Whole Systems Energy Transparency

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
XMOS®

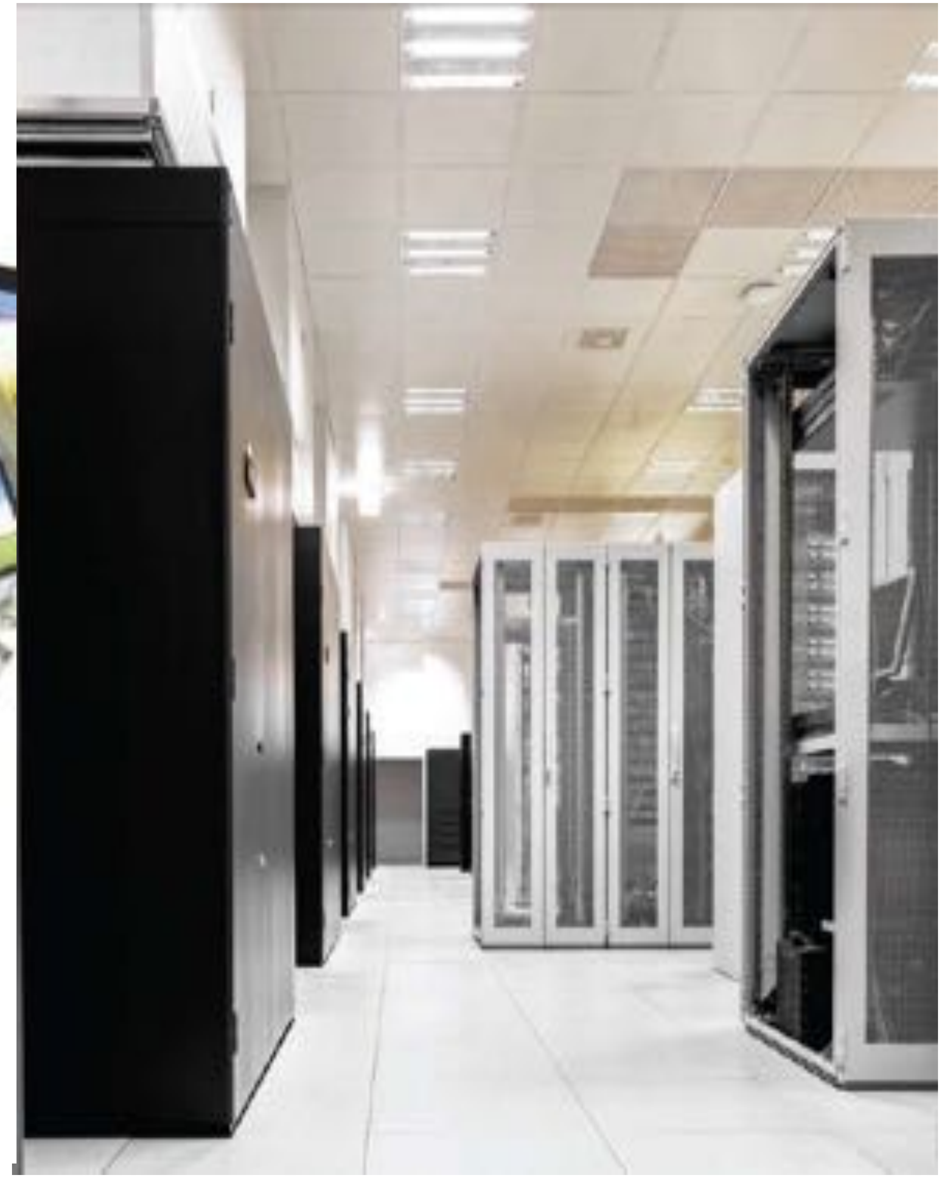
Whole Systems Energy Transparency

More *power* to software developers!

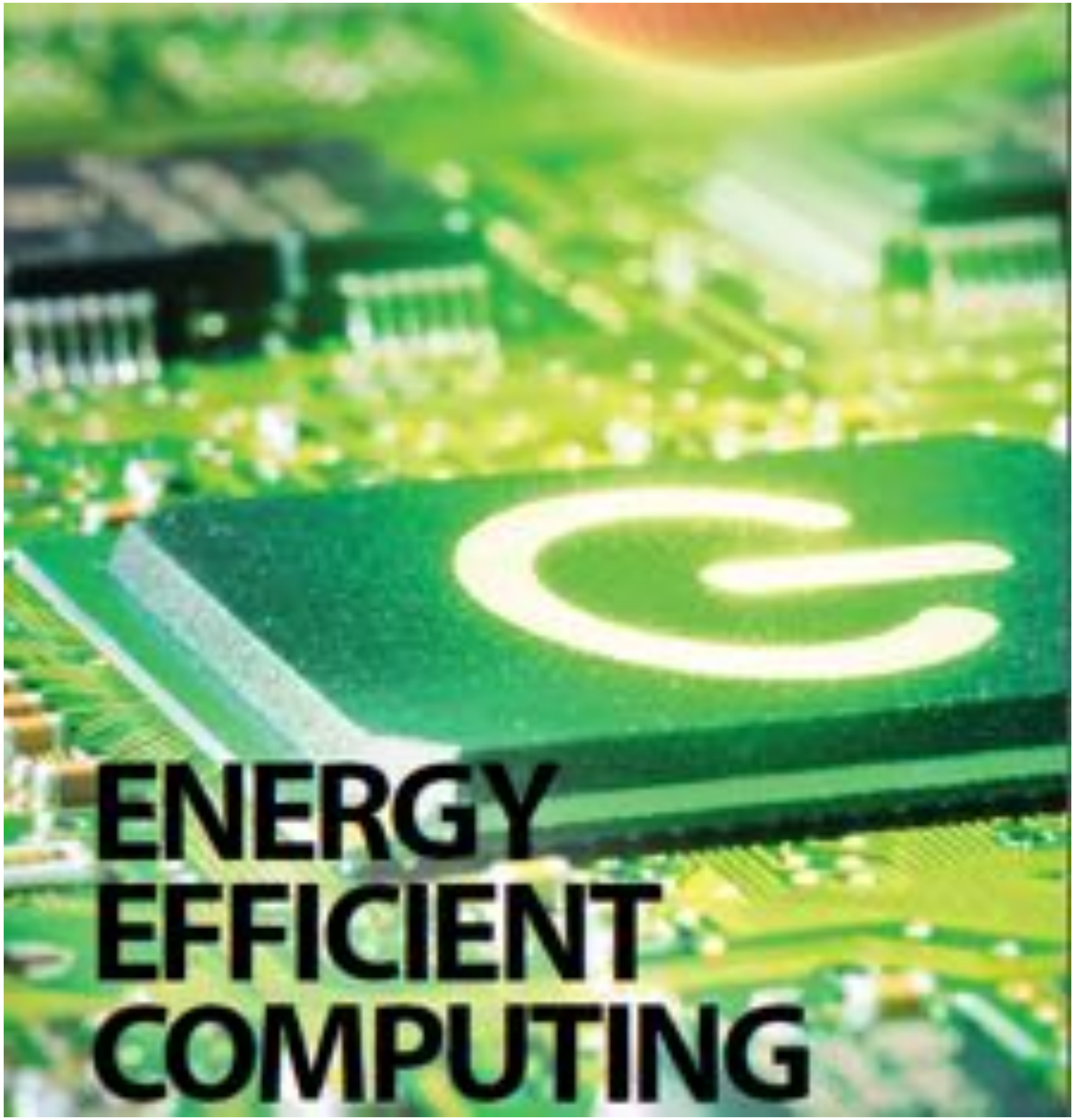
Kerstin Eder

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
XMOS®

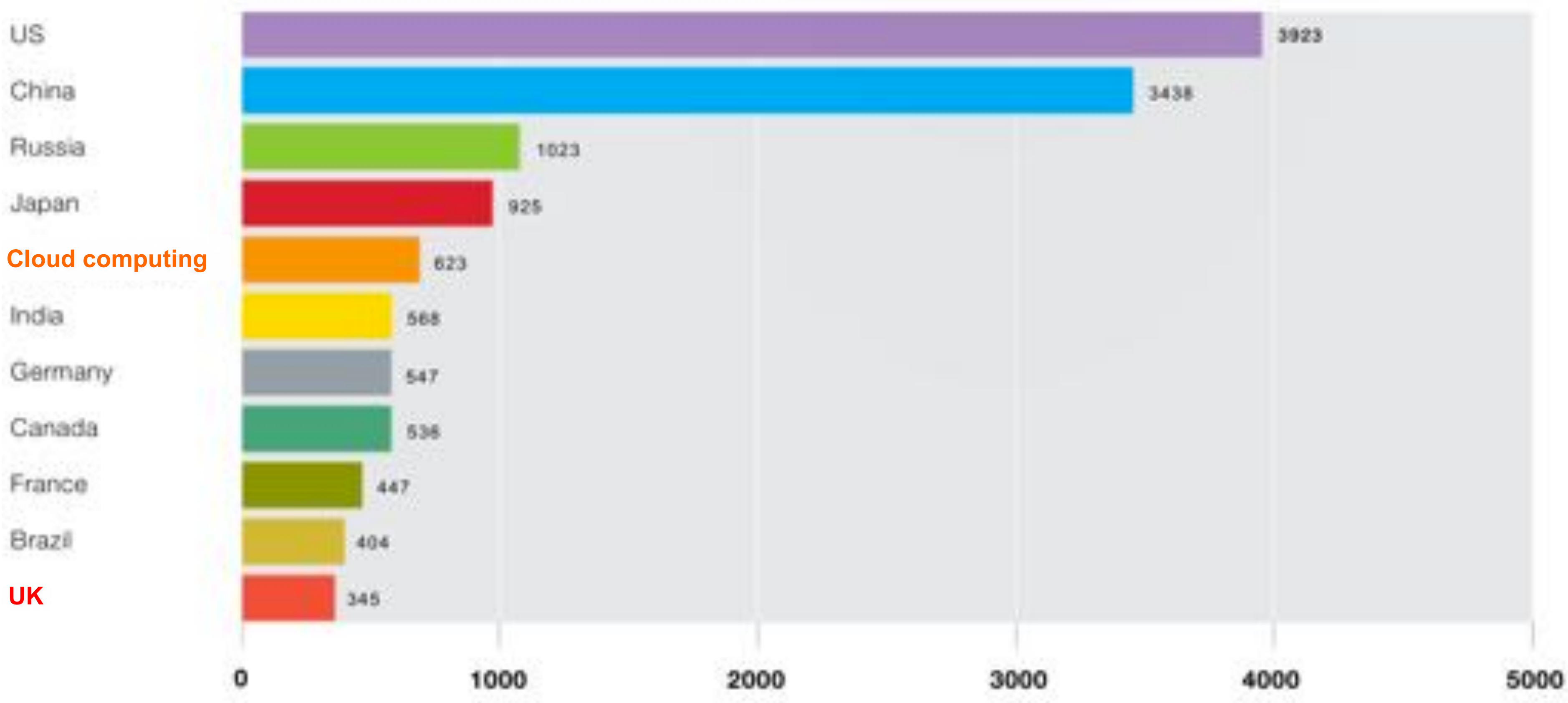


Pictures taken from the Energy Efficient Computing Brochure at:
https://connect.innovateuk.org/documents/3158891/9517074/Energy%20Efficient%20Computing%20Magazine?version=1.0



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

2013

BBC Mobile

News | Sport | Weather | iPlayer | TV |

NEWS TECHNOLOGY

Home World UK England N. Ireland Scotland Wales Business Politics Health Education Sci/Enviro

19 March 2012 Last updated at 17:34

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.

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

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.

Energy Aware System Design

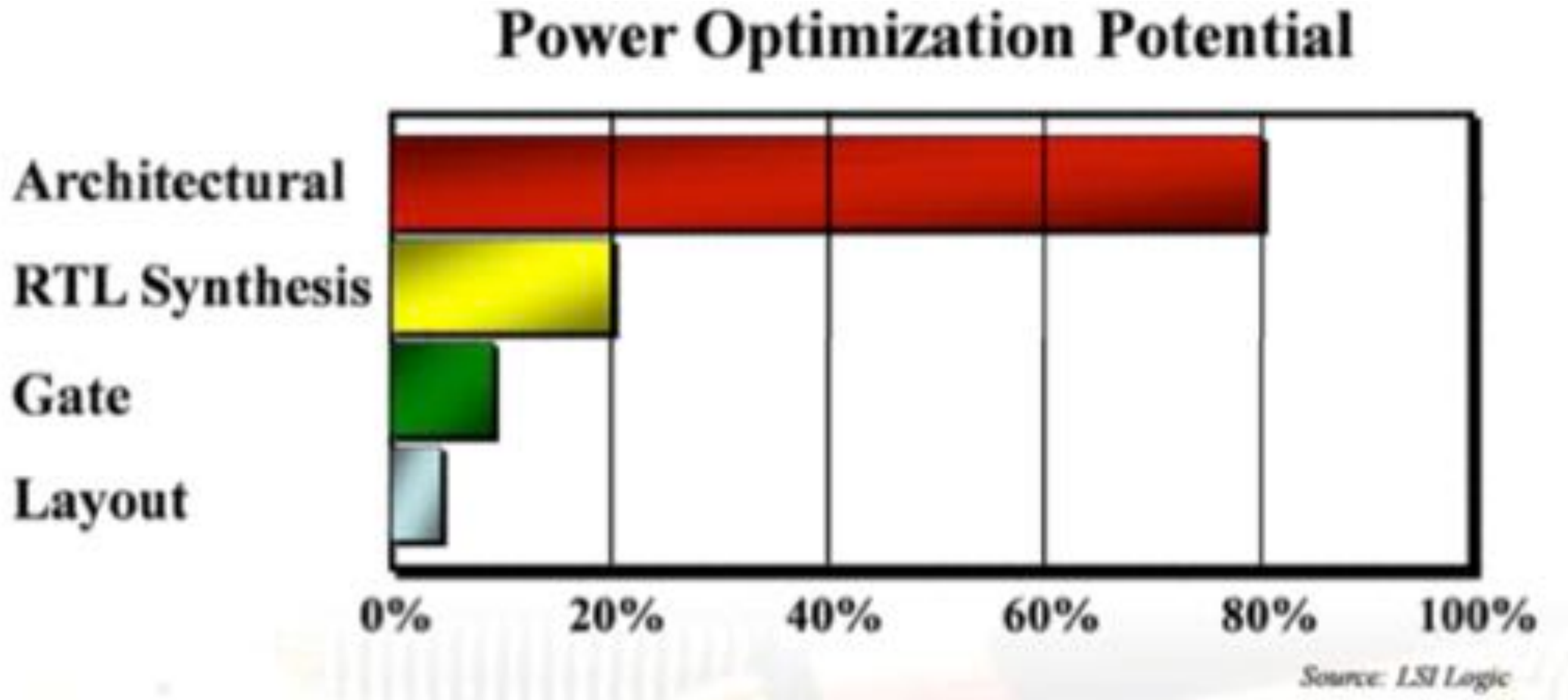
Energy Efficiency of ICT

arChitecture

Blocks

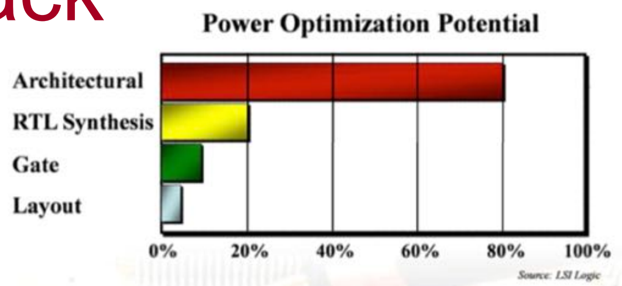
gAtes

Greater Savings at Higher Levels



The Focus is on Software

- Software controls the behaviour of the hardware
- 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 code



Energy Efficiency of ICT

- alGorithms
- soFtware
- compilErs
- Drivers
- arChitecture
- Blocks
- gAtes

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

http://www.clker.com/cliparts/f/0/5/1/1298749934185335695circuit-board.jpg

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 of the progress that has been made and identify ways to minimize the contribution of software to the power dissipation of mixed hardware-software designs.

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?

Picture from www.pd4pic.com

http://scottebates.com/wp-content/uploads/2013/05/transparecny-green.jpg

Energy Transparency

Energy Transparency

Information on energy usage is available for programs:

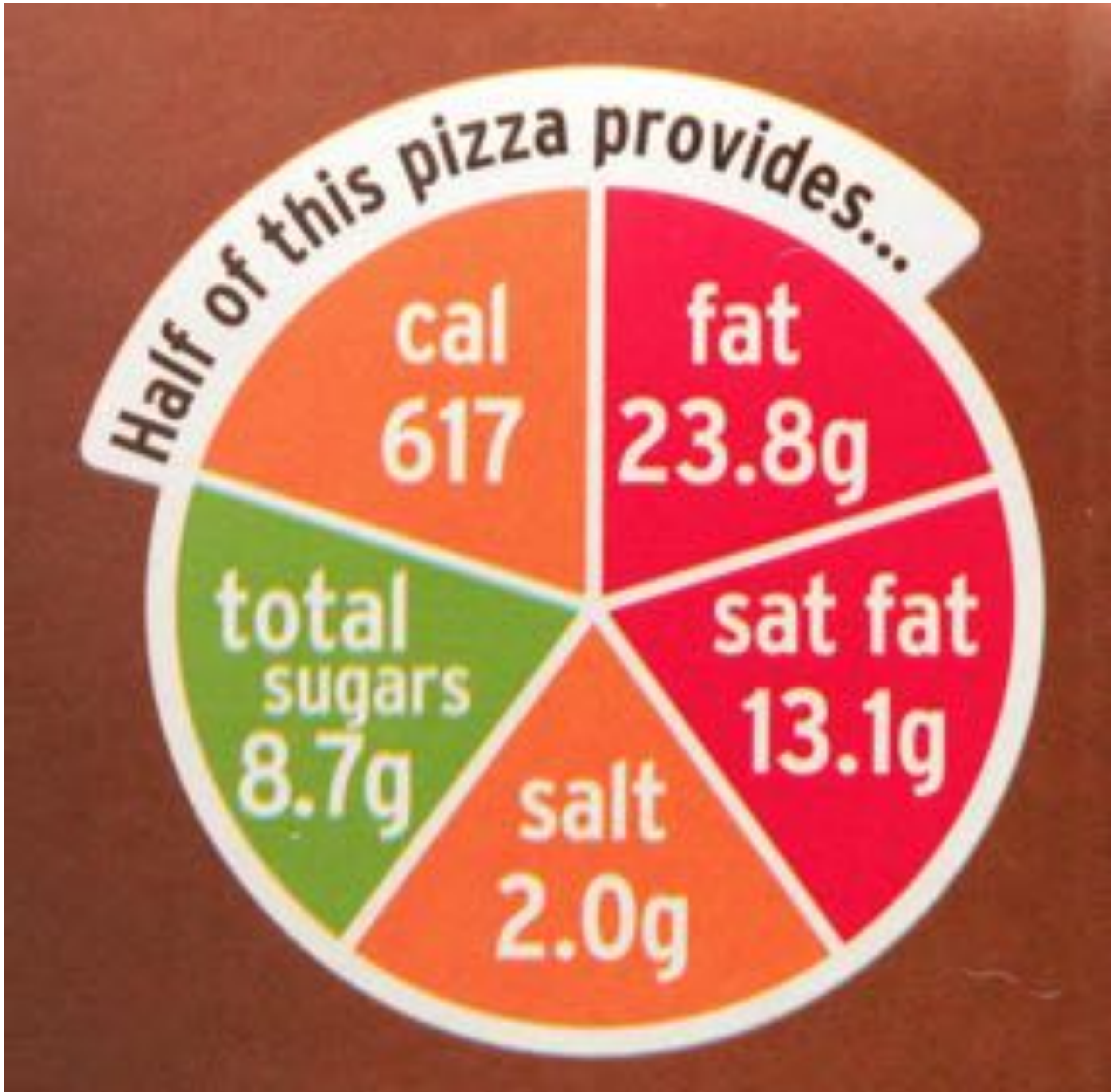
- ideally without executing them, and
- at all levels from machine code to high-level application code.

K. Eder, J.P. Gallagher, P. López-García, H. Muller, Z. Banković, K. Georgiou, R. Haemmerlé, M.V. Hermenegildo, B. Kafle, S. Kerrison, M. Kirkeby, M. Klemen, X. Li, U. Liqat, J. Morse, M. Rhiger, and M. Rosendahl. 2016.

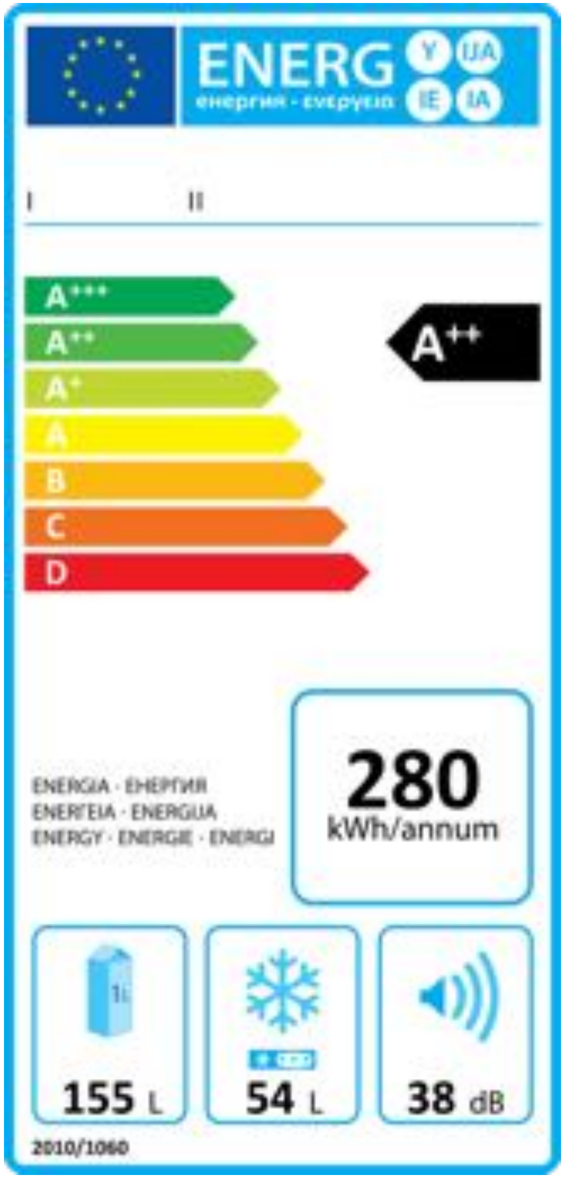
"ENTRA: Whole-systems energy transparency".

