst-case analysis of data dependent energy is infeasible, and that other techniques for energy estimation should be considered.

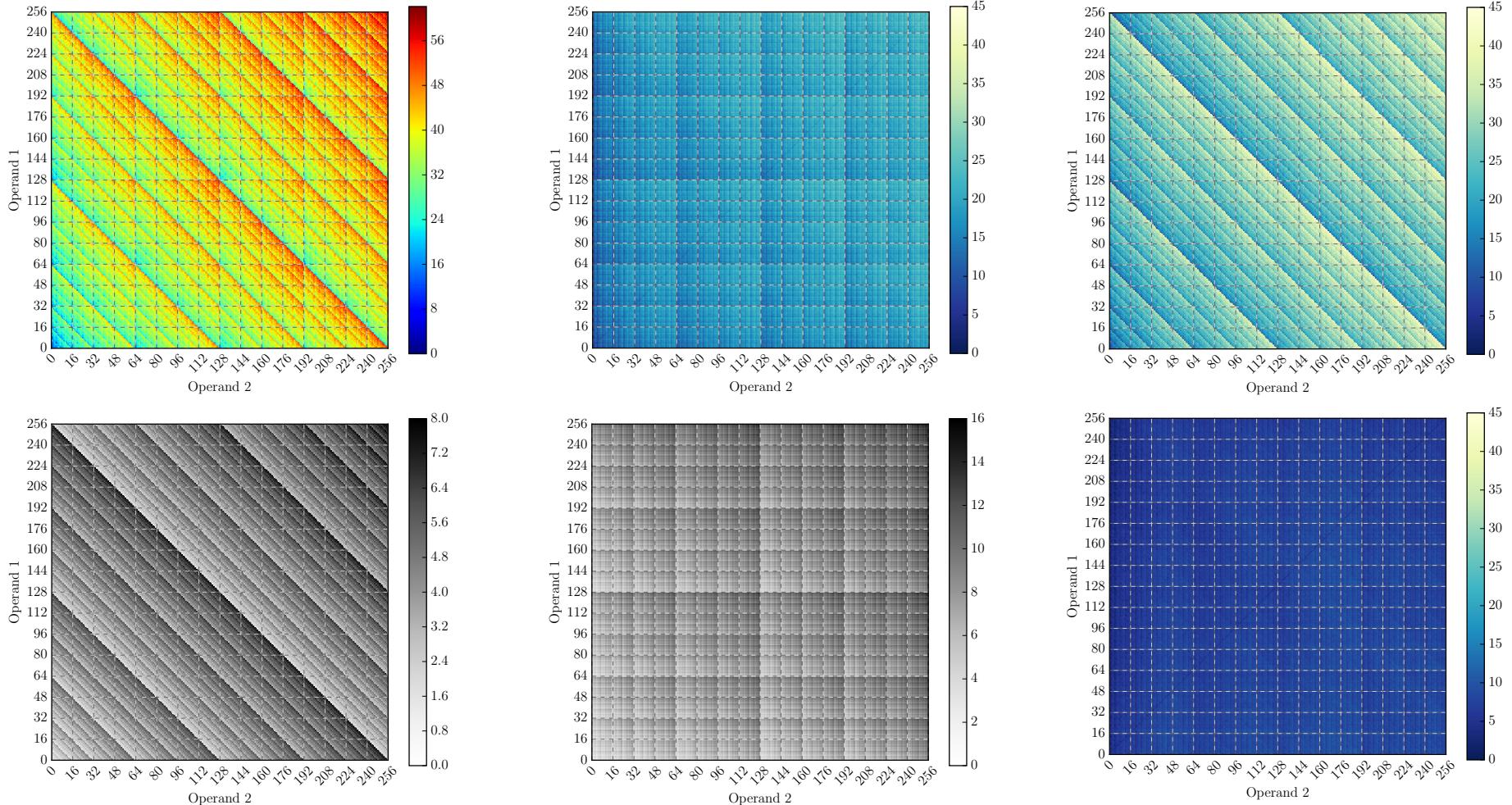
## 1 Introduction

A significant design constraint in the development of embedded systems is that of resource consumption. Software executing on such systems typically has very limited memory and computing power available, and yet must meet the requirements of the system. To aid the design process, analysis tools such as profilers or maximum-stack-depth estimators provide the developer with information allowing them to refine their designs and satisfy constraints.

