Whole Systems Energy Transparency

More *power* to software developers!

(Part II)

Kerstin Eder

Trustworthy Systems Laboratory, University of Bristol
Verification and Validation for Safety in Robots, Bristol Robotics Laboratory

University of BRISTOL
Department of COMPUTER SCIENCE
The Royal Academy of Engineering

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
- Why timing and energy analysis differ

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
- Why timing and energy analysis differ

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)."*

Acknowledgements

The partners in the EU ENTRA project

John Gallagher and team

Pedro López García and team

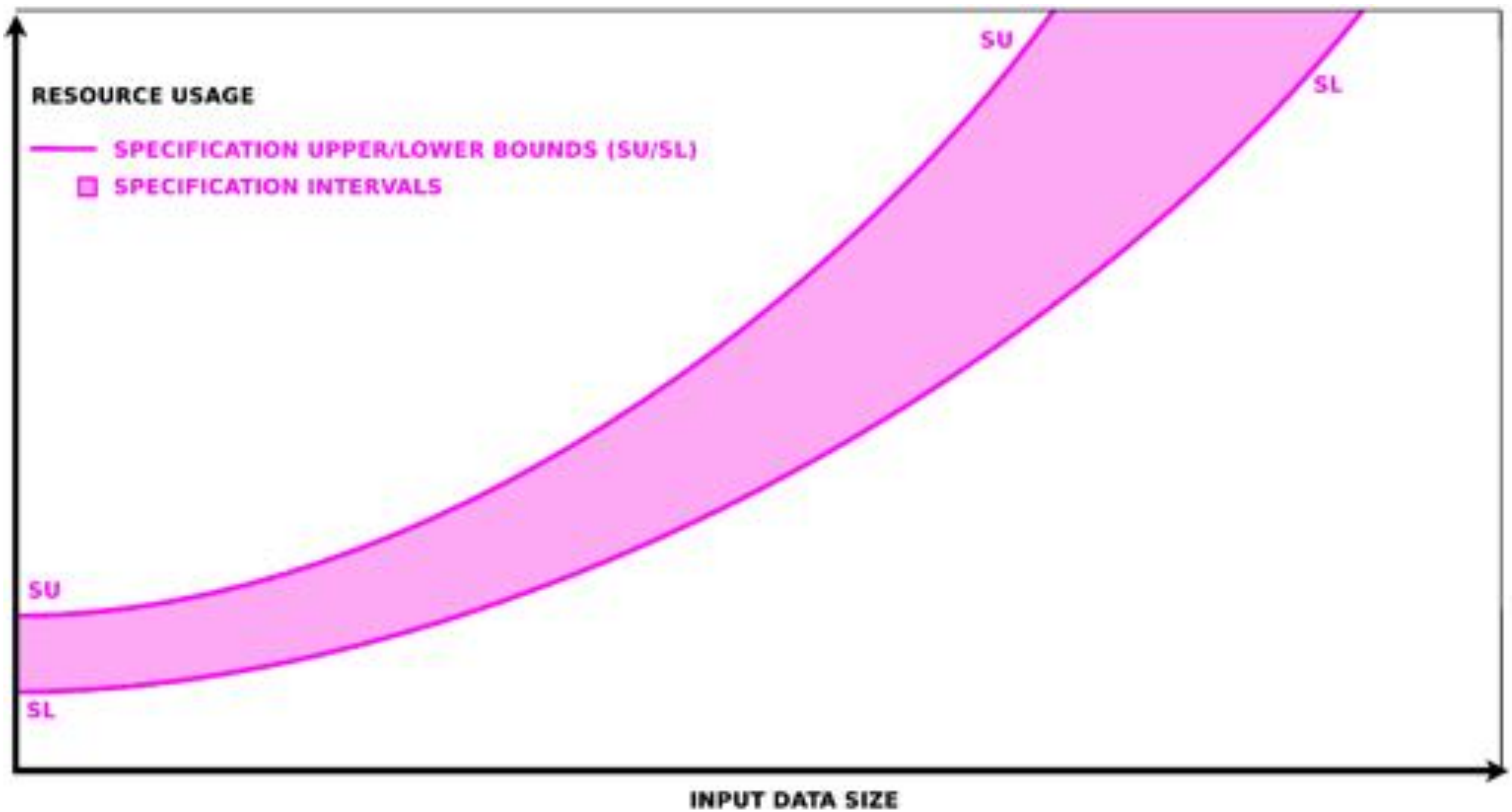
Henk Muller and team

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

SRA for Energy Consumption

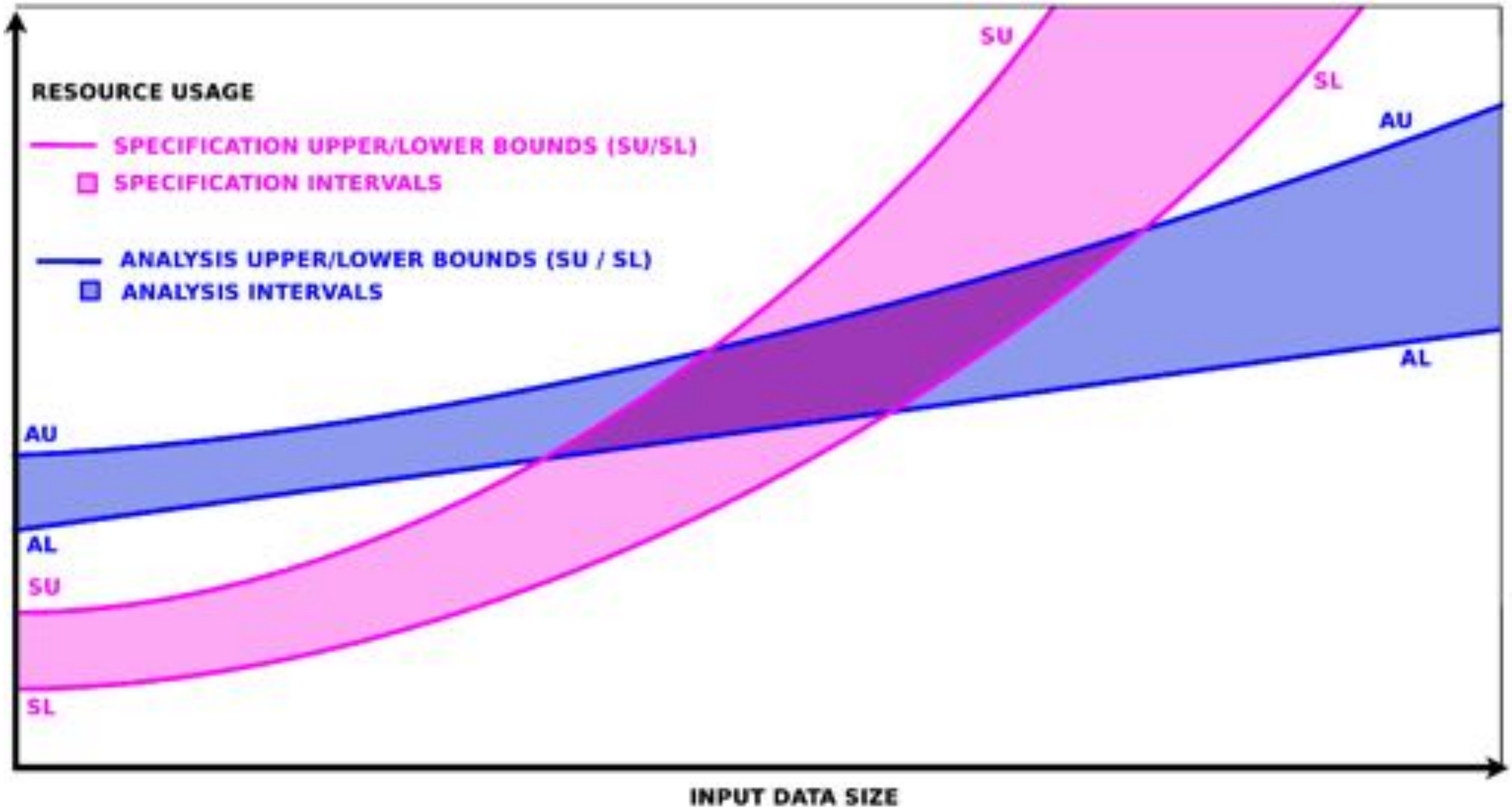
- Adaptation of traditional resource usage analysis techniques to *energy consumption*.
- Techniques automatically infer upper and lower bounds on energy usage of a program.
- Bounds expressed using monotonic arithmetic functions per procedure parameterized by program’s input size.
- Verification can 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



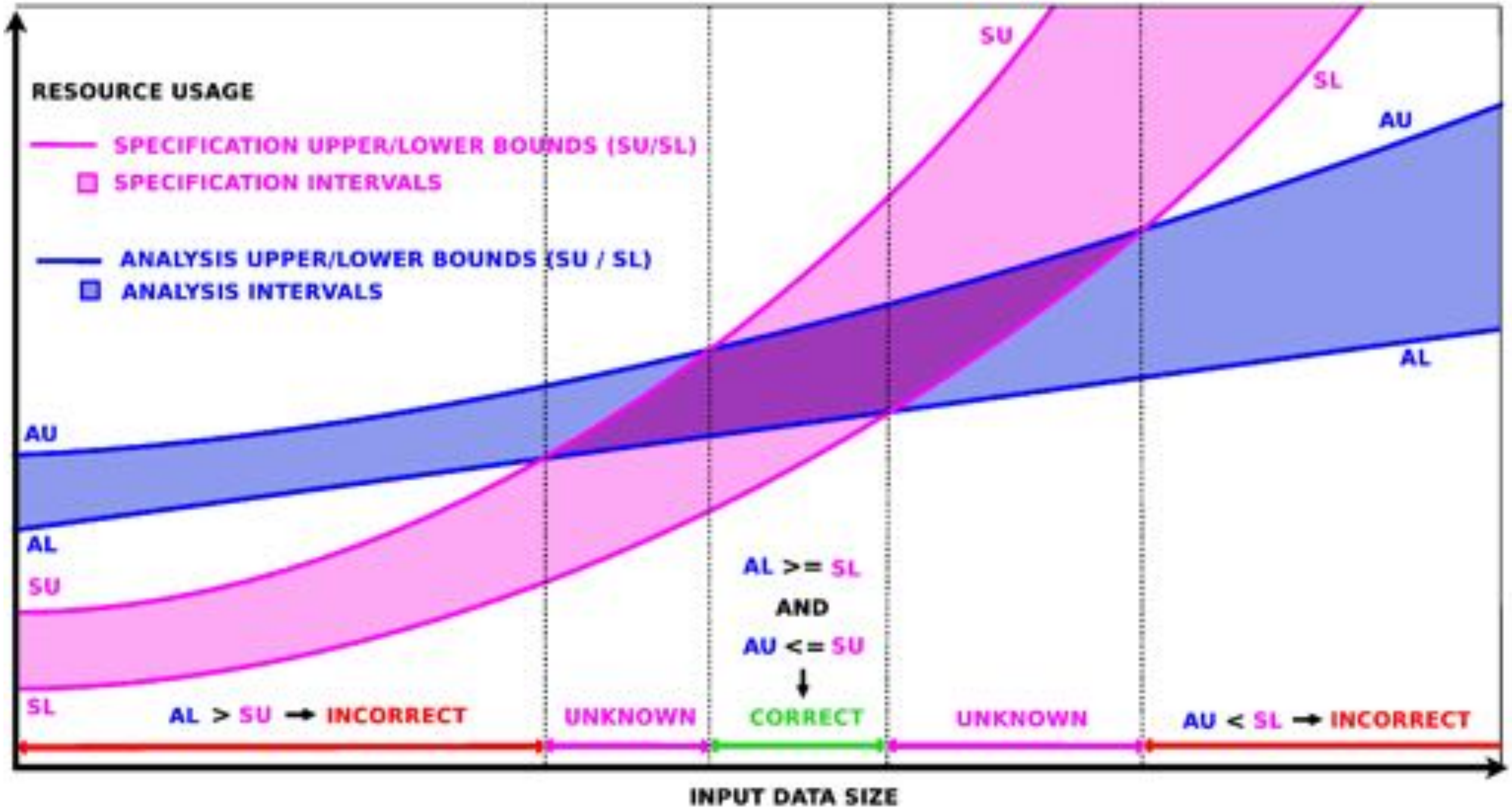
Source: Pedro Lopez Garcia, IMDEA Software Research Institute

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:

$C_{\mathbf{fact}}(x) = C_a + C_b \quad \text{if } x<=0$

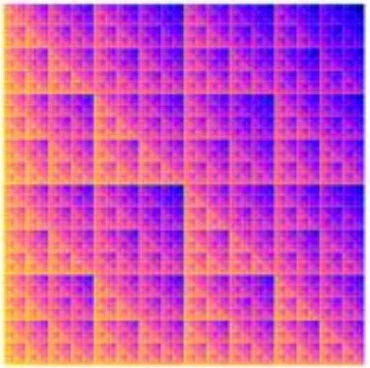
$C_{\mathbf{fact}}(x) = C_a + C_c(x) \quad \text{if } x>0$

$C_c(x) = C_d + C_{\mathbf{fact}}(x-1)$

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

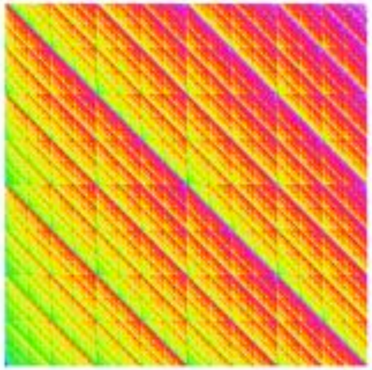
Energy Modelling

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
    - 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

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

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

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:

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

- 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:

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

- 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)`)?

```c
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
        bne     .L3
.L2:
        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

$O_{i,j}$ [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

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:

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

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

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

```
@ 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

$E_{fact(3)}$ = 3*600pJ + 3*300pJ + 500pJ +2*900 + 2*500pJ + 700pJ + 3*20pJ + 10pJ + 30pJ + 2*30pJ + 2*10pJ + 30pJ + 10pJ

= 6920pJ **= 6.92nJ**

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

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.

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

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.

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)*:

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

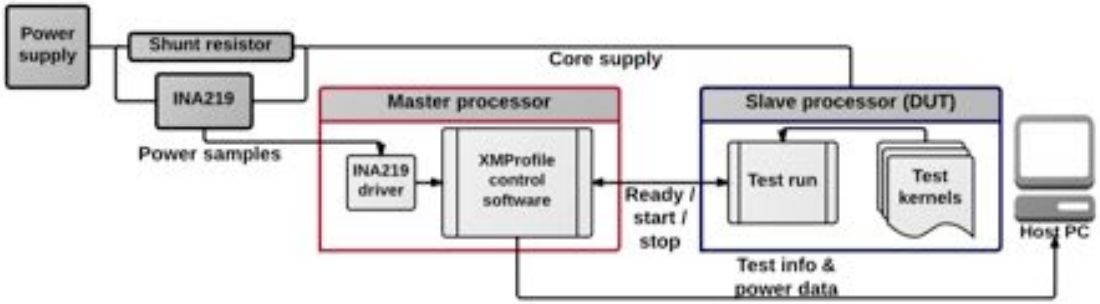
Idle base power and duration

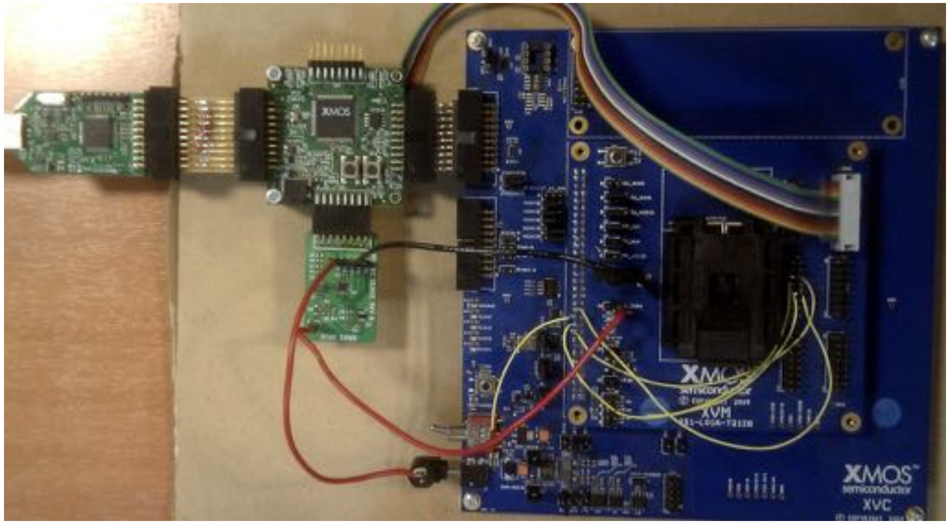
Concurrency cost, instruction cost, generalised overhead, base power and duration

- 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

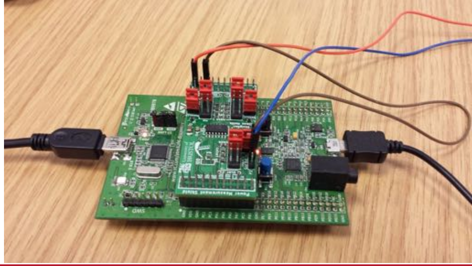
The set up...

