Embedded System Hardware

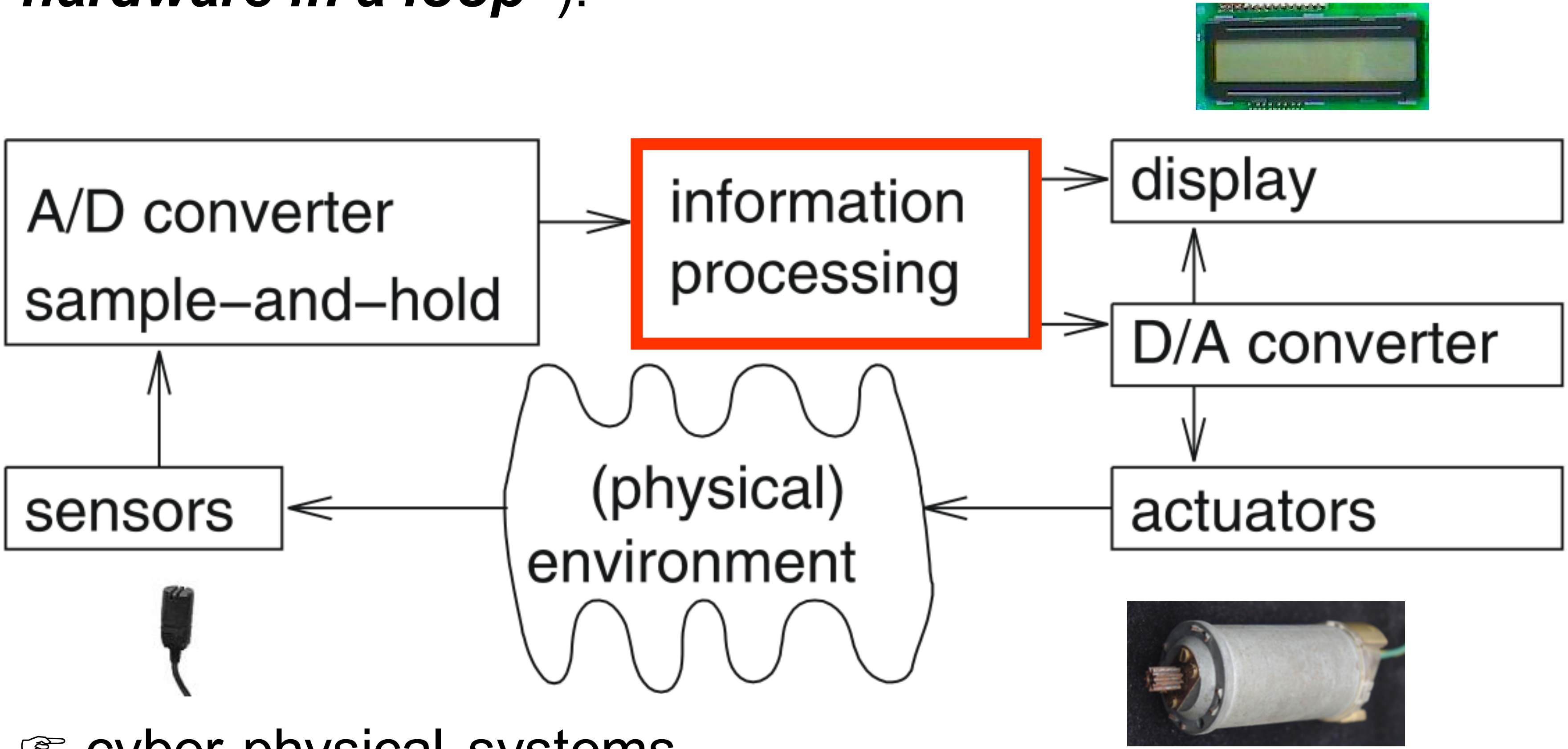
- Processing -

Jian-Jia Chen
(Slides are based on
Peter Marwedel)
Informatik 12
TU Dortmund
Germany

2019年 11 月 06 日

Embedded System Hardware

Embedded system hardware is frequently used in a loop (***"hardware in a loop"***):



☞ cyber-physical systems

Not So Seriously

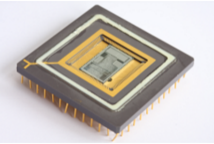
- Dr. Michio Kaku's states in one of his recent books: *"Today, your cell phone has more computer* power *than all of NASA back in 1969, when it placed two astronauts on the moon."*

- Seems hard to believe, we know, but it is actually true – a hand-held apparatus on which we fling birds at pigs has greater computational capabilities than the arsenal of machines used for guiding crafts through outer space some 45 years ago.