A less well studied constraint is the limited energy budgets that deeply embedded systems possess. A typical example would be a wireless sensor powered by battery, that must operate for a minimum period without the battery being replaced. Other examples would be systems dependent on energy harvesting, or systems with low thermal design points that thus have a maximum power dissipation level. These constraints can also be approached with software analysis tools, and several techniques have been developed that allow the estimation of software's energy consumption [17, 7, 18].

Within energy estimation, focus has been given to *Worst Case Energy Consumption* (WCEC): determining the maximum amount of energy that can be consumed during the execution of the software. In this paper, we shall study the calculation of worst case energy, considering only the effects that different software and inputs can have on a system. The objective is to determine whether it is possible to establish upper bounds on energy that is tight to the execution time.

# Impact of datapath switching



# **Energy Consumption Analysis**

## enables energy transparency

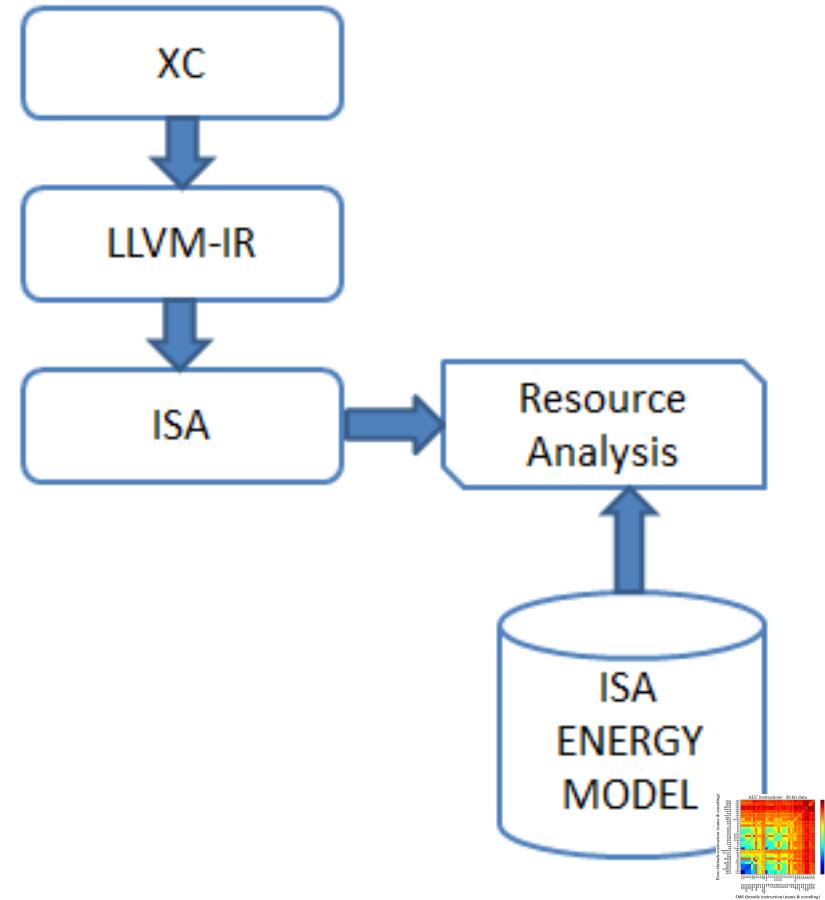


# Energy Consumption Analysis enables energy transparency



# SRA at the ISA Level

- Combine static resource analysis (SRA) with the ISA-level energy model.
- Provide energy consumption function parameterised by some property of the program *or its data*.



# Static Energy Usage Analysis

Original Program:

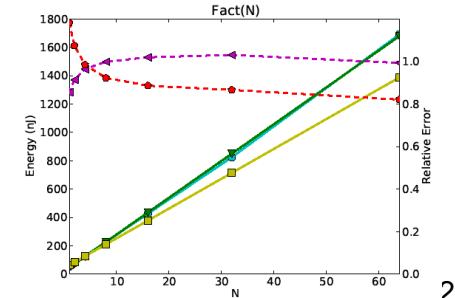
```
int fact (int x) {  
    if (x<=0)a  
        return 1b;  
    return (x *d fact(x-1))c;  
}
```

Extracted Cost Relations:

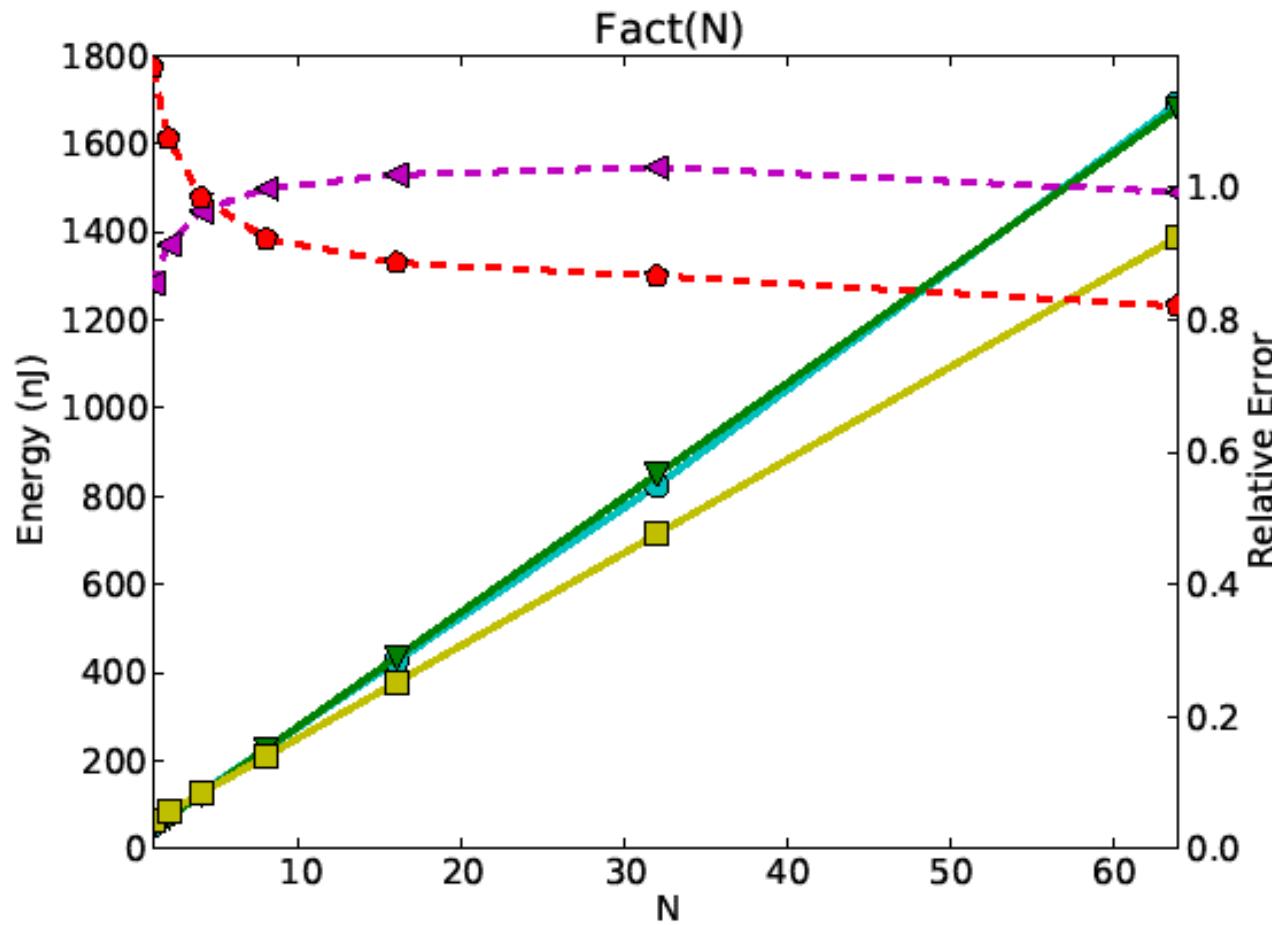
$$\begin{aligned}C_{\text{fact}}(x) &= C_a + C_b && \text{if } x \leq 0 \\C_{\text{fact}}(x) &= C_a + C_c(x) && \text{if } x > 0 \\C_c(x) &= C_d + C_{\text{fact}}(x-1)\end{aligned}$$