Microprocess. Microsyst. 47, PB (November 2016), 278-286. https://doi.org/10.1016/j.micpro.2016.07.003

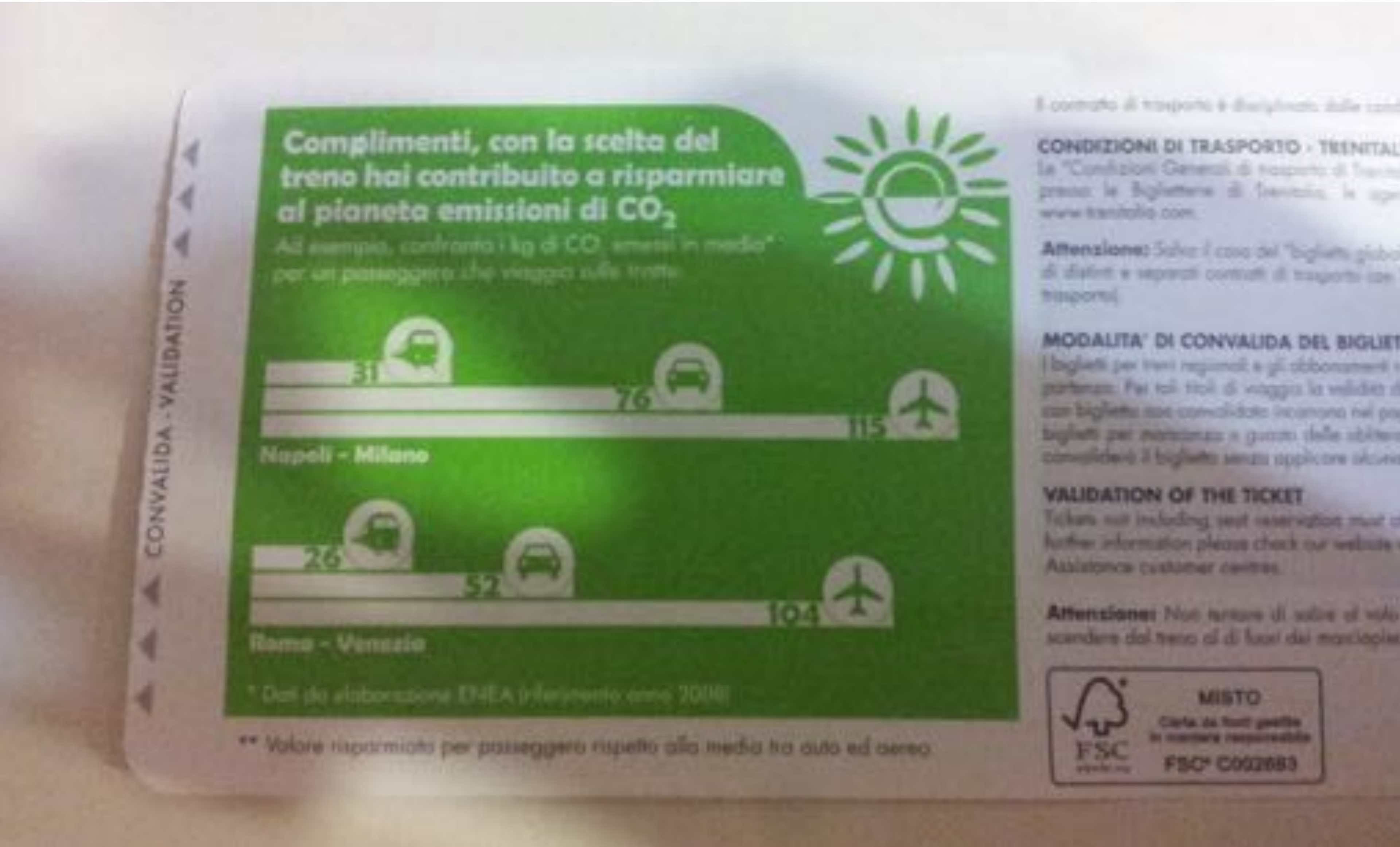
Transparency



Transparency



Transparency



Why Energy Transparency?

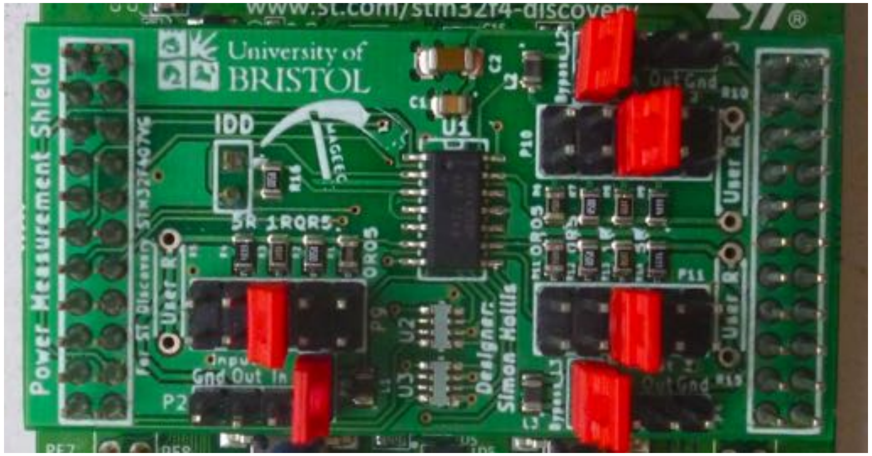
Energy transparency enables a deeper understanding of how algorithms and coding impact on the energy consumption of a computation when executed on hardware.

K. Eder, J.P. Gallagher, P. López-García, H. Muller, Z. Banković, K. Georgiou, R. Haemmerlé, M.V. Hermenegildo, B. Kafle, S. Kerrison, M. Kirkeby, M. Klemen, X. Li, U. Liqat, J. Morse, M. Rhiger, and M. Rosendahl. 2016.

"ENTRA: Whole-systems energy transparency".

Microprocess. Microsyst. 47, PB (November 2016), 278-286. https://doi.org/10.1016/j.micpro.2016.07.003

Measuring the Energy Consumption of Computation



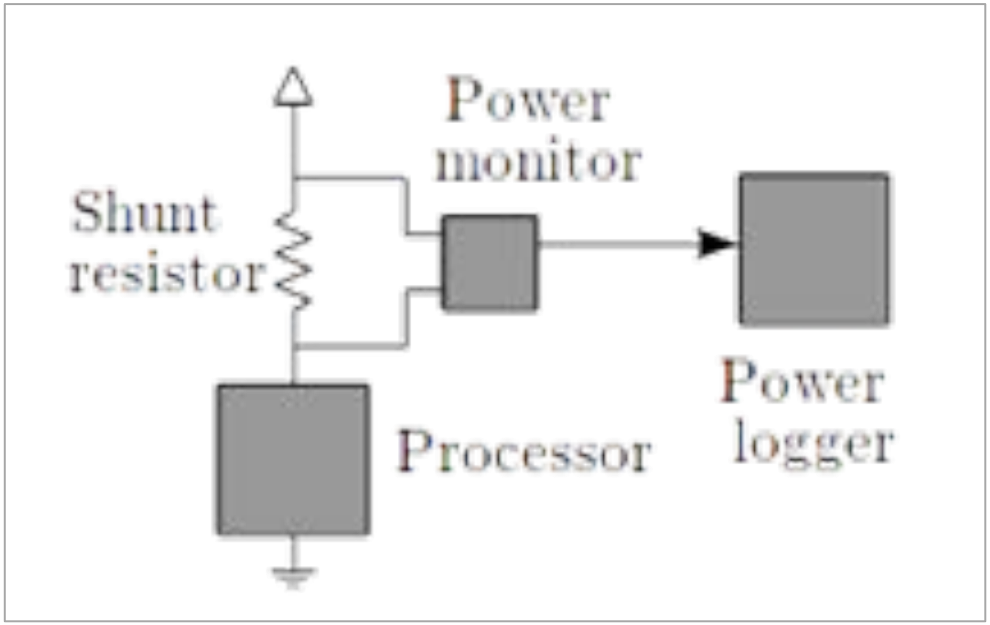
Measuring Power

Measure voltage drop across the resistor

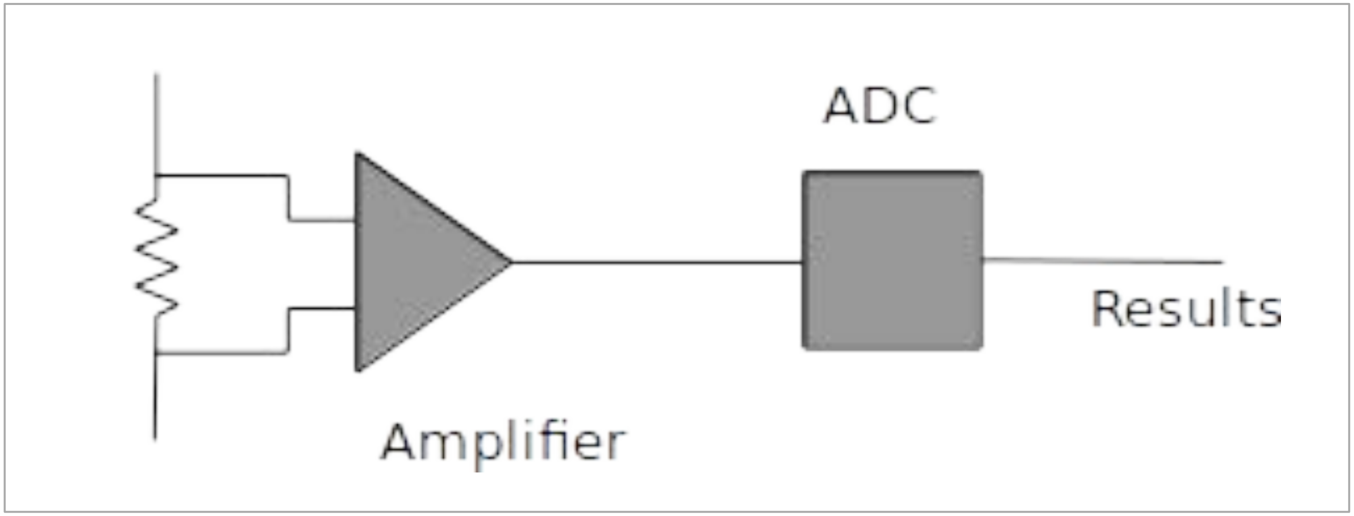
$I = V_{shunt} / R_{shunt}$ to find the current.

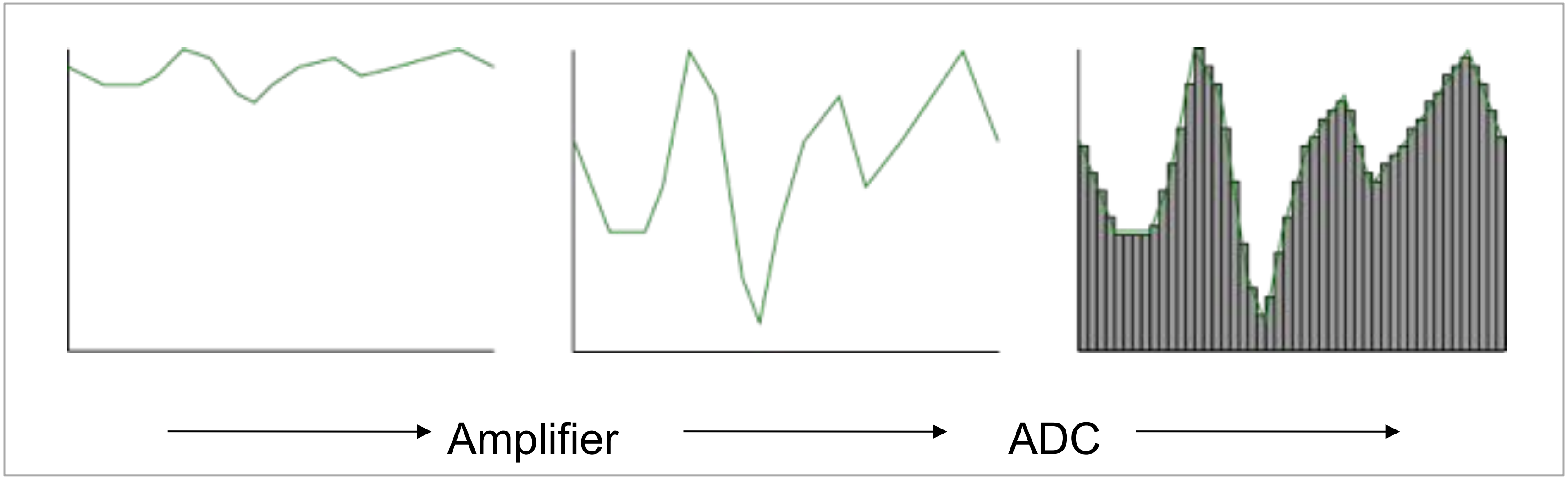
Measure voltage at one side of the resistor

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



The Power Monitor



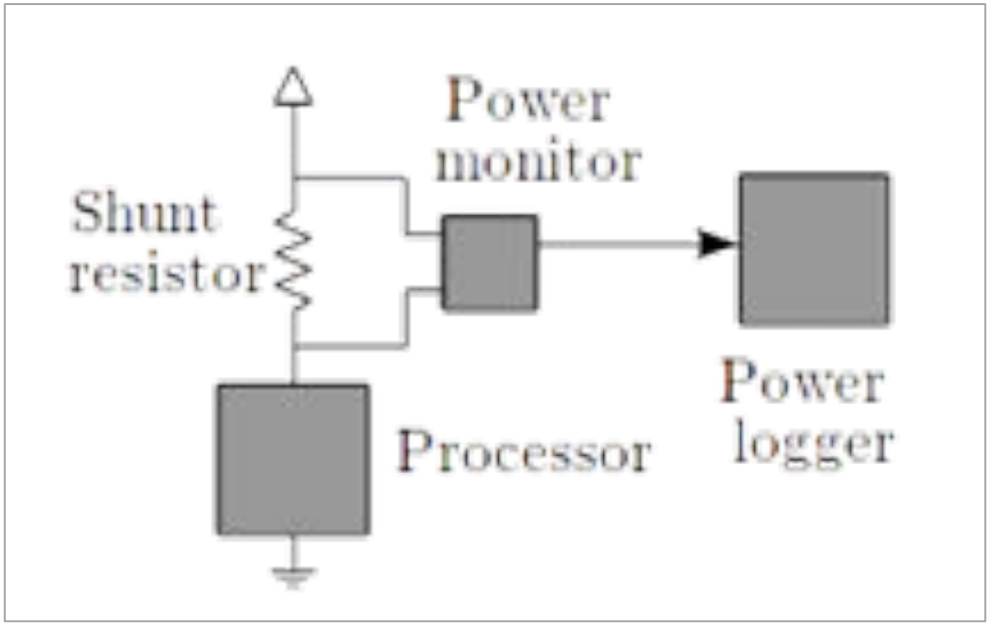


Measuring Power

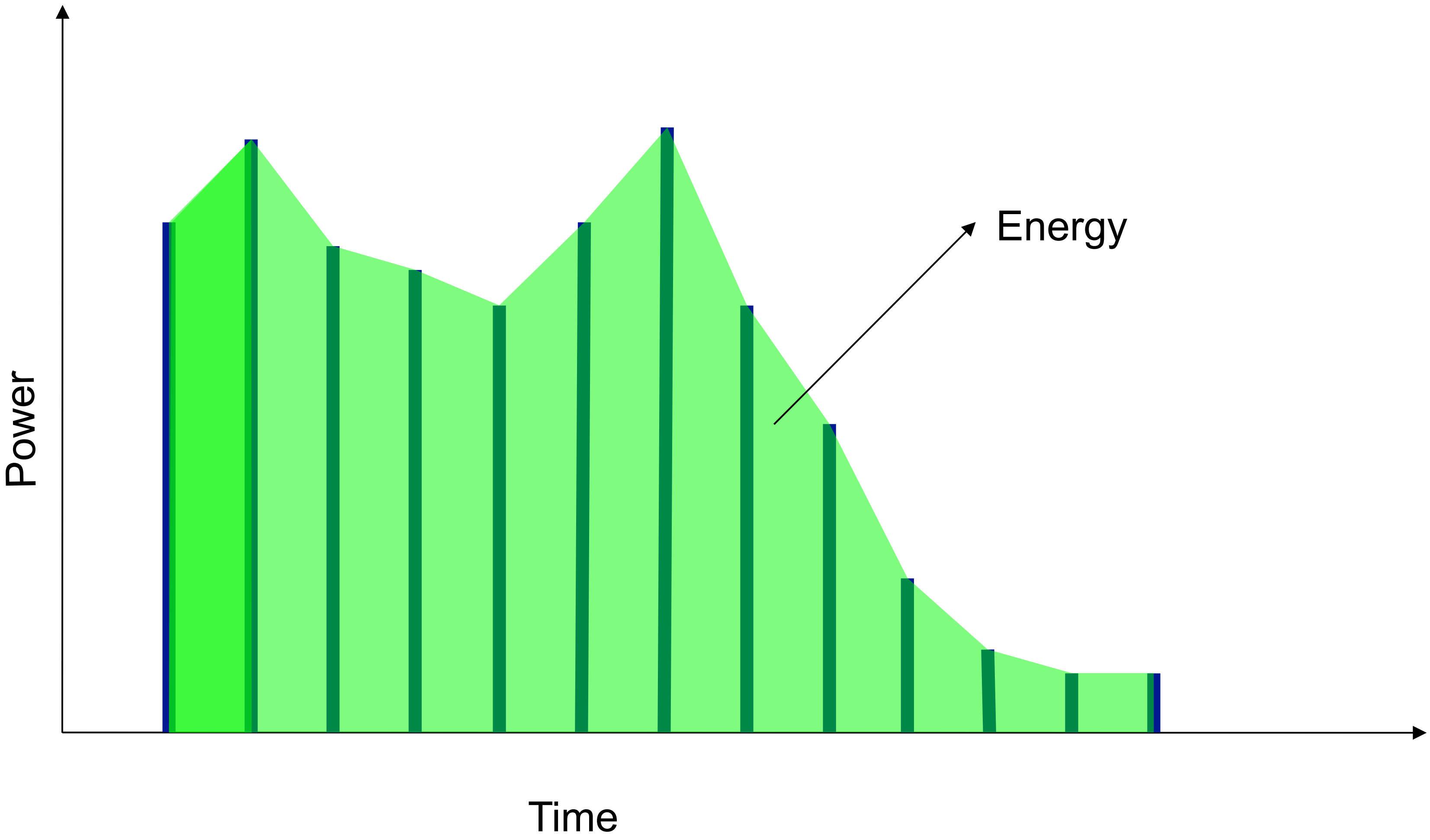
Repeat frequently, timestamp each sample

Measure voltage drop across the resistor — $I = V_{shunt} / R_{shunt}$ to find the current

Measure voltage at one side of the resistor — $P = I \times V$ to calculate the power



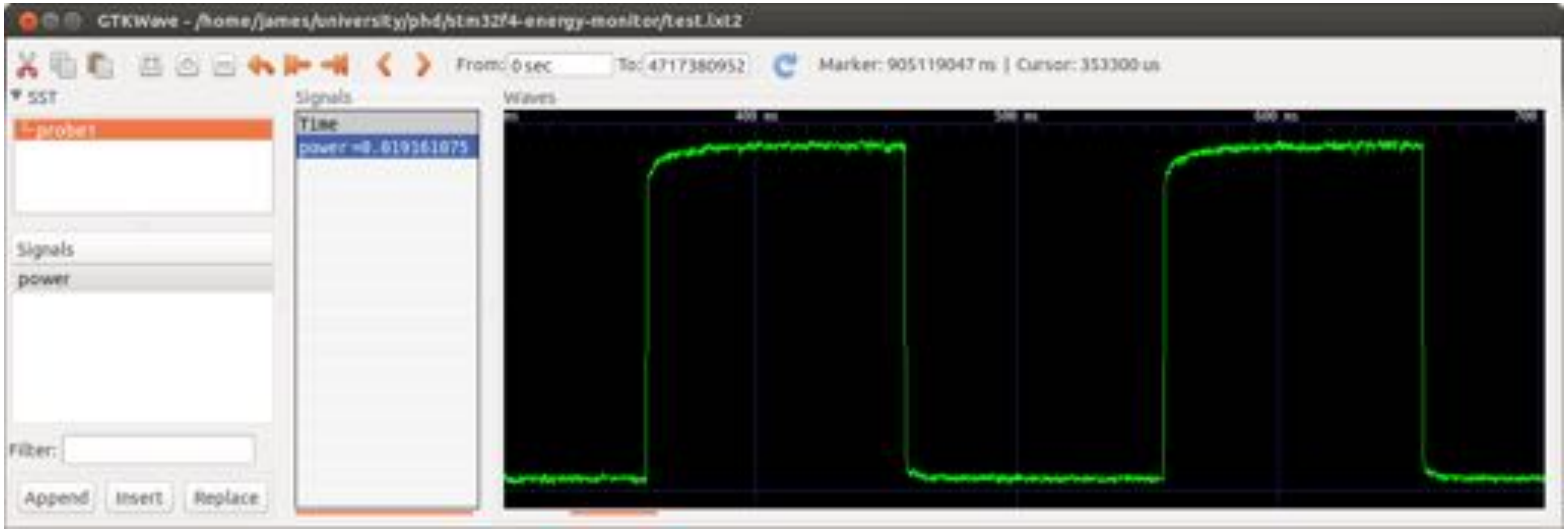
Measuring Energy



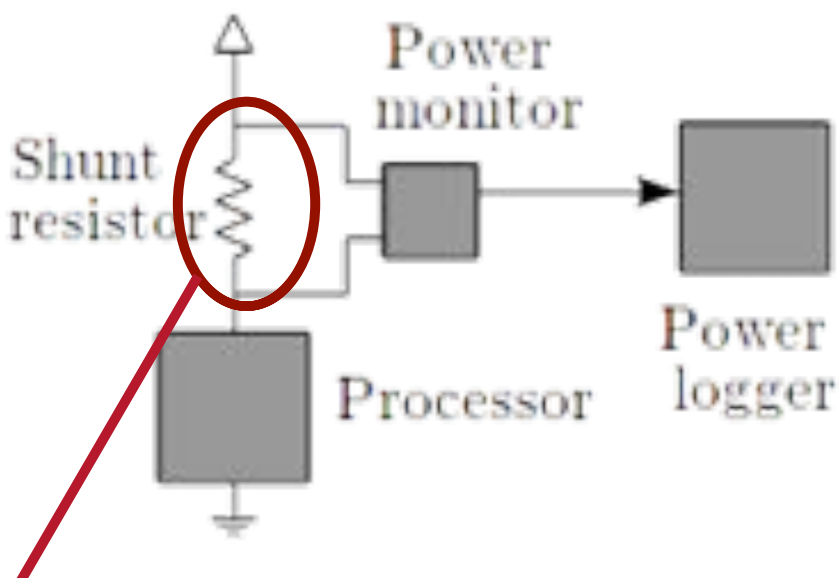
How much data?

