Energy-Aware System Design

(with a focus on software)

Kerstin Eder

Design Automation and Verification, Microelectronics Group

University of BRISTOL

Department of COMPUTER SCIENCE

The Royal Academy of Engineering

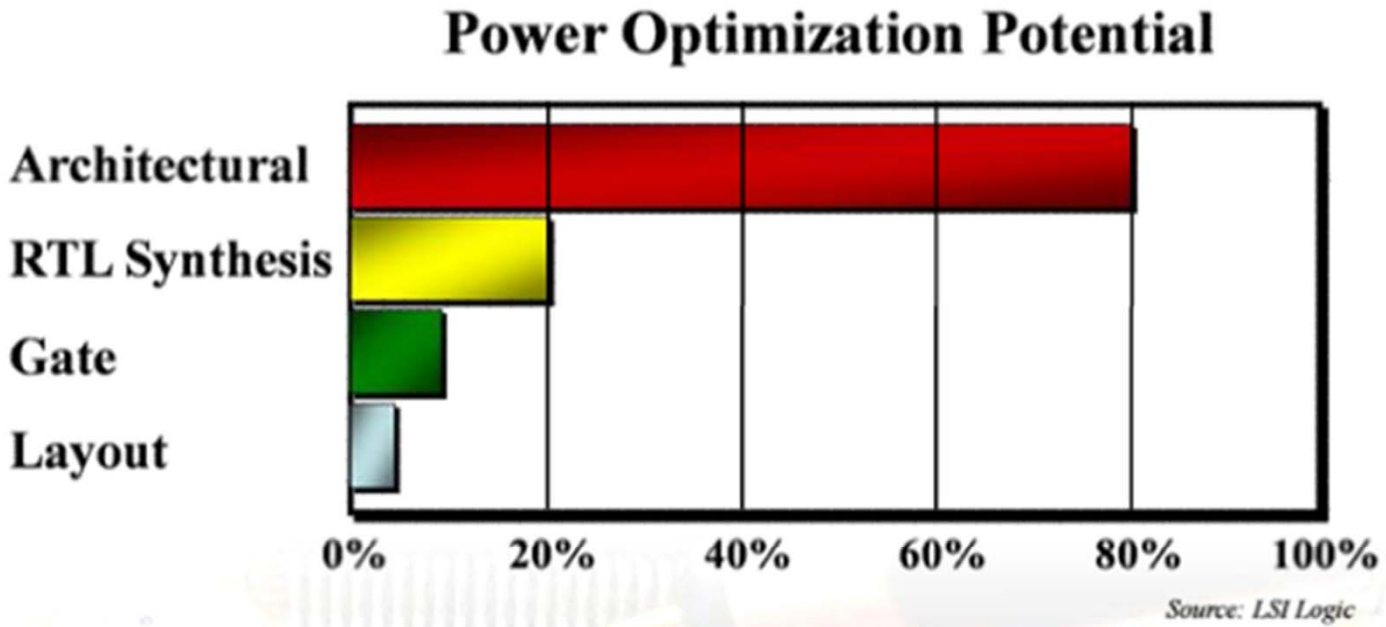
XMOS

“Energy-Aware System Design”

- Power management largely in domain of Hardware Design
  - Considerations to minimize/optimize
    - Dynamic (switching) and static (leakage) power
    - Energy consumption
  - On-chip power management
    - DVFS
    - Modes: on, standby, suspend, sleep, off
- Where can the greatest savings be made?

Greater Savings at Higher Levels

Why Optimize Power at the Architecture?



“with a focus on software”

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

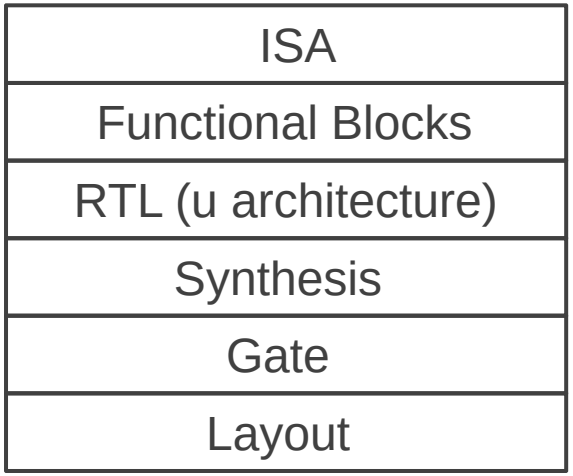
Aligning SW Design Decisions with Energy Efficiency Design Goal

Key steps*:

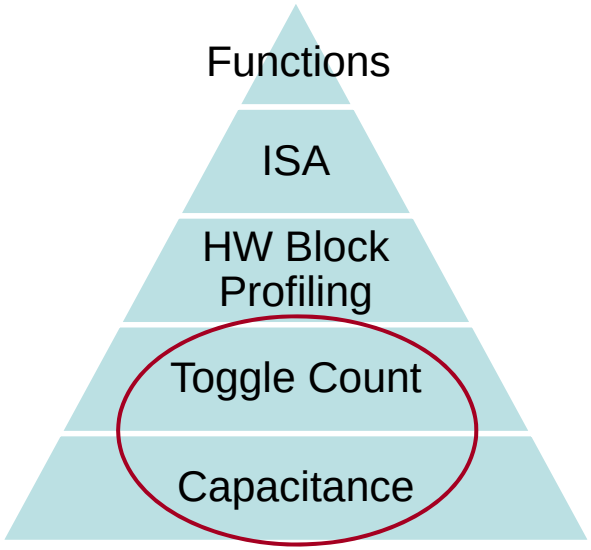
- *"Choose the **best algorithm** for the problem at hand and make sure it **fits well with** the computational **hardware**. Failure to do this can lead to costs far exceeding the benefit of more localized power optimizations.*
- *Minimize **memory size** and expensive **memory accesses** through algorithm 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.