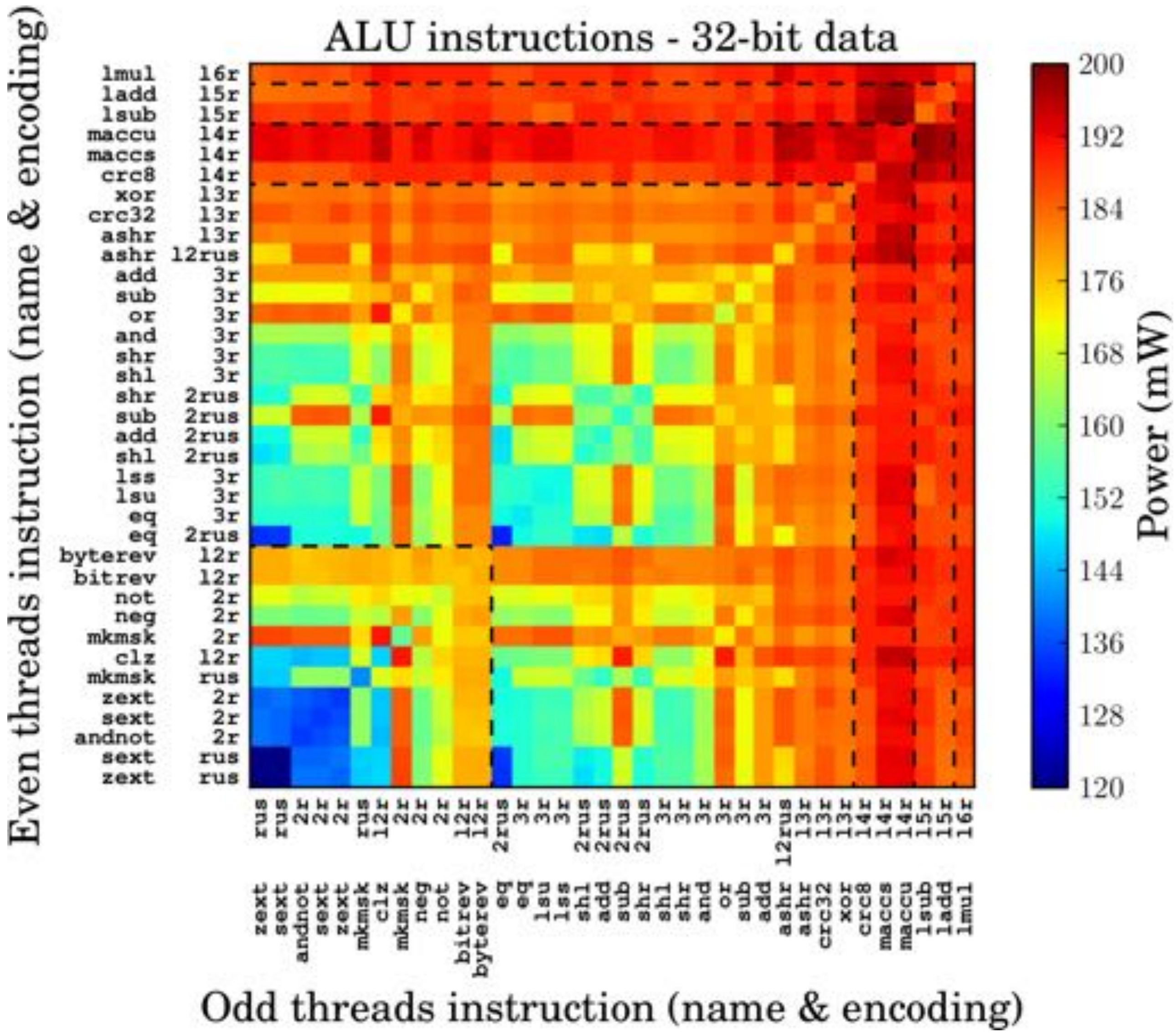




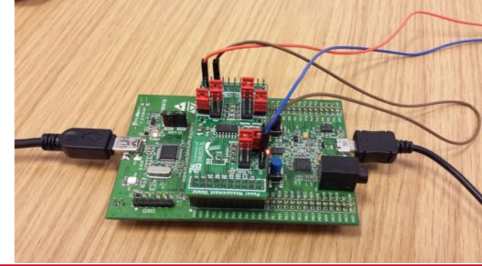
S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages. DOI=10.1145/2700104 http://doi.acm.org/10.1145/2700104

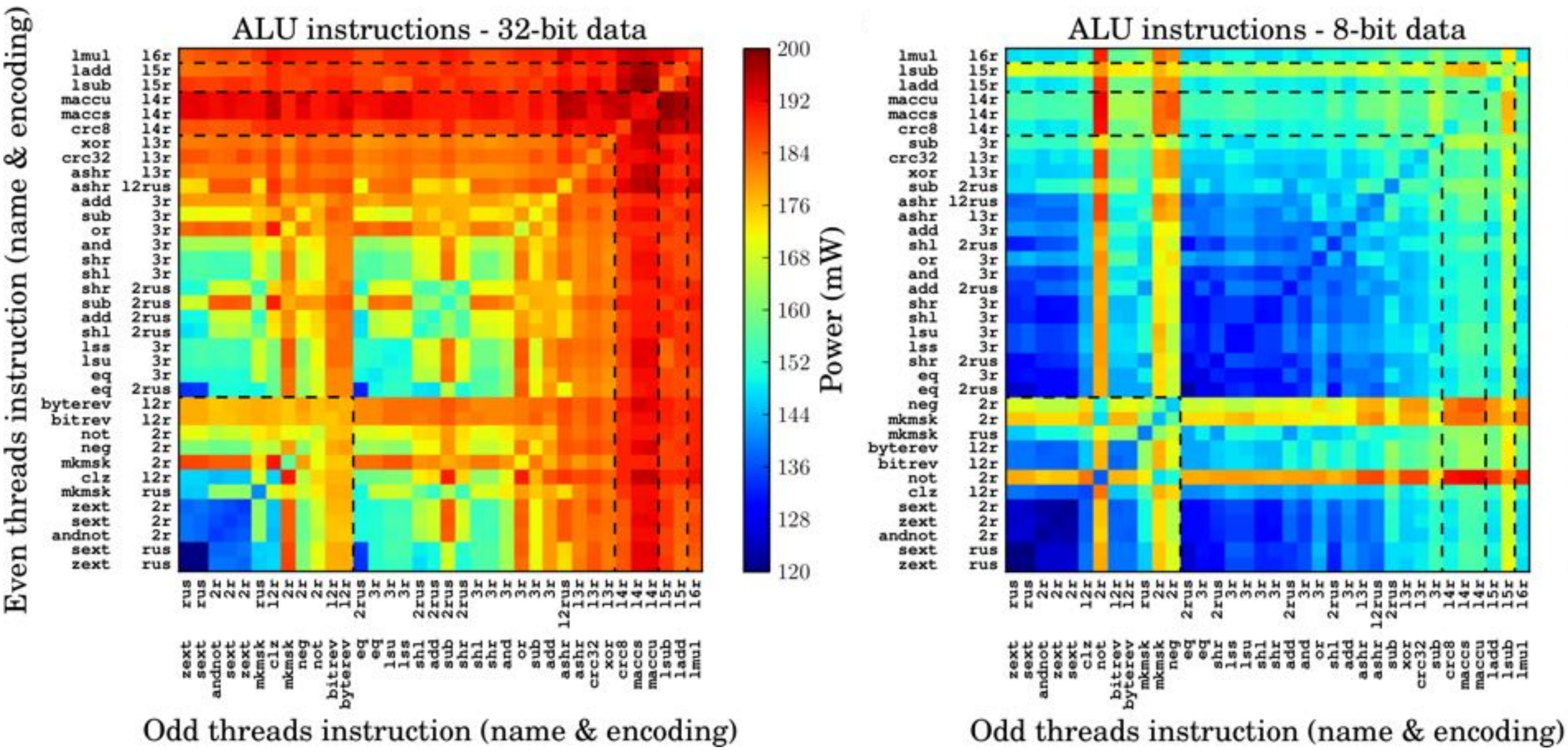
ISA Characterization





ISA Characterization



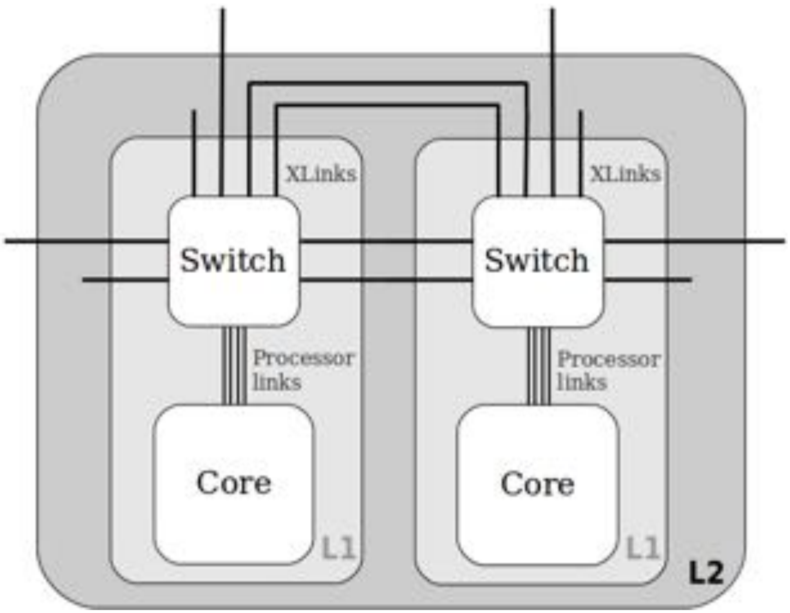


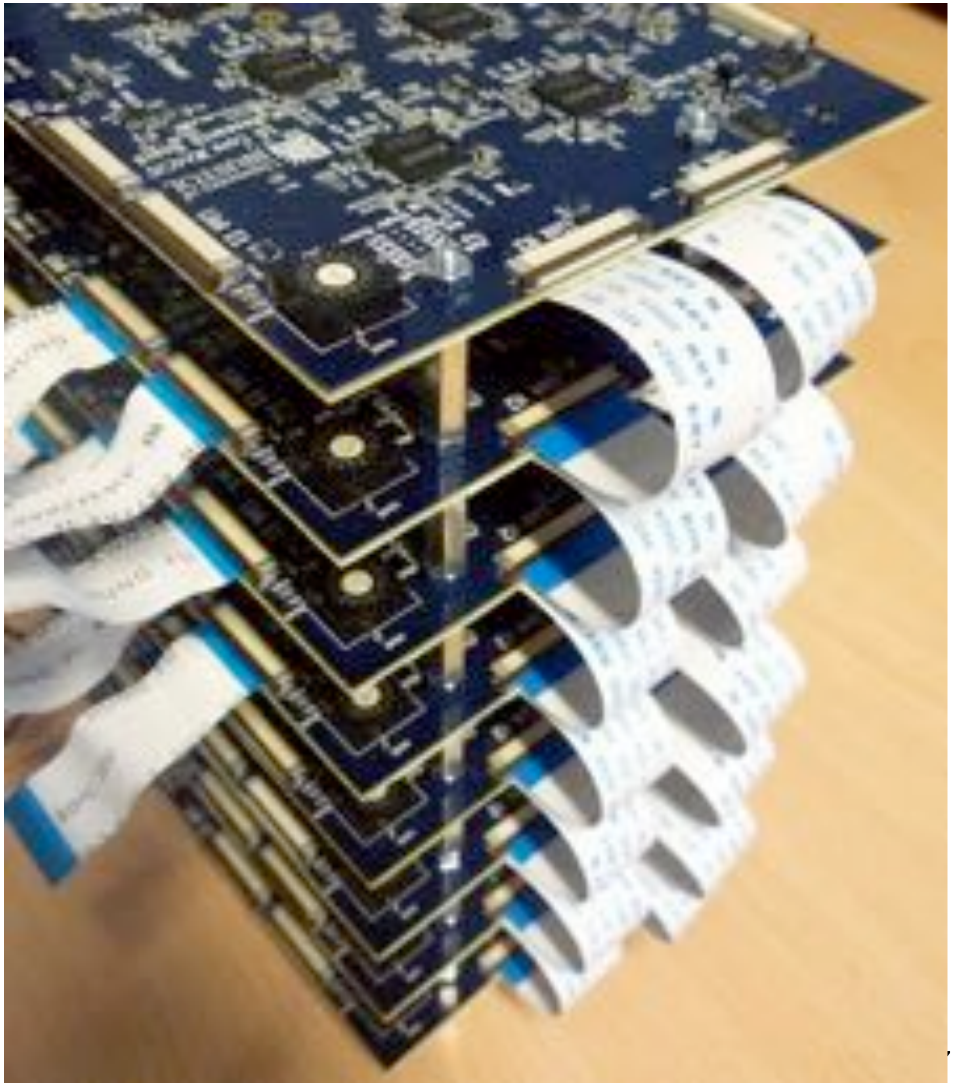
S. Kerrison and K. Eder. 2015. “Energy Modeling of Software for a Hardware Multithreaded Embedded Microprocessor”. ACM Trans. Embed. Comput. Syst. 14, 3, Article 56 (April 2015), 25 pages. DOI=10.1145/2700104 http://doi.acm.org/10.1145/2700104

The Cost of Communication

The Swallow Platform:

- 480 processor embedded system, based on the XMOS XS1 architecture
- 16 cores per slice

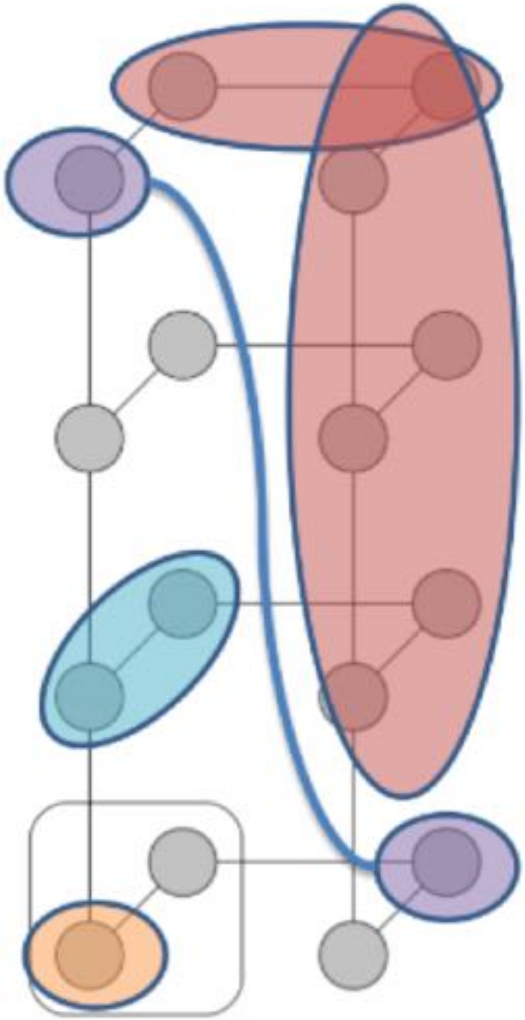




Biquad Filter Example

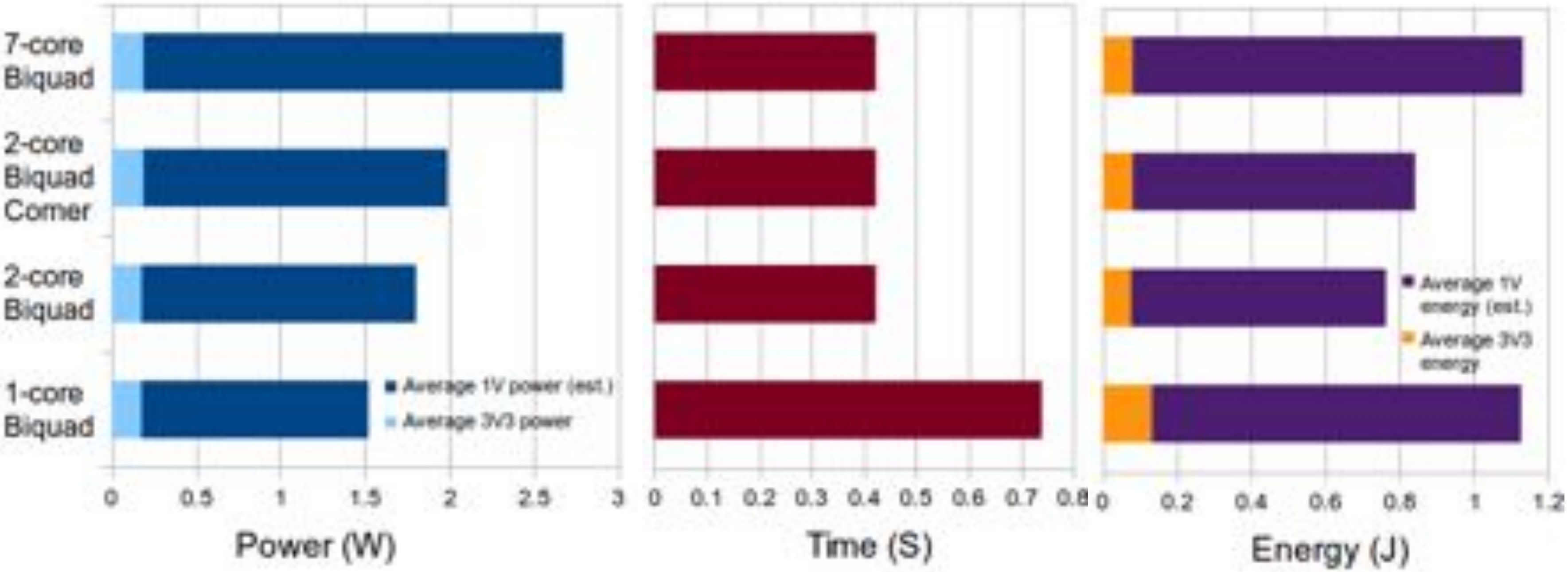
Implemented in various configurations on Swallow:

- 7 threads on 1 core,
- 7 threads across 2 cores,
  - Good spatial locality,
  - Poor spatial locality,
- 7 threads across 7 cores.