http://www.phonearena.com

Efficiency: slide from lecture 1 applied to processing

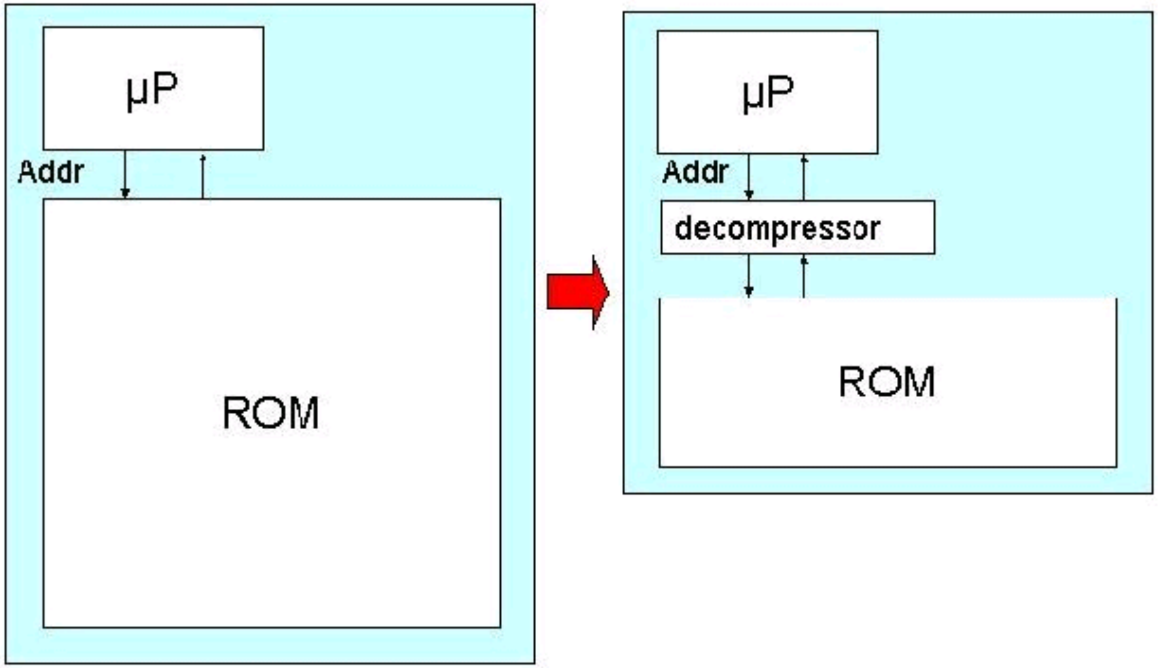
- CPS & ES must be **efficient**
  - Code-size efficient (especially for systems on a chip)
  - Run-time efficient
  - Weight efficient
  - Cost efficient
  - Energy/power efficient



© Graphics: Microsoft, P. Marwedel, M. Engel, 2011

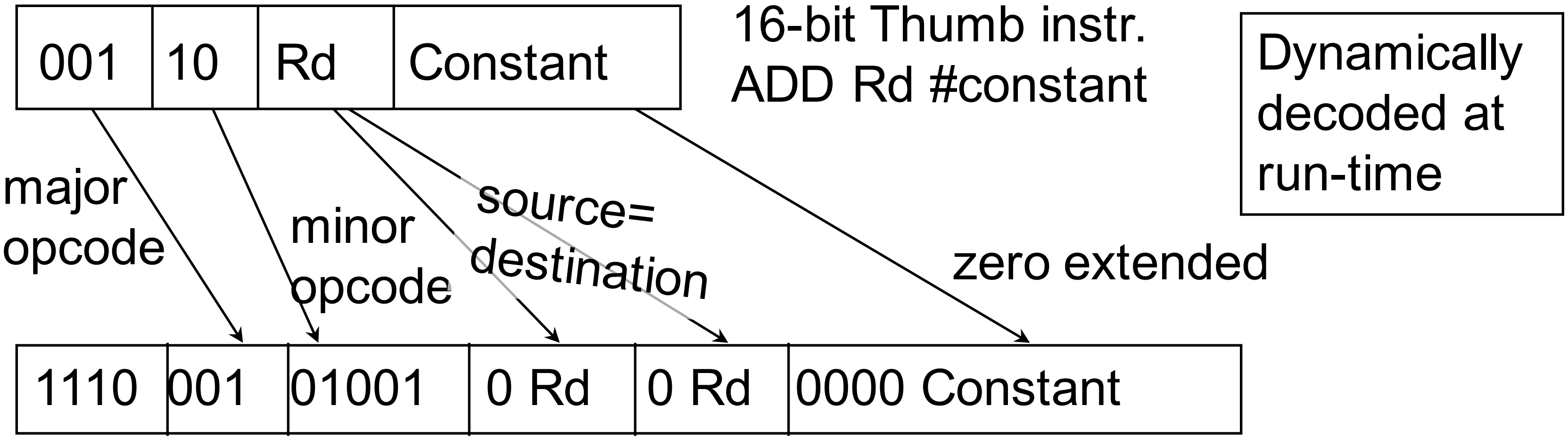
Key requirement #1: Code-size efficiency

- Overview: http://www-perso.iro.umontreal.ca/~latendre/codeCompression/codeCompression/node1.html
- **Compression techniques:** key idea



Code-size efficiency

- **Compression techniques (continued):**
  - 2nd instruction set, e.g. ARM Thumb instruction set:



  - Reduction to 65-70 % of original code size
  - 130% of ARM performance with 8/16 bit memory
  - 85% of ARM performance with 32-bit memory

Same approach for LSI TinyRisc, …
Requires support by compiler, assembler etc.