The HW Design Stack



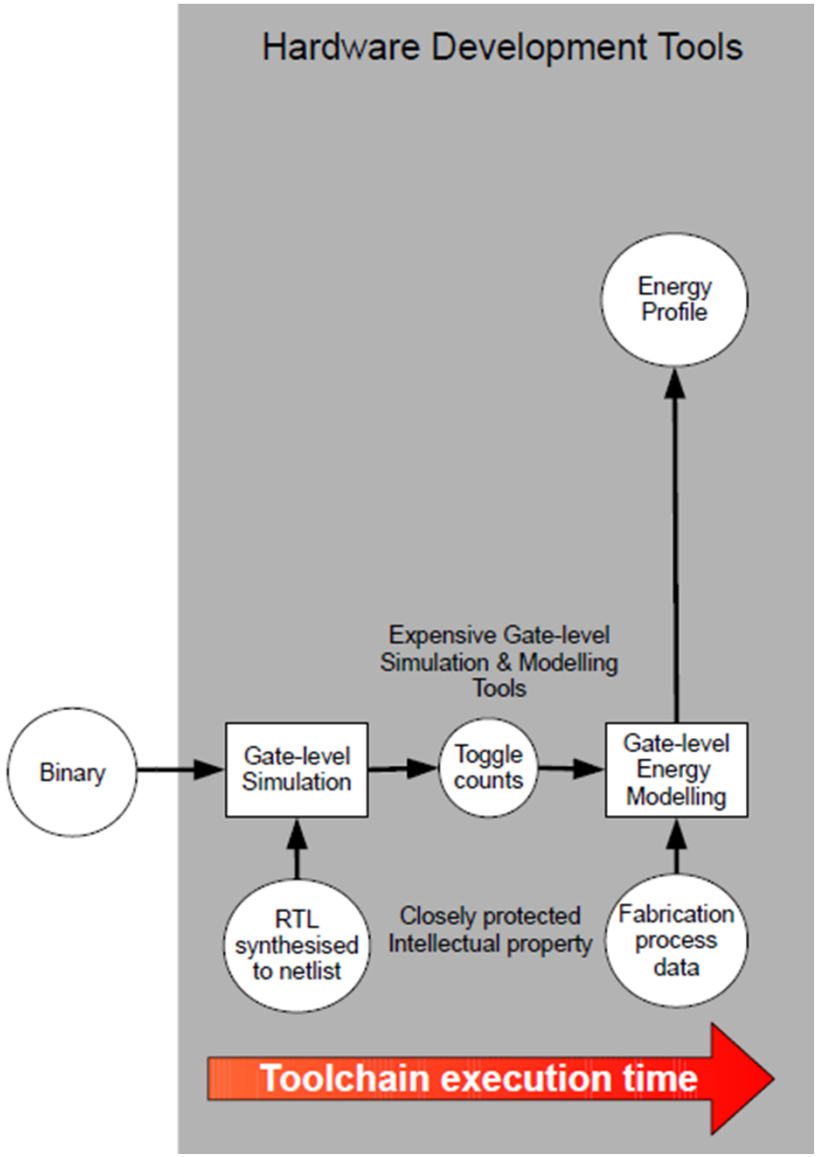
HW Power Analysis



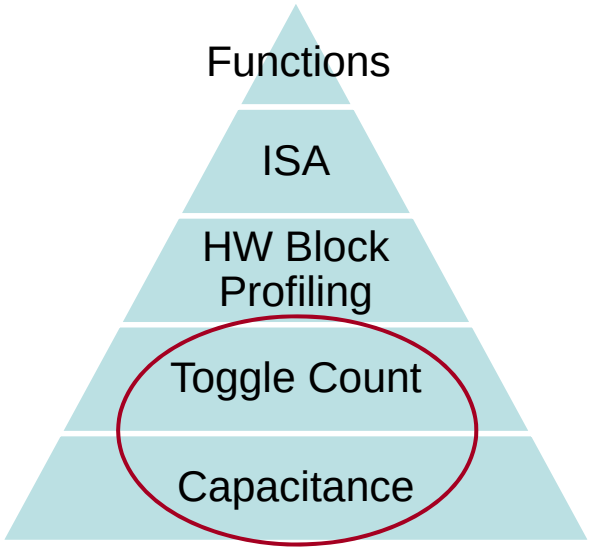
ISA
Functional Blocks
RTL (u architecture)
Synthesis
Gate
Layout