Comms example: Biquad filter

- 7-stage biquad filter implemented in various configurations on Swallow.
- Active cores, latency, contention and under/over-allocation all affect total energy.
- Power, energy & time a valuable triple.

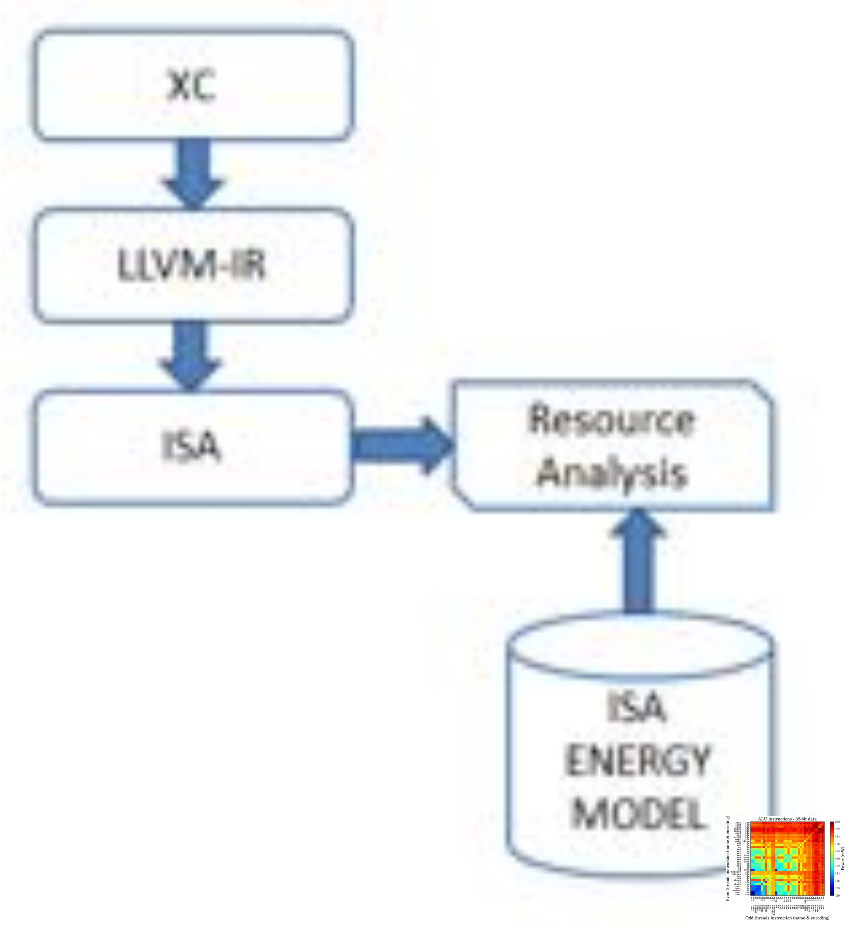


Energy Consumption Analysis
enables energy transparency

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

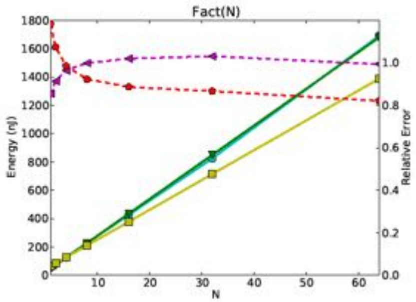
Original Program:

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

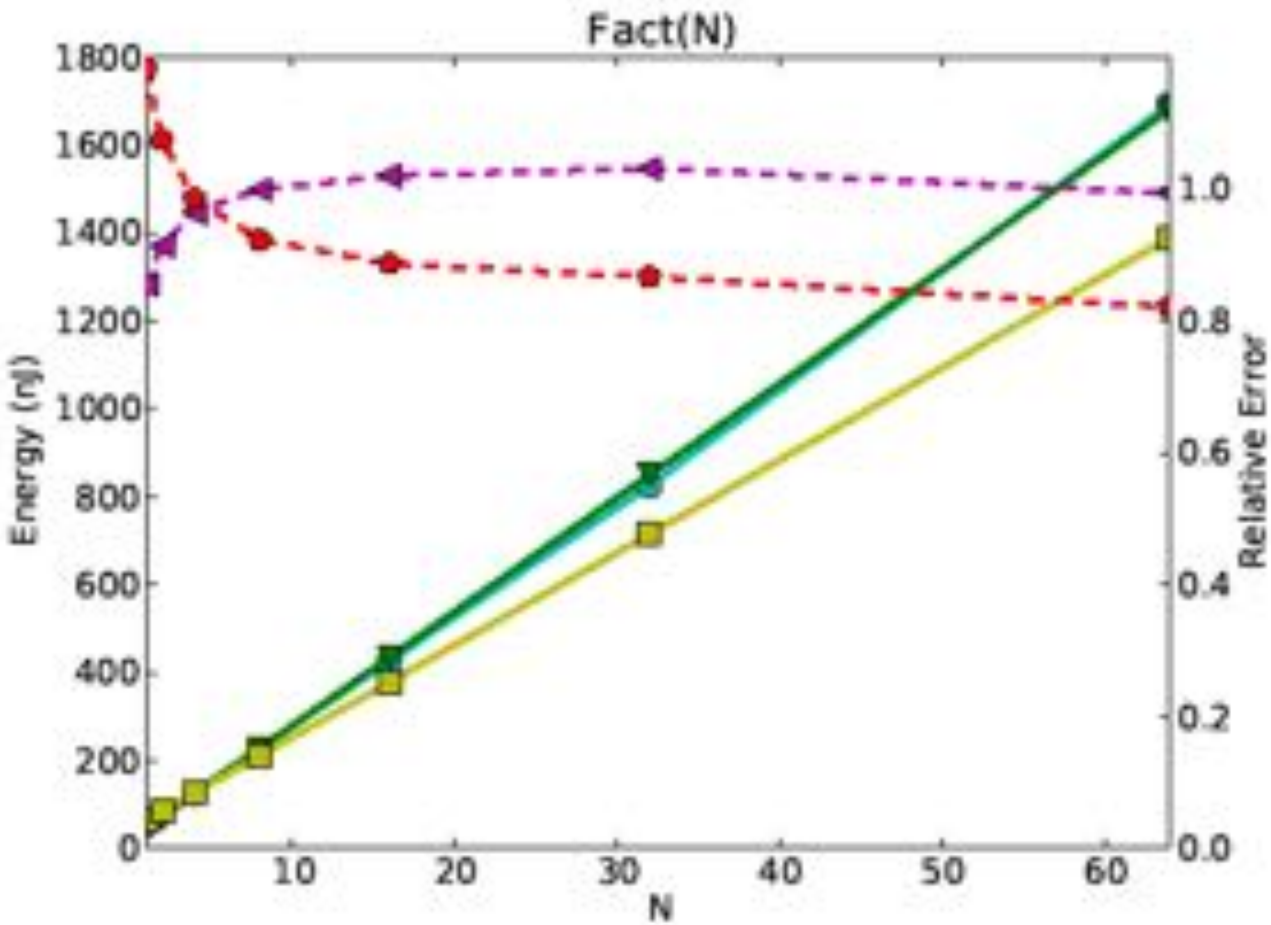
Extracted Cost Relations:

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

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

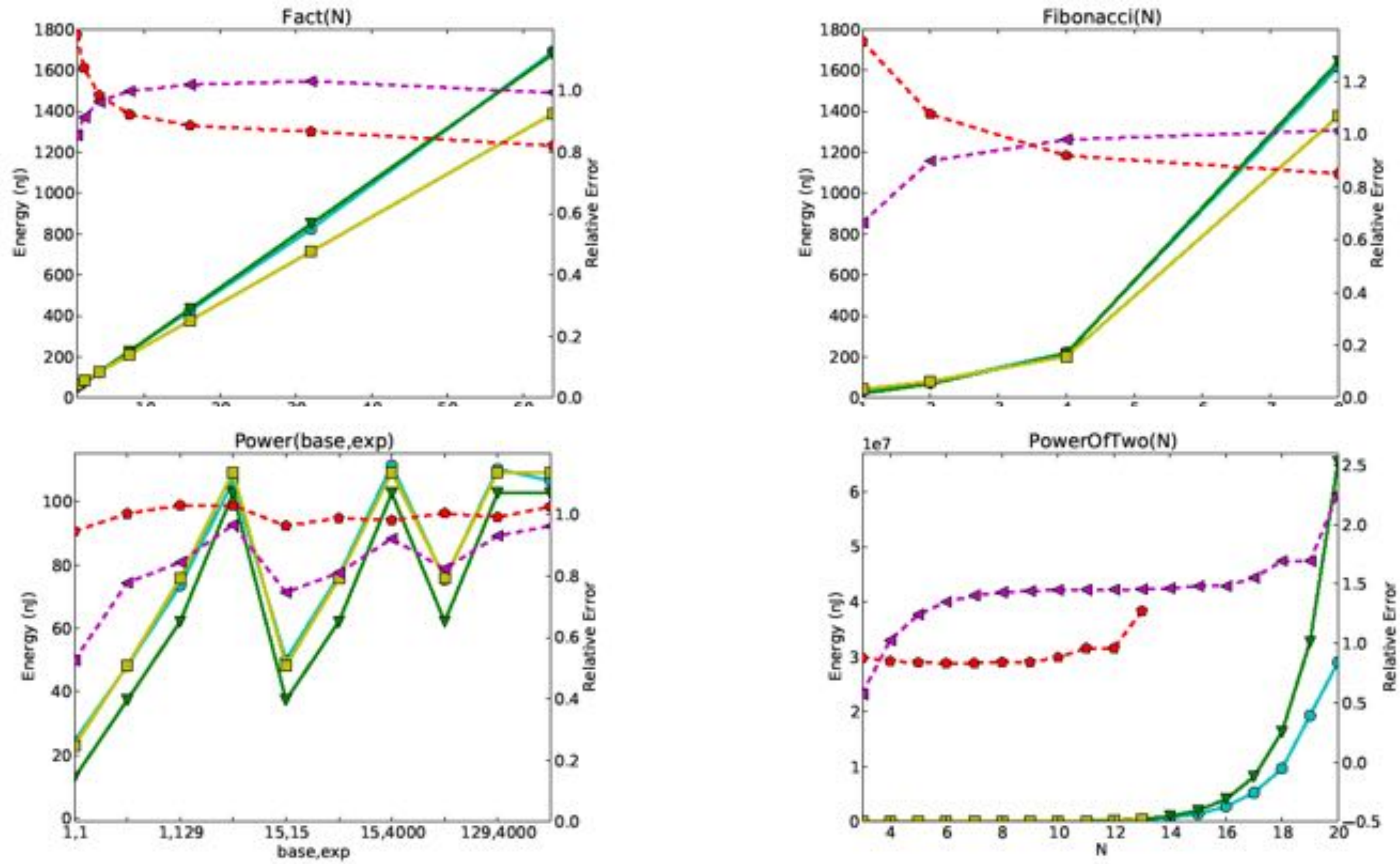


ISA-Level Analysis Results



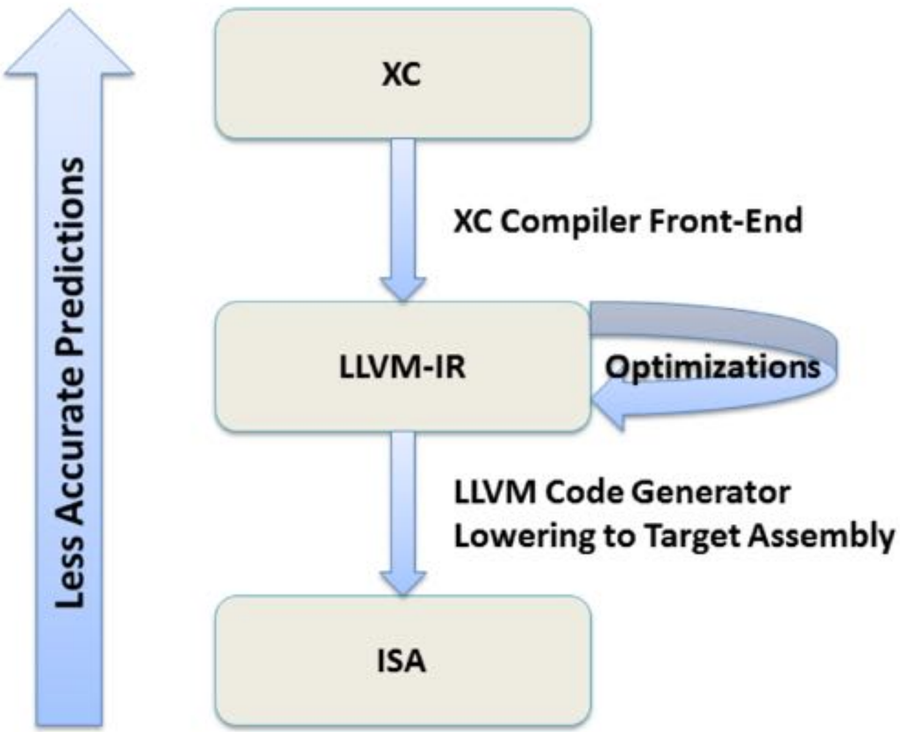
U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, N. Grech, P. Lopez-Garcia, M.V. Hermenegildo and K. Eder. "Energy Consumption Analysis of Programs based on XMOS ISA-Level Models". LOPSTR 2013. LNCS 8901. Springer. DOI: 10.1007/978-3-319-14125-1_5