[ARM, R. Gupta]

Dictionary approach, two level control store

(indirect addressing of instructions)

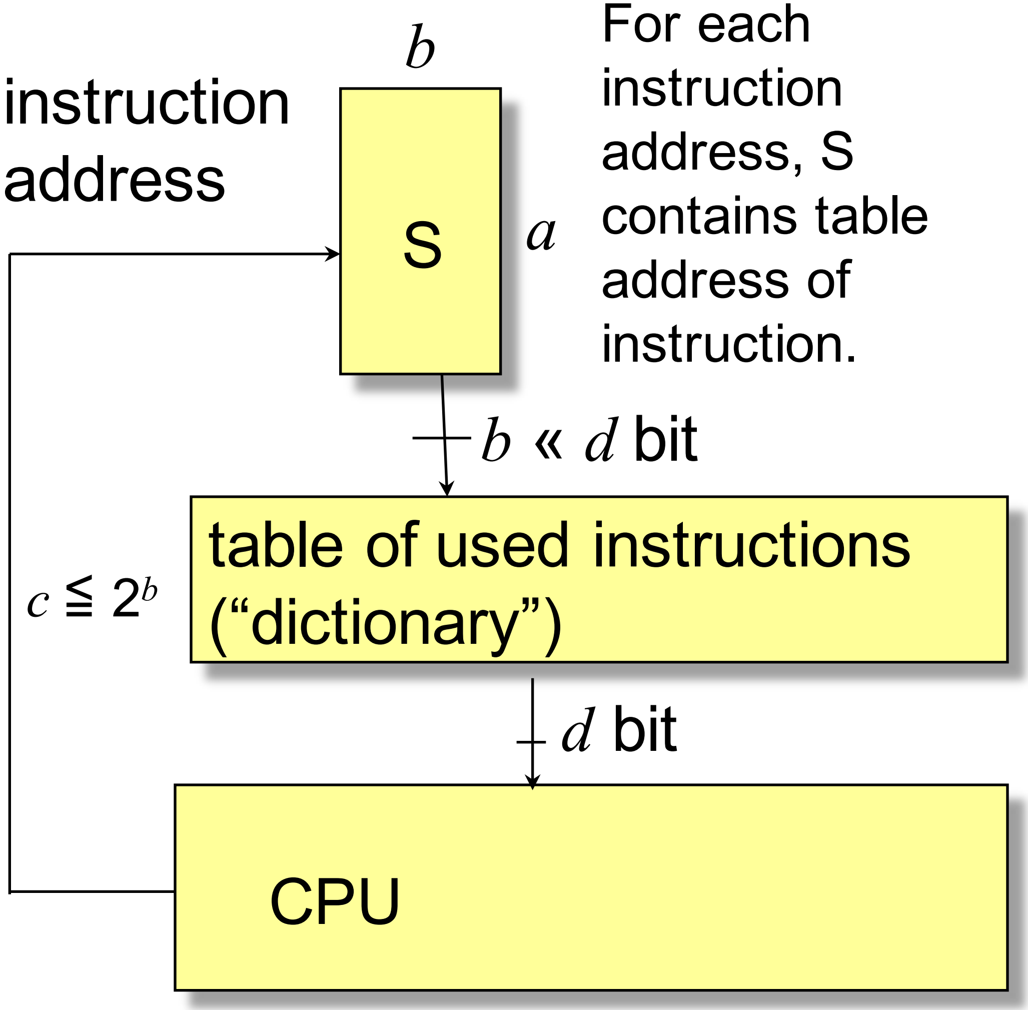
“Dictionary-based coding schemes cover a wide range of various coders and compressors.

Their common feature is that the methods use some kind of a dictionary that contains parts of the input sequence which frequently appear.

The encoded sequence in turn contains references to the dictionary elements rather than containing these over and over.”

[Á. Beszédes et al.: Survey of Code size Reduction Methods, Survey of Code-Size Reduction Methods, *ACM Computing Surveys*, Vol. 35, Sept. 2003, pp 223-267]

Key idea (for *d* bit instructions)



For each instruction address, S contains table address of instruction.

Uncompressed storage of a d-bit-wide instructions requires a x d bits.

In compressed code, each instruction pattern is stored only once.

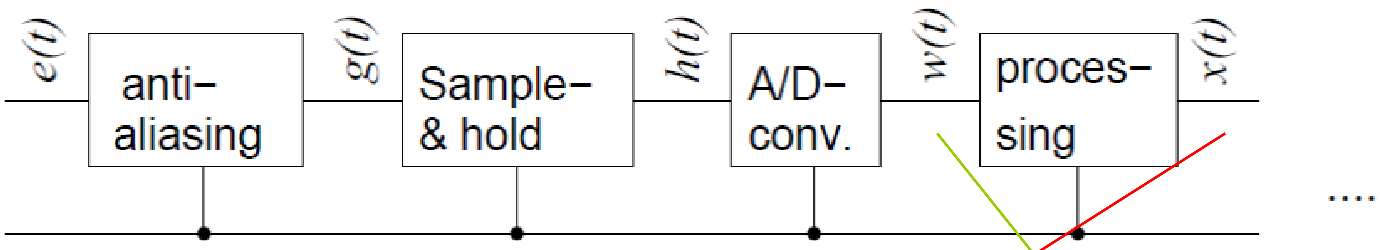
small

Hopefully, $a \times b + c \times d < a \times d$.

Called nanoprogramming in the Motorola 68000.