Hardware Power Analysis

- Late in the design flow
- Computationally expensive
- Slow



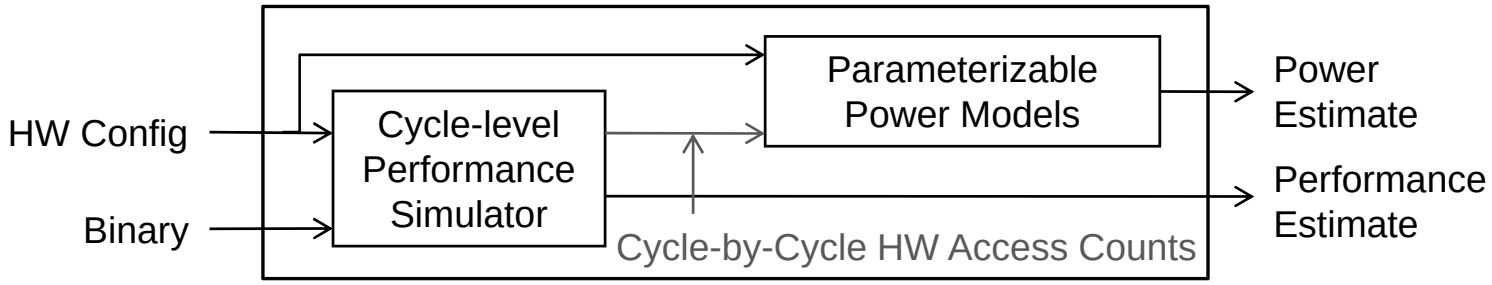
Towards Higher Abstraction Levels



ISA
Functional Blocks
RTL (u architecture)
Synthesis
Gate
Layout

Wattch*

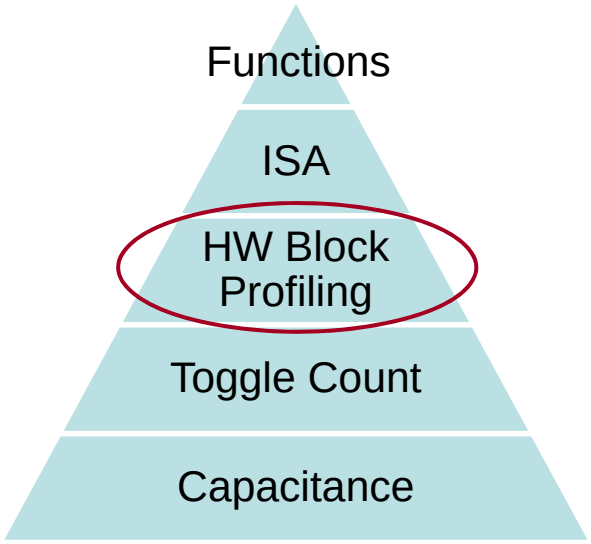
- Simulator for **architectural-level** power analysis
  - power/performance tradeoffs more visible
  - >1000x faster than layout-level power tools
  - accuracy within 10% of estimated values
  - based on SimpleScalar Tool Set



  - requires parametrized power models of common functional blocks (ALU, FPU, RF, I/D Cache) in superscalar processors

* D. Brooks, V. Tiwari, M. Martonosi, “Wattch: a framework for architecture-level power analysis and optimizations,” Proc. 27th International Symp. on Computer Architecture (ISCA), pp. 83-94, 2000.

Towards Higher Abstraction Levels



ISA
Functional Blocks
RTL (u architecture)
Synthesis
Gate
Layout

Instruction-Level Power Analysis*

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.

Approaches at ISA Level

- Low power instruction encoding+
  - Minimize the amount of switching by minimizing Hamming distance between neighbouring instructions
  - Requires profiling to optimize ISA for target applications
  - Up to 62% reduction in switching activity in opcodes

+ S. Woo, J. Yoon and J. Kim, “Low-Power Instruction Encoding Techniques”, Proceedings of the SOC Design Conference, 2001.

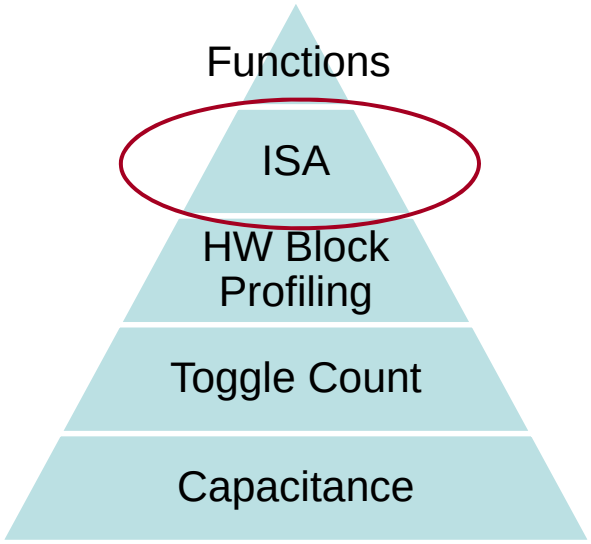
- Partitioning the Register File (RF)*
  - Observation 25% registers account for 83% of RF access time
  - Hot and cold RF regions motivate active and drowsy partitions
  - Requires profiling, recompilation and HW support
  - Average savings above 54% compared to nonpartitioned RF