ISA-Level Analysis Results



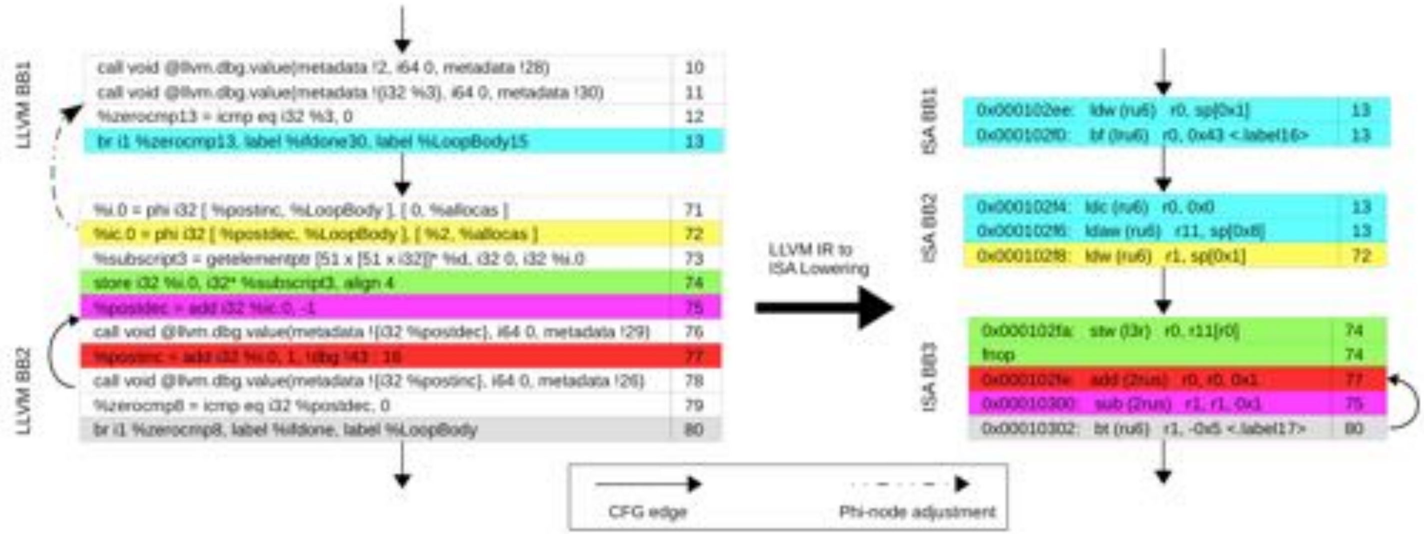
U. Liqat, S. Kerrison, A. Serrano, K. Georgiou, N. Grech, P. Lopez-Garcia, M.V. Hermenegildo and K. Eder. “Energy Consumption Analysis of Programs based on XMOS ISA-Level Models”. LOPSTR 2013.
LNCS 8901. Springer. DOI: [10.1007/978-3-319-14125-1_5](10.1007/978-3-319-14125-1_5)

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

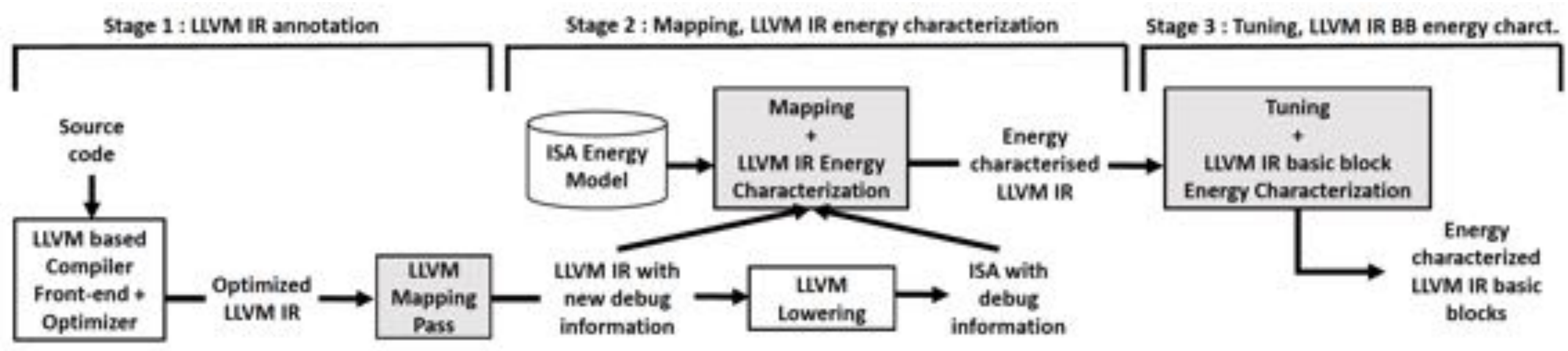


$$E(ir_i) = \sum_{isa_j \in R(ir_i)} E(isa_j)$$

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

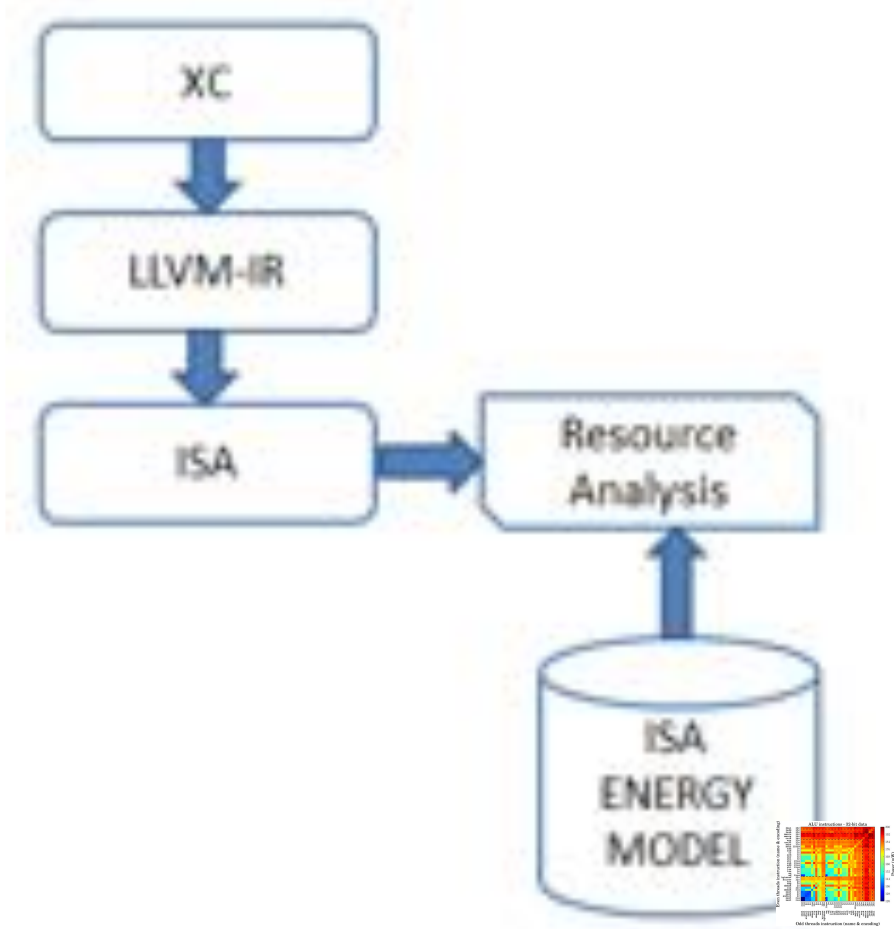
U. Liqat, K. Georgiou, S. Kerrison, P. Lopez-Garcia, J.P. Gallagher, M.V. Hermenegildo, K. Eder. “Inferring Parametric Energy Consumption Functions at Different Software Levels: ISA vs. LLVM IR”. In Proceedings of FOPARA 2015. LNCS 9964. Springer. DOI: 10.1007/978-3-319-46559-3_5 http://arxiv.org/abs/1511.01413

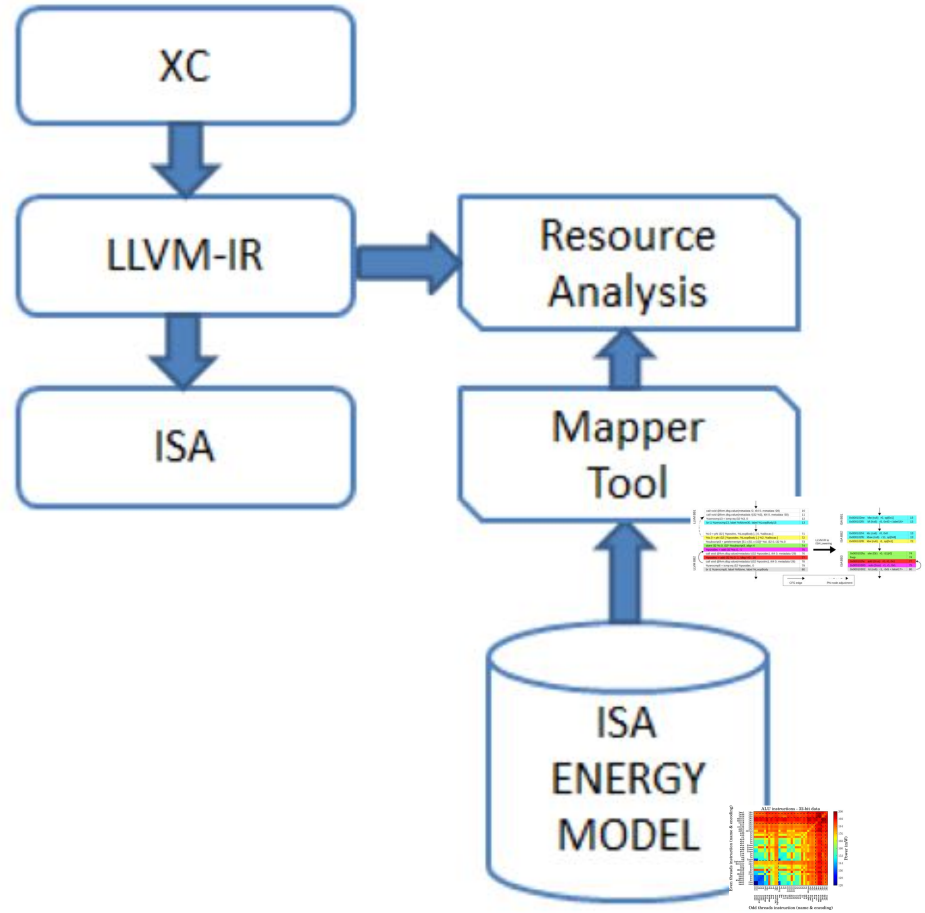
LLVM IR Energy Characterization



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

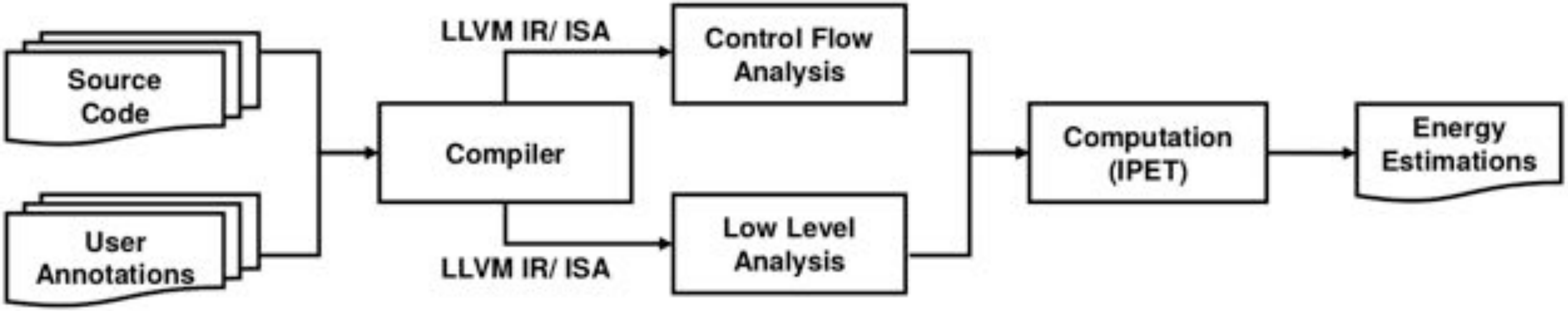
Analysis at the LLVM IR Level





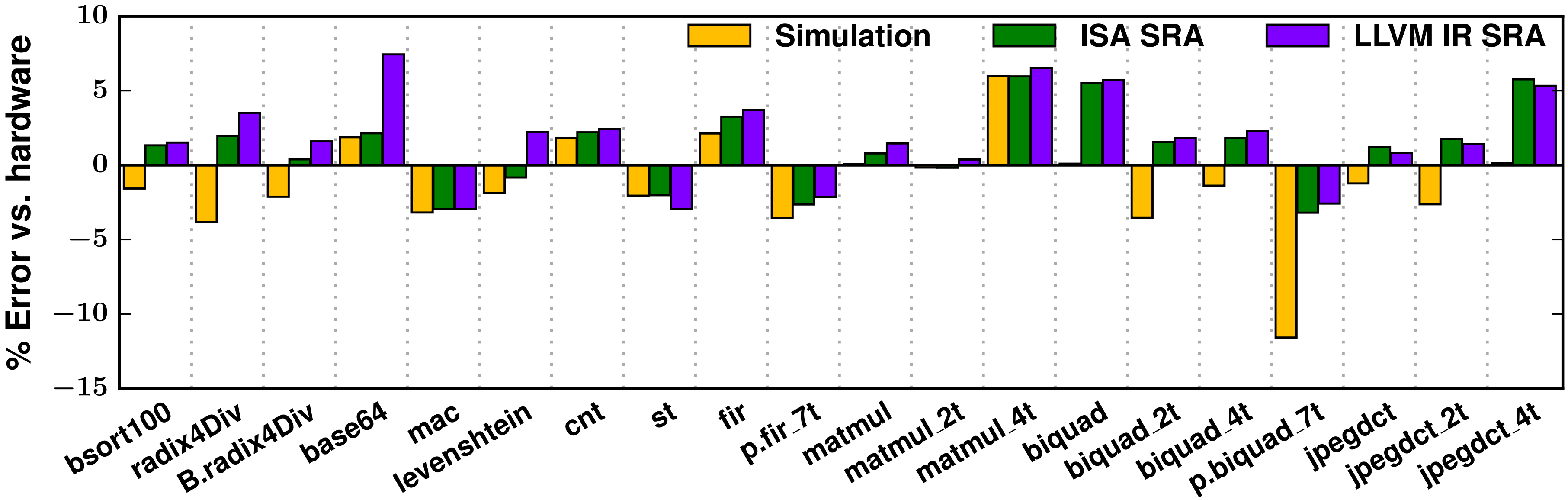
N. Grech, K. Georgiou, J. Pallister, S. Kerrison, J. Morse, K. Eder. 2015. “Static analysis of energy consumption for LLVM IR programs”. In Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems (SCOPES '15). ACM, New York, NY, USA, pages 12-21. http://dx.doi.org/10.1145/2764967.2764974

SRA for Energy Consumption



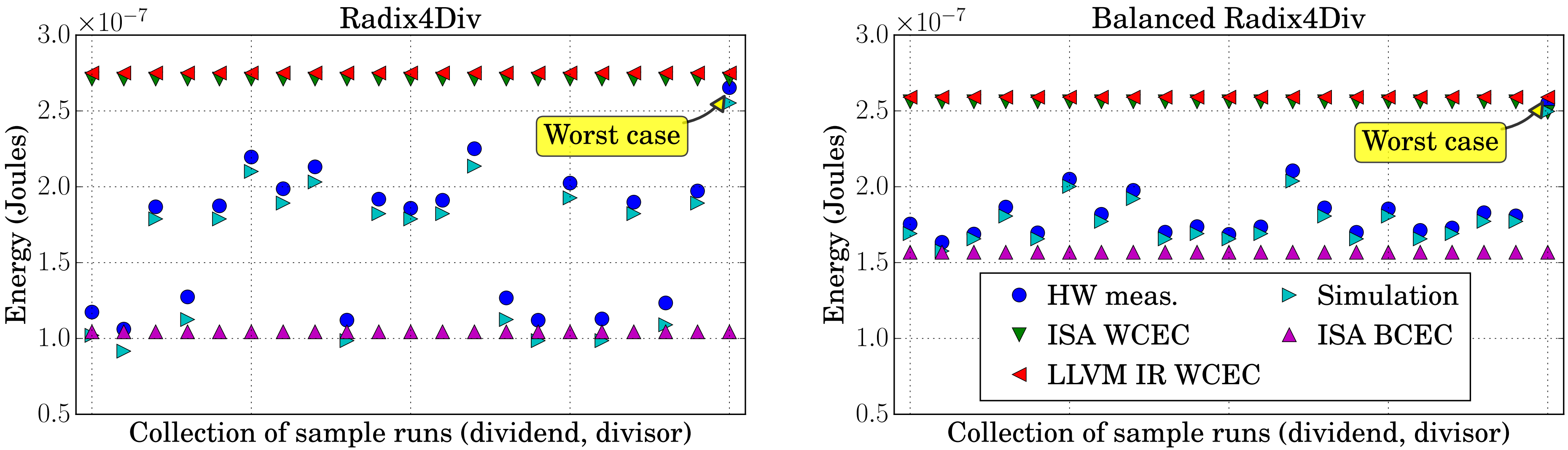
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. “Energy Transparency for Deeply Embedded Programs”. ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