- Substitute  $C_a$ ,  $C_b$ ,  $C_d$  with the **actual energy required to execute the corresponding lower-level (machine) instructions.**
- Solve equation using off-the-shelf solvers.
- Result:  $C_{\text{fact}}(x) = (26x + 19.4) \text{ nJ}$

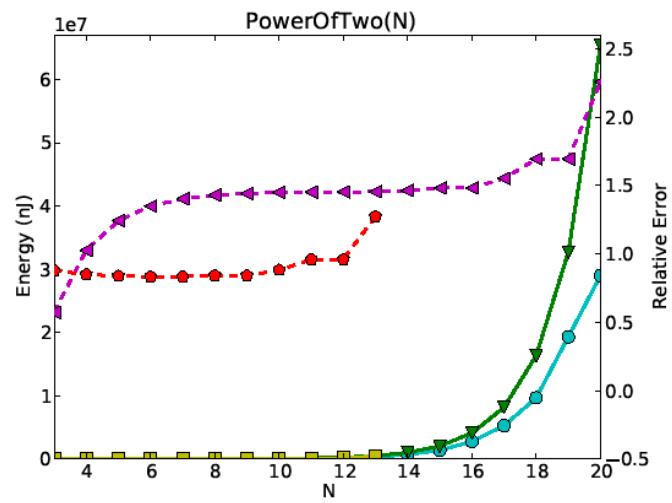
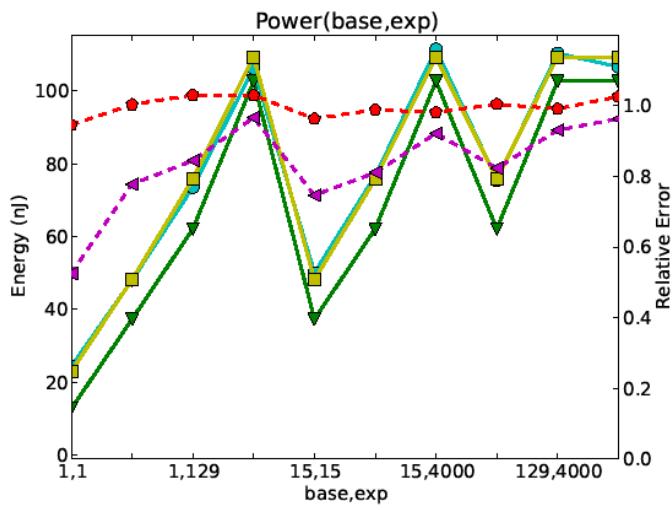
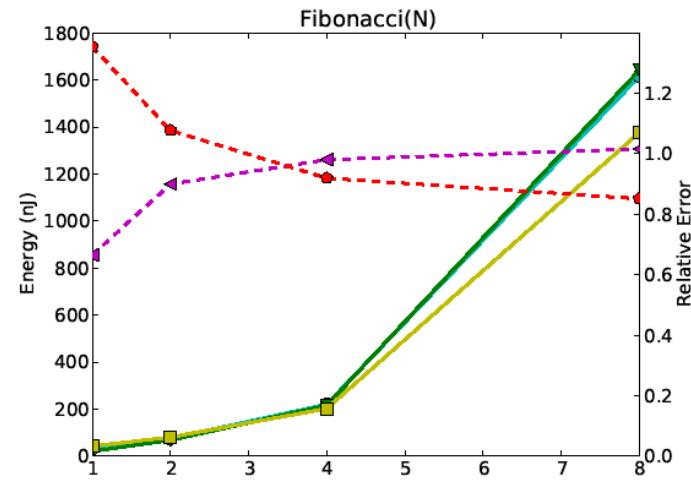
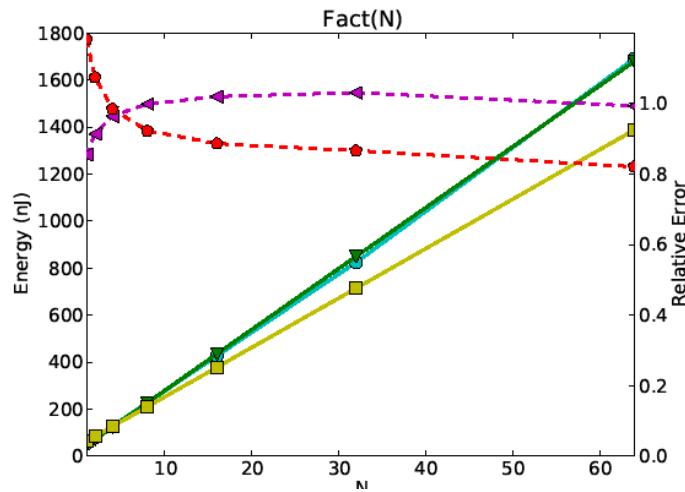
(Note: The above result is based on the XMOS XCore Energy model introduced earlier. It is not using the energy model from the Exercise.)