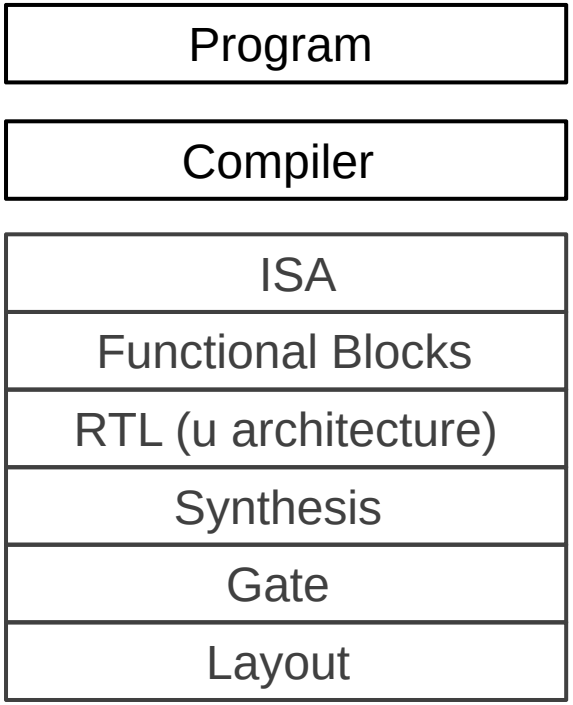
* X. Guan and Y. Fei, “Registeer File Partitioning and Recompilation for Register File Power Reduction”, ACM Transactions on Design Automation of Electronic Systems, 15(3)24, May 2010.

Towards Higher Abstraction Levels

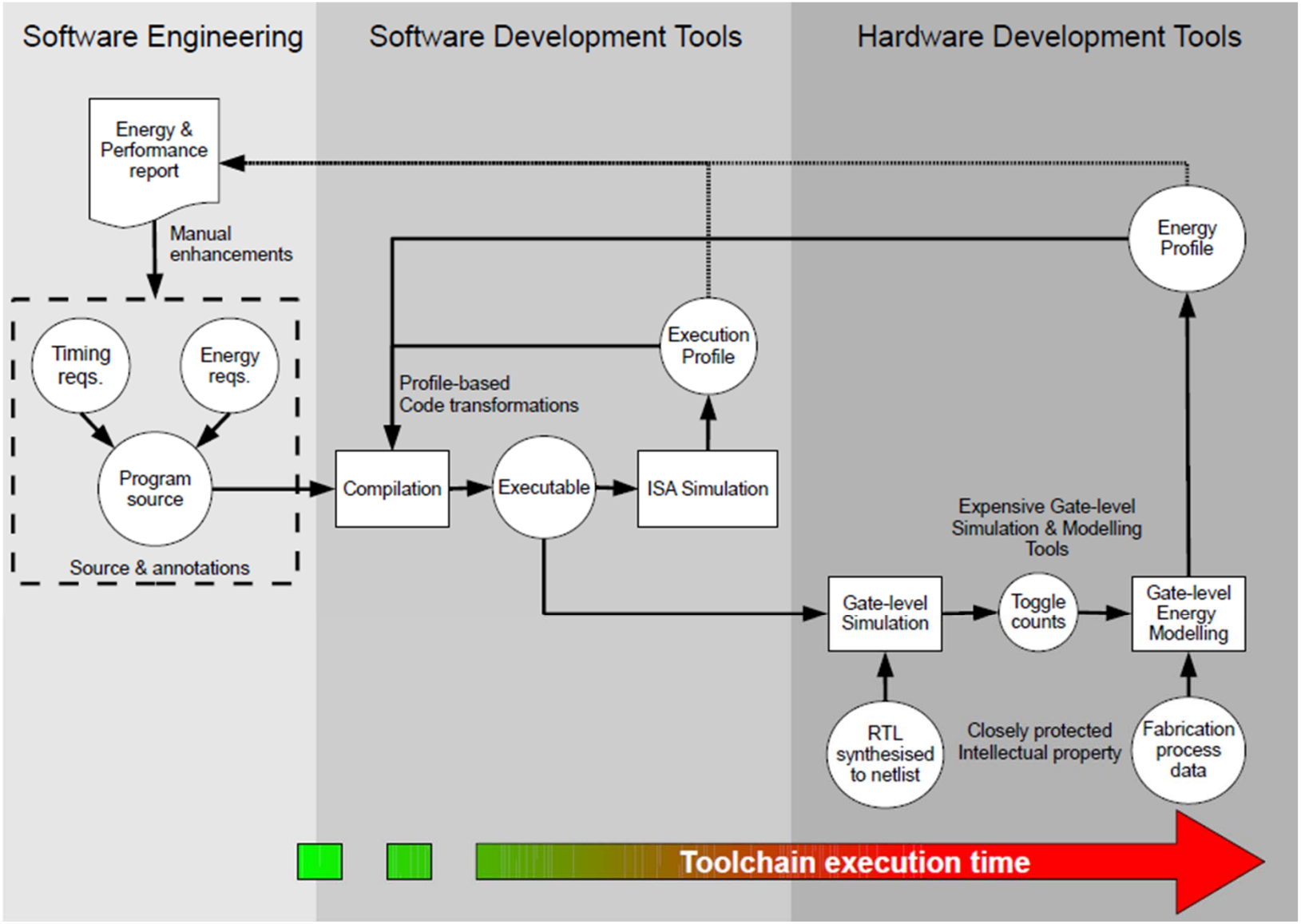
Expose HW properties to enable early system analysis and optimization