SRA for Energy Consumption



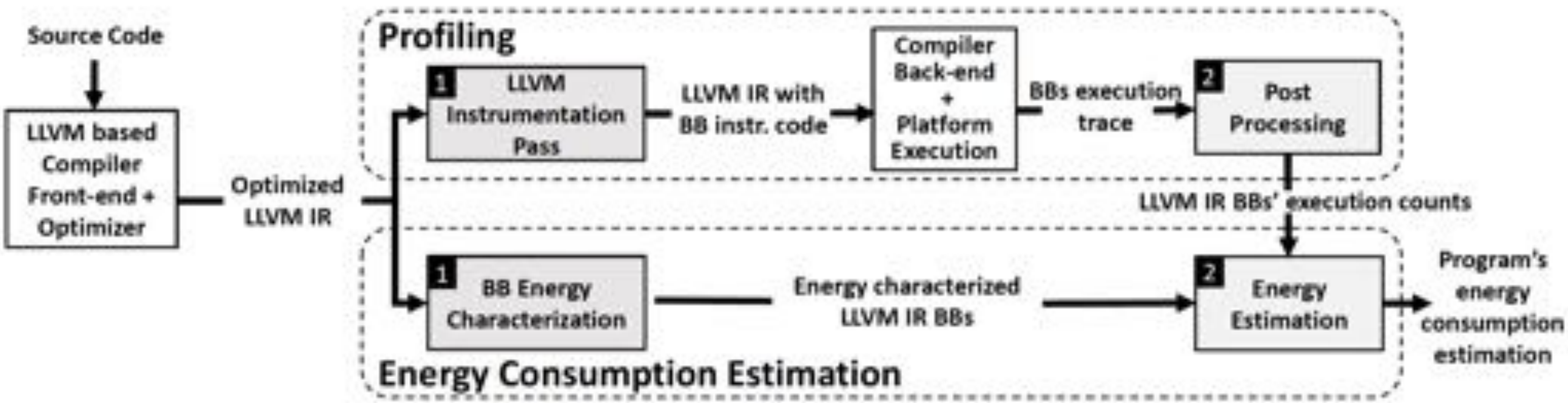
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. "Energy Transparency for Deeply Embedded Programs". ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

EC Static Analysis Results



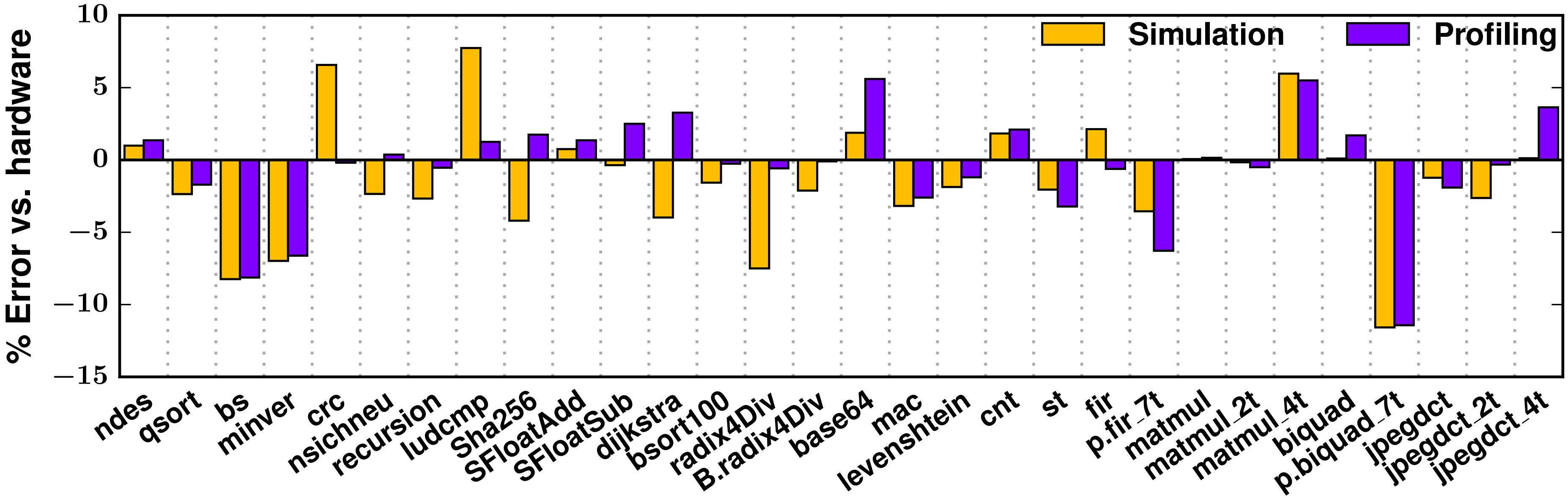
K. Georgiou, S. Kerrison, Z. Chamski and K. Eder. 2017. "Energy Transparency for Deeply Embedded Programs". ACM Trans. Archit. Code Optim. (TACO) 14, 1, Article 8 (March 2017), 26 pages. DOI: https://doi.org/10.1145/3046679. https://arxiv.org/abs/1609.02193

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

Learning Objectives

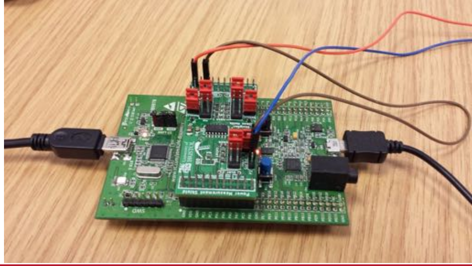
- ✓ Why software is key to energy efficient computing
- ✓ What energy transparency means and why we need energy transparency to achieve energy efficient computing
- ✓ How to measure the energy consumed by software
- ✓ How to estimate the energy consumed by software *without* measuring
- ✓ How to construct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

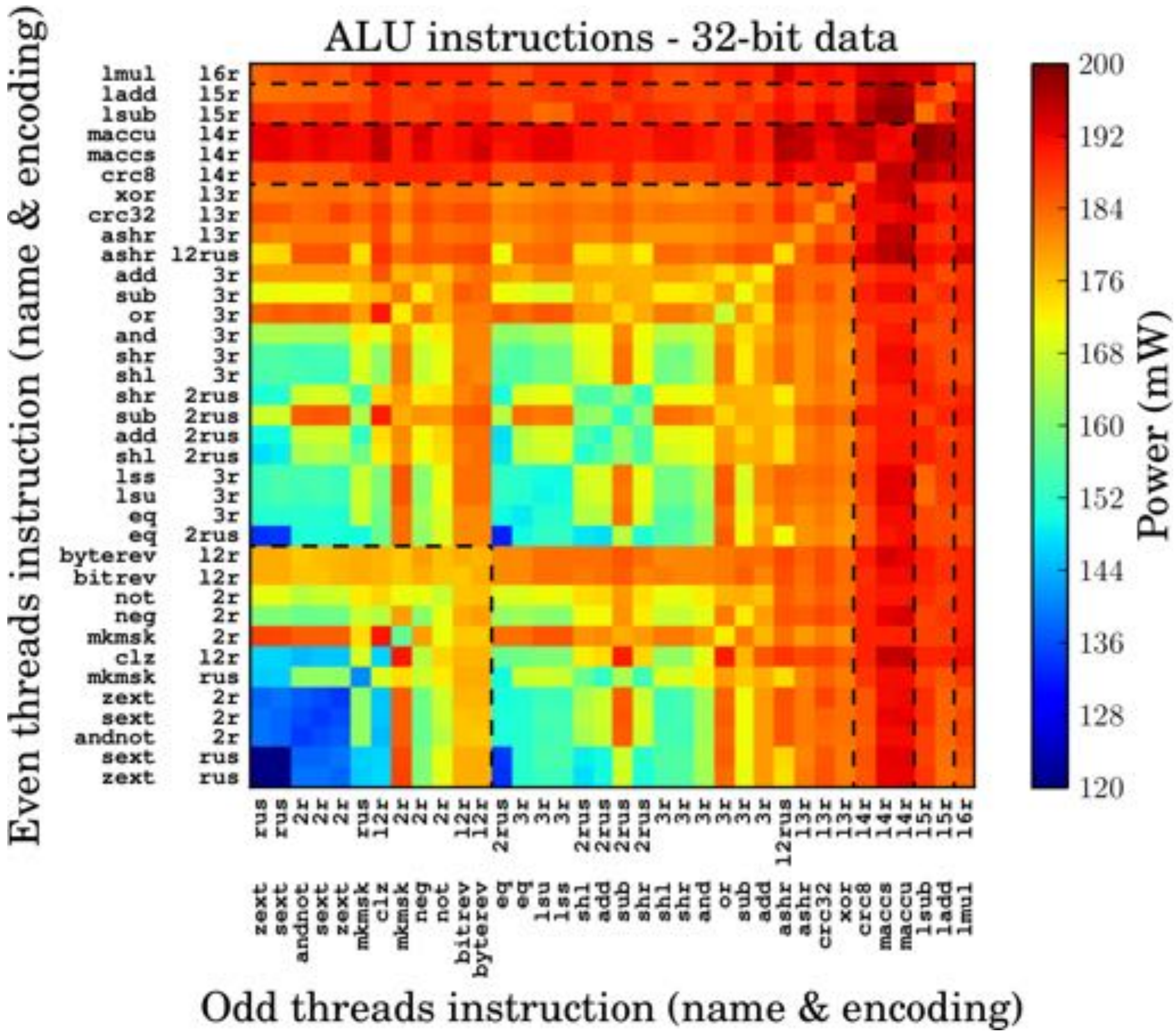
Learning Objectives

- ✓ 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
- Why timing and energy analysis differ

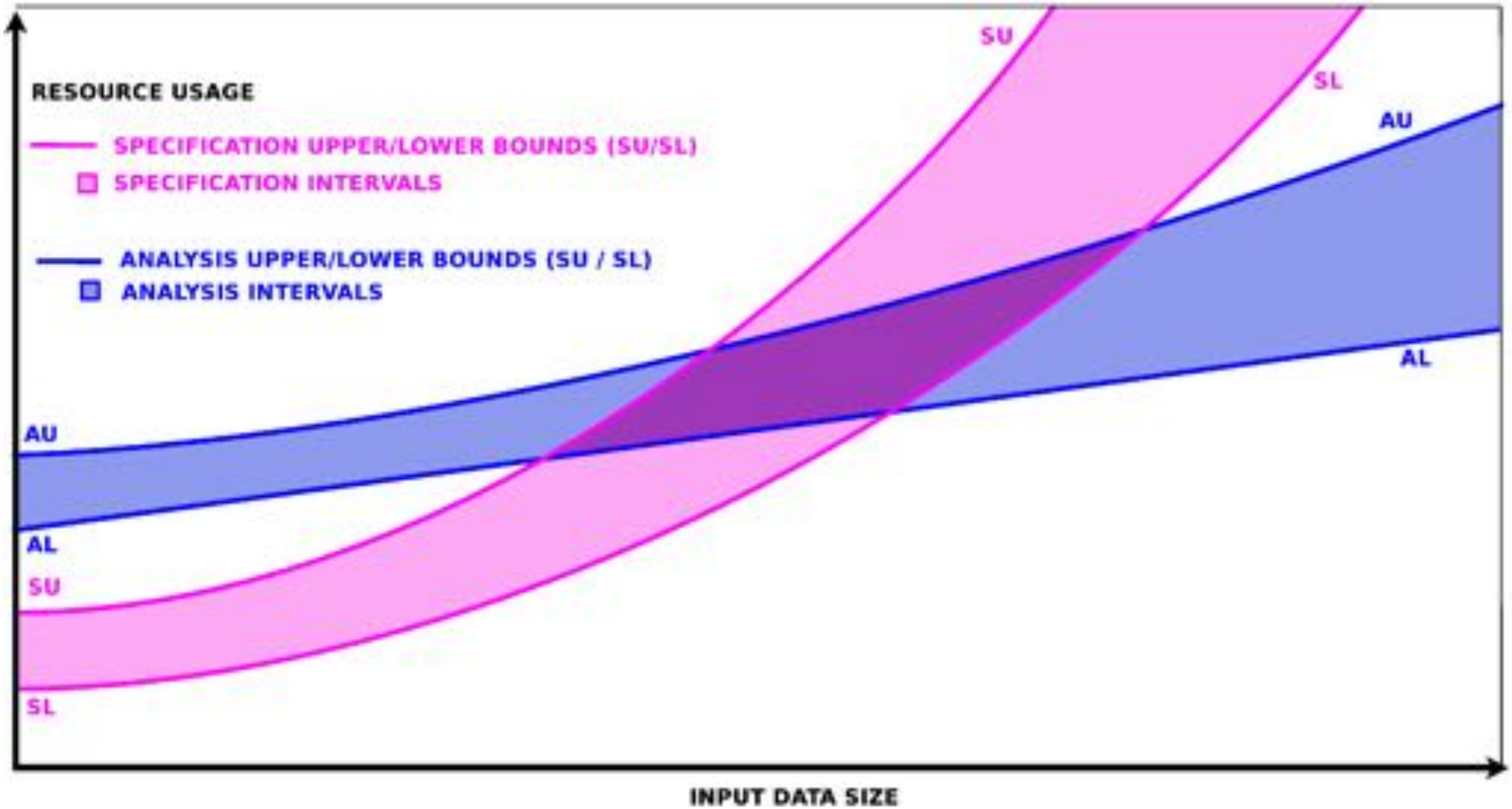
The Worst Case …

ISA Characterization





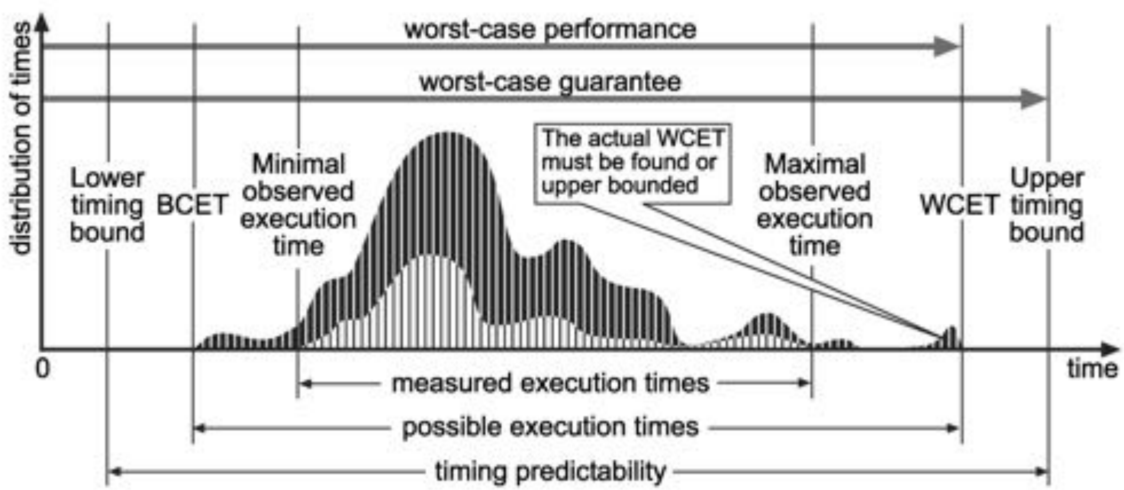
Static Resource *Bound* Analysis



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)

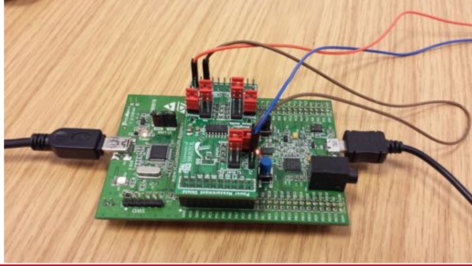
Does this work for energy consumption analysis?

