Energy-Aware Resource Management in Heterogeneous Computing Systems

Part 3: Heterogeneous Hardware

Sven Köhler, Max Plauth, Lukas Wenzel, Prof. Andreas Polze
Operating Systems and Middleware Group
Hasso Plattner Institute

Energy-Aware Computing Team at the Professorship for Operating Systems and Middleware


Sven Köhler


Max Plauth


Lukas Wenzel


Prof. Andreas Polze

We try to understand energy demand as a critical operating resource of computer systems, under the consideration of hardware aspects.

osm.hpi.de/energy

Motivation
How to Influence Power Consumption?

How to control power consumption of computing lab operation?

1. **Work differently:** Use fewer/additional compute resources
   (commonly used to scale depending on operating requirements)

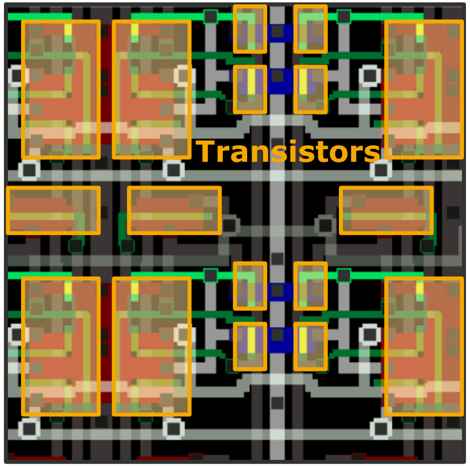
2. **Work another time:** Defer/pick jobs with matching energy profile
   (popular research in embedded systems, requires known and deferrable jobs)

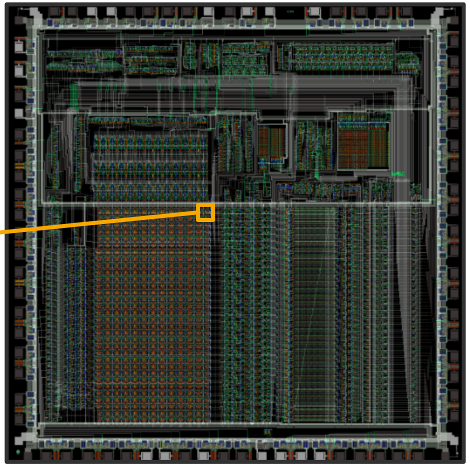
3. **Work elsewhere:** Use other hardware components

Focus of our research at the OSM group

How Compute Hardware Consumes Energy...

*All kinds of **compute hardware** are digital circuits **comprised of transistors**.*





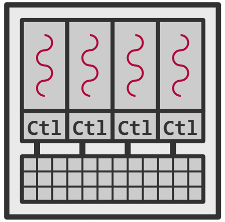
ARM 1 Chip from 1985 © Visual6502.org

***Power is consumed** dynamically **when transistors switch** on or off, and statically due to **leakage currents**.*

$$P \sim U^2 \cdot (C_T \cdot f + G_T) \cdot \boxed{N}$$

Heterogeneous Compute Hardware

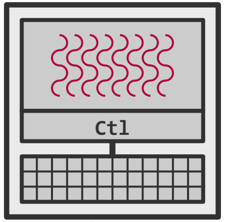
Choosing a suitable type of compute hardware for a workload can reduce its energy consumption.



CPU

Fixed Hardware

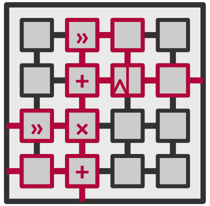
(MIMD)



GPU

Fixed Hardware

(SIMD)



FPGA

Configurable Hardware

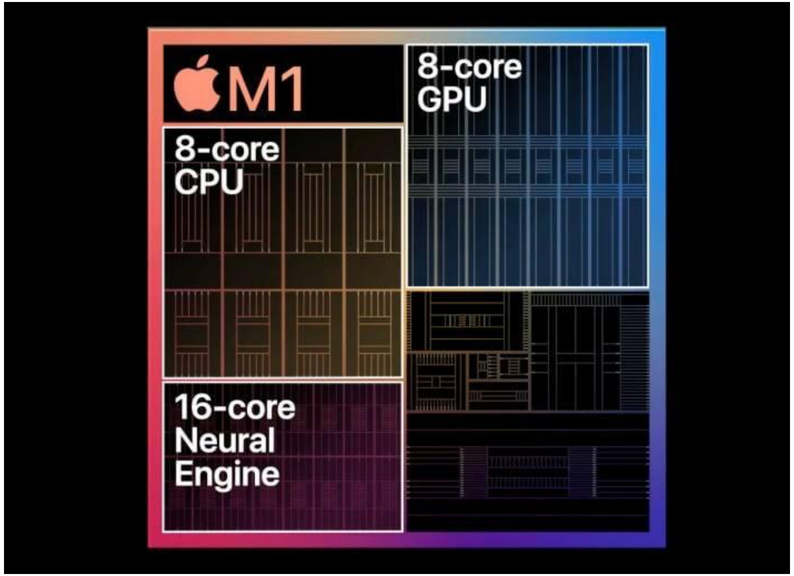
...