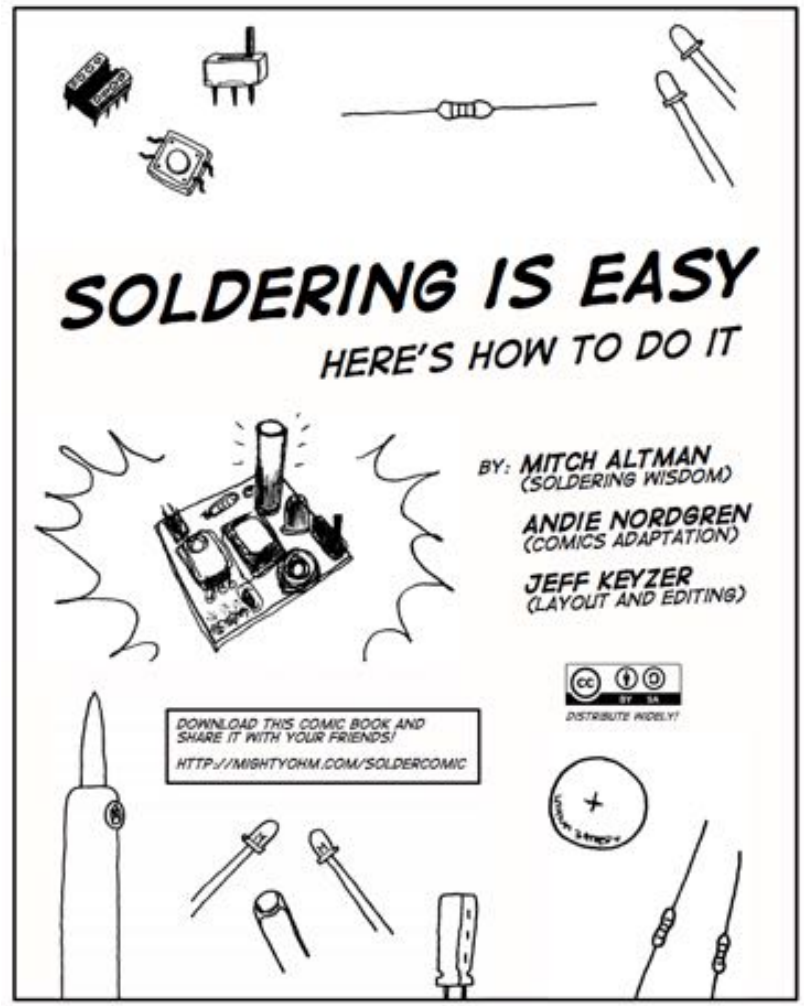
Currently 500,000 Samples/second

6,000,000 S/s possible in bursts

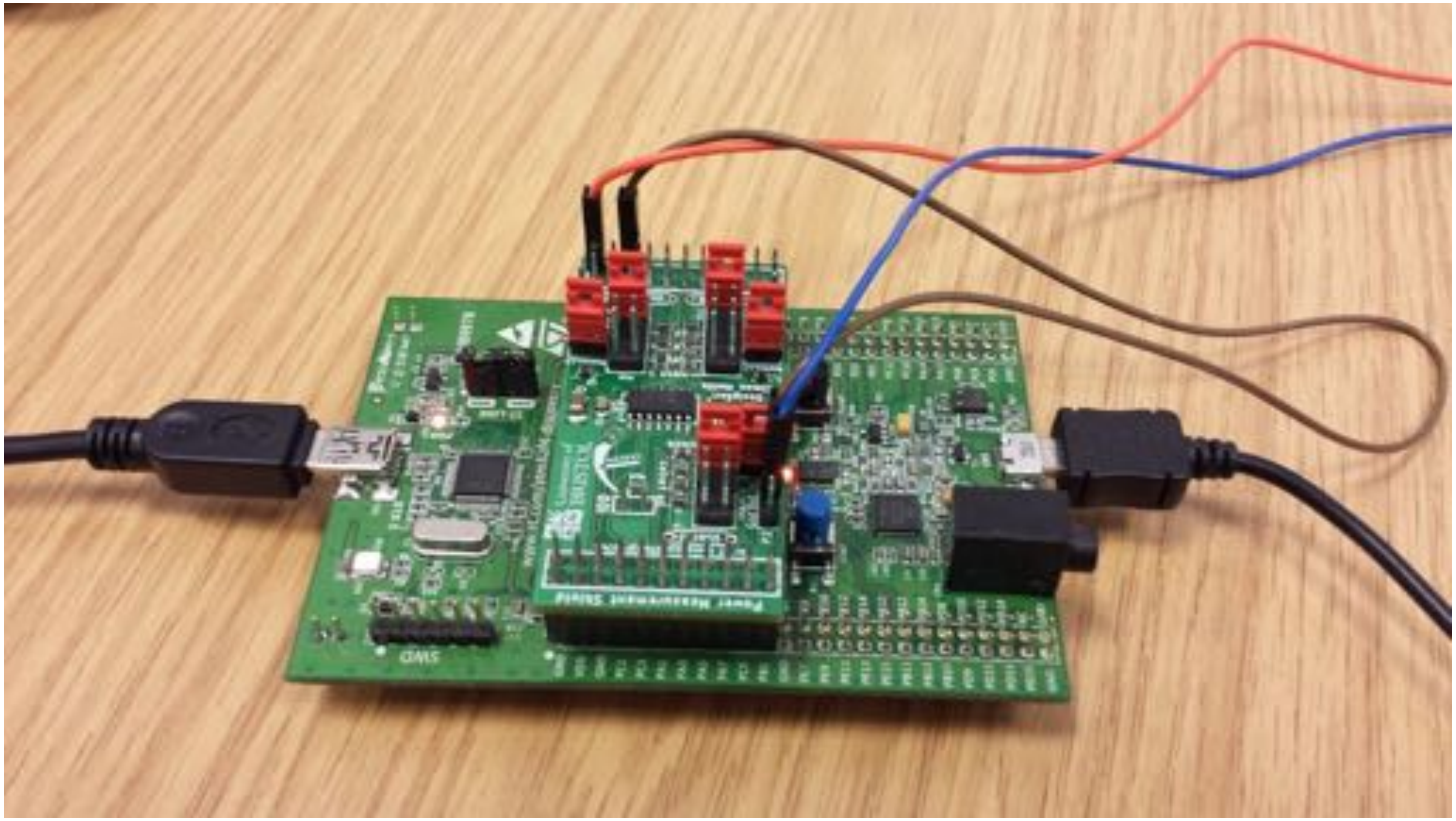


The Showstopper ☹





Open Energy Measurement Board



http://mageec.org/

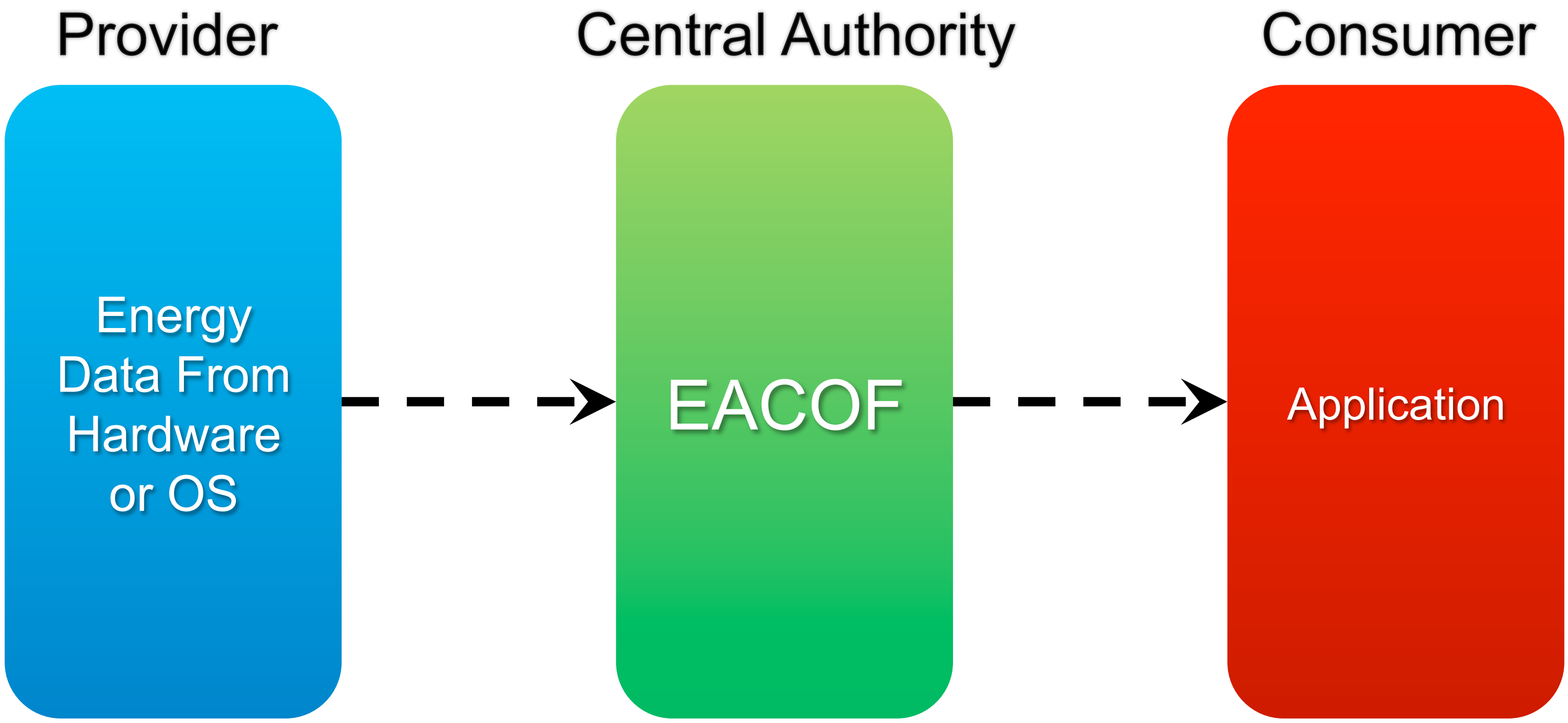
Dynamic Energy Monitoring

The EACOF

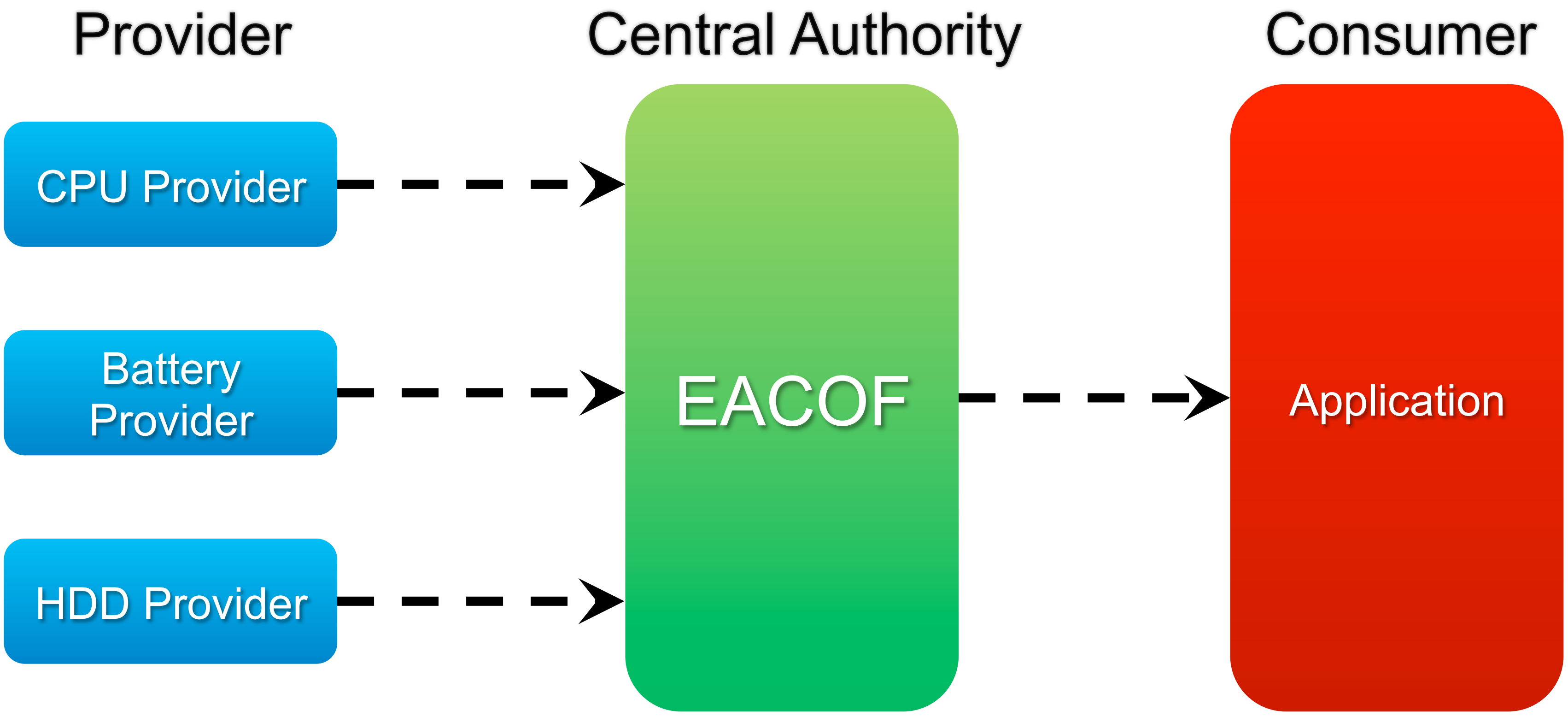
A simple Energy-Aware COmputing Framework

https://github.com/eacof

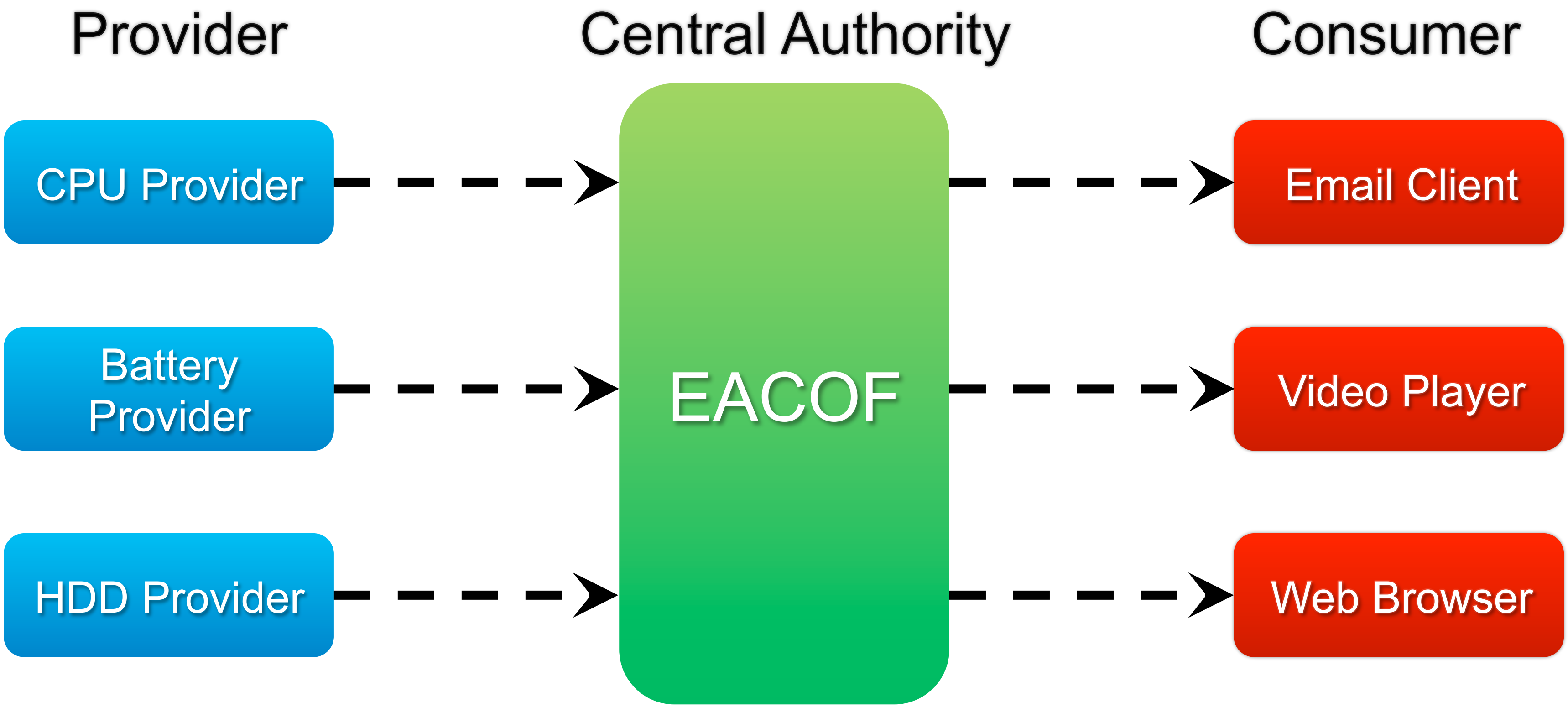
High Level


Provider
Central Authority
Consumer
Energy
Data From
Hardware
or OS
EACOF
Application

Providers


Provider
Central Authority
Consumer
CPU Provider
Battery Provider
HDD Provider
EACOF
Application

Consumers



Provider
Central Authority
Consumer
CPU Provider
Battery Provider
HDD Provider
EACOF
Email Client
Video Player
Web Browser

Comparing Sorting Algorithms

- Sorting of integers in [0,255]

		Data Type											
		uint8_t			uint16_t			uint32_t			uint64_t		
Algorithm	Num Elements	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

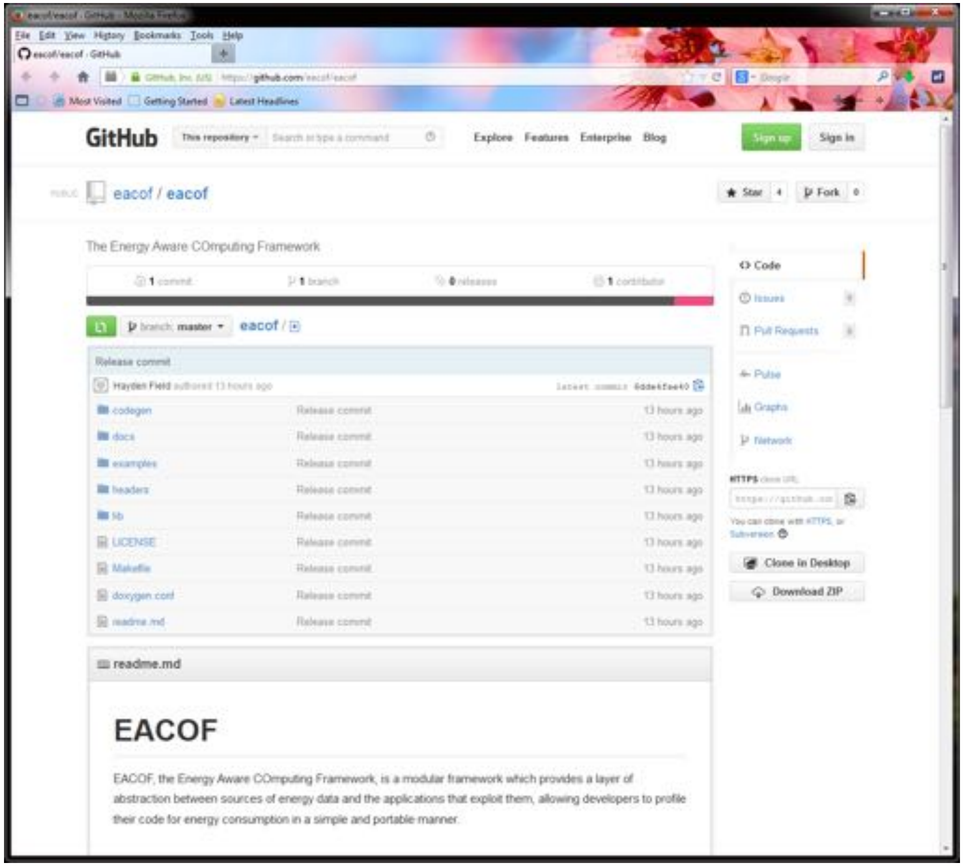
♦ Counting Sort:
75% more energy for 64 bit compared to 8 bit values

• Sorting 64 bit values takes less time than sorting 8 bit values, but consumed more energy

★ Average power variations between algorithms

H. Field, G. Anderson and K. Eder. "EACOF: A Framework for Providing Energy Transparency to enable Energy-Aware Software Development". *29th ACM Symposium On Applied Computing*. pp. 1194–1199. March 2014, ACM.
DOI: 10.1145/2554850.2554920

Invitation: EACOF is open source!



github.com/eacof

Static Analysis for Energy Consumption

ENTRA

Whole-Systems
Energy Transparency

The ENTRA Project

- Whole Systems ENergy TRAnsparency

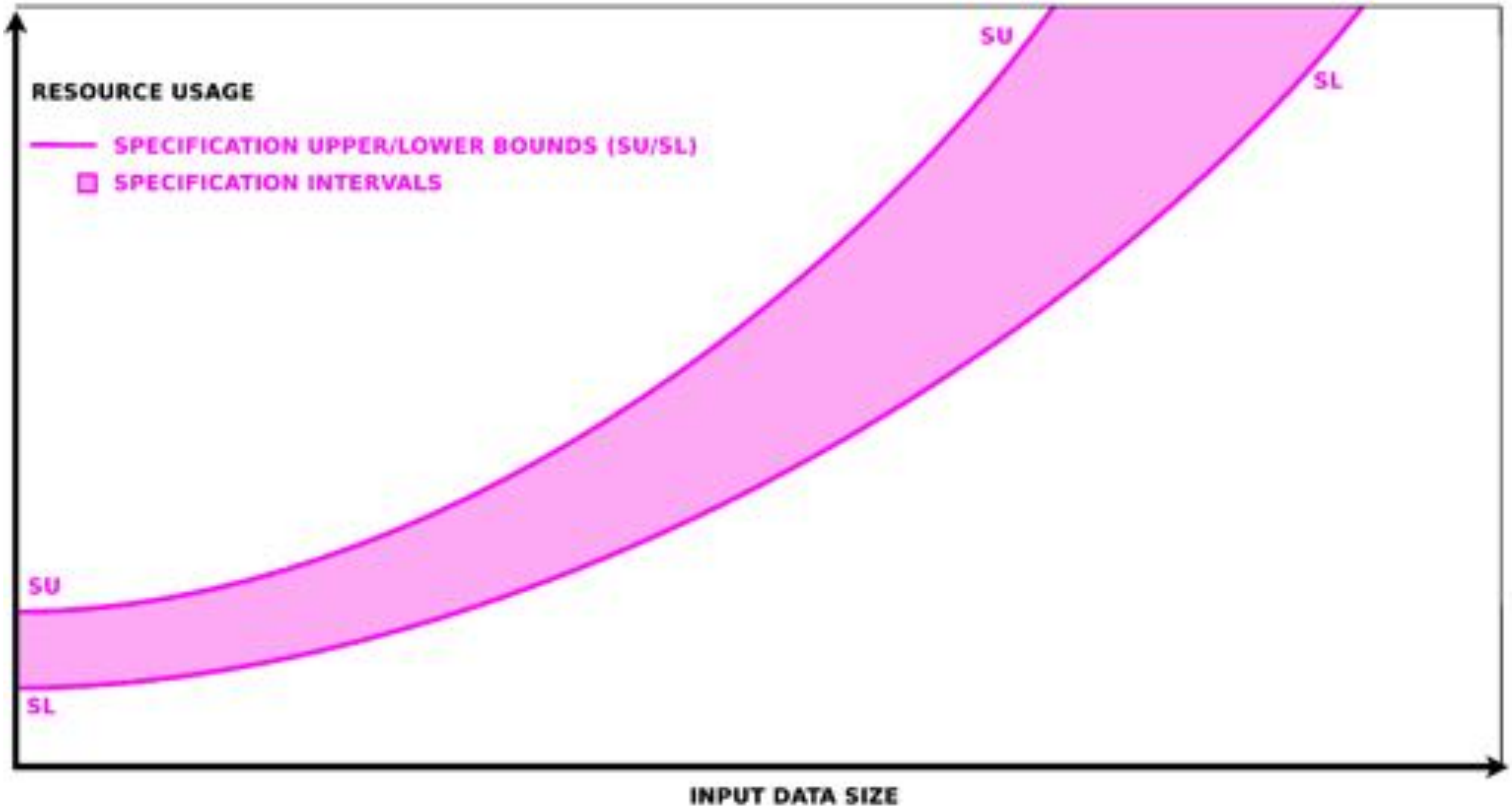
EC FP7 FET MINECC:

*“**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).”*

SRA for Energy Consumption

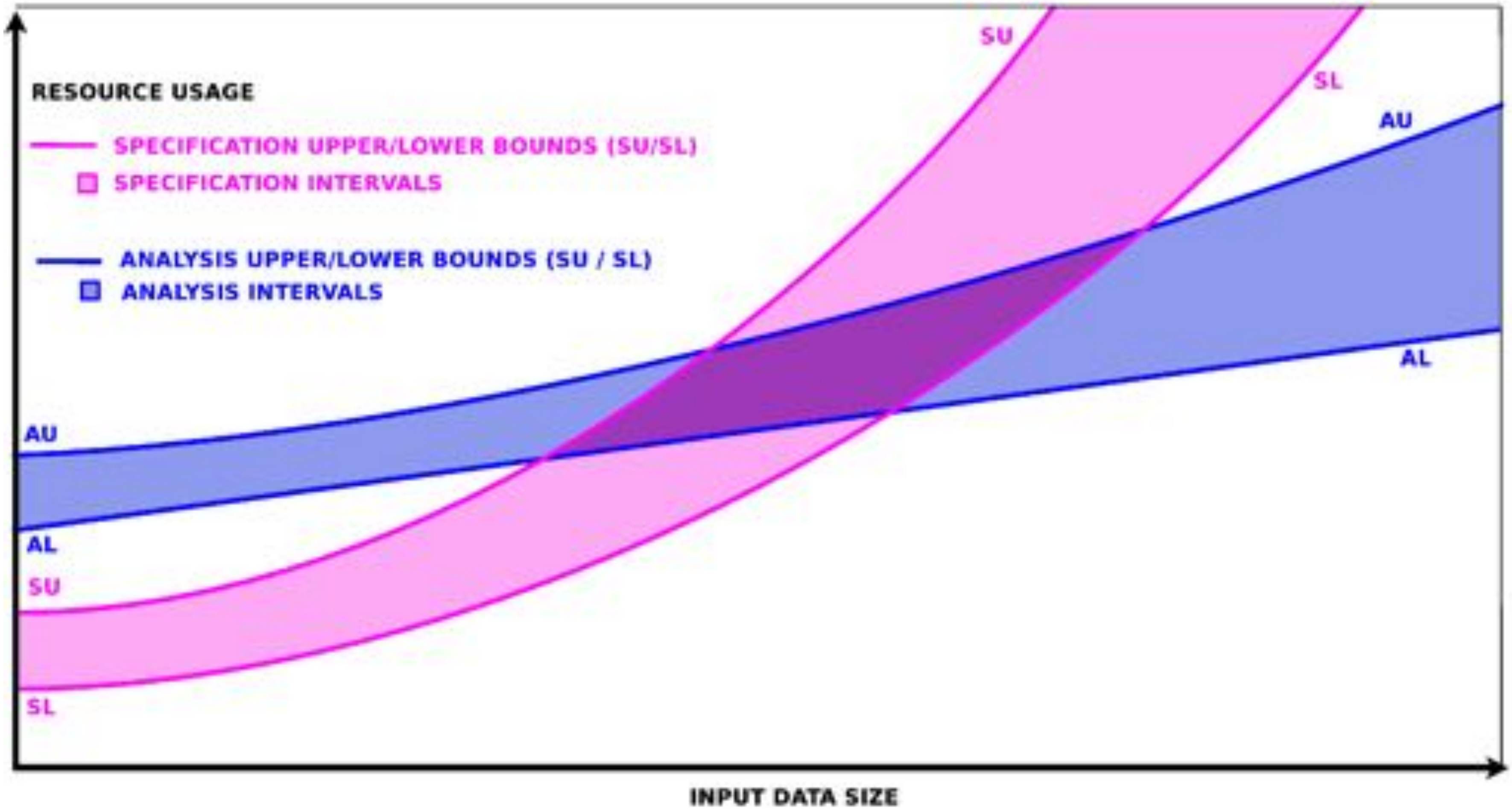
- Adaptation of traditional resource usage analysis techniques to *energy consumption*.
- Techniques automatically infer upper and lower bounds on energy usage of a program.
- Bounds expressed using monotonic arithmetic functions per procedure parameterized by program's input size.
- Verification can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

Specified Resource Usage



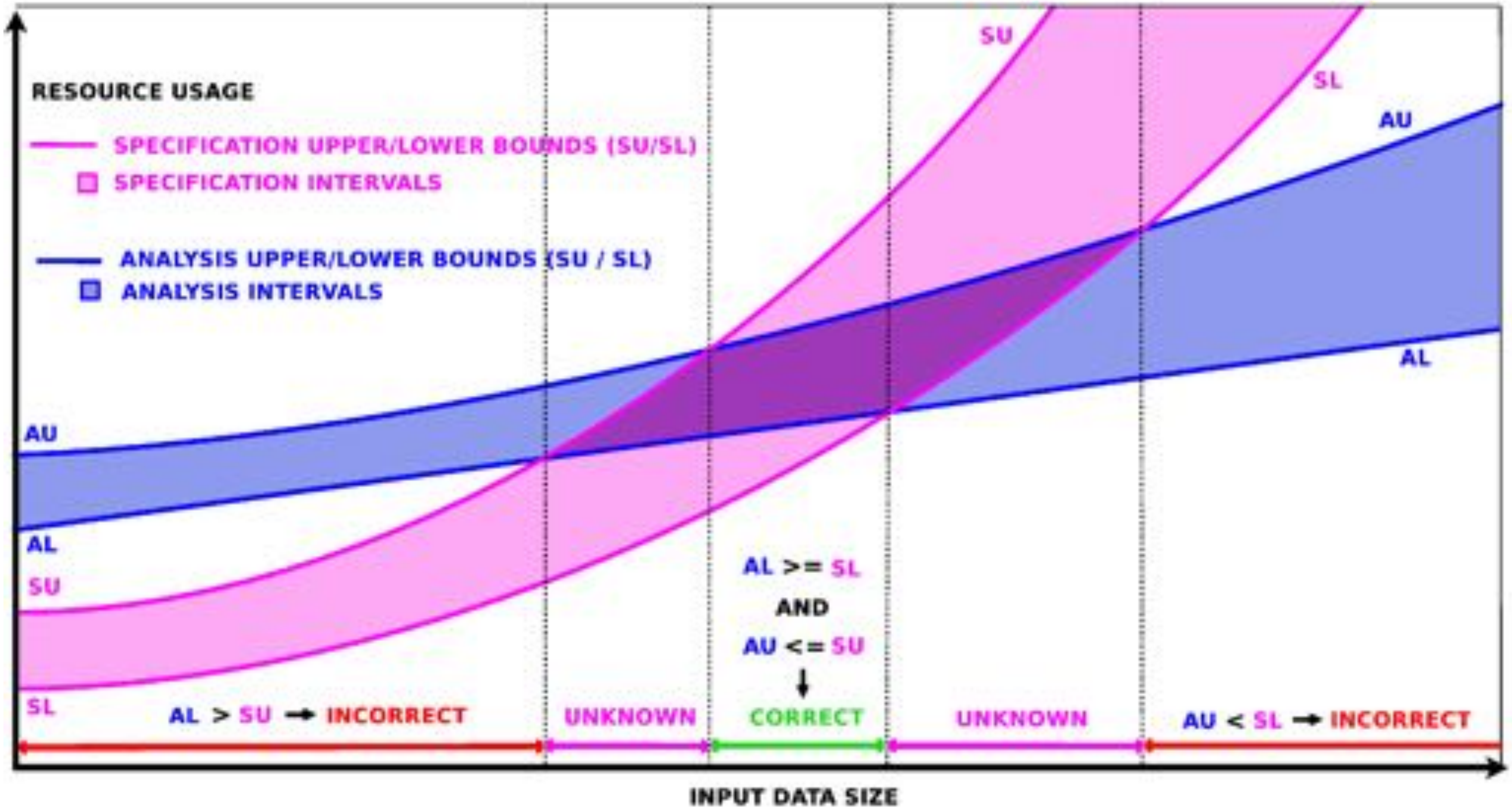
Source: Pedro Lopez Garcia, IMDEA Software Research Institute

Analysis Result



Source: Pedro Lopez Garcia, IMDEA Software Research Institute

Verification



Source: Pedro Lopez Garcia, IMDEA Software Research Institute

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:

$$C_{fact}(x) = C_a + C_b \quad \text{if } x<=0$$
$$C_{fact}(x) = C_a + C_c(x) \quad \text{if } x>0$$
$$C_c(x) = C_d + C_{fact}(x-1)$$

- 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

∧ × +

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

http://www.speechinaction.org/wp-content/uploads/2012/10/dilemma.jpg

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)

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.

XCore Energy Modelling

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

$$E_{\mathrm{p}} = P_{\mathrm{base}} N_{\mathrm{idle}} T_{\mathrm{clk}} + \sum_{t=1}^{N_t} \sum_{i \in \mathrm{ISA}} \left(\left(M_t P_i O + P_{\mathrm{base}} \right) N_{i,t} T_{\mathrm{clk}} \right)$$

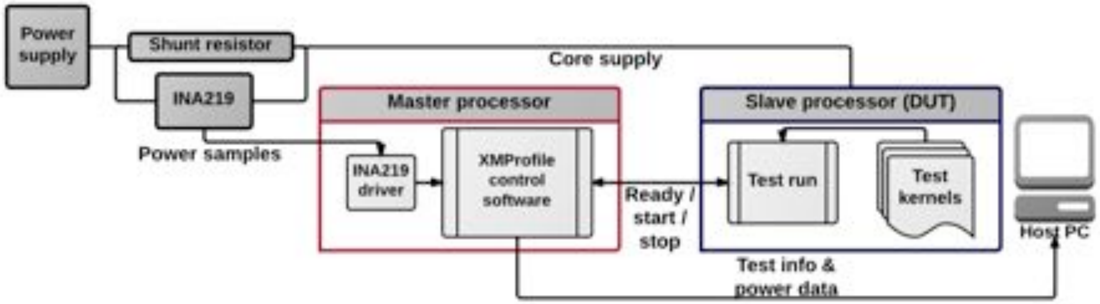
Idle base power and duration

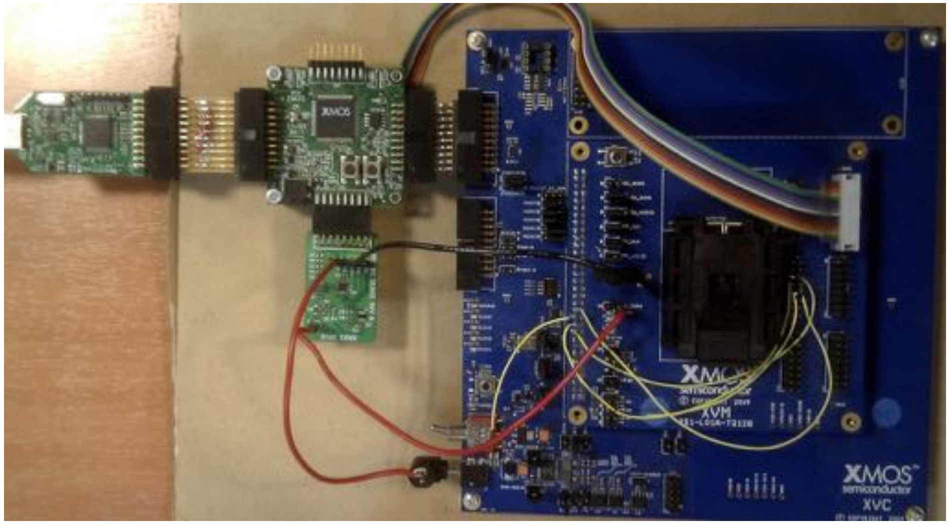
Concurrency cost, instruction cost, generalised overhead, base power and duration

- 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

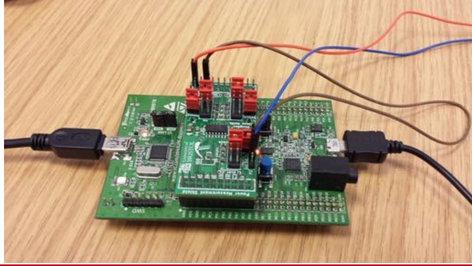
The set up...

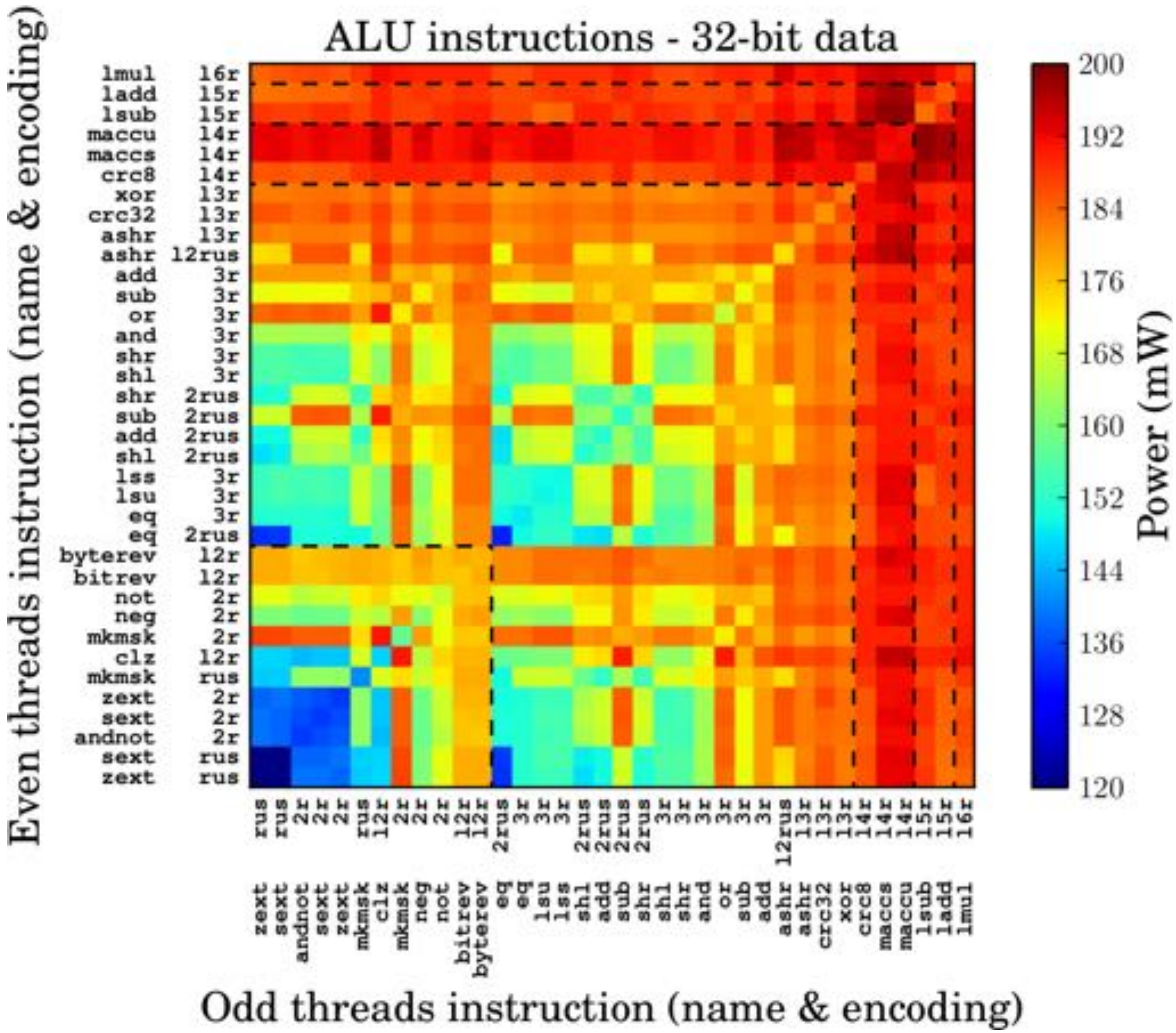




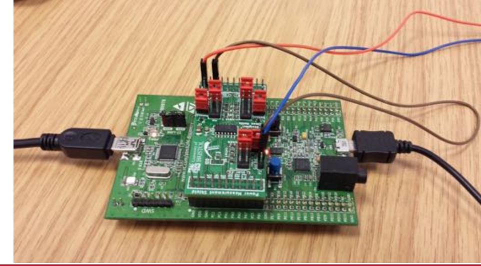
S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages. DOI=10.1145/2700104 http://doi.acm.org/10.1145/2700104

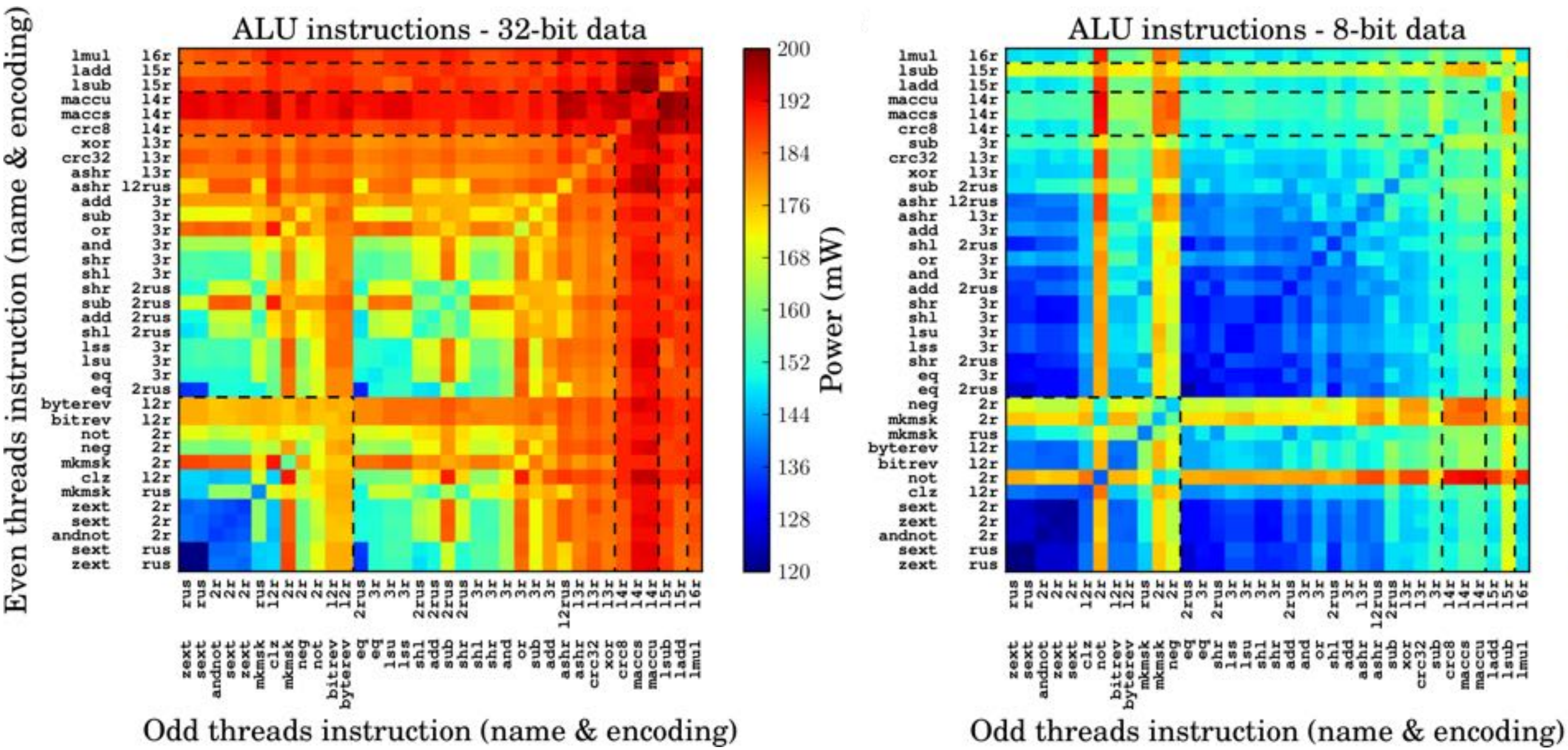
ISA Characterization





ISA Characterization





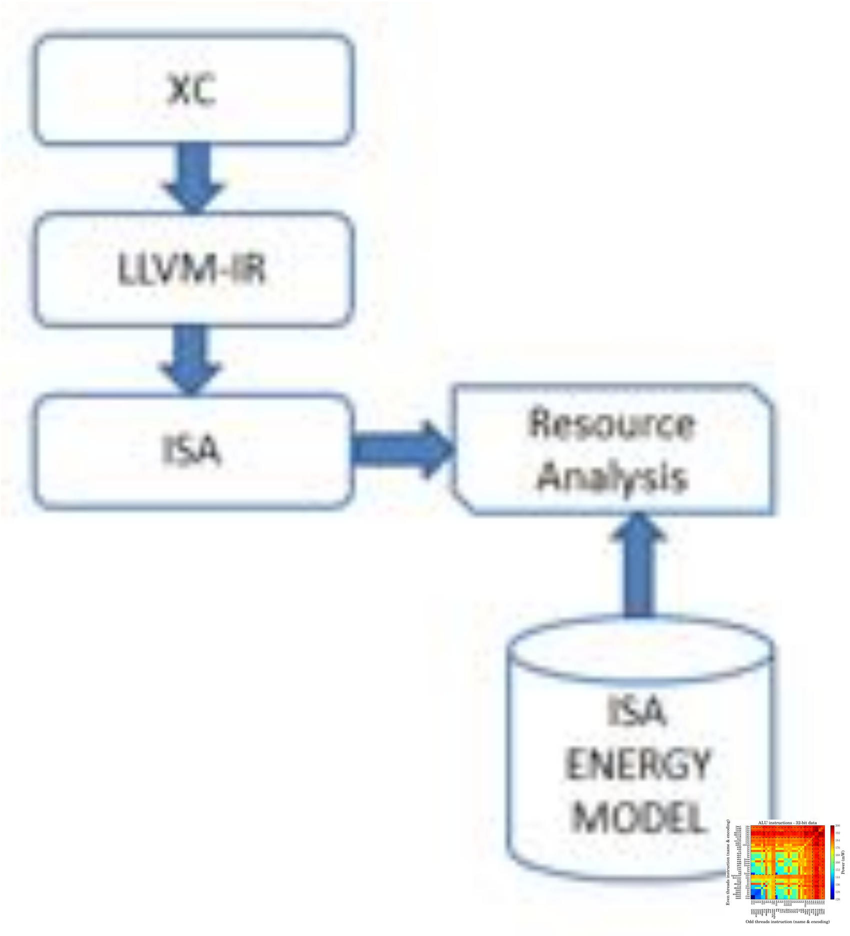
S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages. DOI=10.1145/2700104 http://doi.acm.org/10.1145/2700104