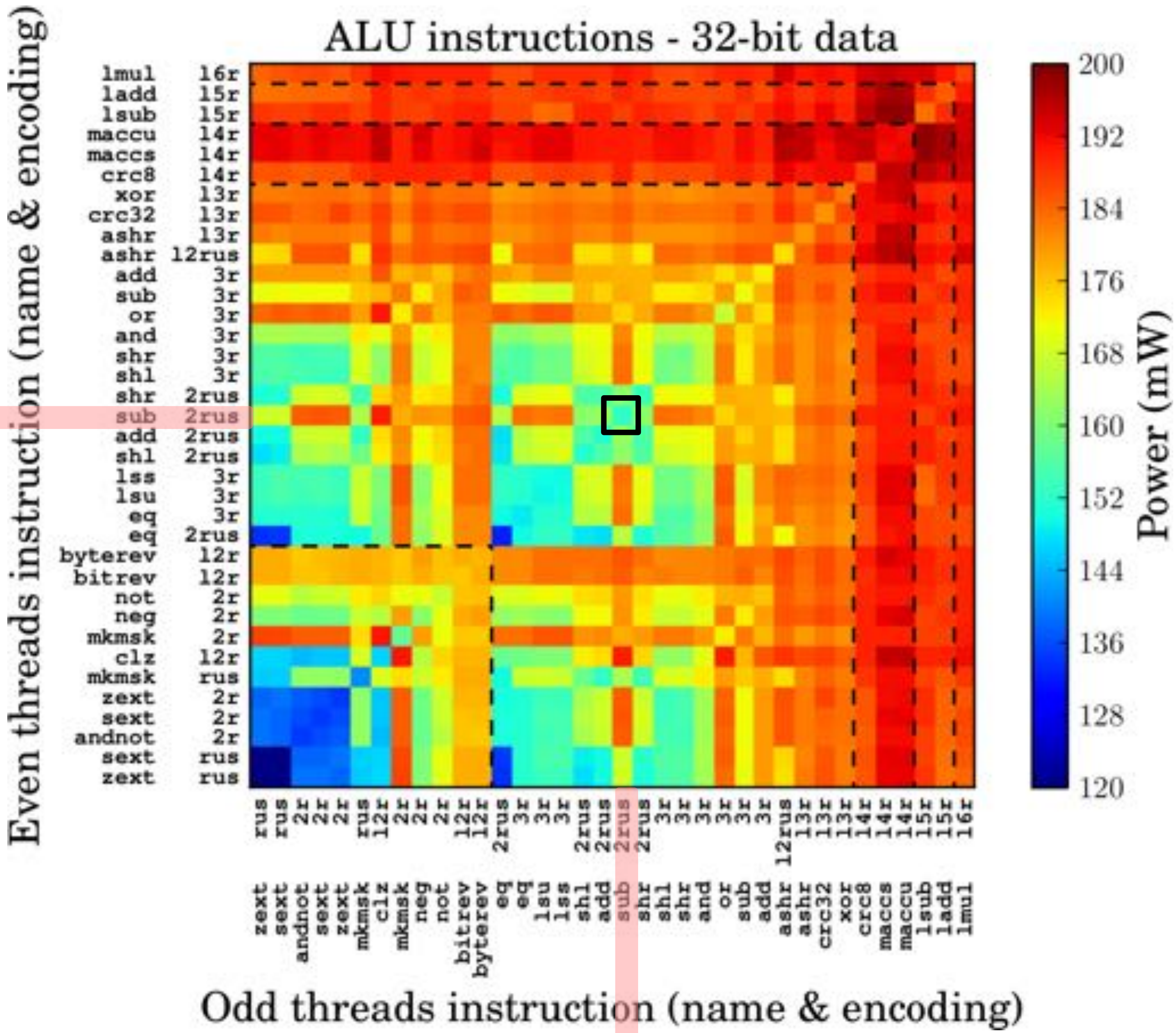
Worst Case Energy Consumption

- WCEC analysis goes well beyond WCET analysis.
  - embedded real-time systems that are timing predictable execute instructions in a fixed number of clock cycles
  - timing variability has mostly been eliminated “by design" through the use of synchronous logic
  - WCET then depends only on the WC execution path
- **But, energy consumption is**