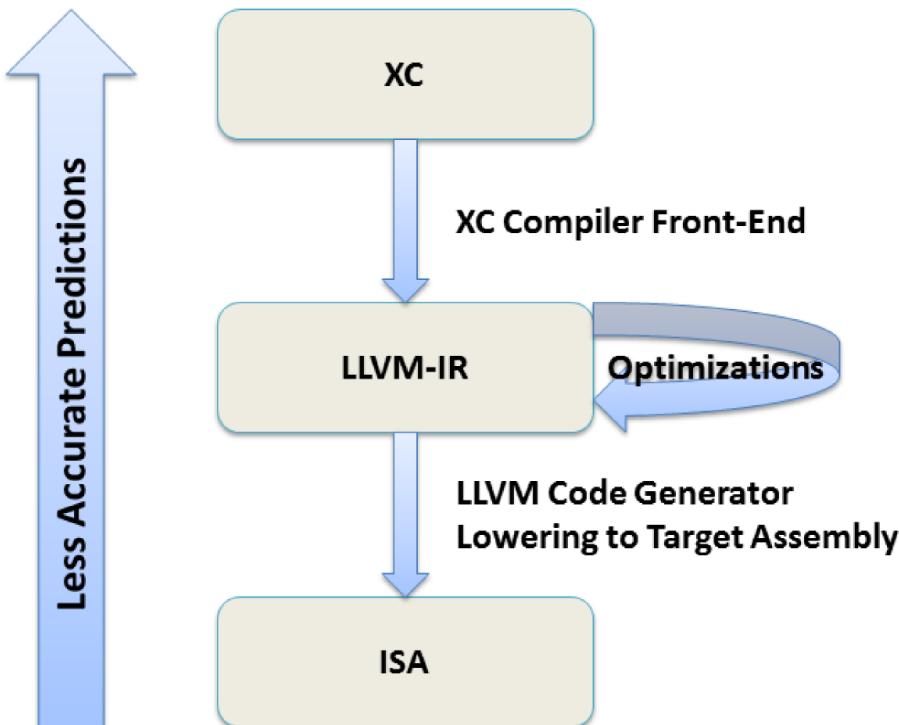
# ISA-Level Analysis Results



# ISA-Level Analysis Results

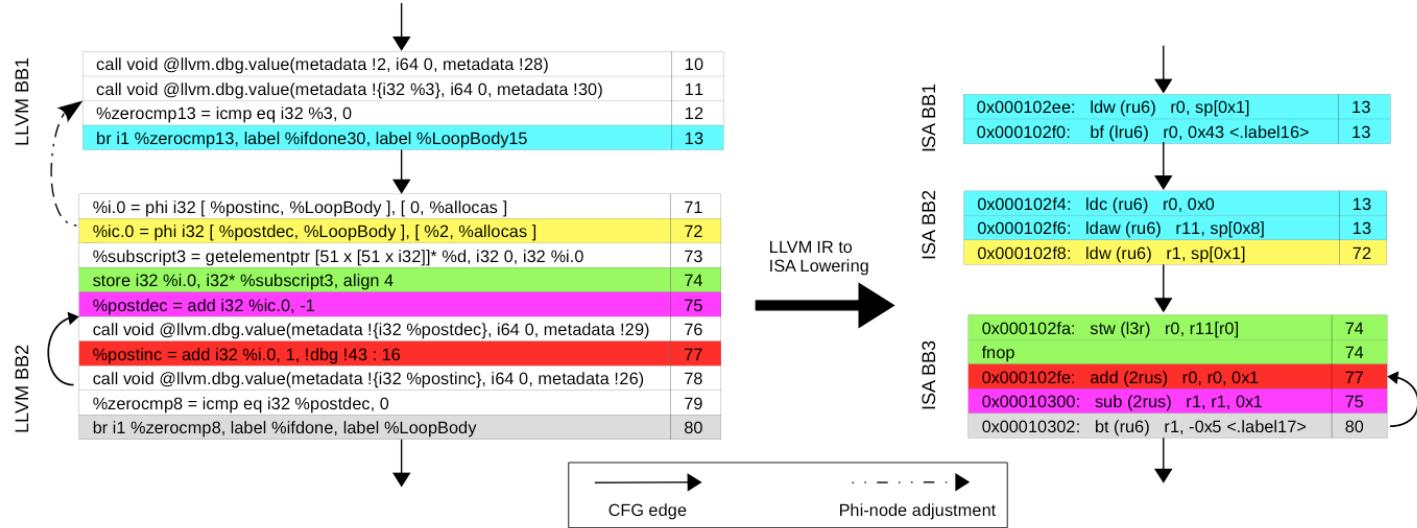


# Analysis Options



- Moving away from the underlying model risks loss of accuracy.
- But it brings us closer to the original source code.