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

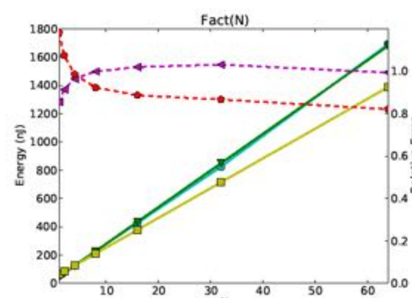
Original Program:

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

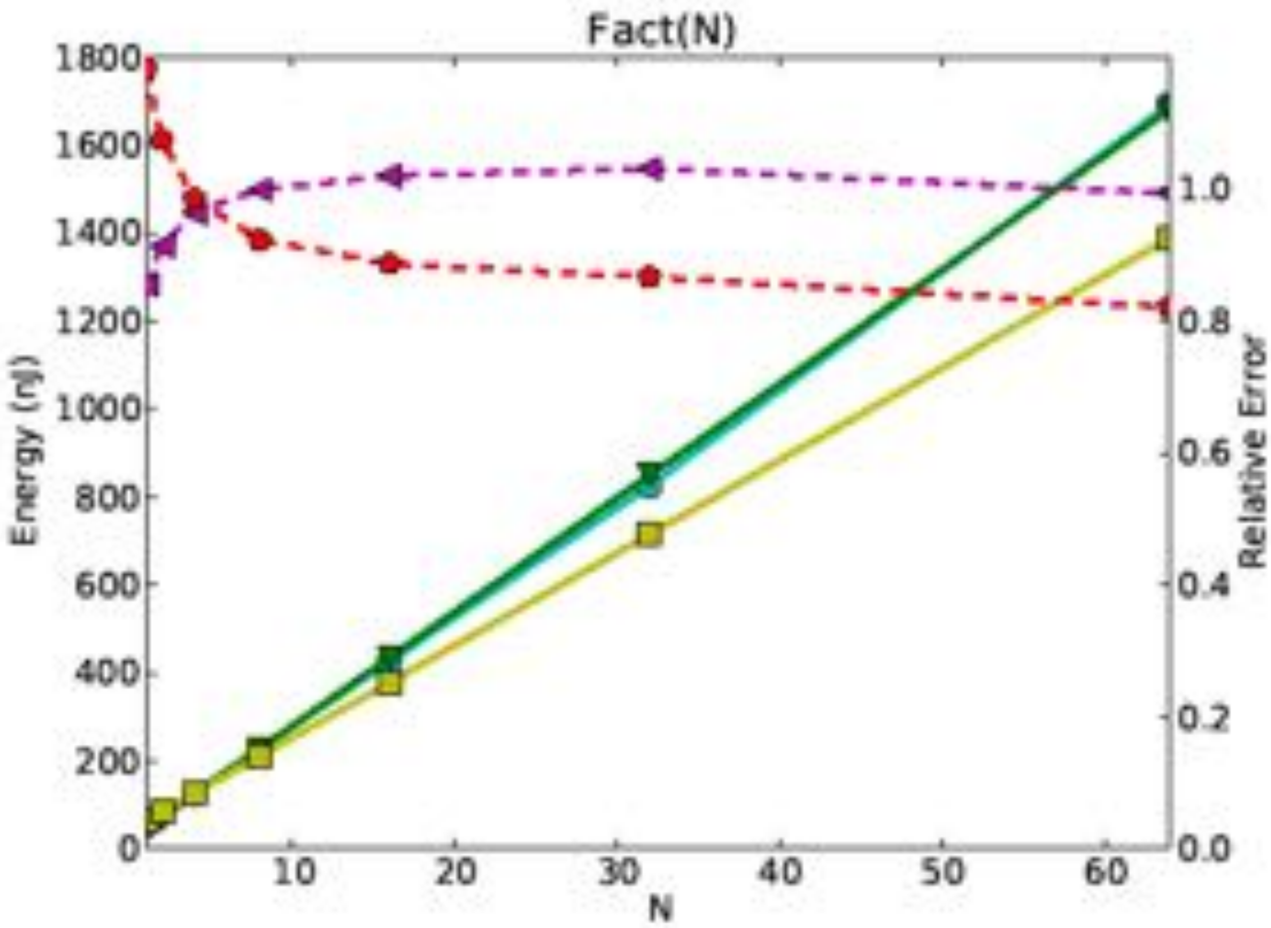
Extracted Cost Relations:

$$C_{\mathbf{fact}}(x) = C_a + C_b \quad \text{if } x<=0$$
$$C_{\mathbf{fact}}(x) = C_a + C_c(x) \quad \text{if } x>0$$
$$C_c(x) = C_d + C_{\mathbf{fact}}(x-1)$$

- 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: $C_{\mathbf{fact}}(x) = (26x + 19.4)$ nJ

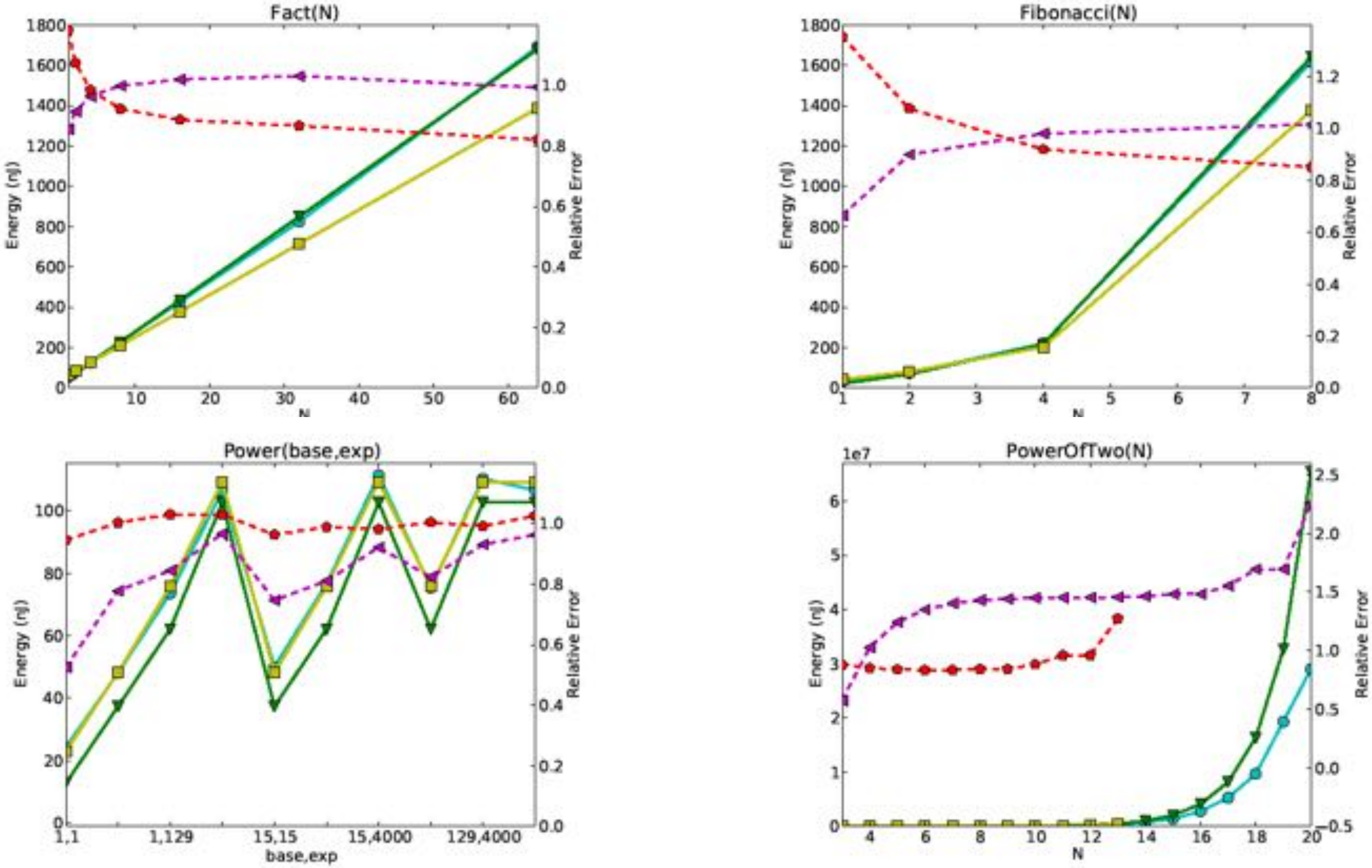


ISA-Level Analysis Results



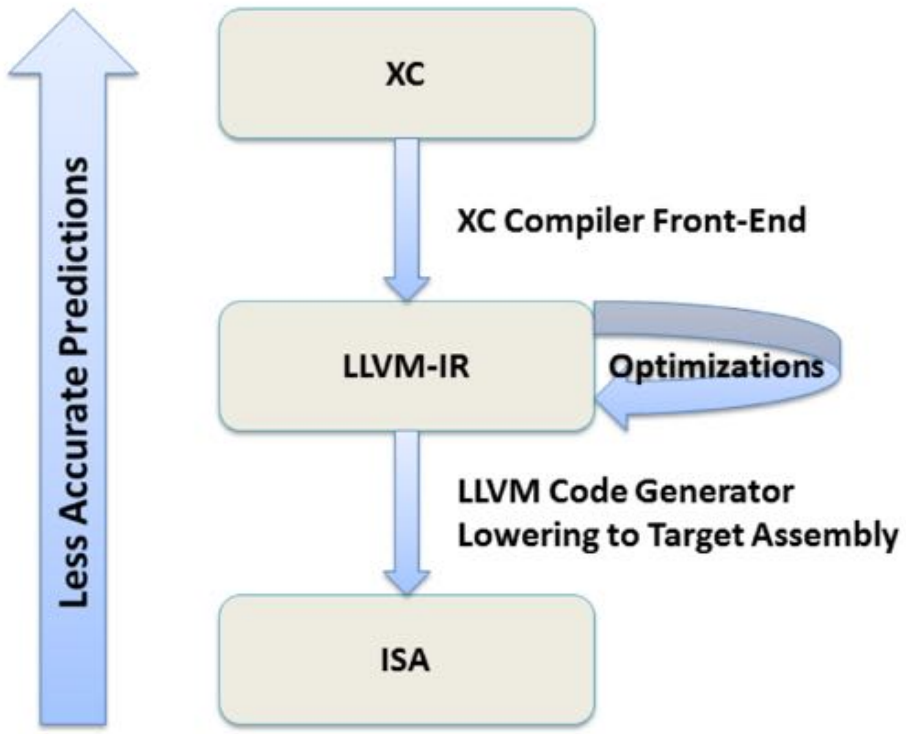
U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, N. Grech, P. Lopez-Garcia, M.V. Hermenegildo and K. Eder. “Energy Consumption Analysis of Programs based on XMOS ISA-Level Models”. LOPSTR 2013. LNCS 8901. Springer. DOI: 10.1007/978-3-319-14125-1_5

ISA-Level Analysis Results



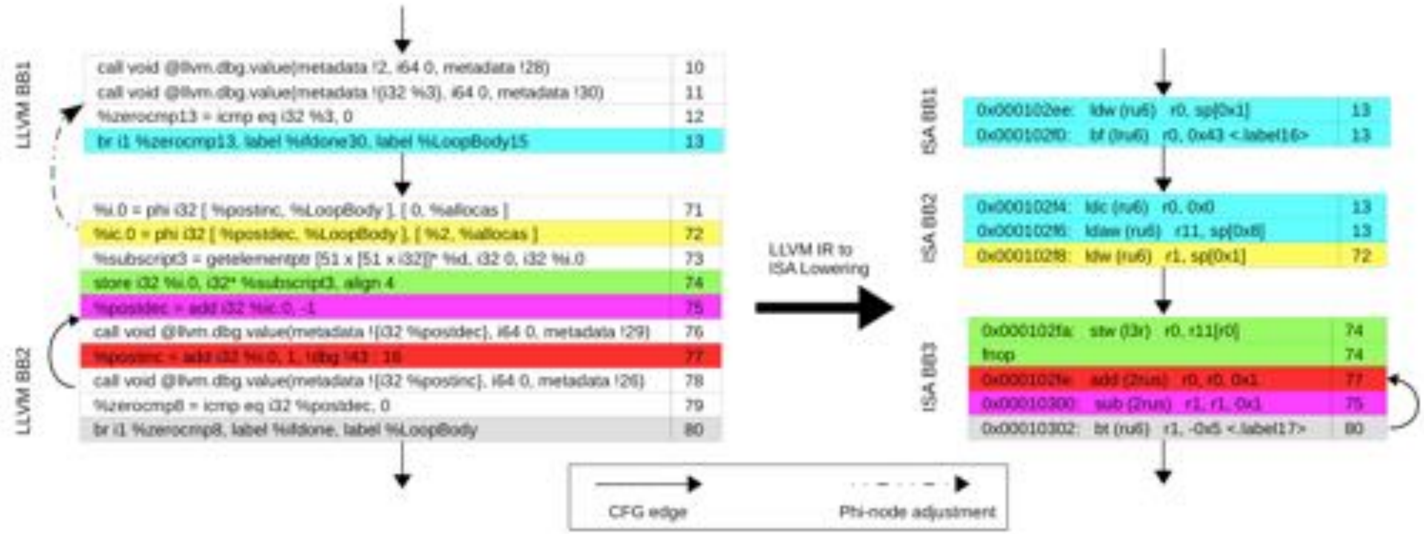
U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, N. Grech, P. Lopez-Garcia, M.V. Hermenegildo and K. Eder. "Energy Consumption Analysis of Programs based on XMOS ISA-Level Models". LOPSTR 2013.

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

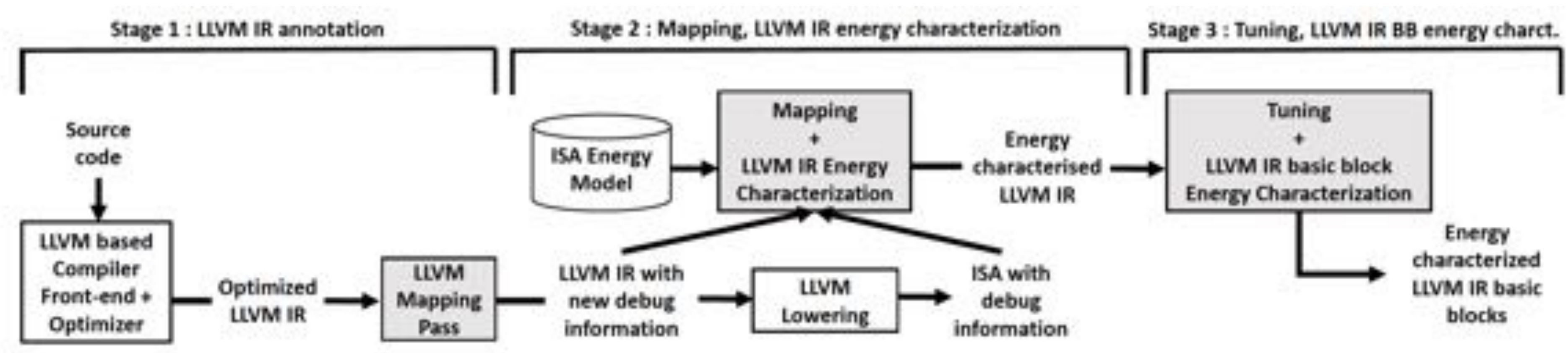


$$E(ir_i) = \sum_{isa_j \in R(ir_i)} E(isa_j)$$

K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

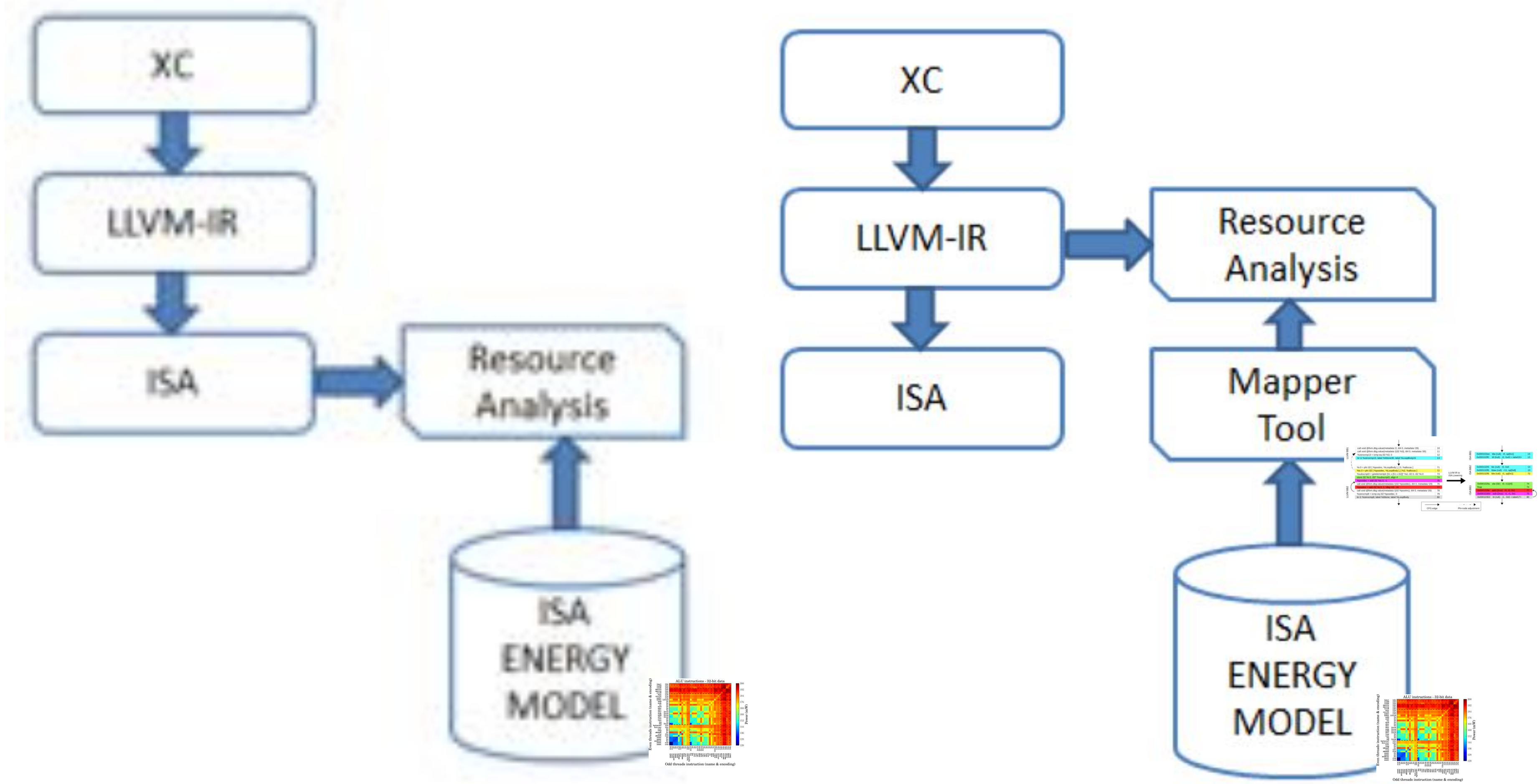
U. Liqat, K. Georgiou, S. Kerrison, P. Lopez-Garcia, J.P. Gallagher, M.V. Hermenegildo, K. Eder. “Inferring Parametric Energy Consumption Functions at Different Software Levels: ISA vs. LLVM IR”. In Proceedings of FOPARA 2015. LNCS 9964. Springer. DOI: 10.1007/978-3-319-46559-3_5 http://arxiv.org/abs/1511.01413

LLVM IR Energy Characterization



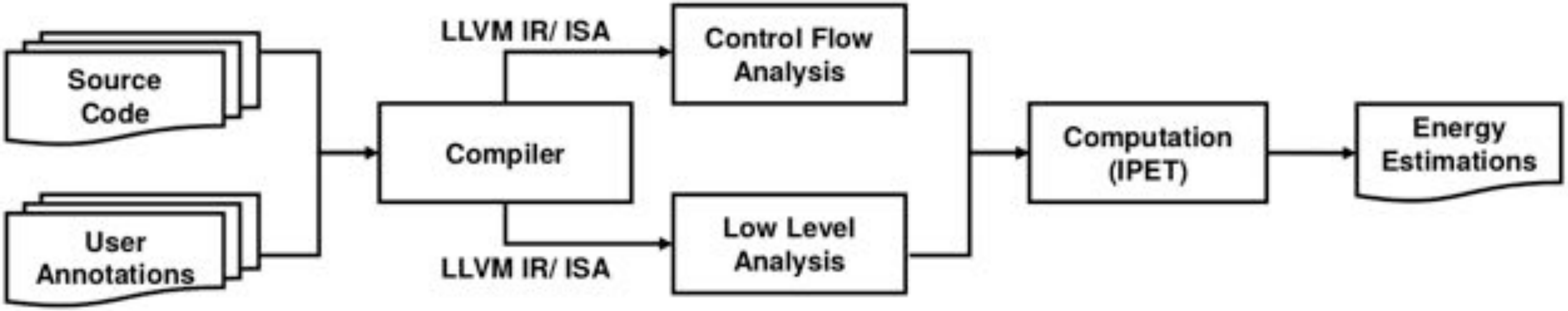
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

Analysis at the LLVM IR Level



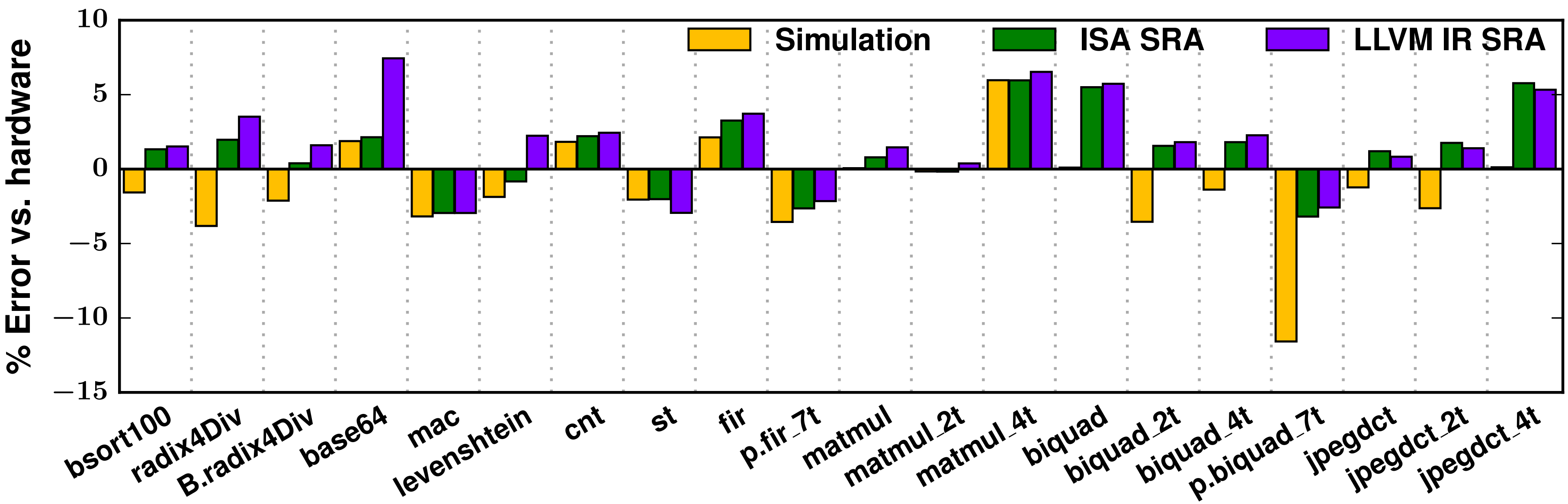
N. Grech, K. Georgiou, J. Pallister, S. Kerrison, J. Morse, K. Eder. 2015. "Static analysis of energy consumption for LLVM IR programs". In Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems (SCOPES '15). ACM, New York, NY, USA, pages 12-21. http://dx.doi.org/10.1145/2764967.2764974