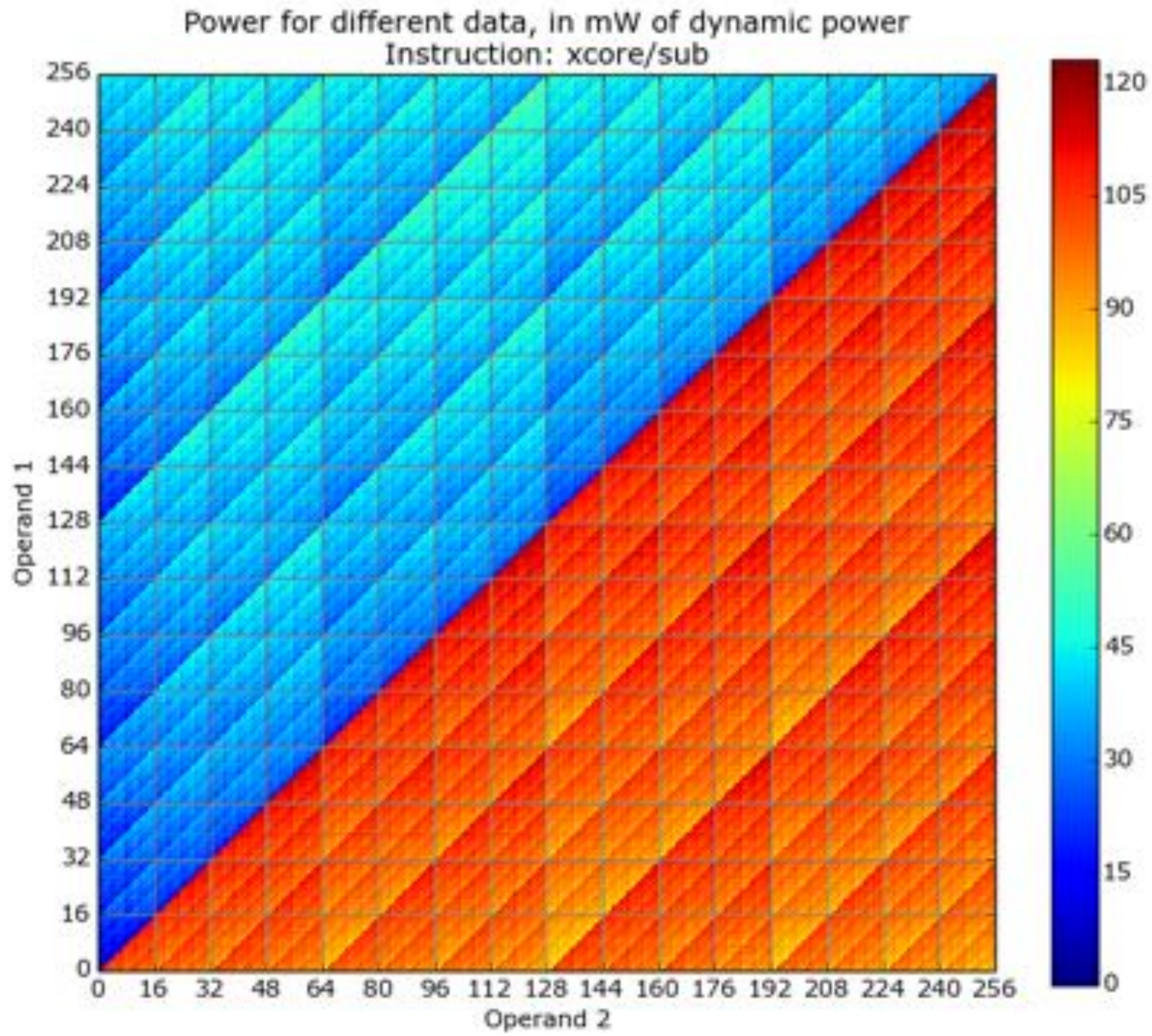
  ***data dependent*.**

ISA Characterization

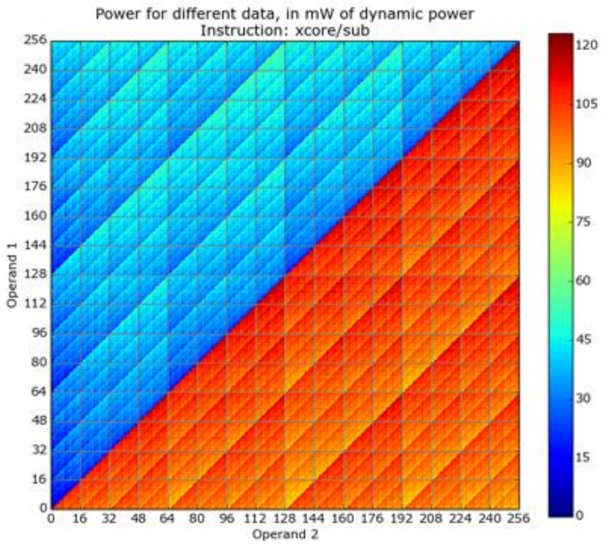


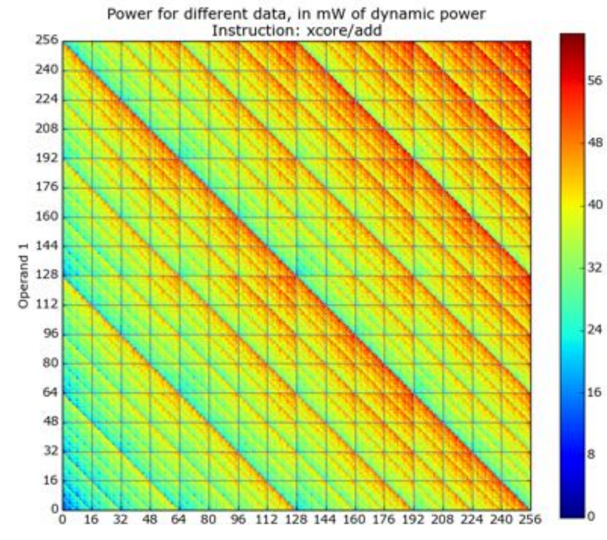


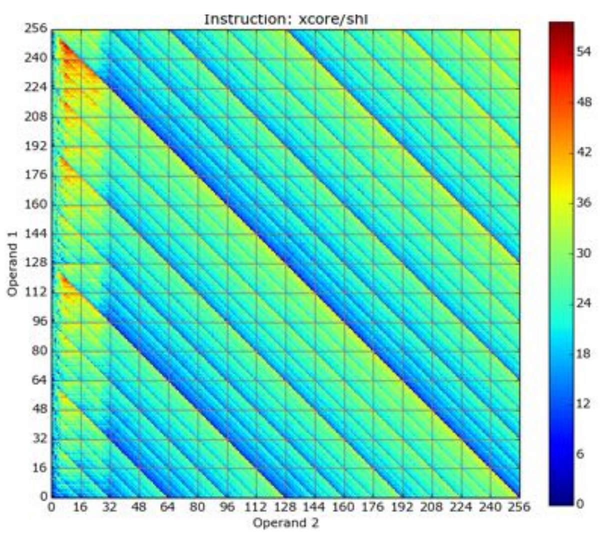
W/A/B-Case Energy Consumption

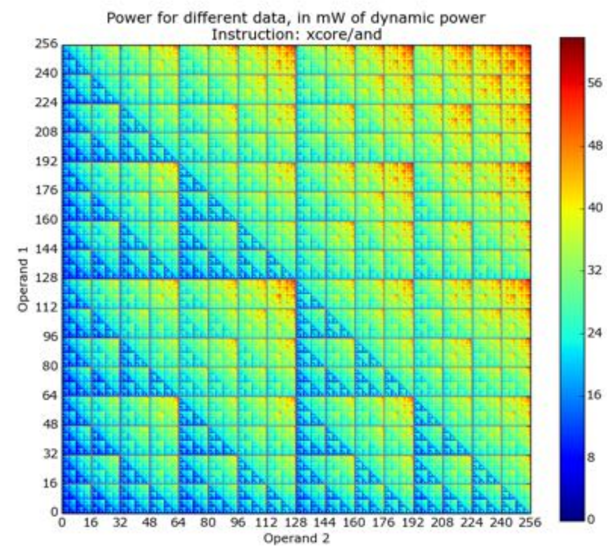


W/A/B-Case Energy Consumption

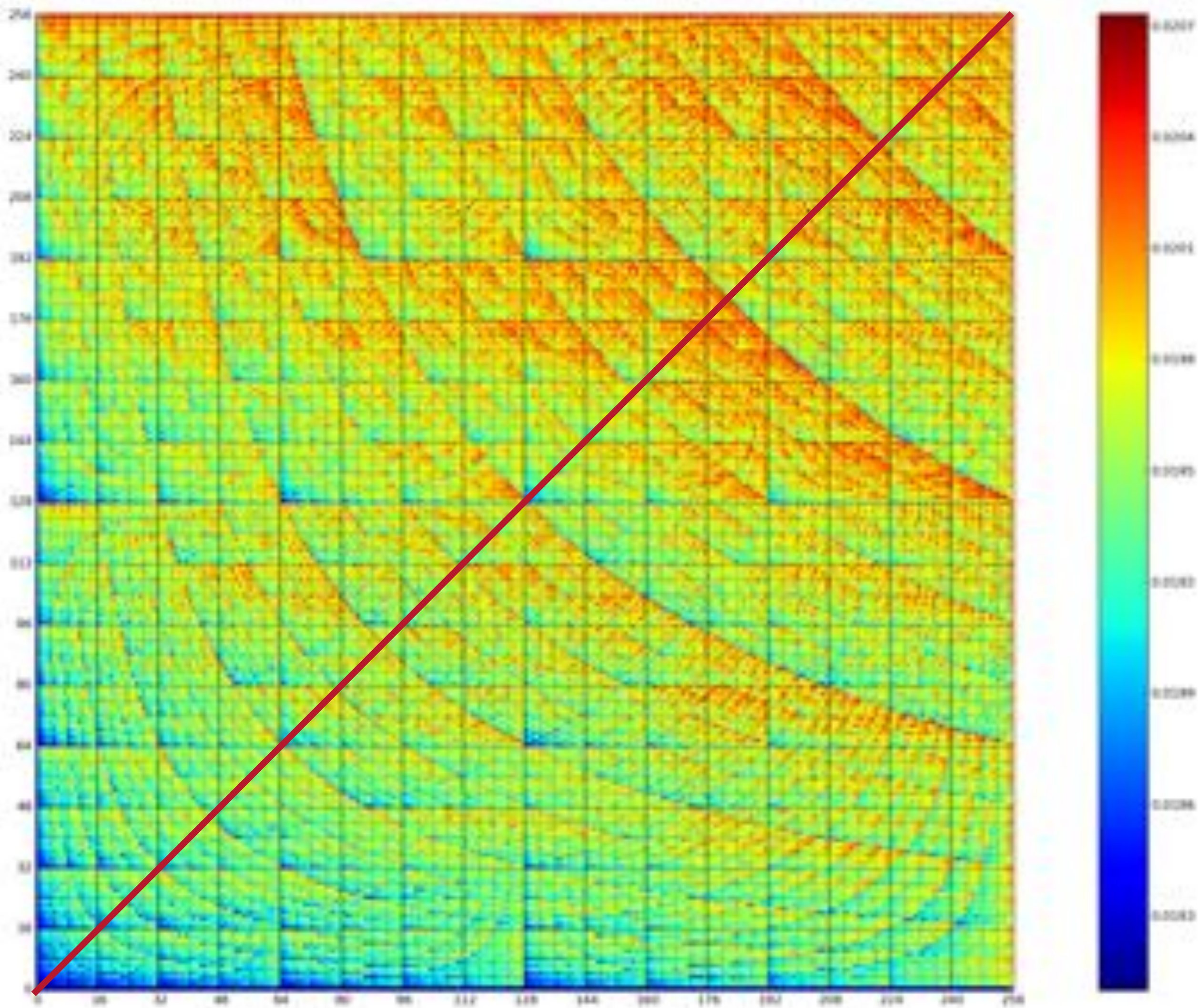




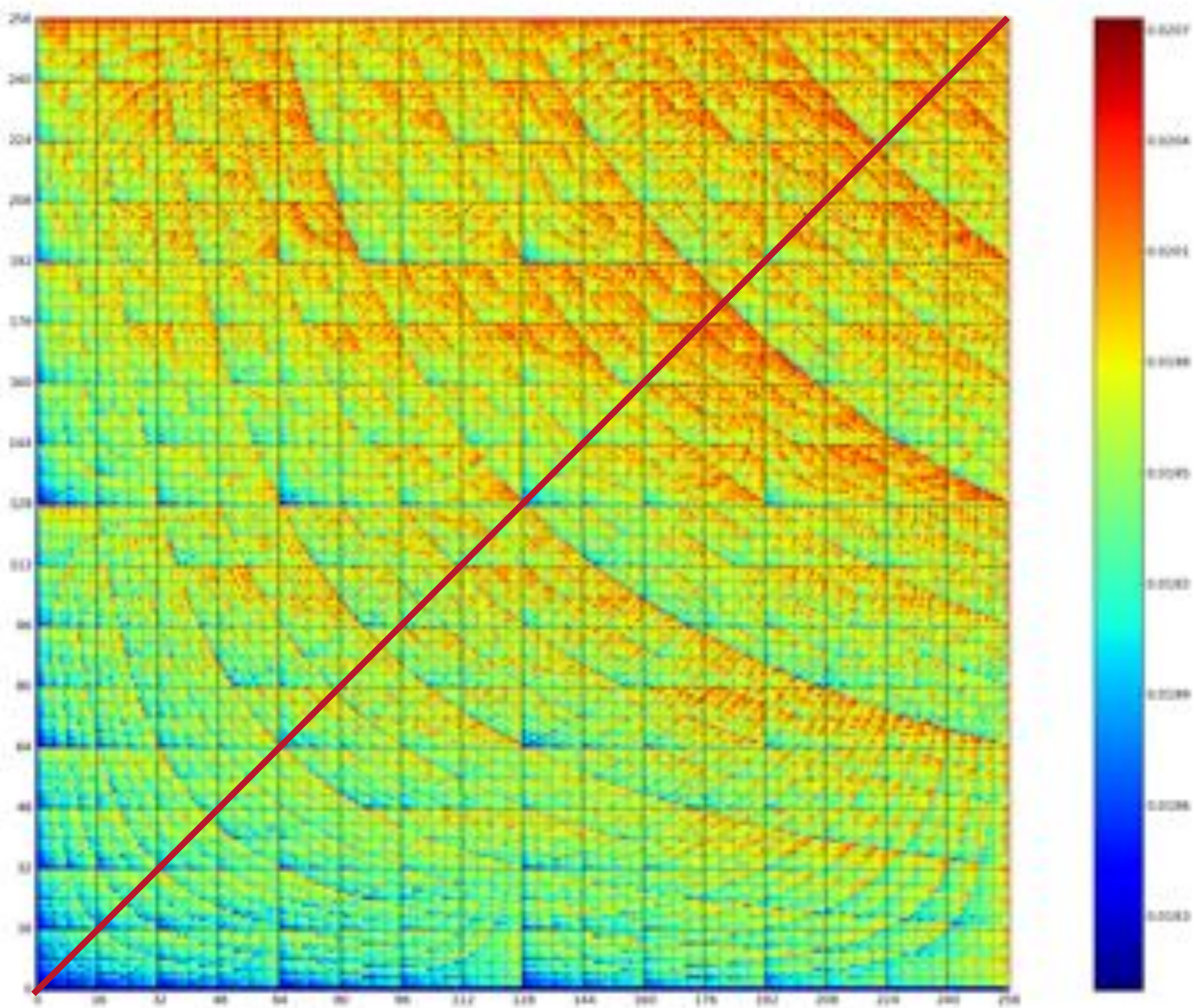




a*b = b*a

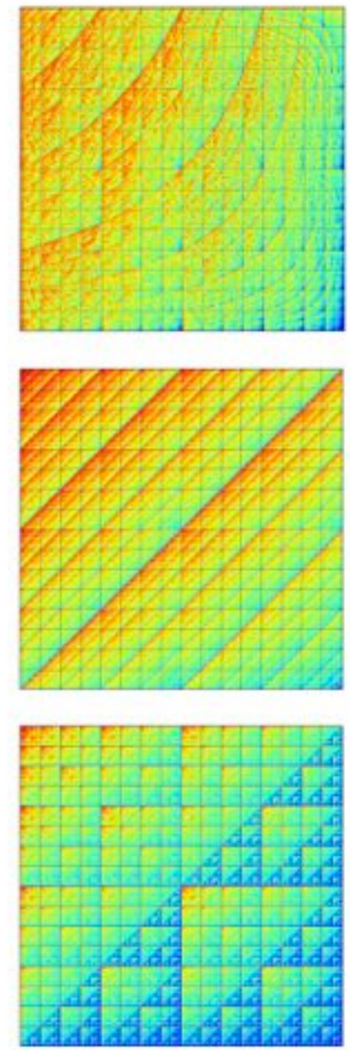


Energy(a*b) ≠ Energy(b*a)

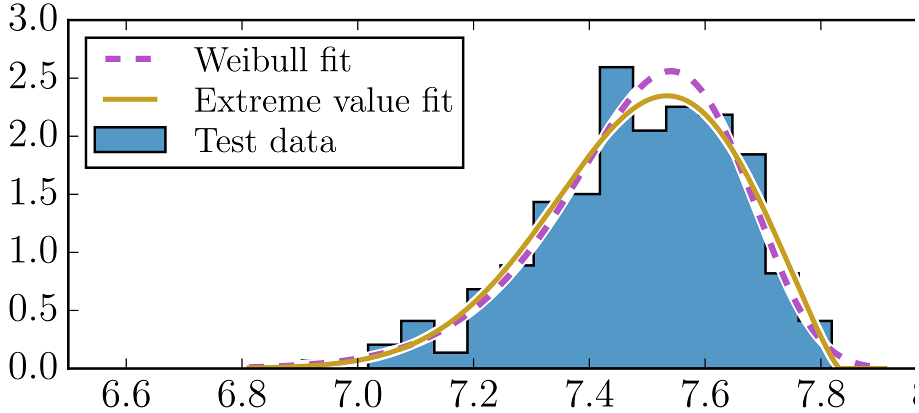


Dynamic Energy can be significant

- Data dependent switching costs can be large, ~30%
- 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.

ACM Reference format:
James Pallister, Steve Kerrison, Jeremy Morse, Kerstin Eder. 2017. Data Dependent Energy Modeling for Worst Case Energy Consumption Analysis. In *Proceedings of SCOPES '17, Sankt Goar, Germany, June 12-14, 2017,* 9 pages.
https://doi.org/10.1145/3078659.3078666

1 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 avail-

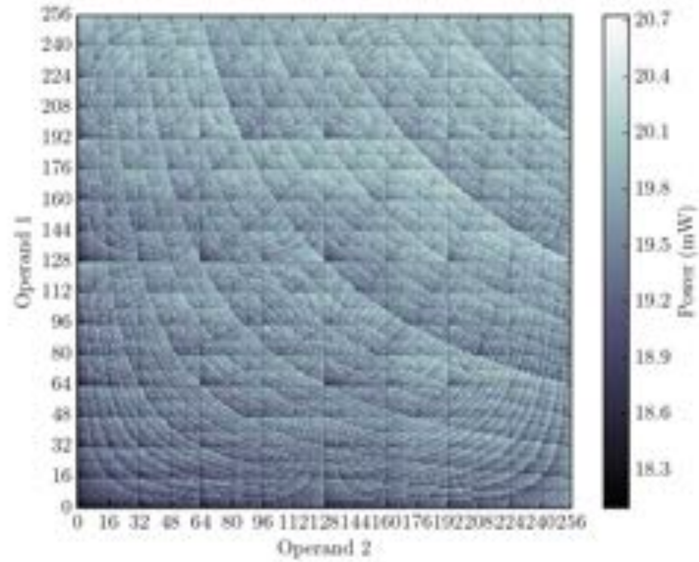


Figure 1: Power map of mul instruction for the AVR processor; total range is 15 % of SoC power.

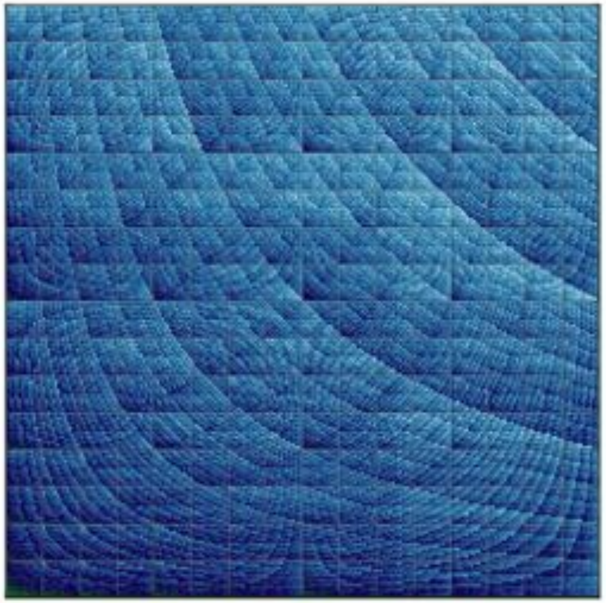
never under-estimates. Current models have not been analysed in this context to provide sufficient confidence, and power figures from manufacturer datasheets are not sufficiently detailed to provide tight bounds.

J. Pallister, S. Kerrison, J. Morse, and K. Eder. 2017. Data Dependent Energy Modeling for Worst Case Energy Consumption Analysis. In *Proceedings of the 20th International Workshop on Software and Compilers for Embedded Systems* (SCOPES '17), Sander Stuijk (Ed.). ACM, New York, NY, USA, 51-59. DOI: https://doi.org/10.1145/3078659.3078666

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?*



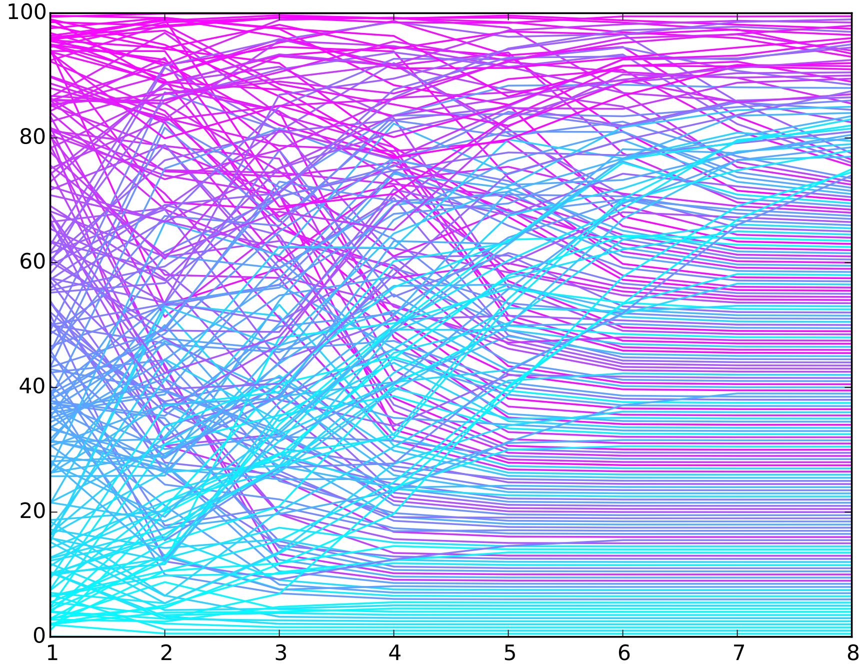
Energy of an Instruction Sequence

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

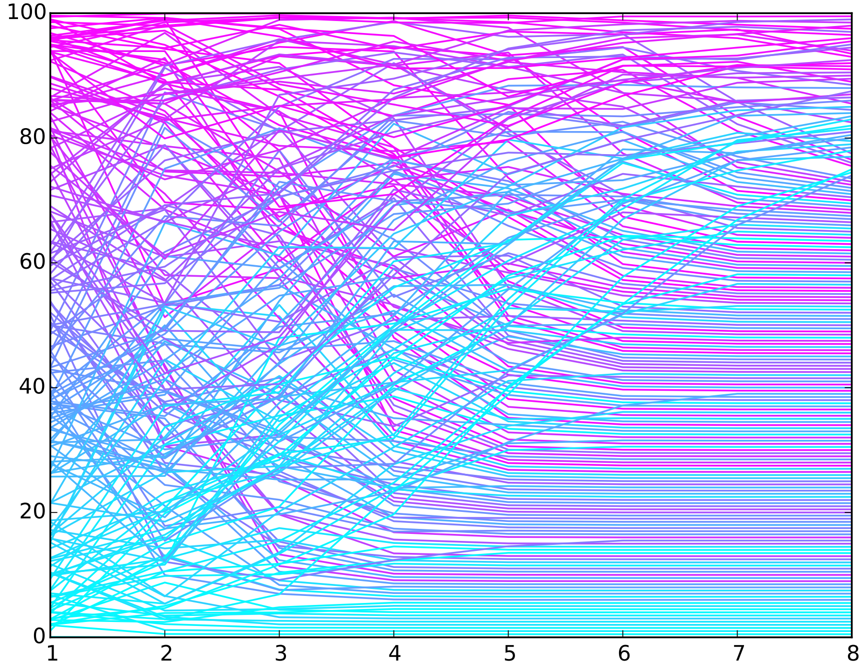


experiments conducted by James Pallister

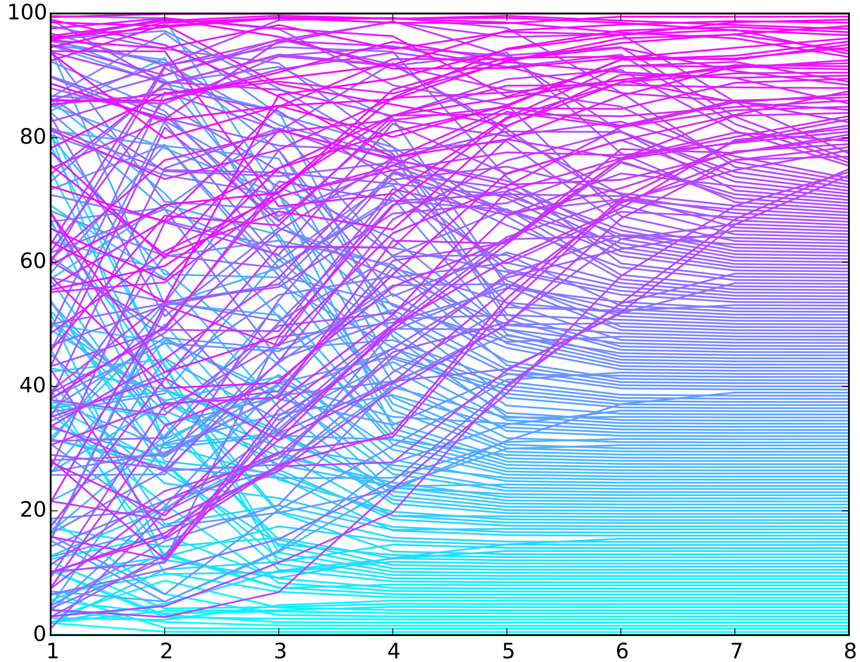
Energy of an Instruction Sequence

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

On the Limitations of Analyzing Worst-Case Dynamic Energy of Processing

JEREMY MORSE, STEVE KERRISON, and KERSTIN EDER, University of Bristol

This article examines dynamic energy consumption caused by data during software execution on deeply embedded microprocessors, which can be significant on some devices. In worst-case energy consumption analysis, energy models are used to find the most costly execution path. Taking each instruction's worst-case energy produces a safe but overly pessimistic upper bound. Algorithms for safe and tight bounds would be desirable. We show that finding exact worst-case energy is NP-hard, and that tight bounds cannot be approximated with guaranteed safety. We conclude that any energy model targeting tightness must either sacrifice safety or accept overapproximation proportional to data-dependent energy.