Key requirement #2: Run-time efficiency
- Domain-oriented architectures -

Example: Filtering in Digital signal processing (DSP)

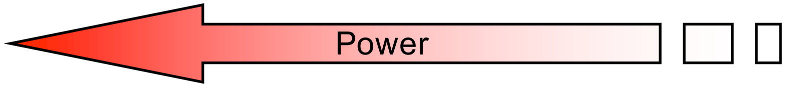


$$x_s = \sum_{k=0}^{n-1} w_{s-k} * a_k$$

Signal at $t=t_s$ (sampling points)

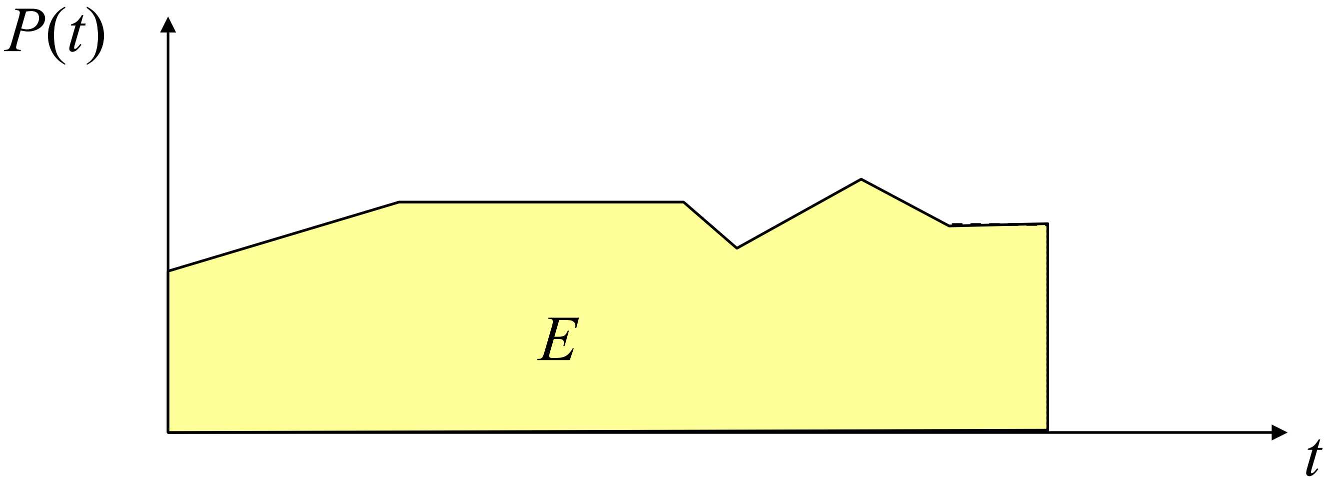
Key requirement #3:
energy-efficient and power efficient

	Relevant during use?		
Execution platform	Plugged	Uncharged periods	Unplug-ged
E.g.	Factory	Car	Sensor
Global warming	☑	☐	☐
Cost of energy	☑	☐	☐
Increasing performance	☑	☑	☑
Problems with cooling, avoiding hot spots	☑	☑	☑
Avoiding high currents & metal migration	☑	☑	☑
Reliability	☑	☑	☑
Energy a very scarce resource	☐	☑	☑



Should we care about energy consumption or about power consumption?

$$E = \int P(t)\, dt$$



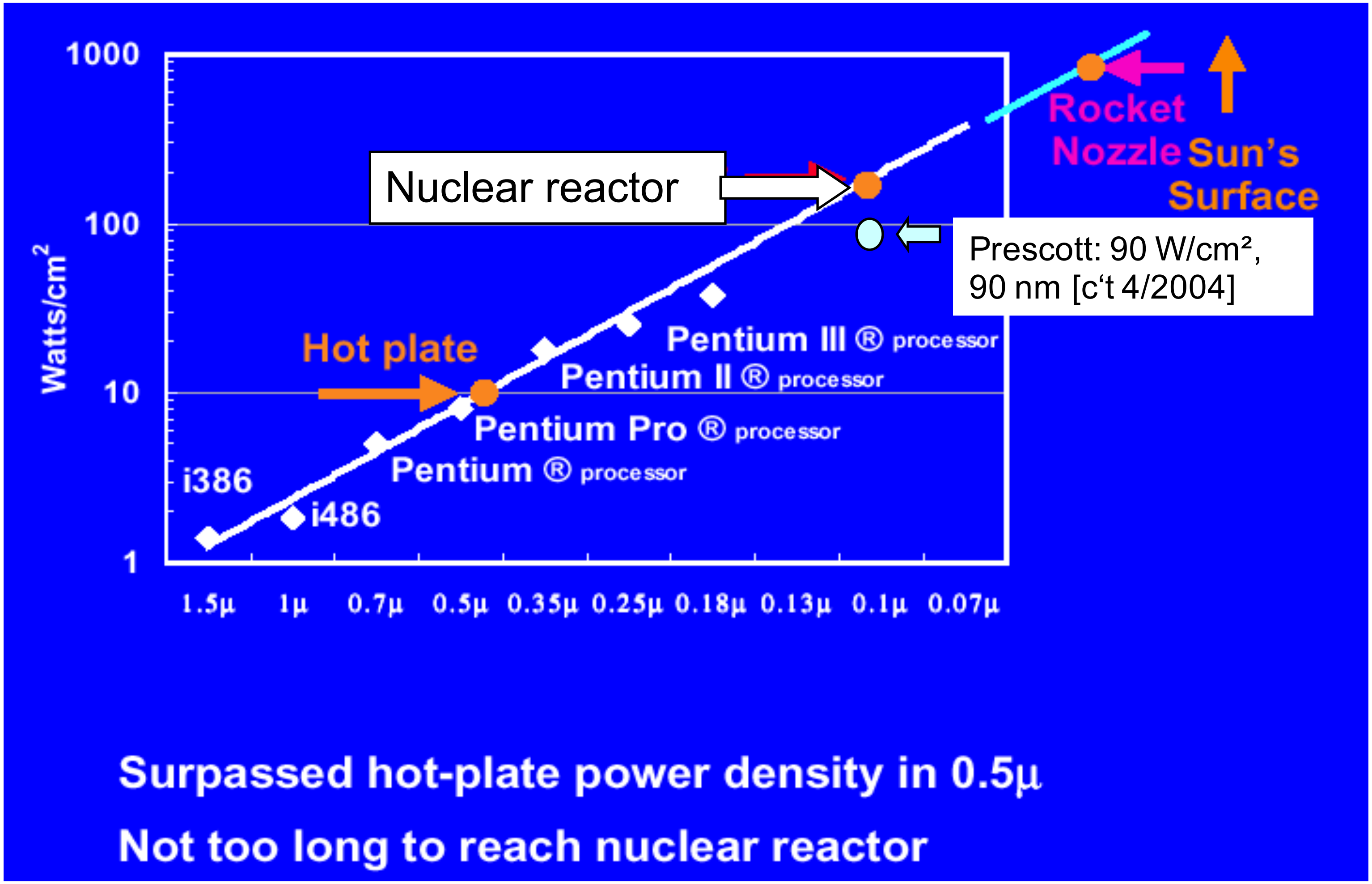
Both are closely related, but still different