SRA for Energy Consumption



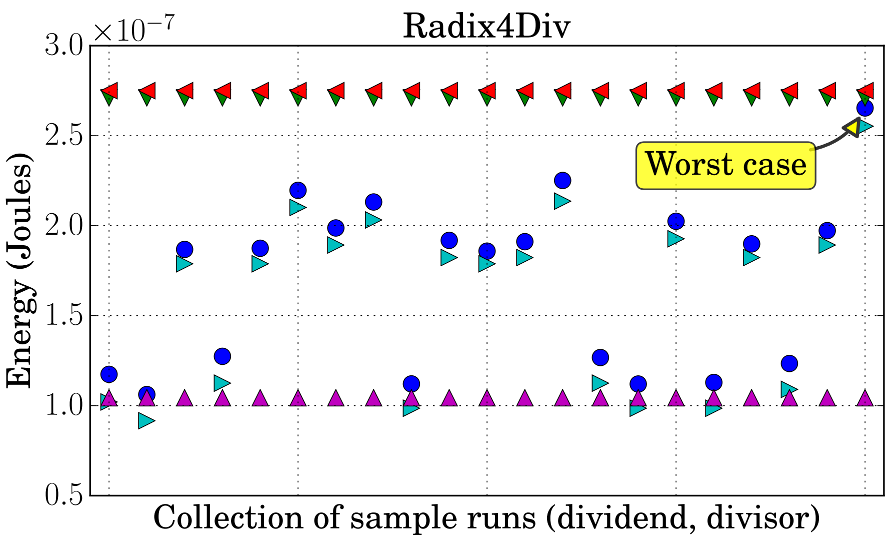
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

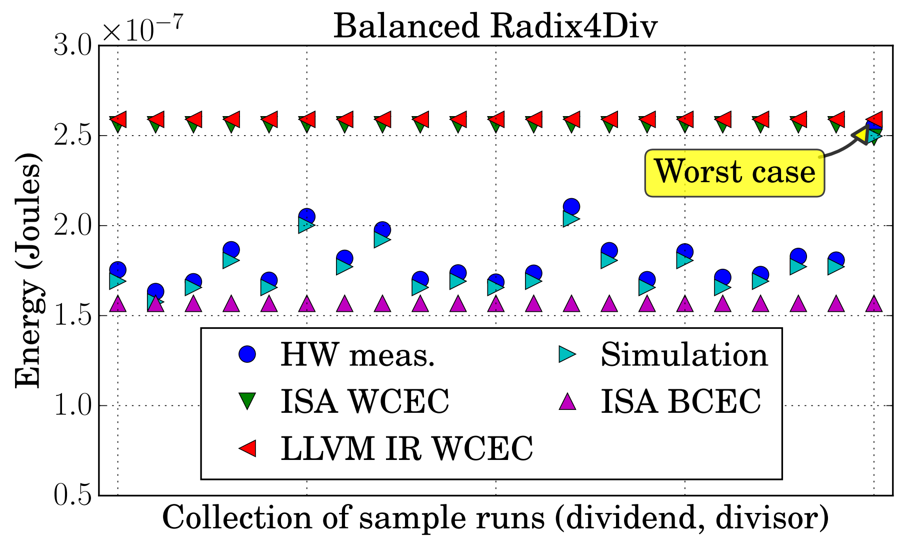
SRA for Energy Consumption



K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

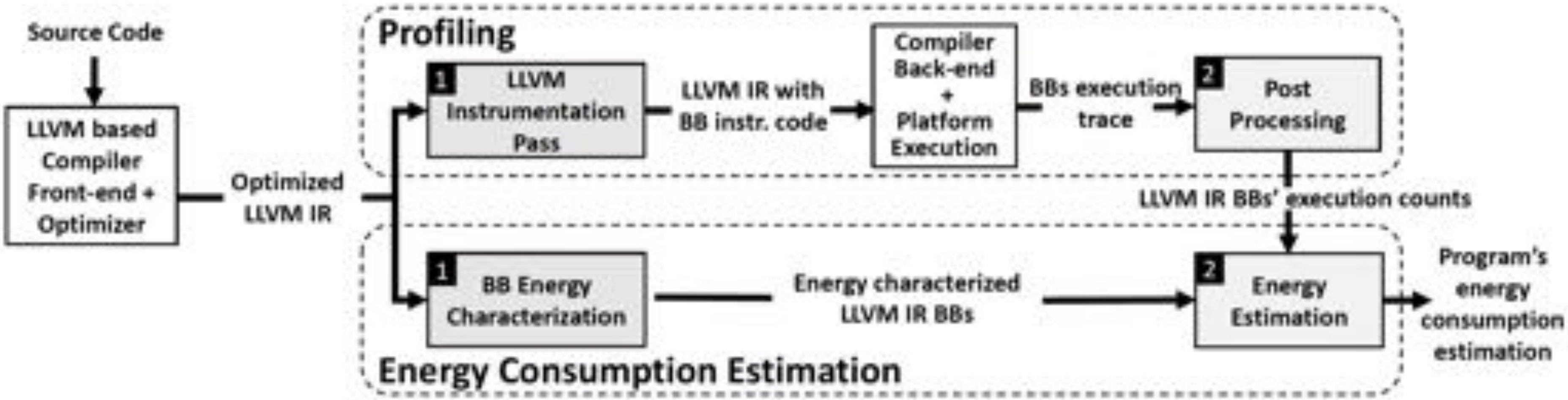
EC Static Analysis Results





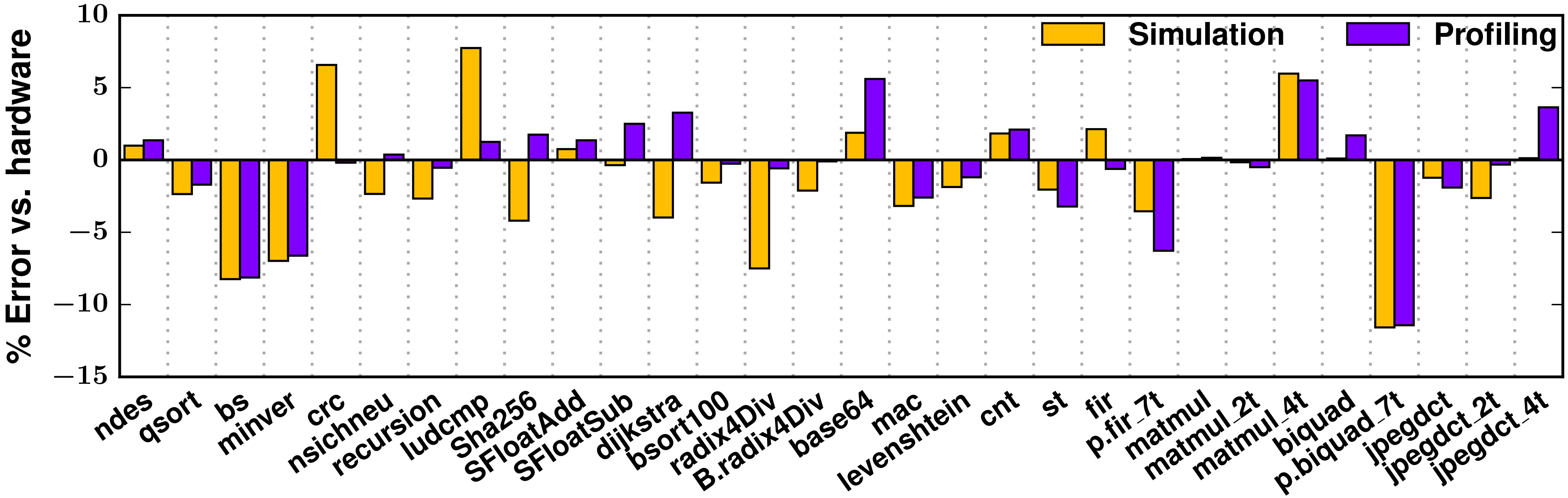
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

Profiling-based Energy Estimation



K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

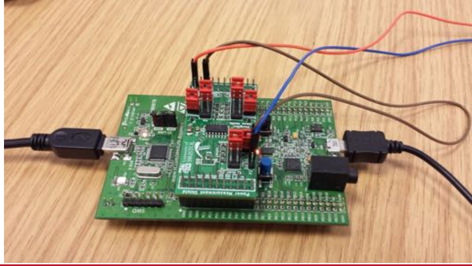
Energy Consumption Profiling

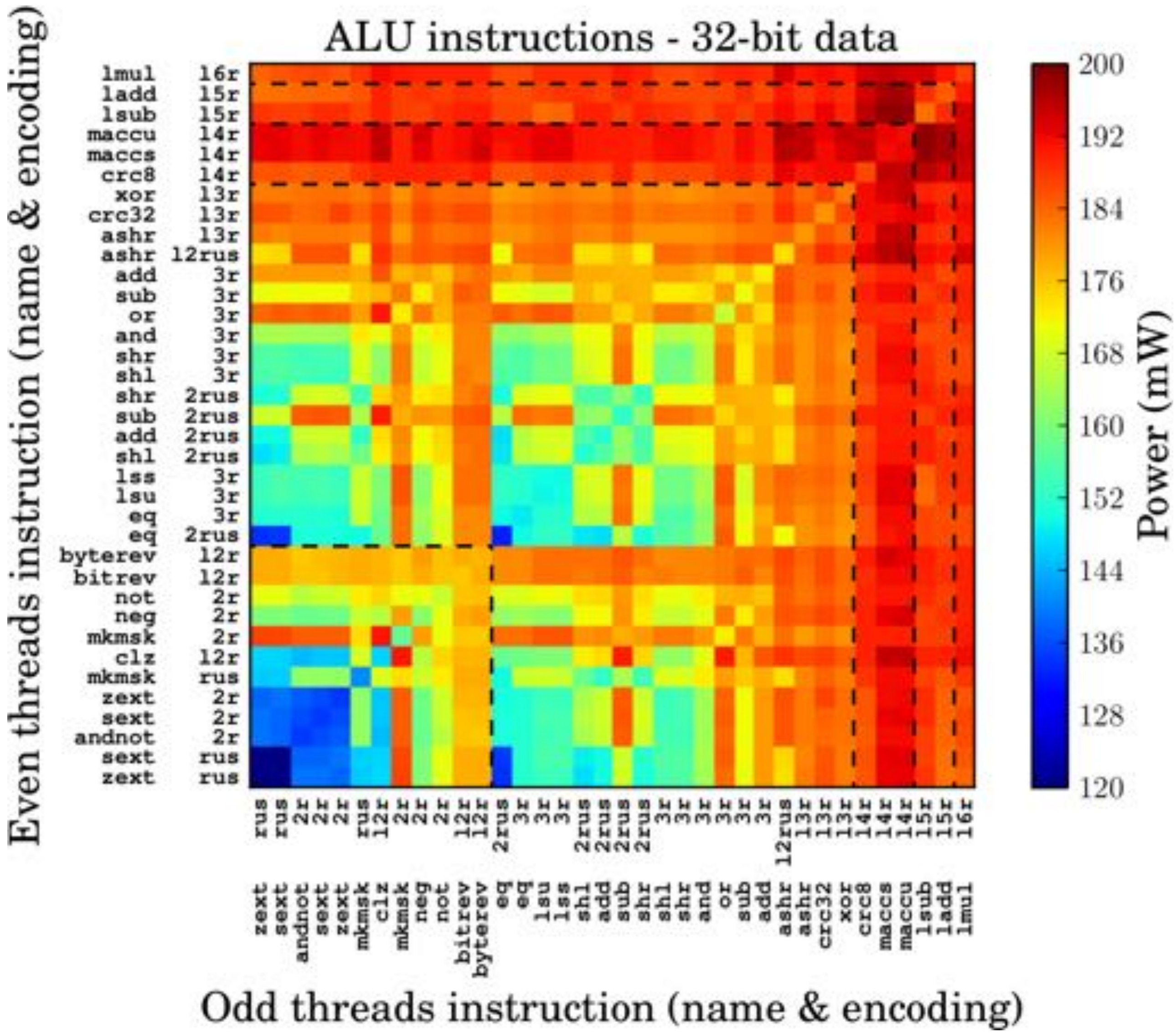


K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. "Energy Transparency for Deeply Embedded Programs". ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

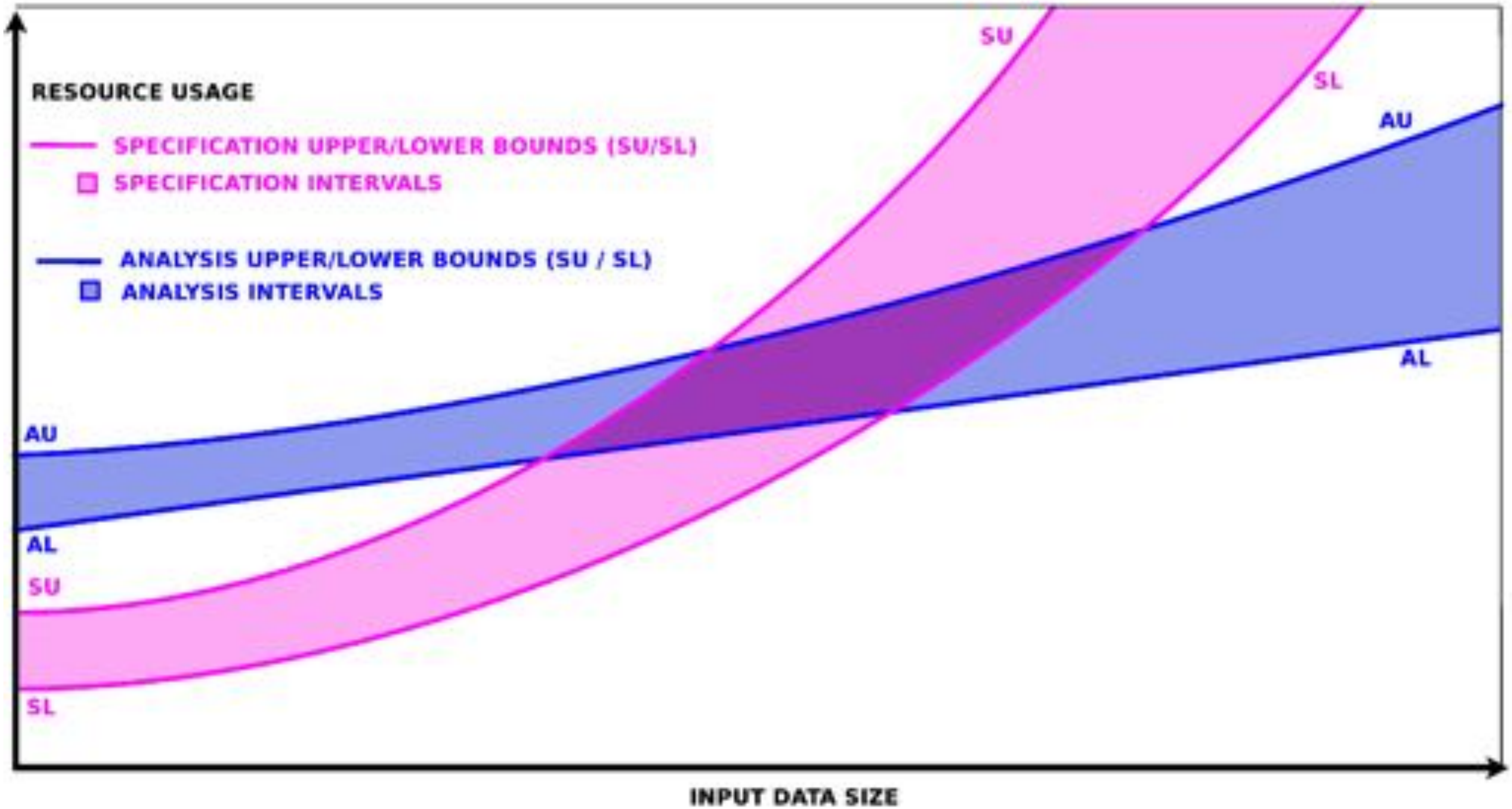
The Worst Case …

ISA Characterization





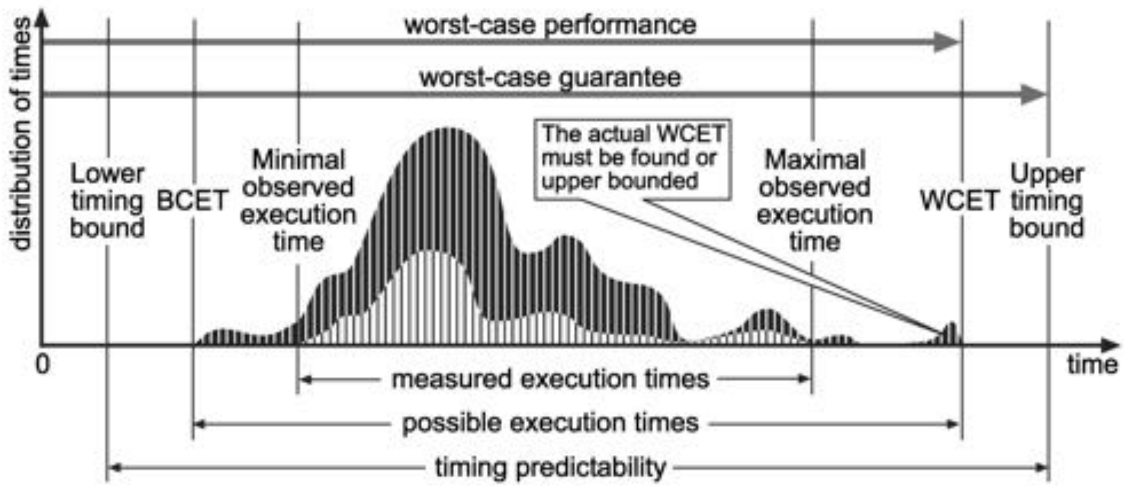
Static Resource ***Bound*** Analysis



Source: Pedro Lopez Garcia, IMDEA Software Research Institute

Worst Case Execution Time

- Worst Case Execution Time (WCET) Analysis:
  - WCET model
  - WCET bounds (often for safety critical applications)
    - 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)

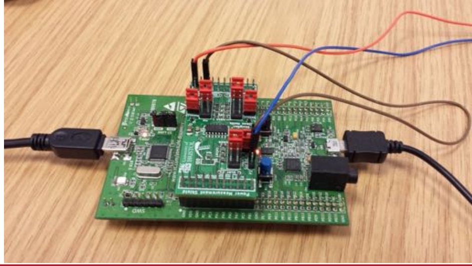
Does this work for energy consumption analysis?

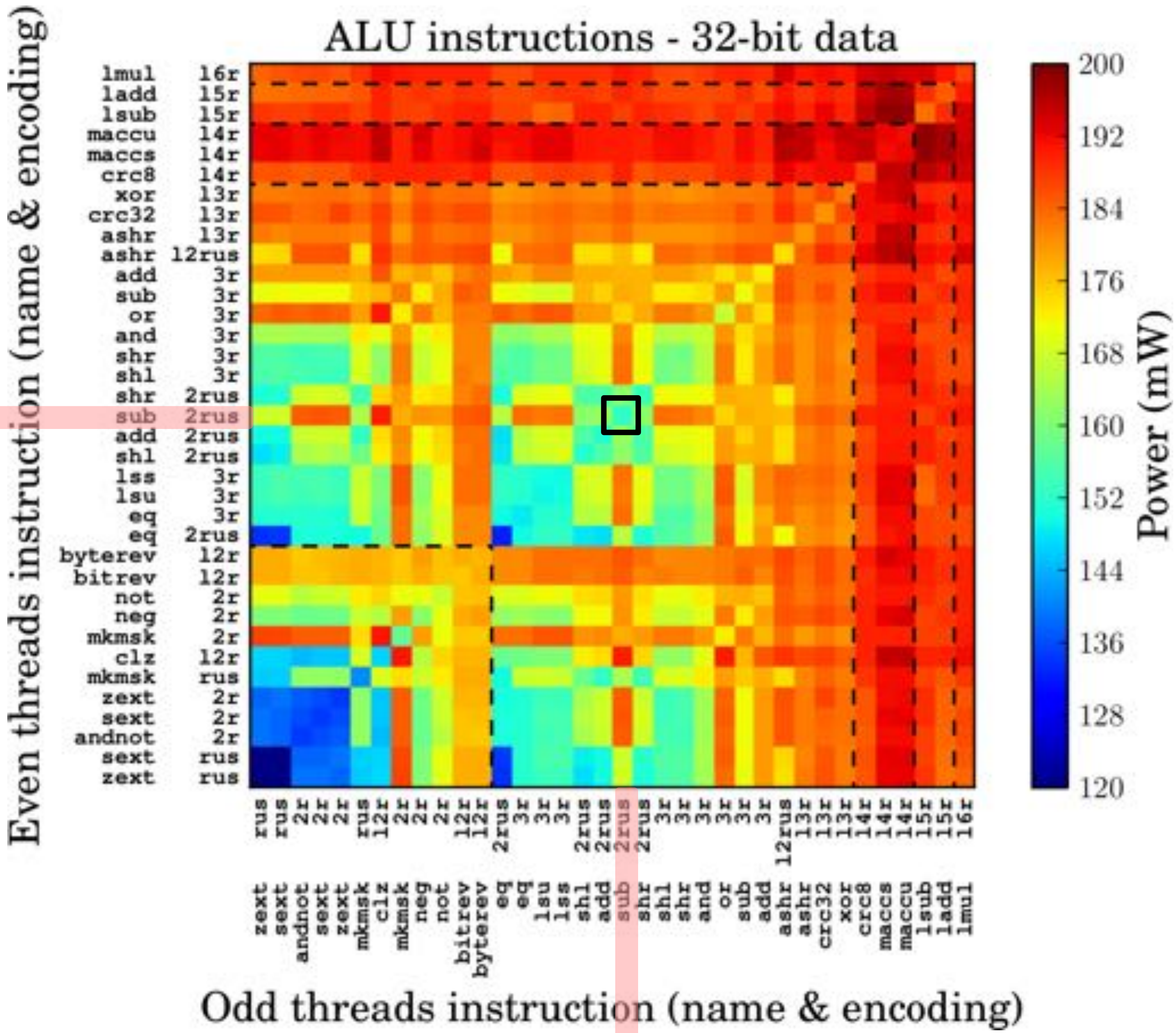
Worst Case Energy Consumption

- WCEC analysis goes well beyond WCET analysis.
  - 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
  - timing variability has mostly been eliminated “by design" through the use of synchronous logic
- **But, energy consumption is**