CCS Concepts: • **Hardware → Chip-level power issues**;

Additional Key Words and Phrases: Energy transparency, complexity, worst case energy consumption

ACM Reference format:
Jeremy Morse, Steve Kerrison, and Kerstin Eder. 2018. On the Limitations of Analyzing Worst-Case Dynamic Energy of Processing. *ACM Trans. Embed. Comput. Syst.* 17, 3, Article 59 (February 2018), 22 pages.
https://doi.org/10.1145/3173042

1 INTRODUCTION

A significant design constraint in the development of embedded systems is that of resource consumption. Software executed on embedded hardware typically has very limited memory and computing performance 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

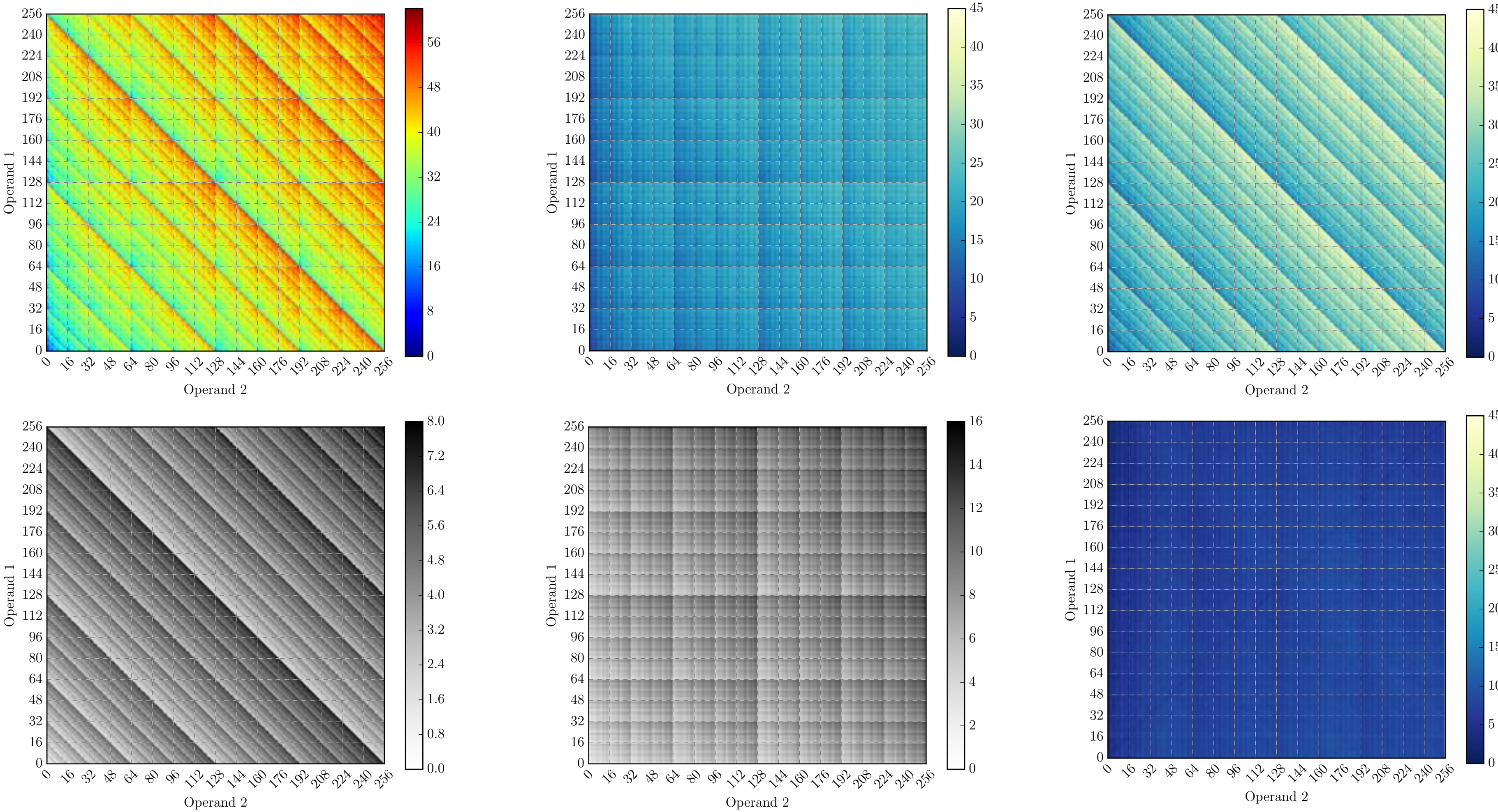
J. Morse, S. Kerrison, and K. Eder. 2018. On the Limitations of Analyzing Worst-Case Dynamic Energy of Processing. ACM Trans. Embed. Comput. Syst. 17, 3, Article 59 (February 2018), 22 pages. DOI: https://doi.org/10.1145/3173042

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. 2018. On the Limitations of Analyzing Worst-Case Dynamic Energy of Processing. ACM Trans. Embed. Comput. Syst. 17, 3, Article 59 (February 2018), 22 pages. DOI: https://doi.org/10.1145/3173042

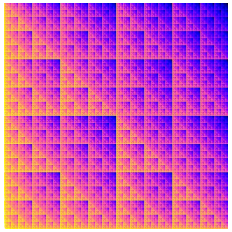
Impact of Datapath Switching



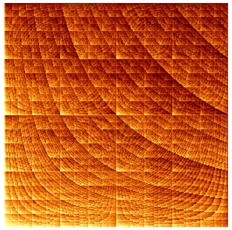
J. Morse, S. Kerrison, and K. Eder. 2018. On the Limitations of Analyzing Worst-Case Dynamic Energy of Processing. ACM Trans. Embed. Comput. Syst. 17, 3, Article 59 (February 2018), 22 pages. DOI: https://doi.org/10.1145/3173042

Summing up

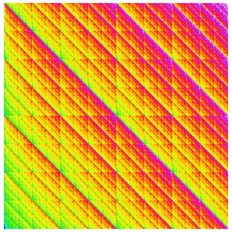
- To achieve *Energy Transparency*
  - Energy modelling is a challenge
    - Fundamental research questions
      - data-dependent energy models
      - compositional
      - probabilistic techniques



^



×



\+

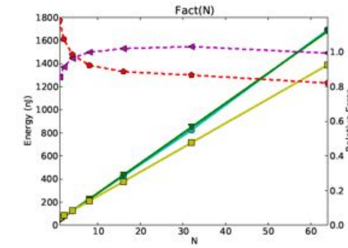
  - Analysis techniques for energy consumption
    - SRA works best for IoT-type systems
    - Hybrid, profiling-based techniques for more complex architectures

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
- ✓ Why timing and energy analysis differ

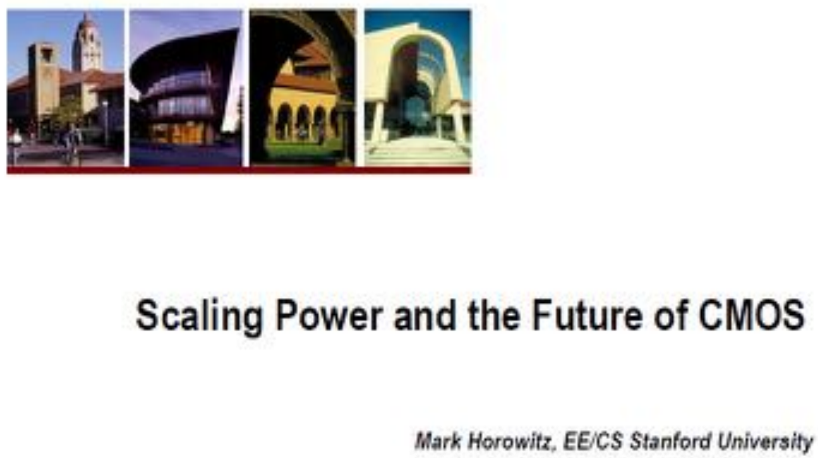
Towards Energy Aware Software Engineering

Energy Transparency

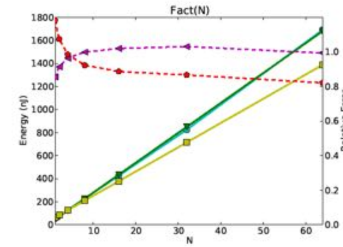


- For HW designers:
  "Power is a 1st and last order design constraint."
  [Dan Hutcheson, VLSI Research, Inc., E^{3}S Keynote 2011]

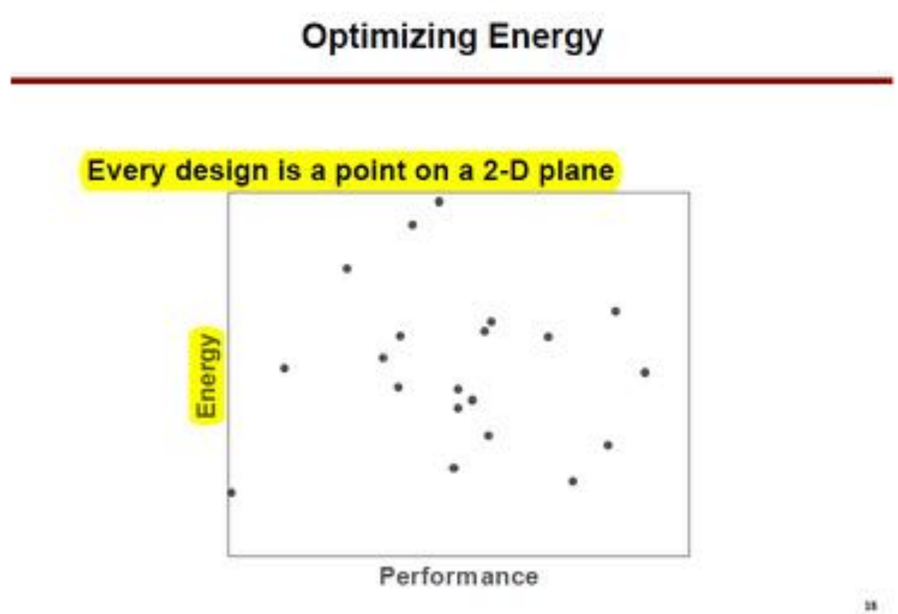
- "Every design is a point in a 2D plane."
  [Mark Horowitz,E^{3}S 2009]



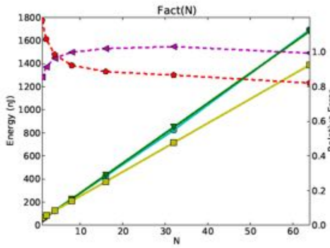
Energy Transparency



- 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]

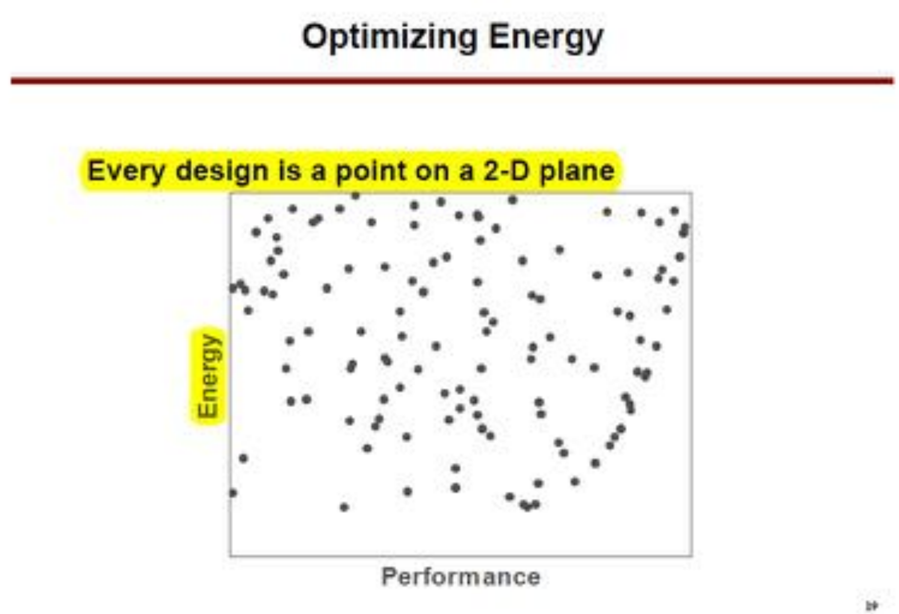


Energy Transparency

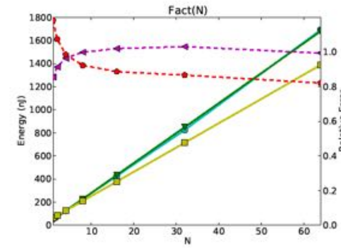


- For HW designers:
  “Power is a 1st and last order design constraint.”
  [Dan Hutcheson, VLSI Research, Inc., E³S Keynote 2011]

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

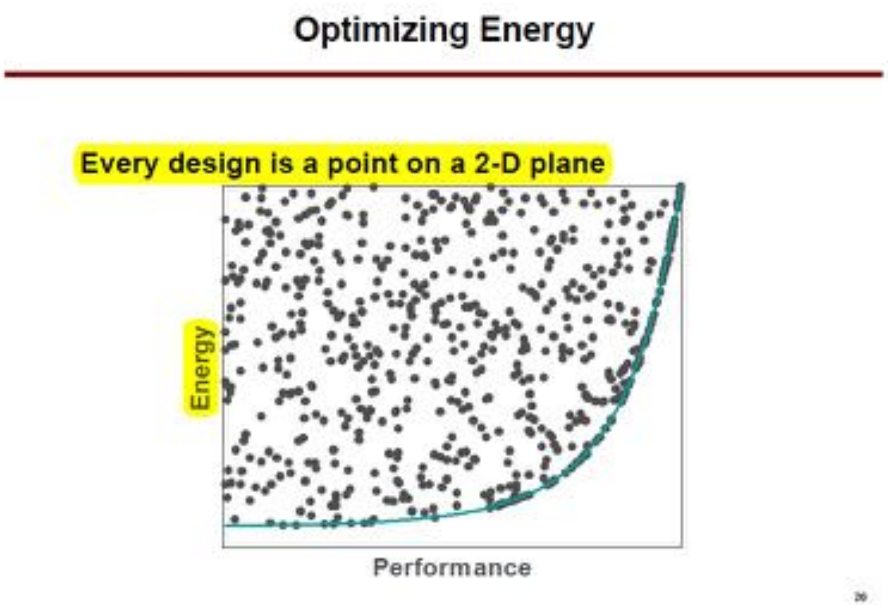


Energy Transparency



- 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

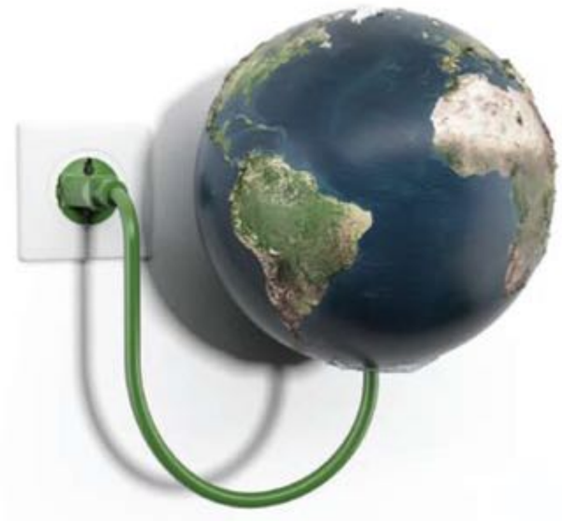
```
in 5pJ do {...}
```

- Full **Energy Transparency** from HW to SW
- New programming models

"Cool" code for green software

A cool programming competition!

Energy rated software



We aim to promote energy efficiency to a 1st class SW design goal!

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®

TEAMPLAY Time, Energy and security Analysis for Multi/Many-core heterogeneous PLAtforms

EMC²

ARTEMIS

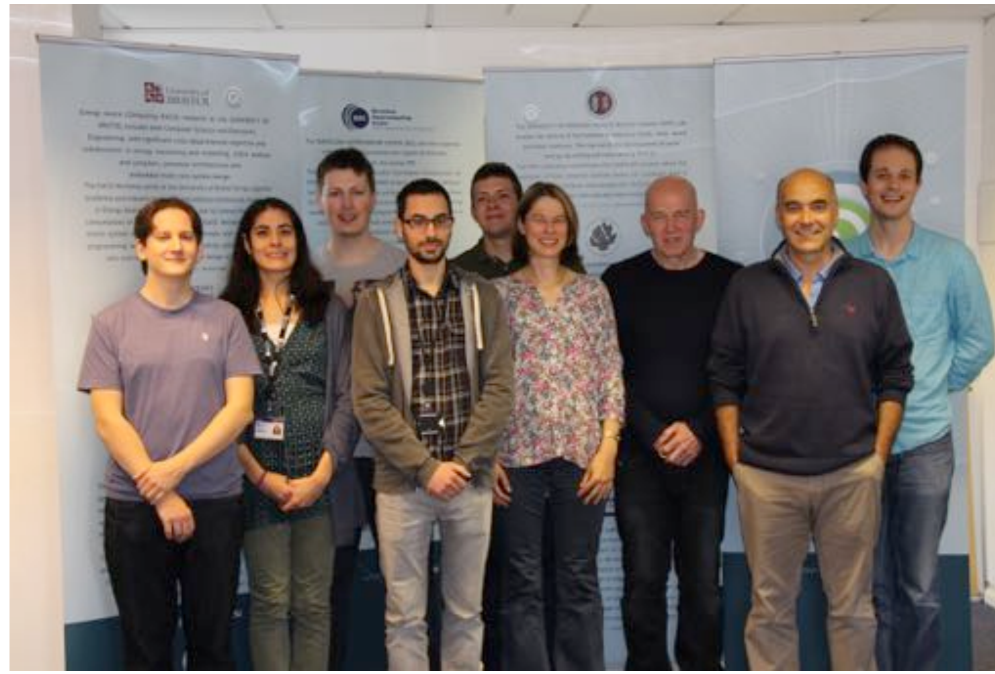
Innovate UK Technology Strategy Board

EPSRC Engineering and Physical Sciences Research Council

ICTenergy

XMOS®

The Royal Academy of Engineering



Kerstin.Eder@bristol.ac.uk