# Energy Consumption of LLVM IR

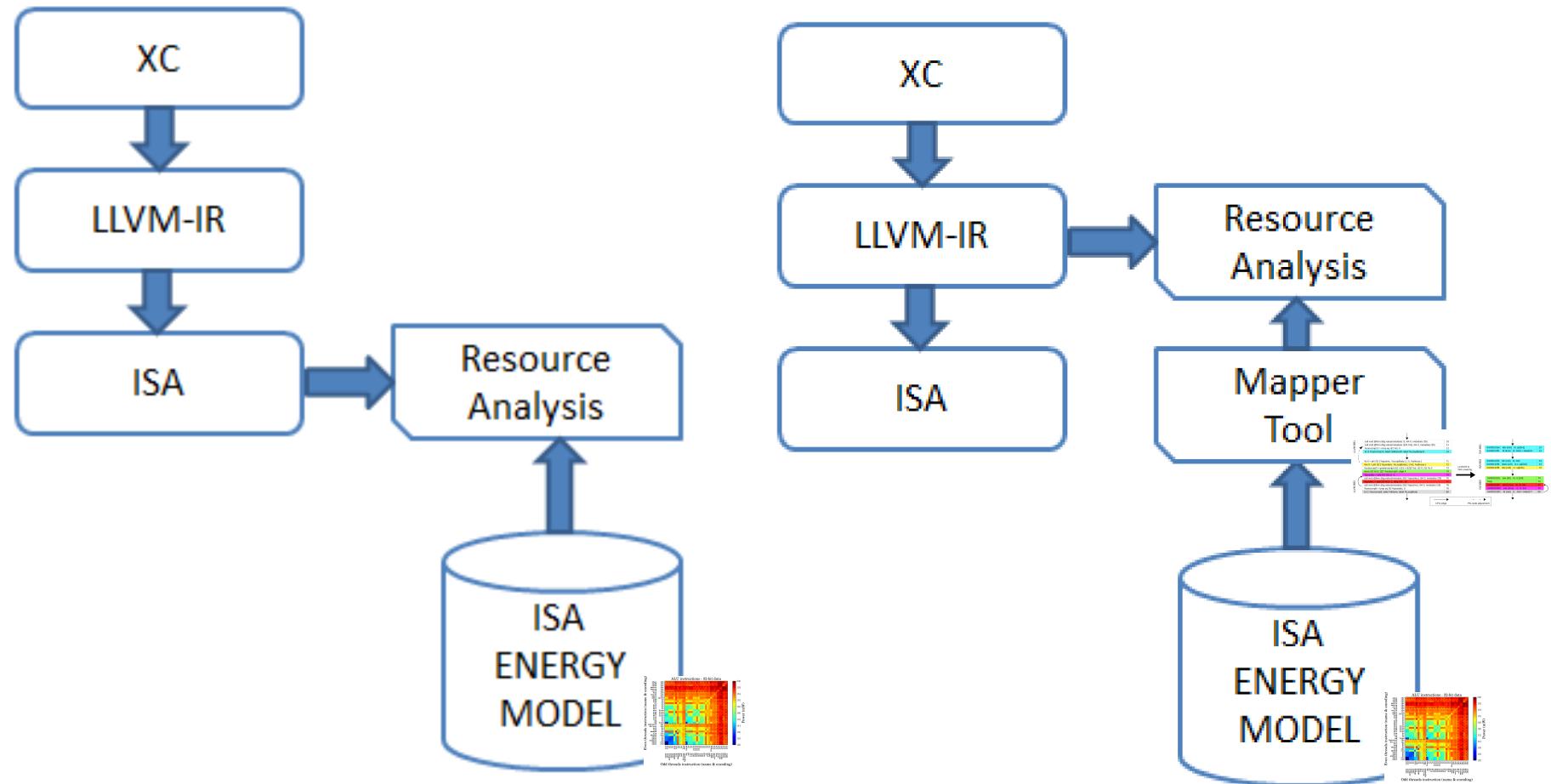


$$E(ir_i) = \sum_{isa_j \in S} E(isa_j)$$

K. Georgiou, S. Kerrison and K. Eder, Oct 2015. “On the Value and Limits of Multi-level Energy Consumption Static Analysis for Deeply Embedded Single and Multi-threaded Programs”. <http://arxiv.org/abs/1510.07095>

U. Liqat, K. Georgiou, S. Kerrison, P. Lopez-Garcia, J.P. Gallagher, M.V. Hermenegildo, K. Eder. Inferring Parametric Energy Consumption Functions at Different Software Levels: ISA vs. LLVM IR. In Proceedings of FOPARA 2015. <http://arxiv.org/abs/1511.01413>

# Analysis at the LLVM-IR Level



N. Grech, K. Georgiou, J. Pallister, S. Kerrison, J. Morse, K. Eder. 2015. Static analysis of energy consumption for LLVM IR programs. In Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems (SCOPES '15). ACM, New York, NY, USA, pages 12-21. <http://dx.doi.org/10.1145/2764967.2764974>

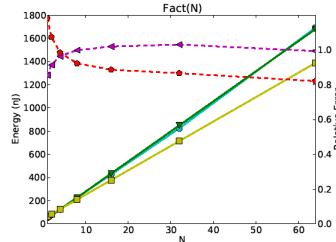
# Learning Objectives

---

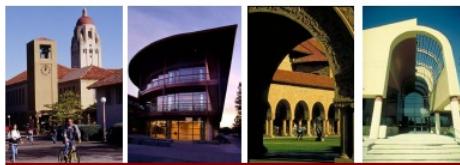
- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- ✓ How to measure the energy consumed by software
- ✓ How to estimate the energy consumed by software *without* measuring
- ✓ How to construct energy consumption models

# Towards Energy Aware Software Engineering

# Energy Transparency



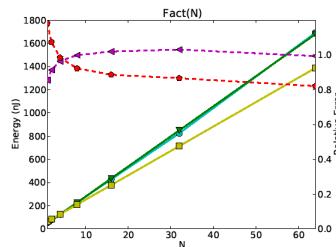
- For HW designers:  
“Power is a 1<sup>st</sup> and last order design constraint.”  
[Dan Hutcheson, VLSI Research, Inc., E<sup>3</sup>S Keynote 2011]
- “Every design is a point in a 2D plane.”  
[Mark Horowitz, E<sup>3</sup>S 2009]



## Scaling Power and the Future of CMOS

Mark Horowitz, EE/CS Stanford University

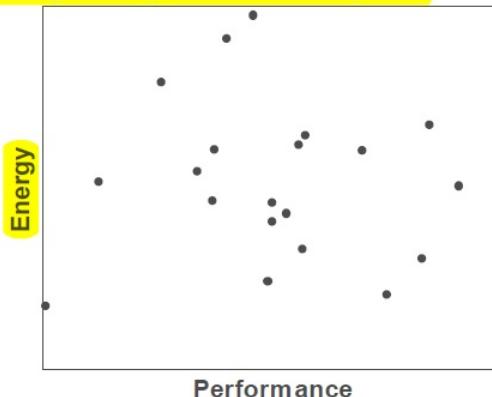
# Energy Transparency



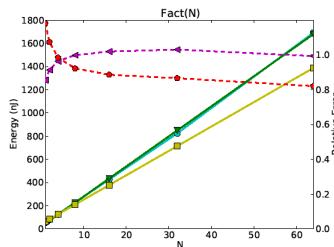
- For HW designers:  
“Power is a 1<sup>st</sup> and last order design constraint.”  
[Dan Hutcheson, VLSI Research, Inc., E<sup>3</sup>S Keynote 2011]
- “Every design is a point in a 2D plane.”  
[Mark Horowitz, E<sup>3</sup>S 2009]