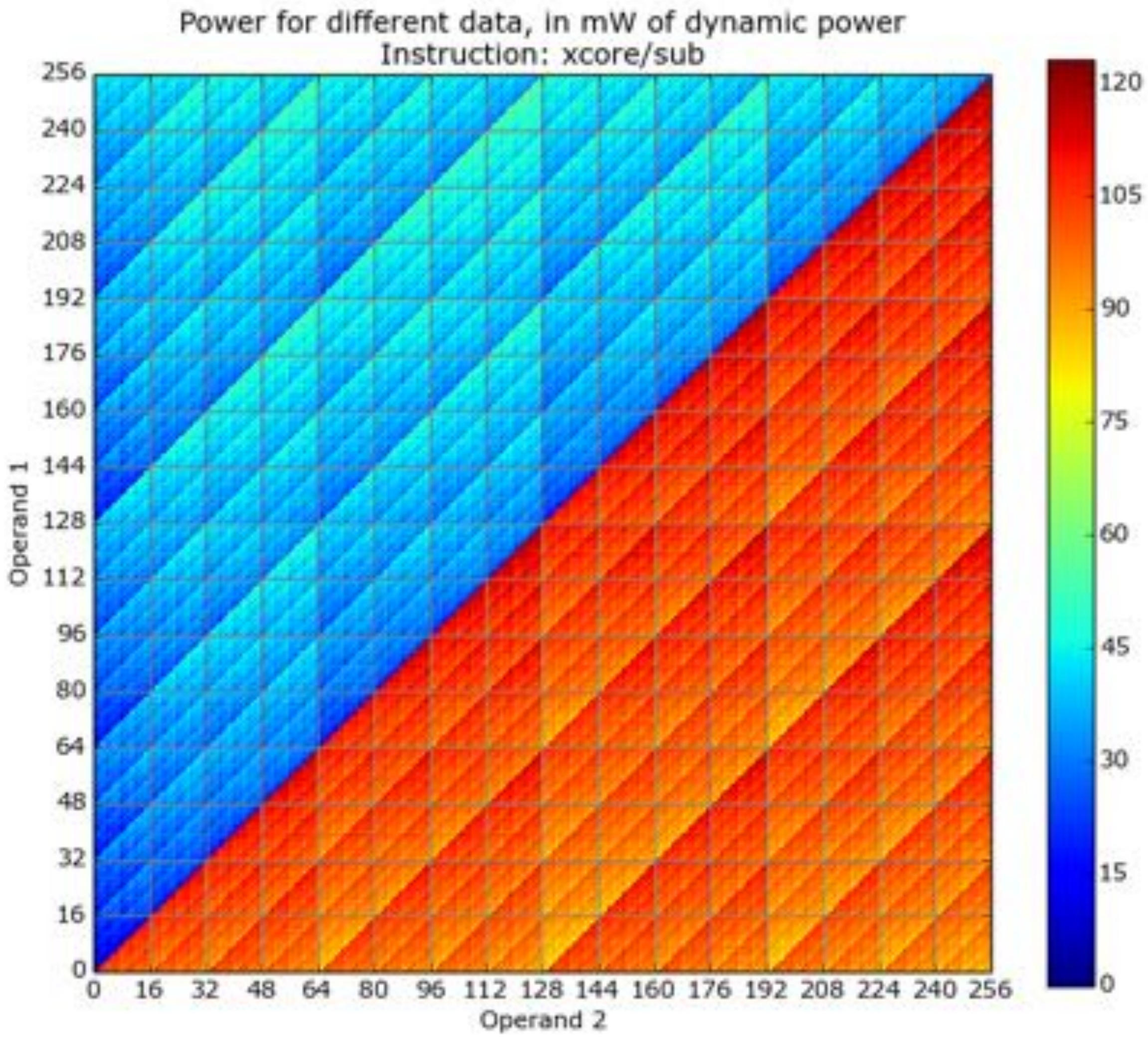
  ***data dependent*.**

ISA Characterization

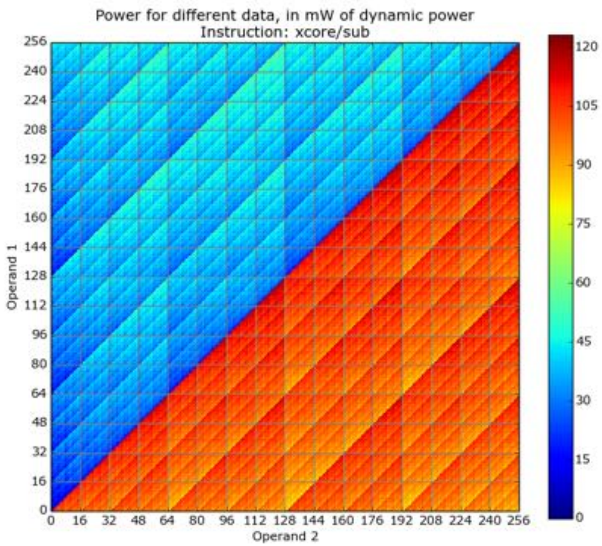


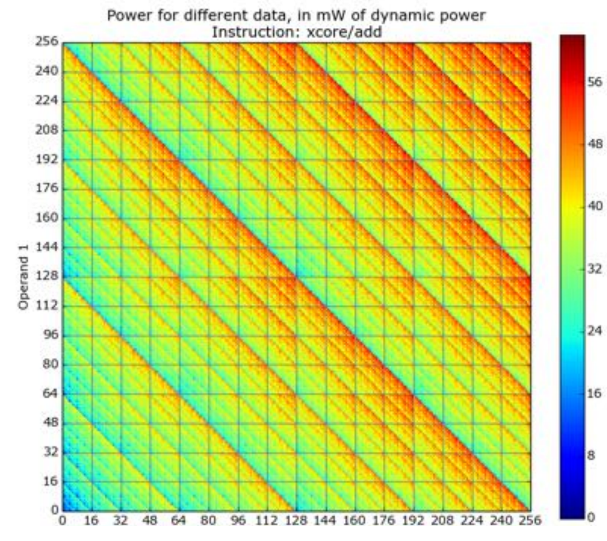


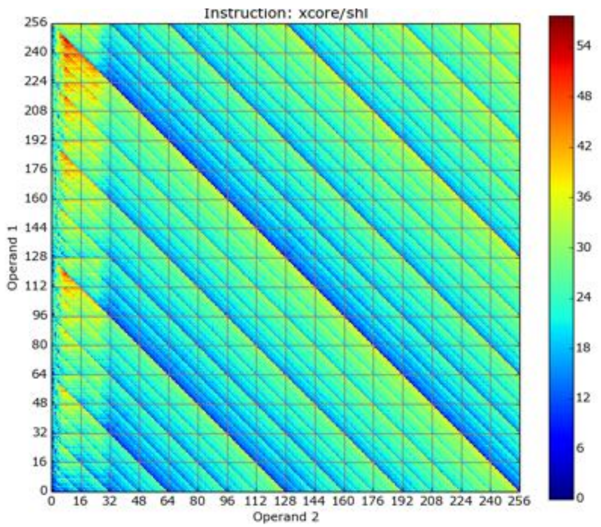
W/A/B-Case Energy Consumption

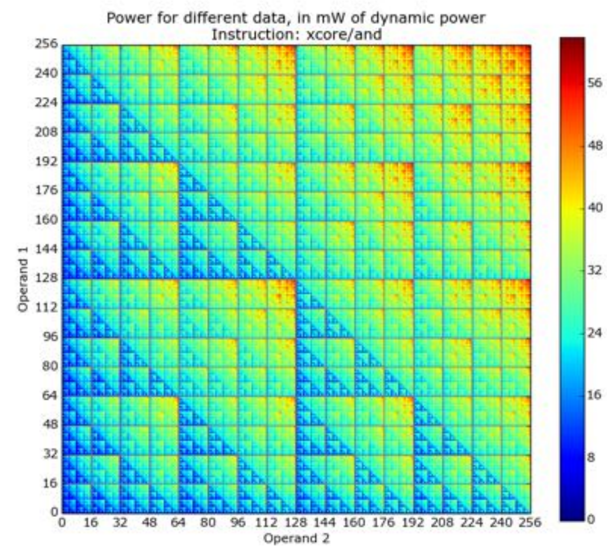


W/A/B-Case Energy Consumption

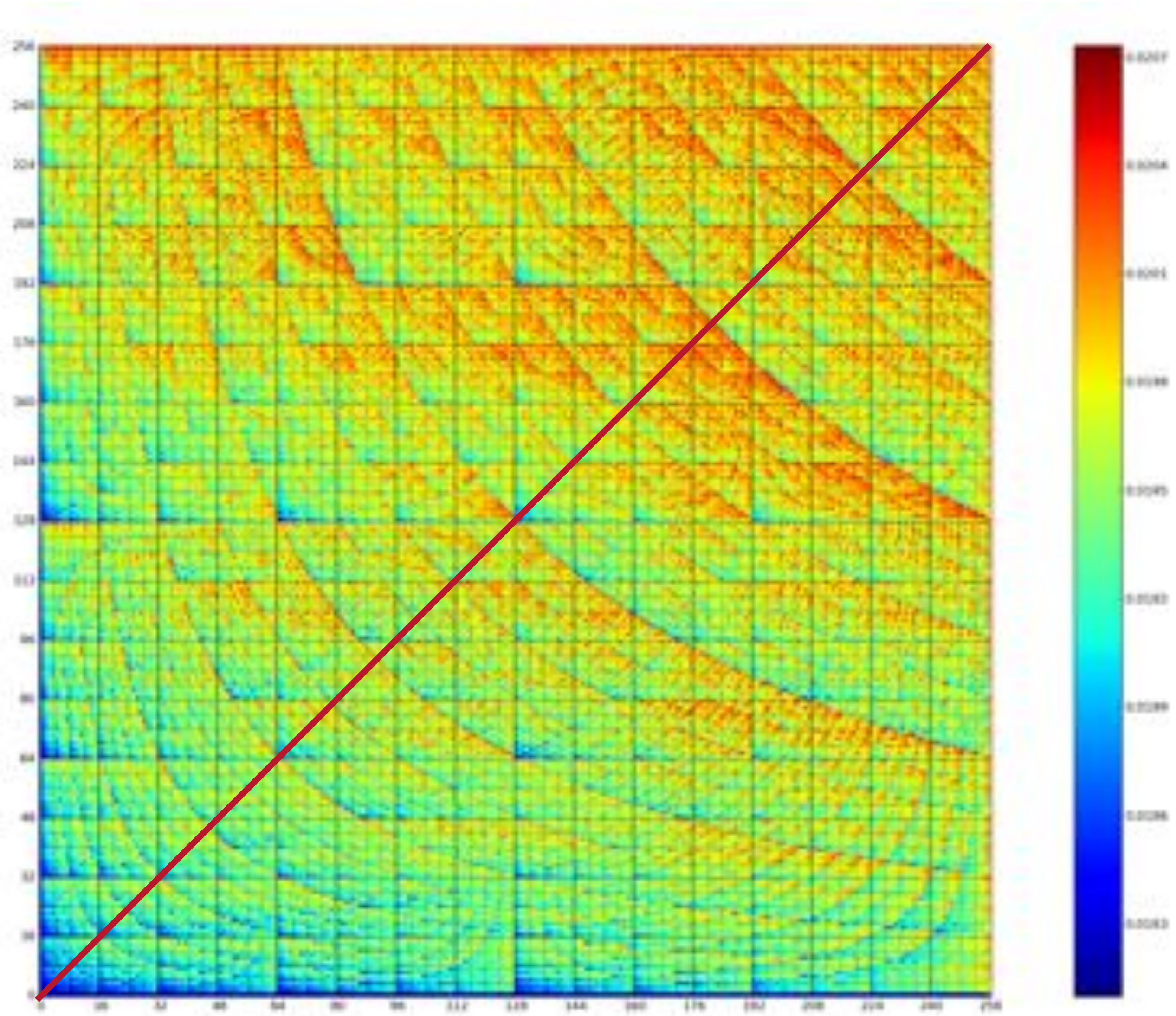




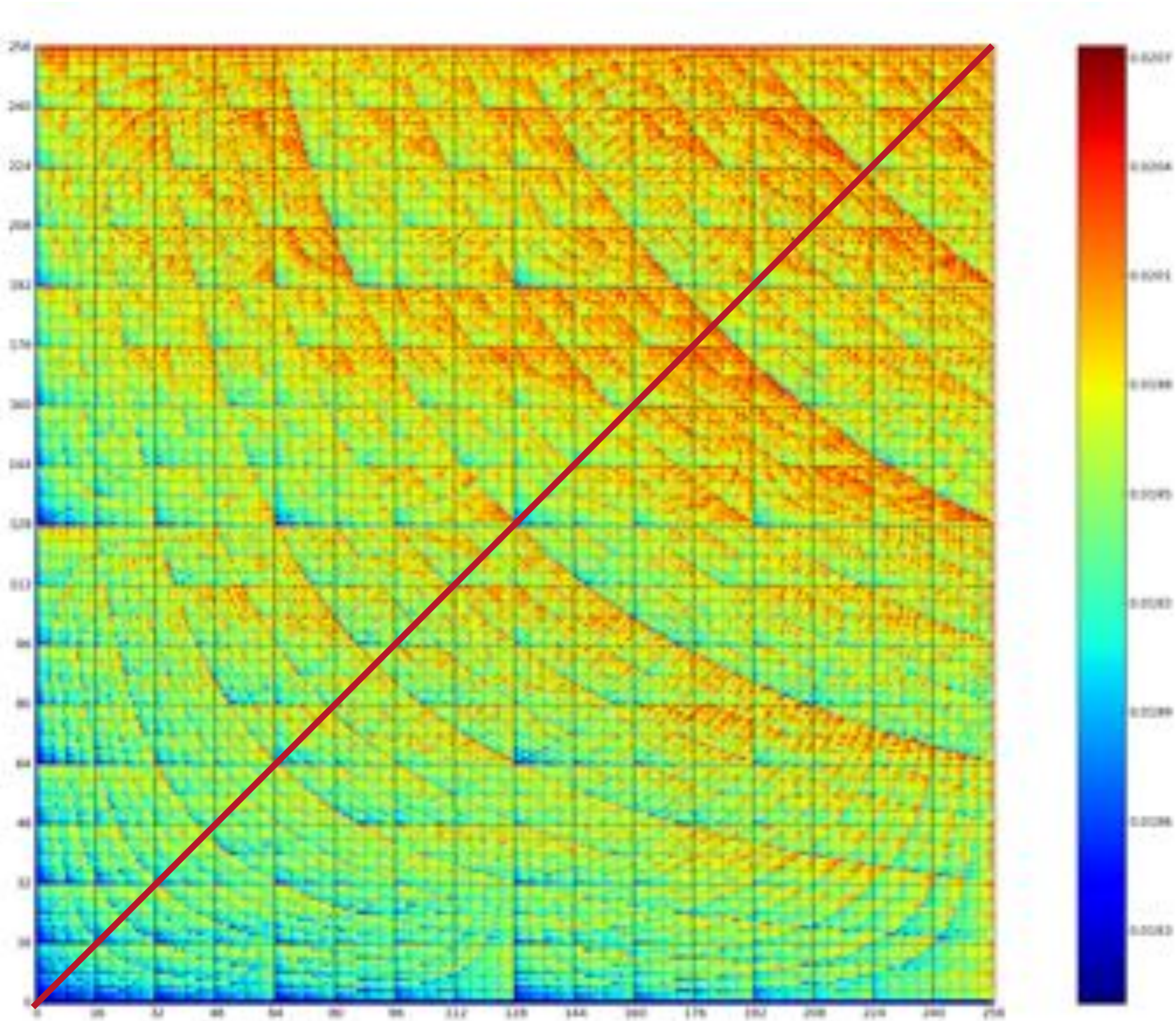




a*b = b*a

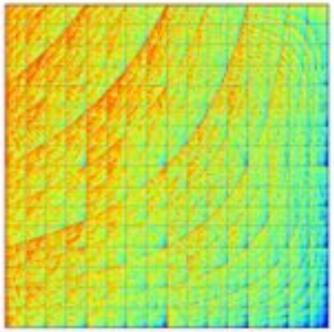


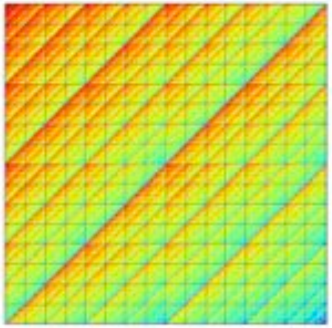
Energy(a*b) ≠ Energy(b*a)

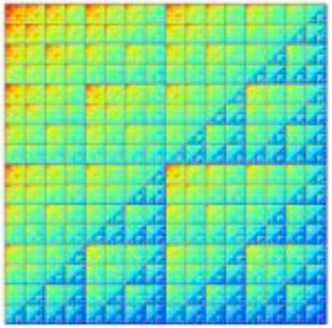


Dynamic Energy is significant

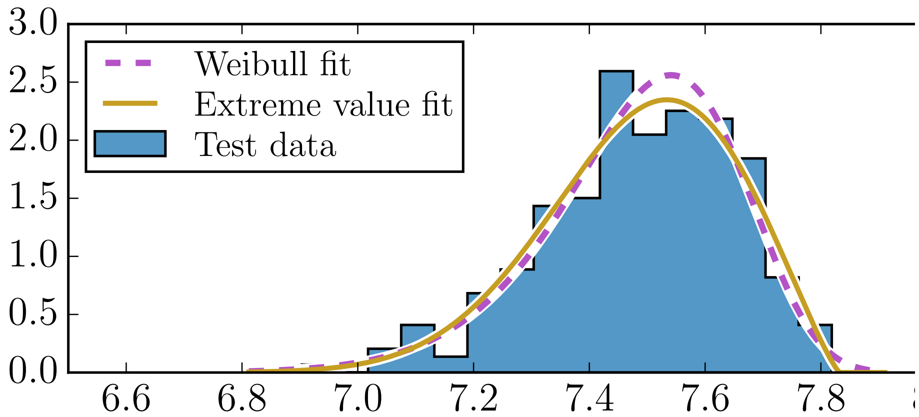
- Data dependent switching costs can be large
- Some instructions can cause as much dynamic energy as static (sub)
- How can we account for context-dependent switching costs?
- **Can WCEC be safe and tight?**







Statistical Energy Modelling



Energy distribution (nJ) for AVR shl

- Many instructions exhibit statistical properties
- Different instruction distributions can be composed
- Can statistically impossible energy be considered a safe upper bound?

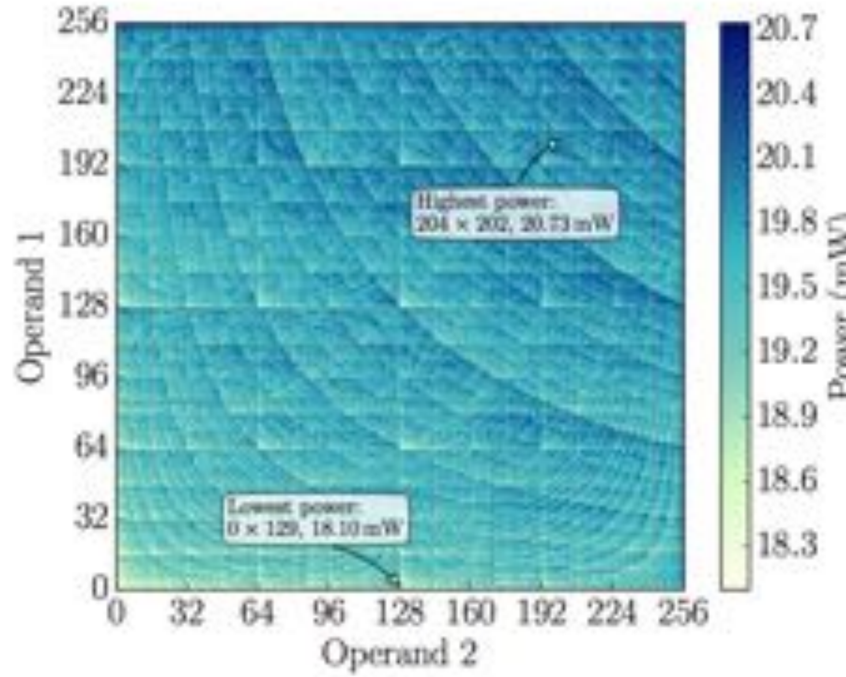
Data Dependent Energy Modeling for Worst Case Energy Consumption Analysis

James Pallister, Steve Kerrison, Jeremy Morse, Kerstin Eder
Department of Computer Science, University of Bristol, BS8 1UB, UK
firstname.lastname@bristol.ac.uk

ABSTRACT

Safely meeting Worst Case Energy Consumption (WCEC) criteria requires accurate energy modeling of software. We investigate the impact of instruction operand values upon energy consumption in cacheless embedded processors. Existing instruction-level energy models typically use measurements from random input data, providing estimates unsuitable for safe WCEC analysis.

We examine probabilistic energy distributions of instructions and propose a model for composing instruction sequences using distributions, enabling WCEC analysis on program basic blocks. The worst case is predicted with statistical analysis. Further, we verify that the energy of embedded benchmarks can be characterised as a distribution, and compare our proposed technique with other methods of estimating energy consumption.