Should we care about energy consumption or about power consumption (2)?

- Minimizing **power consumption** important for
  - design of the power supply & regulators
  - dimensioning of interconnect, short term cooling
- Minimizing **energy consumption** important due to
  - restricted availability of energy (mobile systems)
  - cooling: high costs, limited space
  - thermal effects
  - dependability, long lifetimes

☞ **In general, we need to care about both**

PCs: Problem: Power density increasing



Prescott: 90 W/cm², 90 nm [c't 4/2004]

Surpassed hot-plate power density in 0.5µ

Not too long to reach nuclear reactor

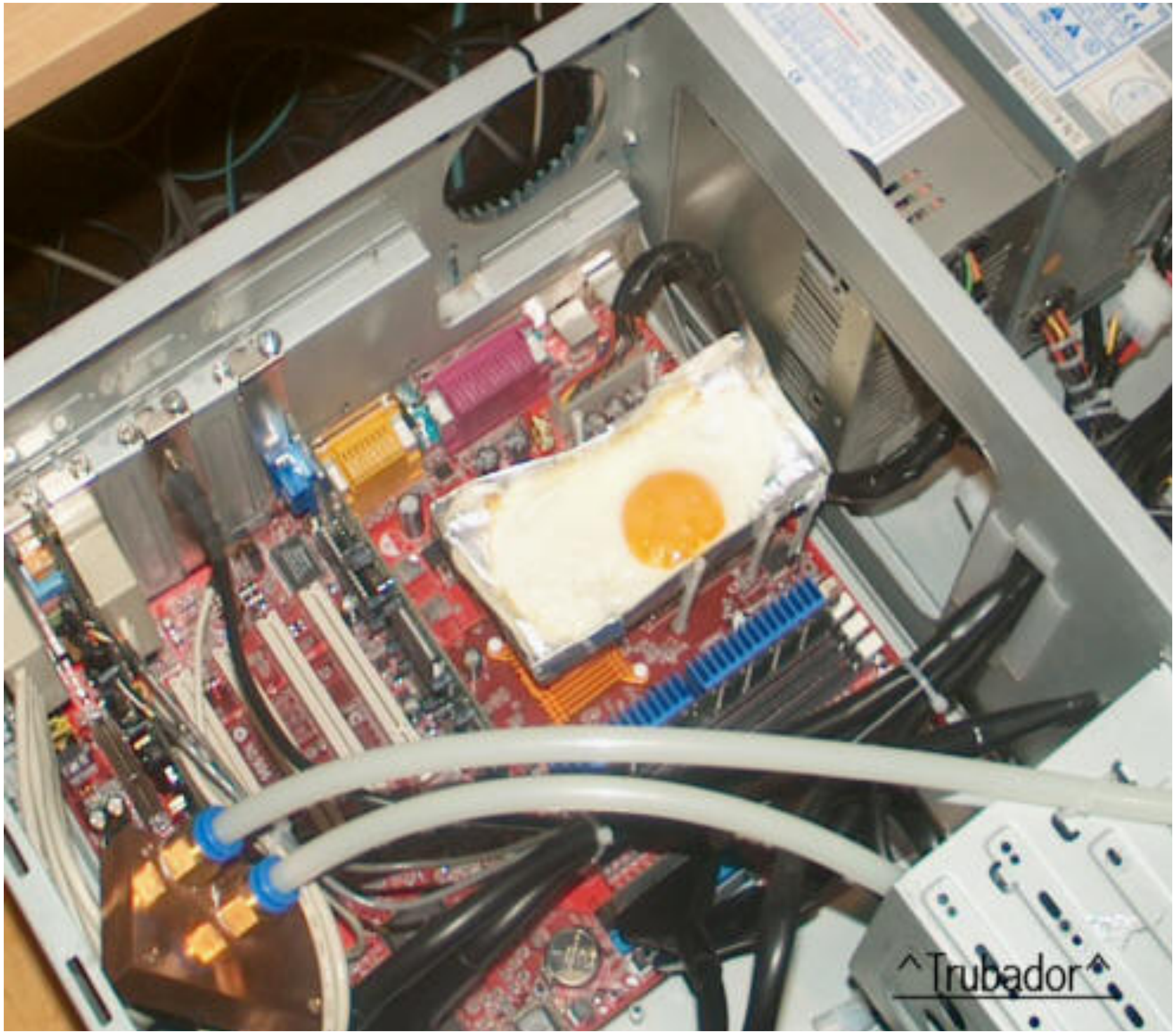
© Intel
M. Pollack,
Micro-32

PCs: Just adding transistors would have resulted in this:

Reuters: December 9, 2004: Men should keep their laptops off their laps because they could damage fertility, an expert said on Thursday. Laptops, which reach high internal operating temperatures, can heat up the scrotum which could affect the quality and quantity of men’s sperm. “The increase in scrotal temperature is significant enough to cause changes in sperm parameters,” said Dr Yefim Sheynkin, an associate professor of urology at the State University of New York at Stony Brook.