HW to SW

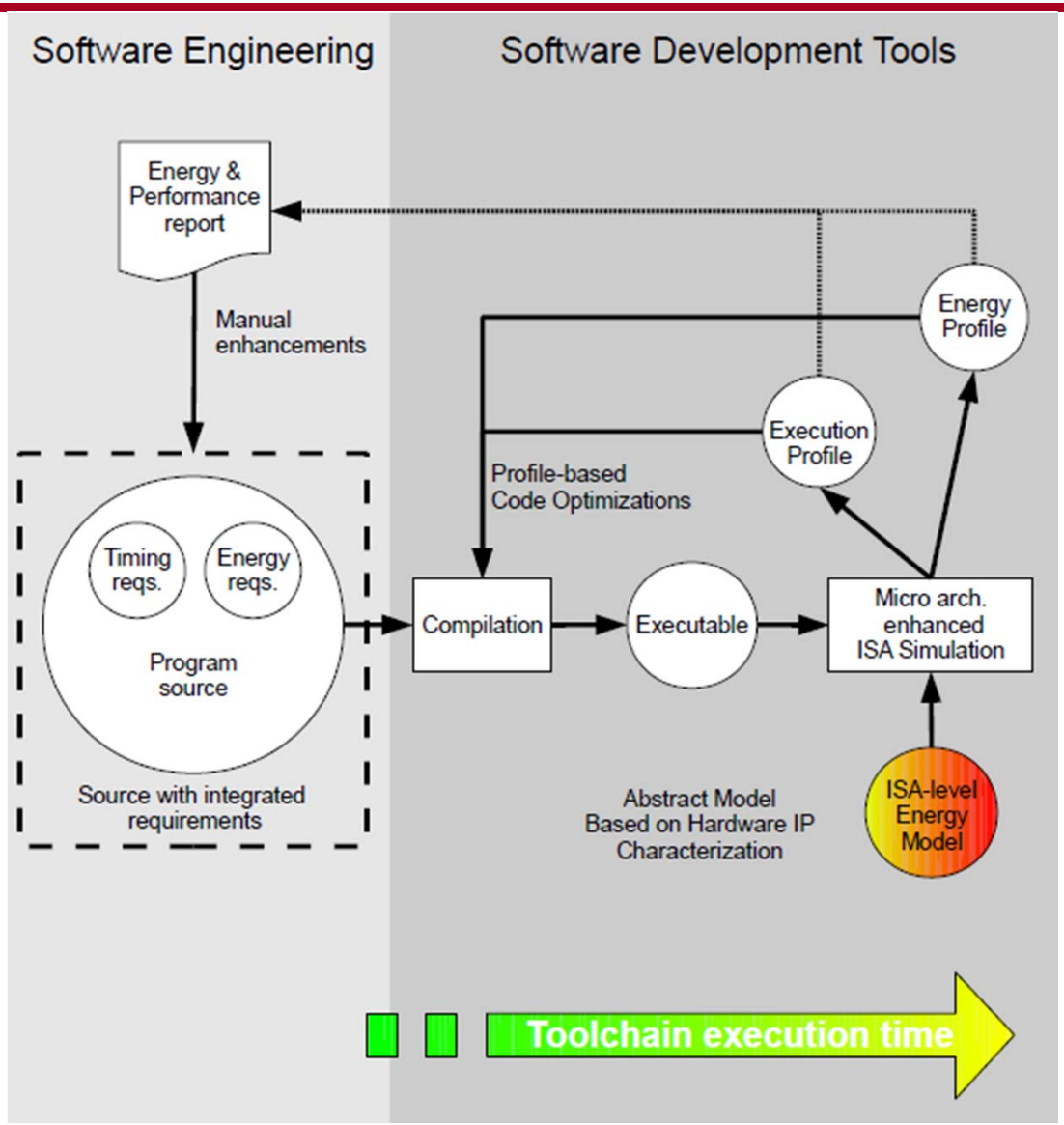




SW Energy Analysis



Shortening the “deep loop”



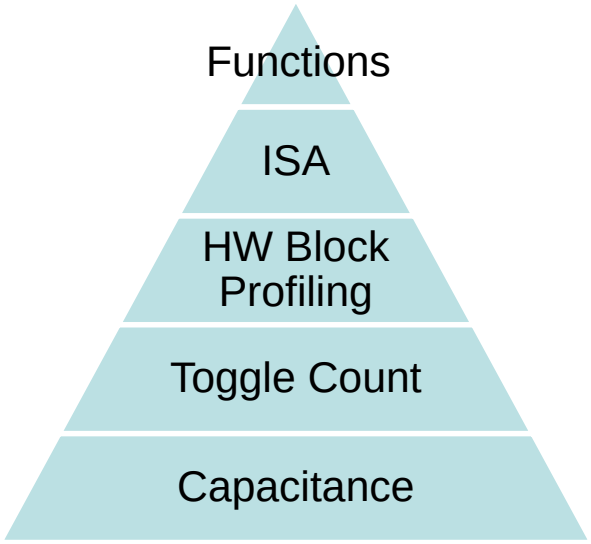
- Early in the design flow
- Computationally less expensive
- Fast
- Less accurate