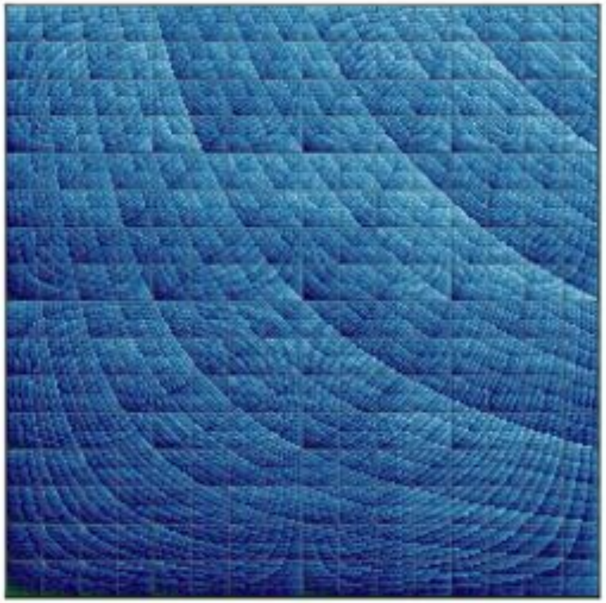


Accepted for publication at 20th International Workshop on Software and Compilers for Embedded Systems (SCOPES 2017). Preprint available at: https://arxiv.org/abs/1505.03374

Data Dependent Energy Modelling

Critical questions for WCEC modelling:

- *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?*



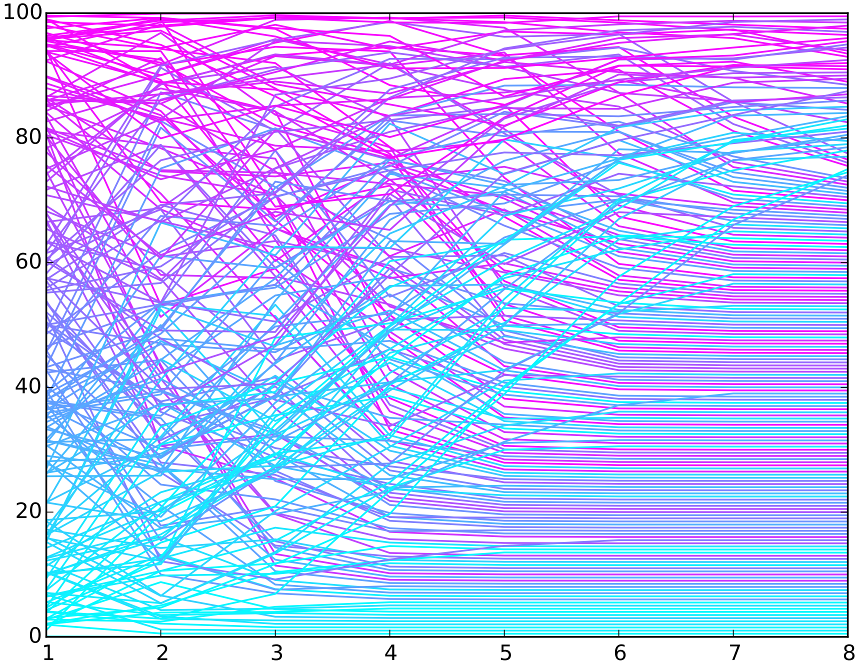
Energy of an Instruction Sequence

100 data values provided to a sequence of 8 instructions
ranking of the instruction sequence's energy up to instruction x

Energy of an Instruction Sequence

100 data values provided to a sequence of 8 instructions
ranking of the instruction sequence's energy up to instruction x

by input

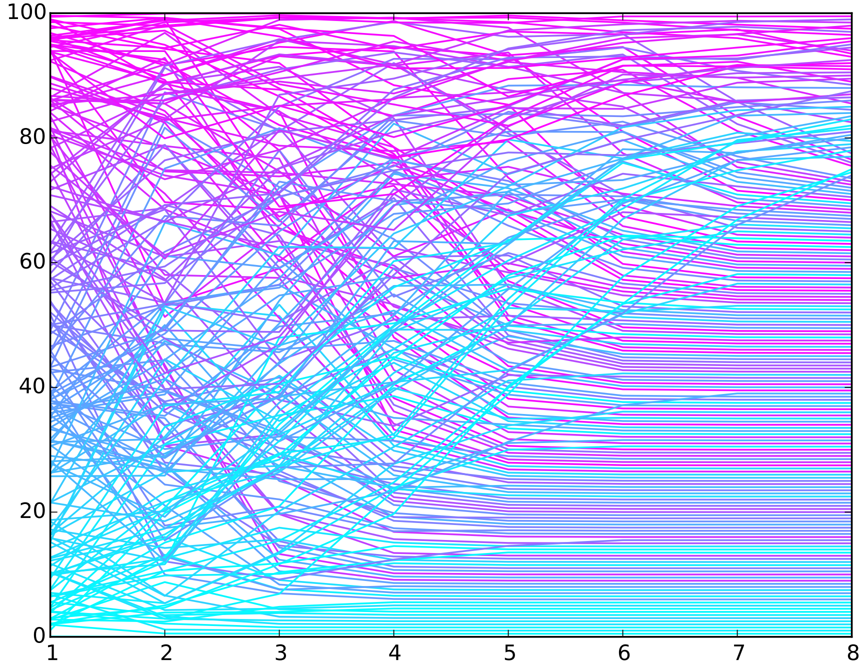


experiments conducted by James Pallister

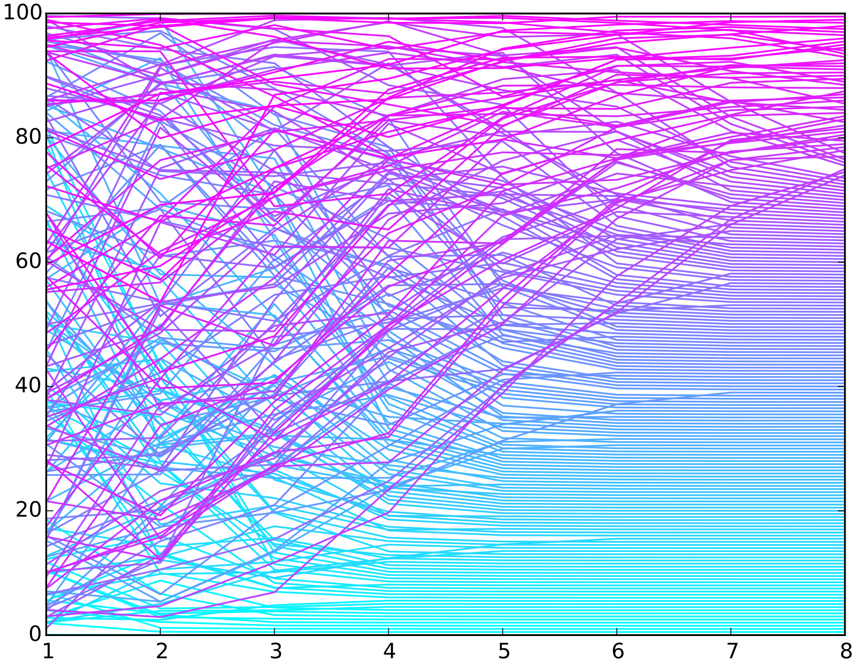
Energy of an Instruction Sequence

100 data values provided to a sequence of 8 instructions
ranking of the instruction sequence's energy up to instruction x

by input



and by output



experiments conducted by James Pallister

Complexity Analysis

- Determining switching costs is NP-hard
  - Amount of computation required increases exponentially with program size
  - Problem cannot be approximated accurately
- No algorithm can efficiently find dynamic energy, so other questions must be posed
  - *Is a less general solution acceptable?*
  - *What level of inaccuracy can be tolerated?*

J. Morse, S. Kerrison and K. Eder. 2016. "On the infeasibility of analysing worst case dynamic energy". (under review) http://arxiv.org/abs/1603.02580

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.

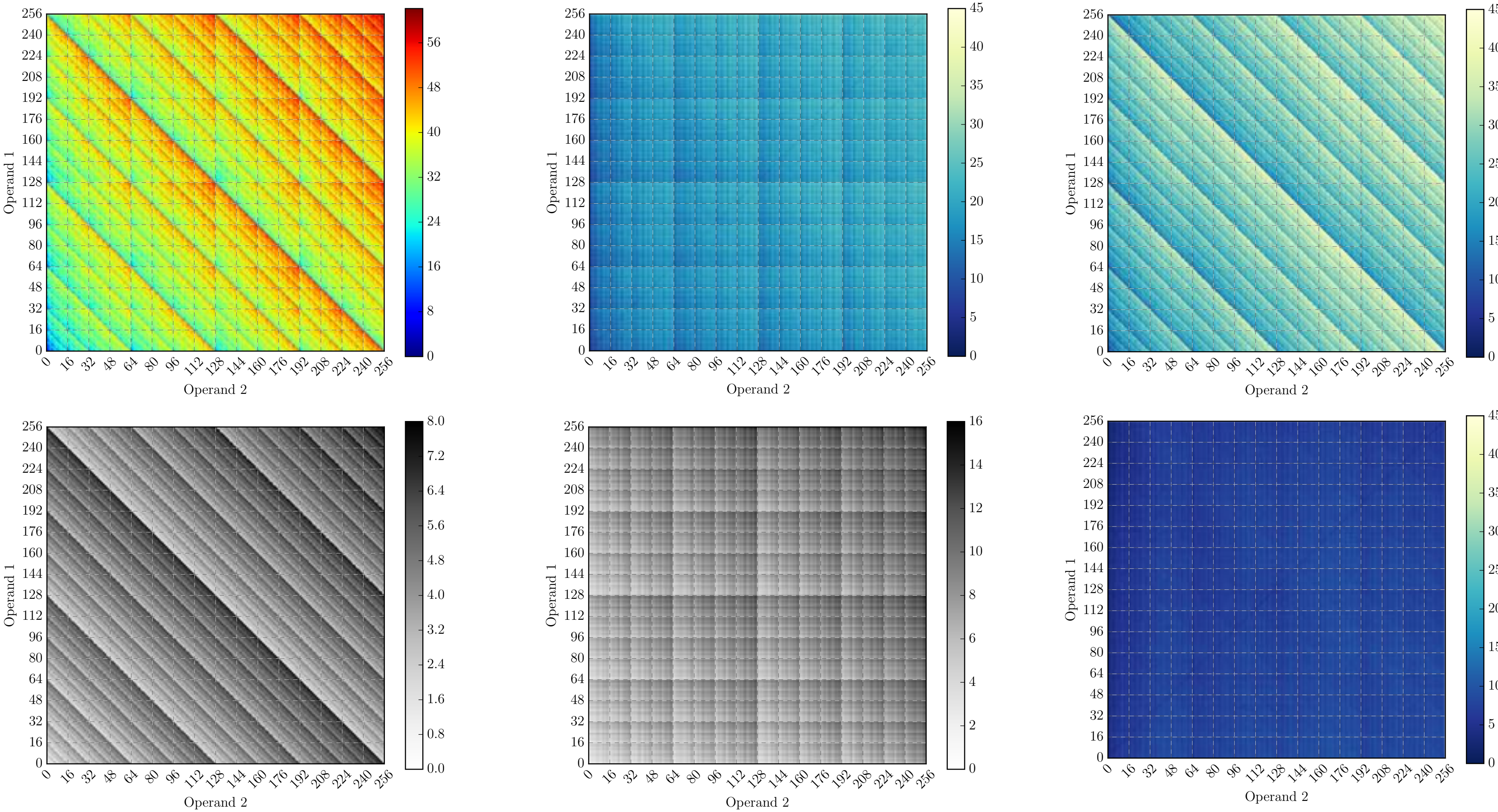
1 Introduction

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

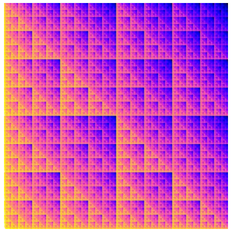
Impact of Datapath Switching



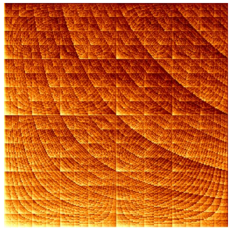
J. Morse, S. Kerrison and K. Eder. 2016. "On the infeasibility of analysing worst case dynamic energy". (under review) http://arxiv.org/abs/1603.02580

Summing up

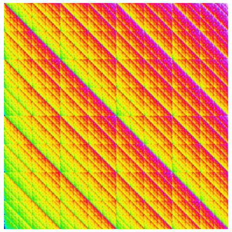
- To achieve *Energy Transparency*
  - Energy modelling is a huge challenge
    - Fundamental research questions
      - data-dependent energy models
      - compositional
      - probabilistic techniques



^



×

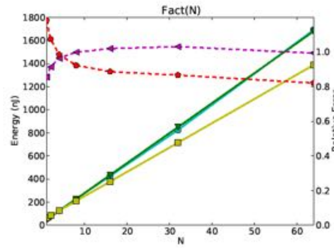


\+

  - Analysis techniques for energy consumption
    - SRA works best for IoT-type systems
    - Hybrid, profiling-based techniques for more complex architectures

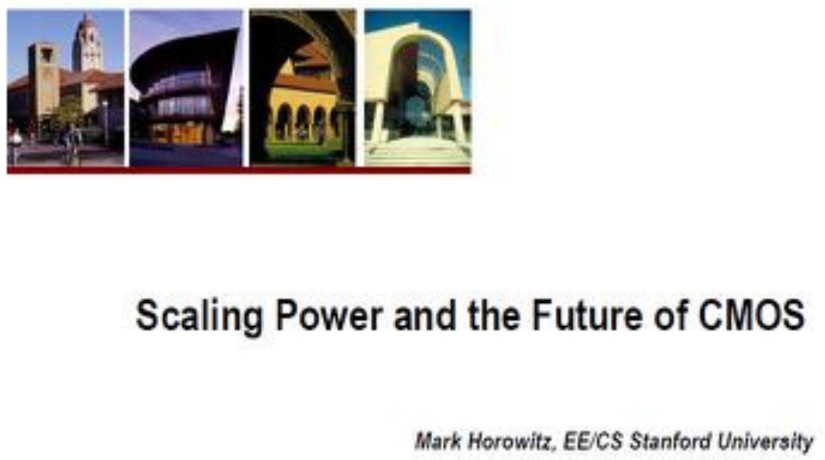
Towards Energy Aware Software Engineering

Energy Transparency

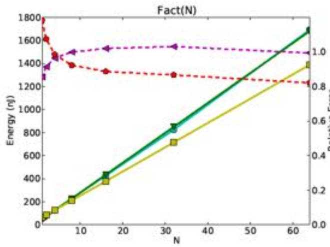


- For HW designers:
  "Power is a 1st and last order design constraint."
  [Dan Hutcheson, VLSI Research, Inc., E^{3}S Keynote 2011]

- "Every design is a point in a 2D plane."
  [Mark Horowitz,E^{3}S 2009]

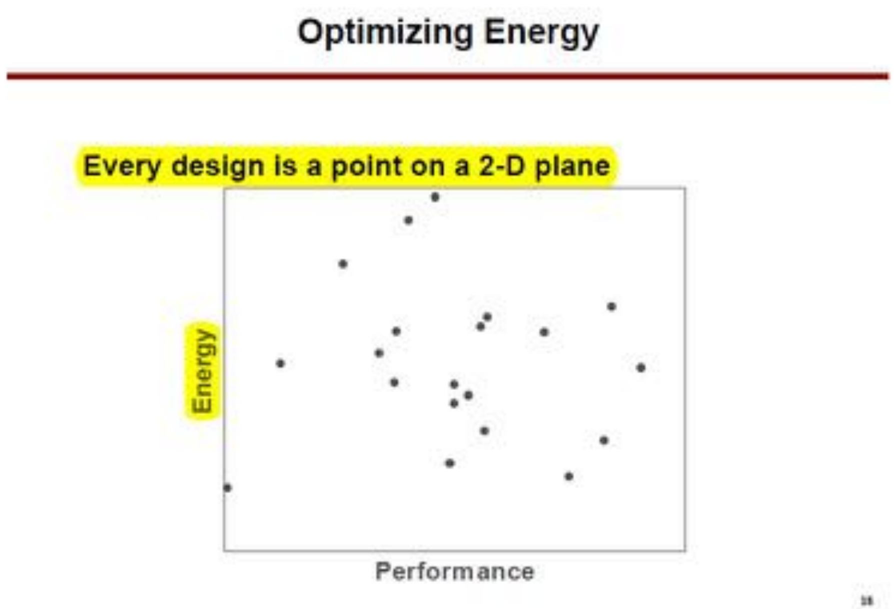


Energy Transparency

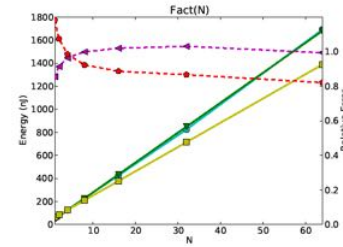


- For HW designers:
  “Power is a 1st and last order design constraint.”
  [Dan Hutcheson, VLSI Research, Inc., E³S Keynote 2011]

- “Every design is a point in a 2D plane.”
  [Mark Horowitz, E³S 2009]

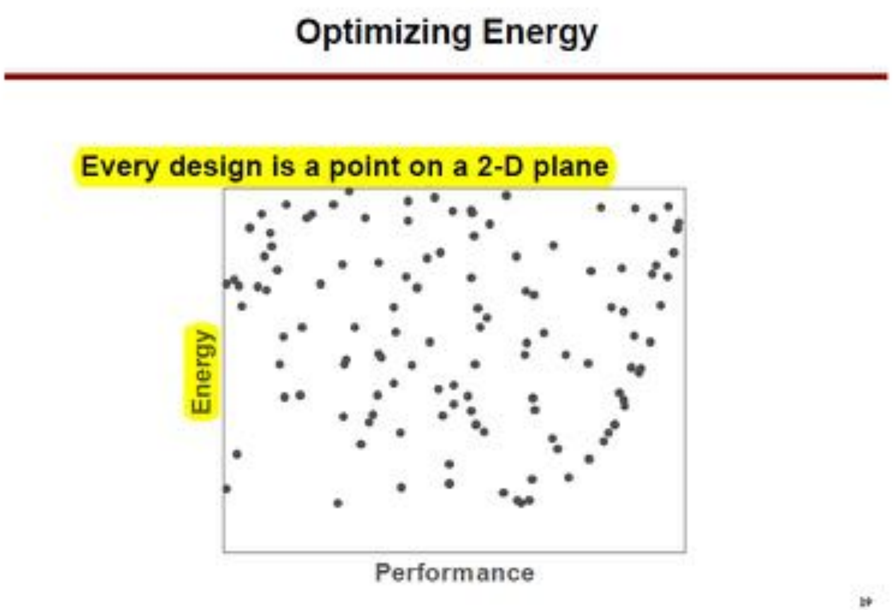


Energy Transparency

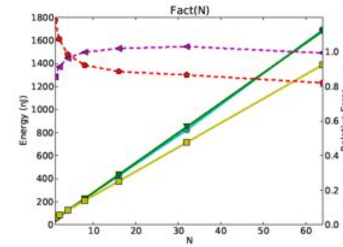


- For HW designers:
  “Power is a 1st and last order design constraint.”
  [Dan Hutcheson, VLSI Research, Inc., E3S Keynote 2011]

- “Every design is a point in a 2D plane.”
  [Mark Horowitz,E3S 2009]

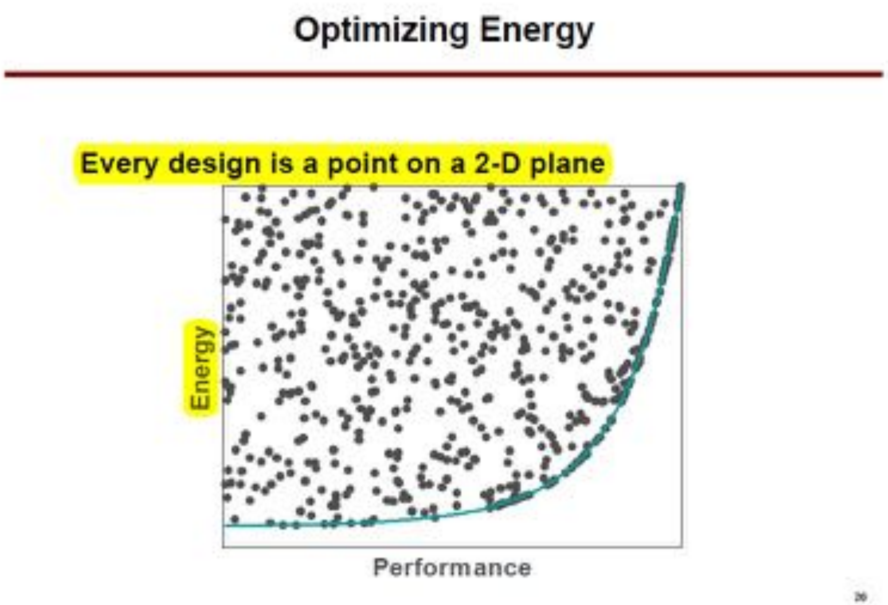


Energy Transparency



- For HW designers:
  “Power is a 1st and last order design constraint.”
  [Dan Hutcheson, VLSI Research, Inc., E3S Keynote 2011]

- “Every design is a point in a 2D plane.”
  [Mark Horowitz,E3S 2009]



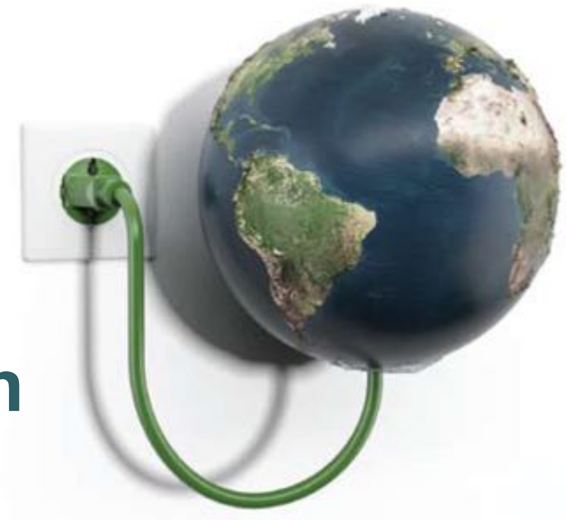
More POWER to SW Developers

```
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 1st class SW design goal is an urgent research challenge.

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

Thank you for your attention

ENTRA Whole-Systems Energy Transparency

MAGEEC

EMBECOSM®

EMC²

ARTEMIS

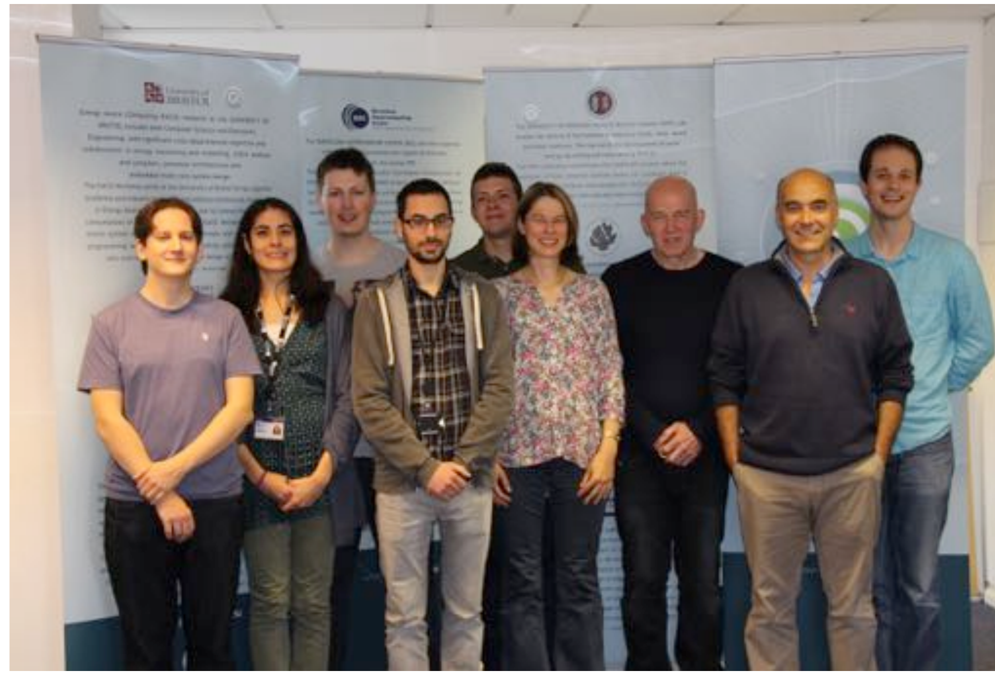
Innovate UK
Technology Strategy Board

EPSRC
Engineering and Physical Sciences Research Council

ICTenergy

XMOS®

The Royal Academy of Engineering



Kerstin.Eder@bristol.ac.uk





February 2010, acmqueue Interview with Steve Furber
The designer of the ARM chip shares lessons on energy-efficient computing at: http://queue.acm.org/detail.cfm?id=1716385

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.

Steve Furber