How do We Now Cook

PCs: Surpassed hot (kitchen) plate …? Why not use it?

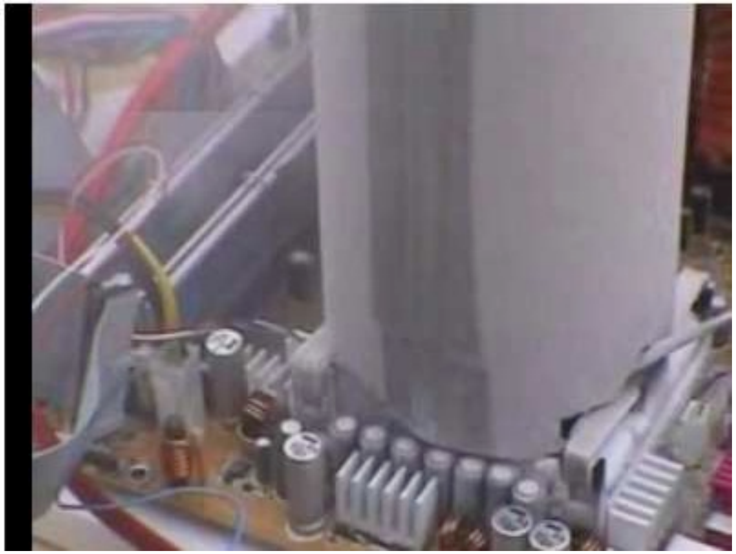


Strictly speaking, energy is not “consumed”, but converted from electrical energy into heat energy

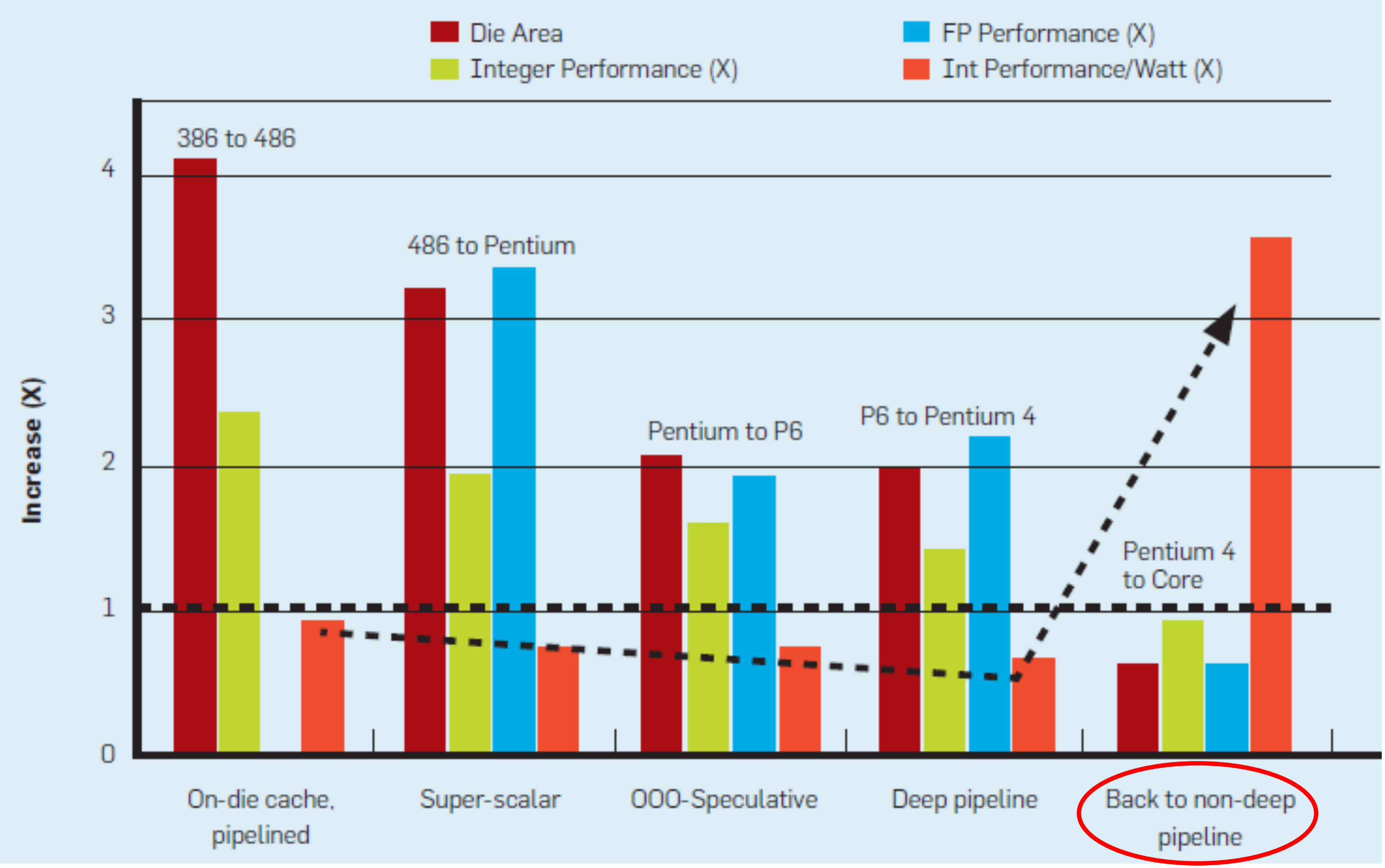
http://www.phys.ncku.edu.tw/~htsu/humor/fry_egg.html

Cooling Matters

- Thermoelectric cooling
- Liquid cooling
- Refrigeration cooling
- etc.