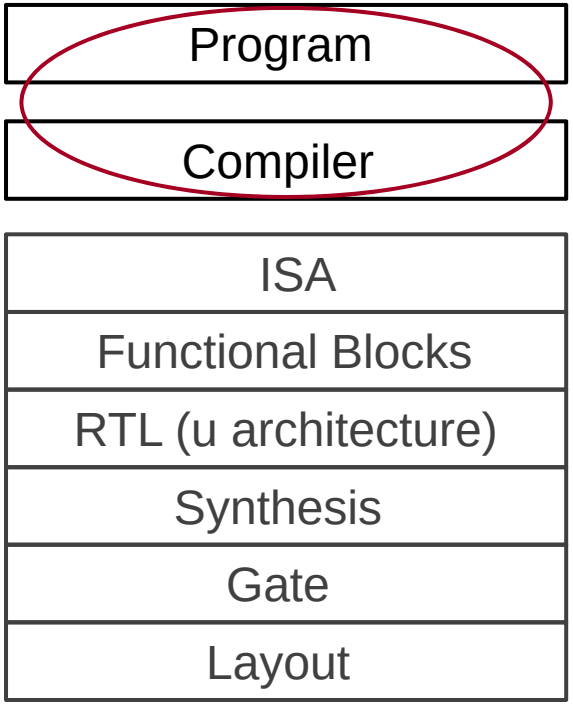
A Considerable Challenge

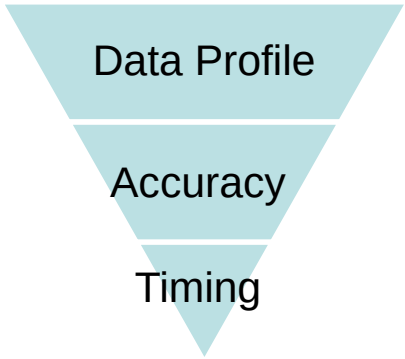
Towards Higher Abstraction Levels

Expose HW properties to enable early system analysis and optimization

HW to SW







Energy-Aware Software Design

- New languages give more control to programmers
  - Exploiting **energy/accuracy** tradeoffs at the SW level*
  - Type system supports “precise” or “approximate” data types
  - Provide support for “approximate” computation
  - Programmers annotate code
  - Type checker ensures separation of “precise” and “approx” code

* A. Sampson, W. Dietl, E. Fortuna, D. Gnanapragasam, L. Ceze and D. Grossman, “EnerJ: Approximate Data Types for Safe and General Low-Power Computation” In Proc. of PLDI, June 2011.

- HW to provide approximate storage/computations
  - Reduce FP power consumption by minimising data bit width+
  - Up to 66% reduction in energy/operation without loss of accuracy

\+ J. Y. F. Tong, D. Nagle and R.A. Rutenbar, “Reducing Power by Optimizing the Necessary Precision/Range of Floating-Point Arithmetic”, IEEE Transactions on VLSI Systems, 8(3), pp 273-286, June 2000.

Requires collaboration between SW and HW Engineers

More “**Power**” to SW Developers

- Requires taking **more responsibility** for
  - Algorithms (accuracy/fit to HW/encoding)
  - Data types (width/accuracy)
  - Budgets (time/power/energy)