## Optimizing Energy

Every design is a point on a 2-D plane



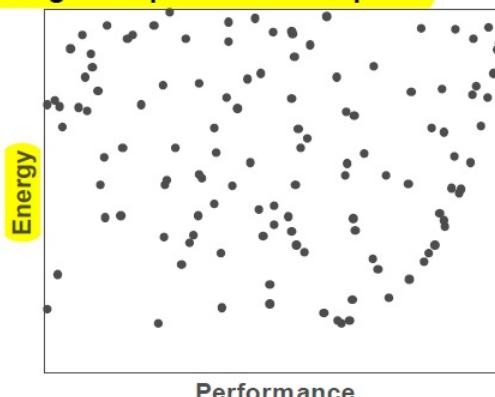
# Energy Transparency



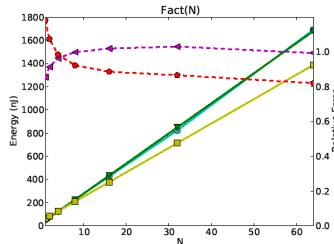
- For HW designers:  
“Power is a 1<sup>st</sup> and last order design constraint.”  
[Dan Hutcheson, VLSI Research, Inc., E<sup>3</sup>S Keynote 2011]
- “Every design is a point in a 2D plane.”  
[Mark Horowitz, E<sup>3</sup>S 2009]

## Optimizing Energy

Every design is a point on a 2-D plane

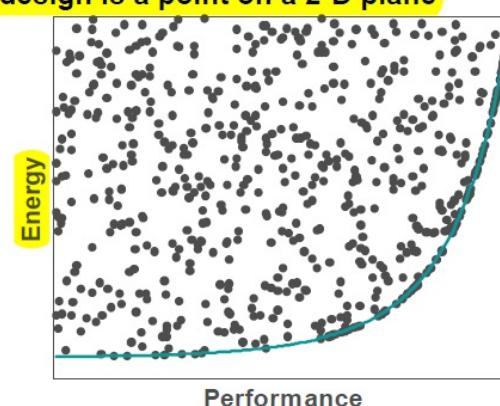


# Energy Transparency



- For HW designers:  
“Power is a 1<sup>st</sup> and last order design constraint.”  
[Dan Hutcheson, VLSI Research, Inc., E<sup>3</sup>S Keynote 2011]
- “Every design is a point in a 2D plane.”  
[Mark Horowitz, E<sup>3</sup>S 2009]

## Optimizing Energy



# More POWER to SW Developers

```
in 5pJ do {...}
```

- Full **Energy Transparency** from HW to SW
- Location-centric programming model

## “Cool” code for green software

A cool programming competition!

Promoting energy efficiency to a 1<sup>st</sup> class SW design goal is still a very important research challenge.



**If you want an ultimate low-power system, then you have to worry about *energy usage at every level in the system design*, and you have to get it right from top to bottom, because any level at which you get it wrong is going to lose you perhaps an order of magnitude in terms of power efficiency.**

The hardware technology has a first-order impact on the power efficiency of the system, but you've also got to have software at the top that avoids waste wherever it can. You need to avoid, for instance, anything that resembles a polling loop because that's just burning power to do nothing.

I think one of the hard questions is whether you can pass the responsibility for the software efficiency right back to the programmer.

**Do programmers really have any understanding of how much energy their algorithms consume?**

I work in a computer science department, and it's not clear to me that we teach the students much about how long their algorithms take to execute, let alone how much energy they consume in the course of executing and how you go about optimizing an algorithm for its energy consumption.

Some of the responsibility for that will probably get pushed down into compilers, but I still think that fundamentally, at the top level, **programmers will not be able to afford to be ignorant about the energy cost of the programs they write.**

What you need in order to be able to work in this way at all is instrumentation that tells you that running this algorithm has this kind of energy cost and running that algorithm has that kind of energy cost.

**You need tools that give you feedback and tell you how good your decisions are.**

Currently the tools don't give you that kind of feedback.

# Thank you for your attention



cādence®

XMOS®



The Royal Academy  
of Engineering



[Kerstin.Eder@bristol.ac.uk](mailto:Kerstin.Eder@bristol.ac.uk)