Special-Purpose Accelerators

- Cryptography
- Compression
- Neural Networks

Heterogeneous Compute Hardware

Choosing a suitable type of compute hardware for a workload can reduce its energy consumption.



... or any combination of these!

= Heterogeneous System

Energy-Aware Resource Management

Lukas Wenzel,
OSM Group

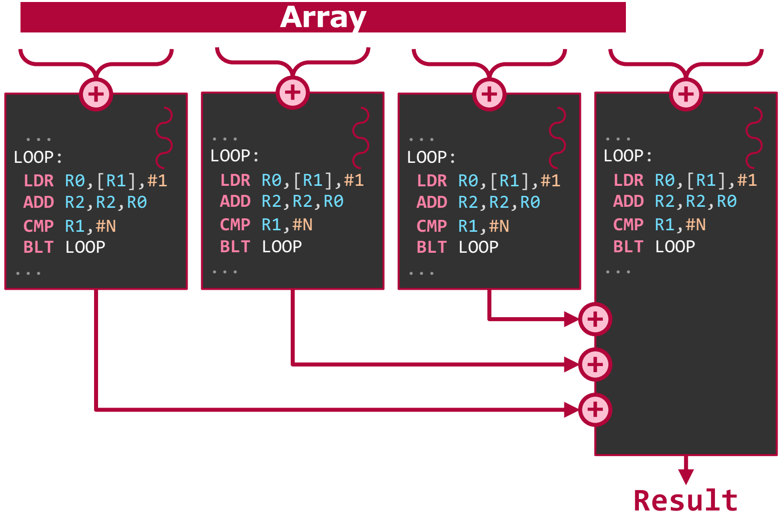
Example
Array Sum on a CPU and GPU

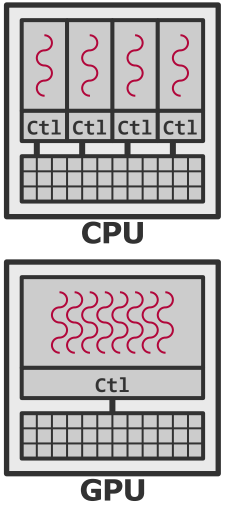
Workload: *Accumulate all values in a contiguous array*

```
int array[N];
int sum = 0;

for (int i = 0; i < N; ++i)
  sum += array[i];
```

- Partition Array according to available execution threads
- Accumulate local sums
- Compute global sum from local sums





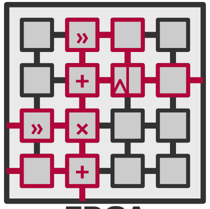
Example
Array Sum on an FPGA

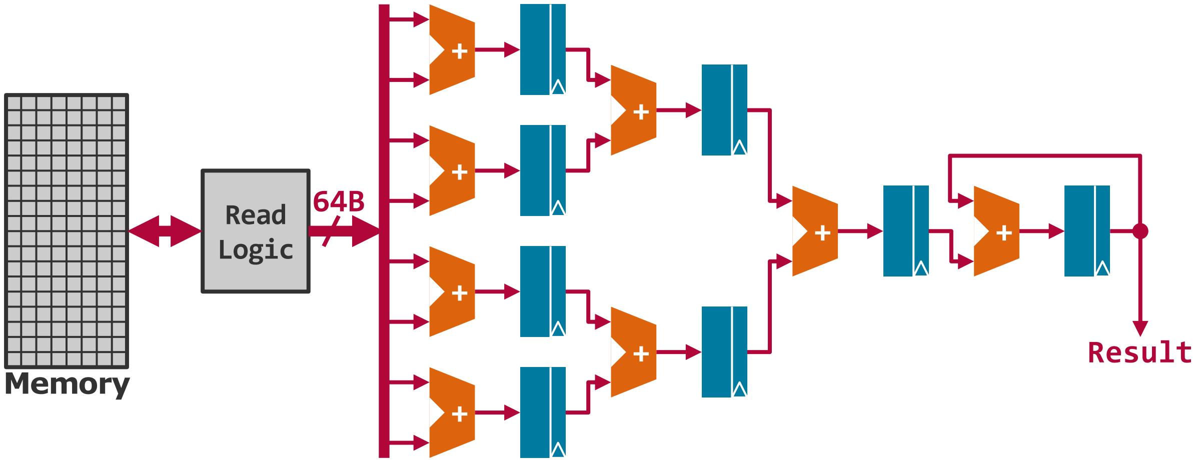
Workload: ***Accumulate all values in a contiguous array***

```
int array[N];
int sum = 0;

for (int i = 0; i < N; ++i)
  sum += array[i];
```