in 15ms do {...}

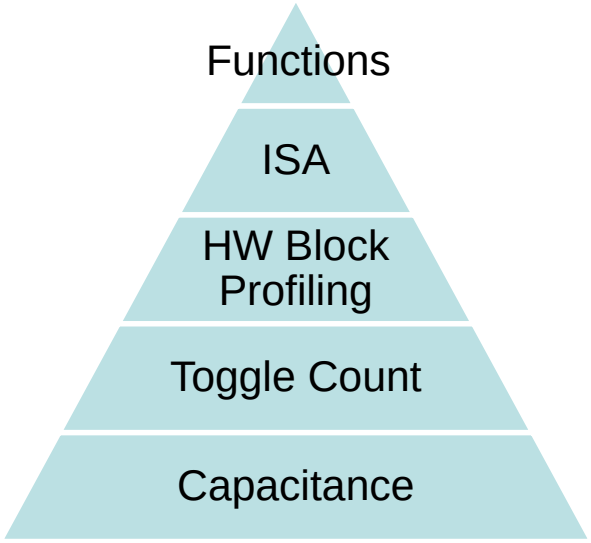
in 29000mJ do {...}

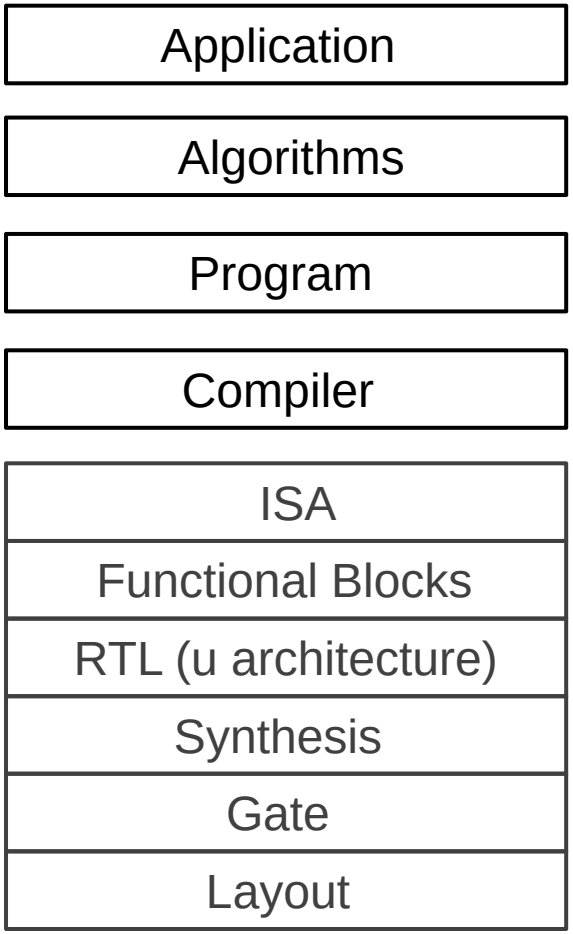
- **Toolchain** must enable early design space exploration
  - Communicate power/energy data from HW to SW developers
  - Determine impact on power/energy consumption
  - Develop intuition for power/energy budget of applications
  - Two stages:
    - Present energy consumption profile to software developer
    - Enable the compiler and “ee” optimizer to use this data
      - Complement static analysis with dynamic profiling

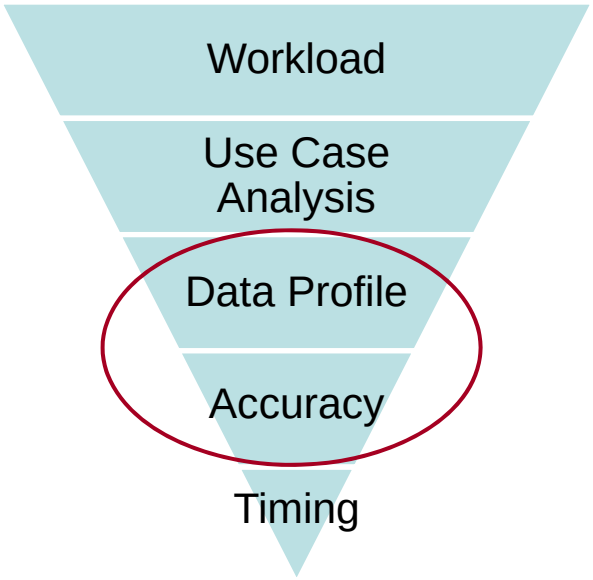
Towards Higher Abstraction Levels

Expose HW properties to enable early system analysis and optimization

HW to SW



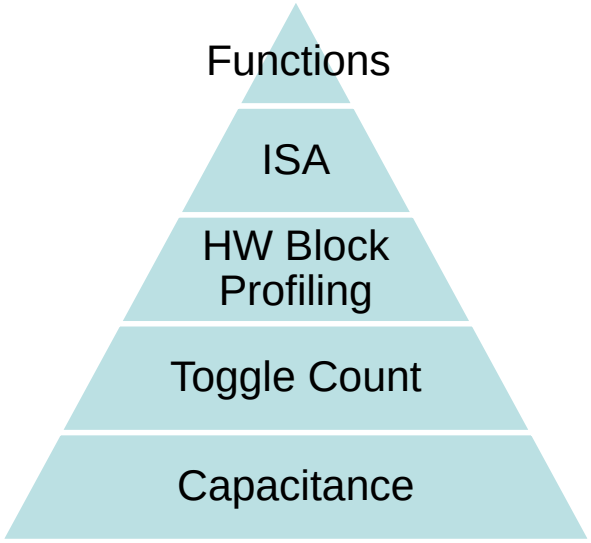


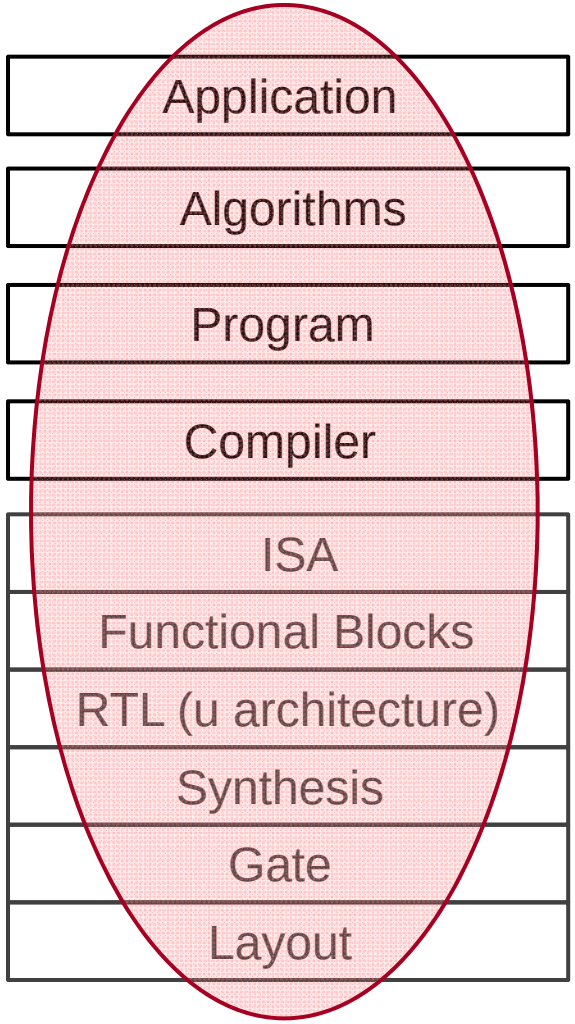


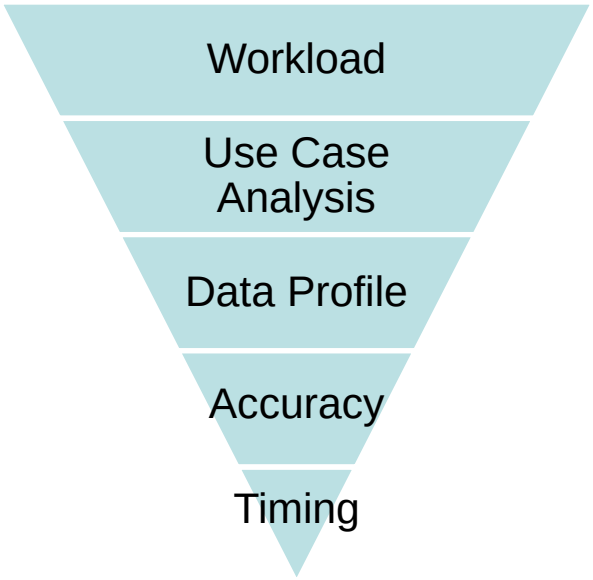
A Holistic “Systems” Approach

Expose HW properties to enable early system analysis and optimization

HW to SW







SW to HW

Design system tailored to end user needs

Energy Efficient SW

Promoting Energy Efficiency to a 1st Class System Design Goal

Challenges:

- Shortening the “deep loop” in the **Toolchain**
  - early estimation/optimization of time/power/energy
- Re-evaluation of SOA for **multi-threading and multi-core**
- More **“power” to SW engineers** and programmers
  - expressive power of languages
  - resource budgets (WCET, power, energy)

  The **Energy efficient Software Challenge**
- Increasing impact of static (leakage) power motivates **programming paradigm shift**
  - from always on “Busy Waiting” style
  - to default off “Event-driven” programming
- **Education** and **Collaboration**

LOW POWER

Lack of software support marks the low power scorecard at DAC

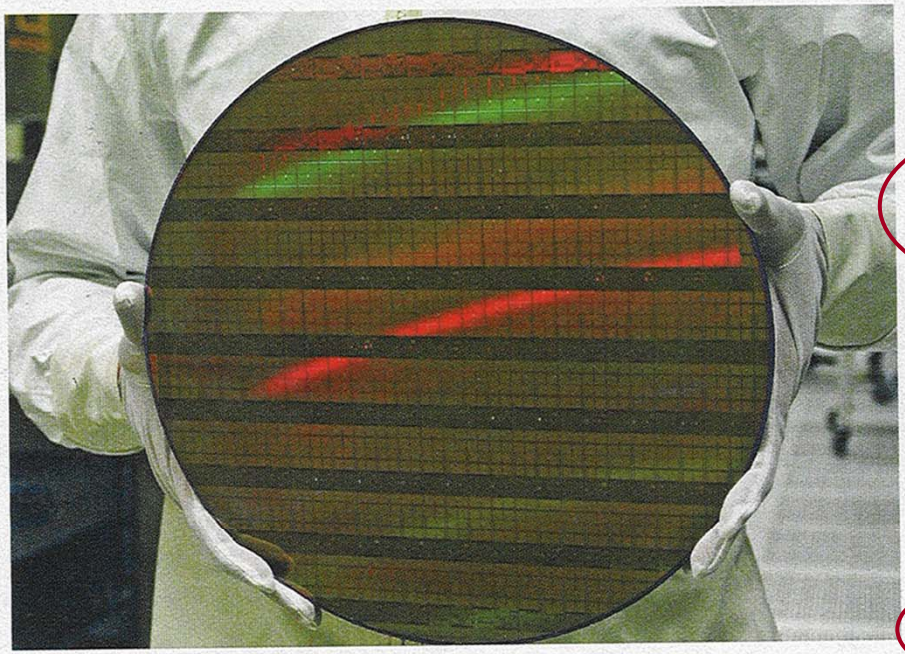
One of the panels at the Design Automation Conference (DAC), which took place in California in early June, set out to get an idea of how well the industry is doing at delivering lower-power systems.

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



Intel waits for better low-power software control

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

During an afternoon panel discussion 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

www.electronicsweekly.com/ew-blogs/

Energy Aware COmputing

- New initiative at Bristol
- Kick off with 3 dedicated EACO workshops
  - http://www.bristol.ac.uk/ias/workshops/current-workshops/energy-aware-computing.html
  - Sponsored by the IAS

- Intellectual Challenges
  - Incremental improvements
  - Radically new innovative approaches
- Interested collaborators please contact: Kerstin.Eder@bristol.ac.uk

Thank you

Any questions?

Acknowledgements: I gratefully acknowledge the feedback and contributions towards this presentation from the Microelectronics Research Group and from the Engineers at XMOS. The Royal Academy of Engineering are funding my secondment to XMOS via the Industrial Secondment Scheme.

The Royal Academy of Engineering

XMOS®