- Hardware memory interfaces usually supply multiple `int`s per transfer
- Reduction network and accumulator implemented directly in hardware
  - ➢ No instruction decoding / scheduling overhead

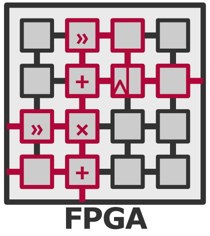


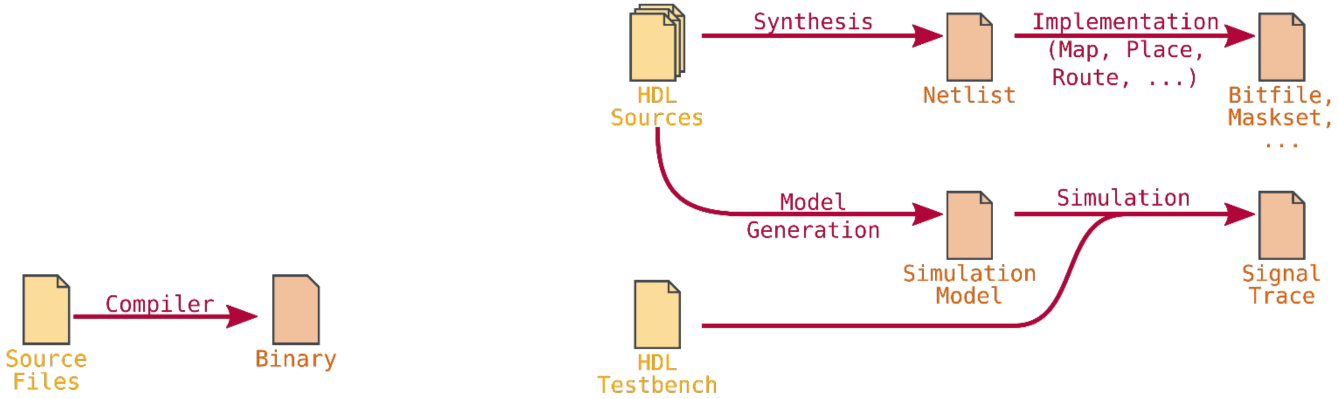


FPGA Hardware Design Process

FPGA development requires a **skillset not** usually **found among software developers**:

- Proficiency in a hardware development language and associated workflows
- Understanding of hardware design tradeoffs
- Experience with common design patterns and idioms in hardware architecture



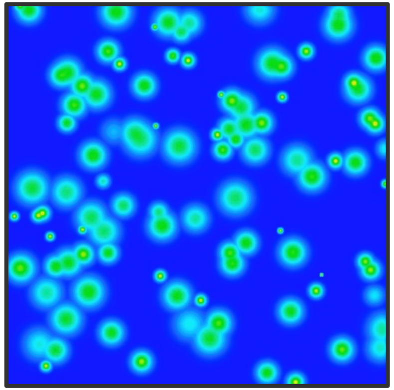


Software Development vs. Hardware Development Workflow

Case Study
Heatmap Simulation

Workload: ***Heatmap Simulation (continuously applied convolution filter)***

Device	Size (MCells)	Type	Throughput (MCells/s)		Power Consumption (W)	Energy Efficiency (MCells/Ws)	
NVidia Tesla K20Xm	256	float (32bit)	1127,88	**×1.0**	60	18,80	**×1.0**
Intel E5-2630 v4	256	float (32bit)	1435,48	**×1.3**	85	16,89	**×0.9**
Intel E5-2630 v4	256	char (8bit)	1420,29	**×1.3**	85	16,71	**×0.9**
Xilinx XCKU060	1024	char (8bit)	2209,35	**×2.0**	9,5	232,56	**×12**



- FPGAs can save significant **logic overhead** w.r.t. general purpose hardware
- FPGAs operate at about one **order of magnitude lower clock frequencies** than ASICs

$$P \sim U^2 \cdot (C_T \cdot f + G_T) \cdot N$$

> **Superior power efficiency** compared to equivalent implementations on general purpose hardware

Conclusion

- **FPGAs** are **configurable devices** to realize custom execution logic
  - ➢ **Custom logic** can **save overhead** w.r.t. general-purpose devices

- **Hardware design** is a significantly **more complex** / costly process than software development

FPGAs are an important building block in energy-efficient heterogeneous systems