Keep it simple, stupid (KISS)

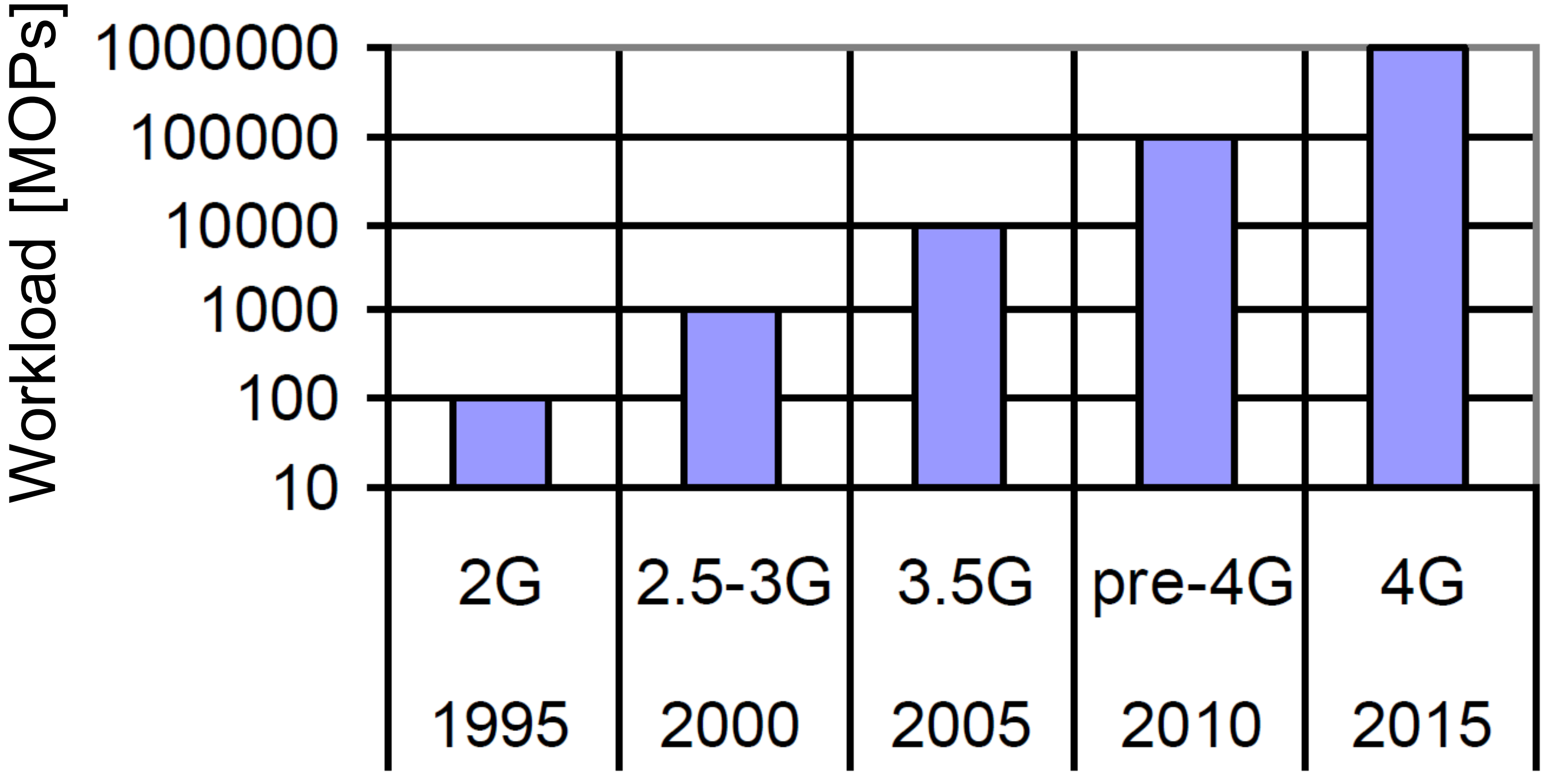


S. Borkar, A. Chien: The future of microprocessors, *Communications of the ACM*, May 2011

© ACM, 2011

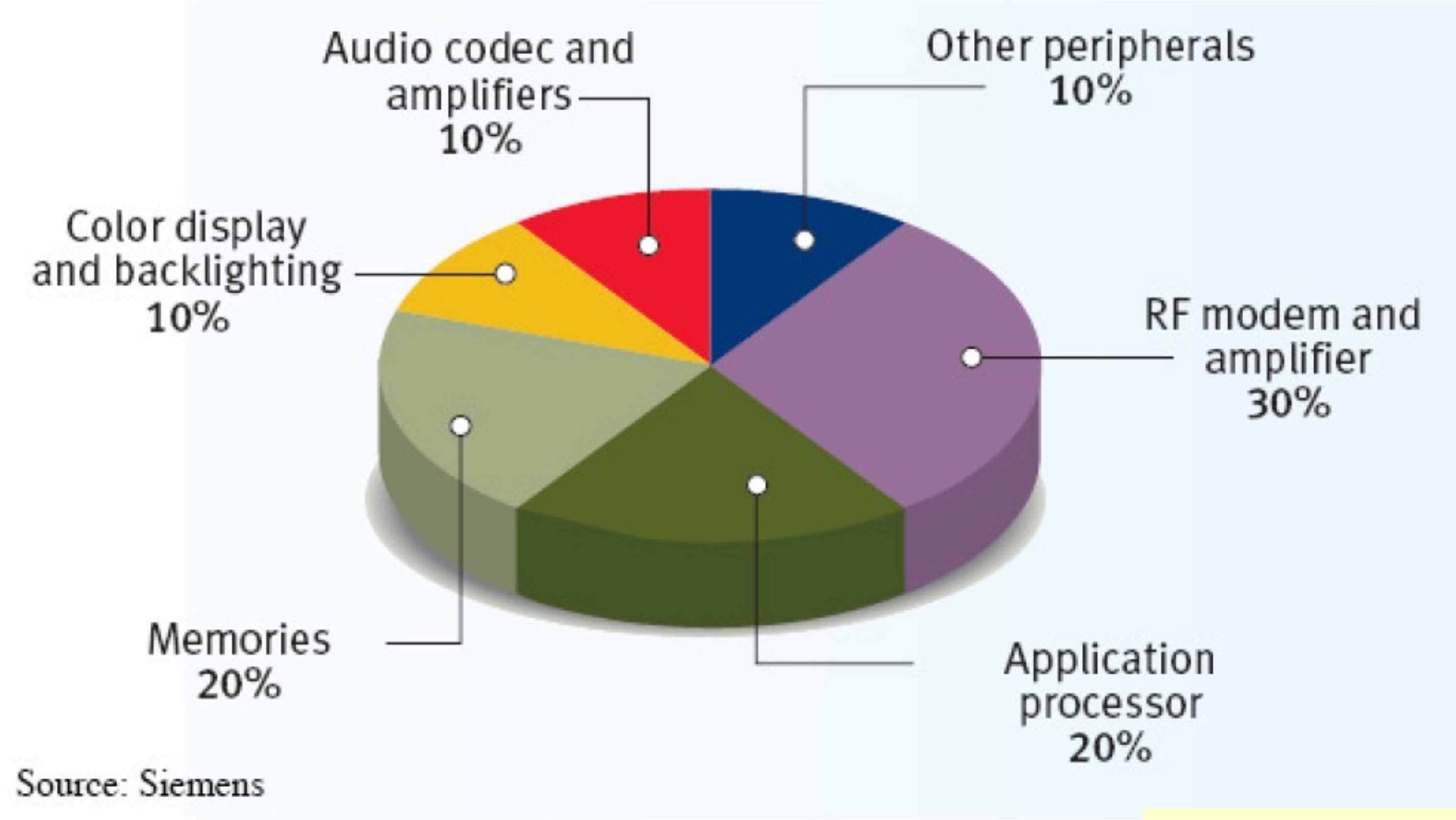
Mobile phones: Increasing performance requirements

C.H. van Berkel: Multi-Core for Mobile Phones, DATE, 2009;



Many more instances of the power/energy problem

Mobile phones: Where does the power go?

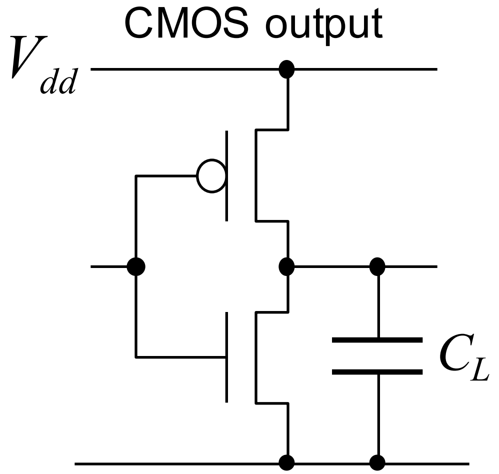


Source: Siemens

[O. Vargas: Minimum power consumption in mobile-phone memory subsystems; Pennwell Portable Design - September 2005;]

Prerequisite: CMOS Circuits
Static and dynamic power consumption

- Dynamic power consumption: Power consumption caused by charging capacitors when logic levels are switched.



$P = \alpha\, C_L\, V_{dd}^2\; s$ with

α : switching activity

C_L : load capacitance

V_{dd} : supply voltage

s : clock frequency

☞Decreasing V_{dd} reduces P quadratically

- Static power consumption (caused by leakage current): power consumed in the absence of clock signals
- Leakage becoming more important due to smaller devices

How to make systems energy efficient: Fundamentals of dynamic voltage scaling (DVS)

Power consumption of CMOS circuits (ignoring leakage):

$P = \alpha \, C_L V_{dd}^2 \, s$ with

α : switching activity

C_L : load capacitance

V_{dd} : supply voltage

s : clock frequency

Delay for CMOS circuits:

$$\tau = k \, C_L \frac{V_{dd}}{(V_{dd} - V_t)^2}$$ with

V_t : threshhold voltage

(V_t < than V_{dd})

☞Decreasing V_{dd} reduces P quadratically, while the run-time of algorithms is only linearly increased

How to make systems energy efficient: Fundamentals of dynamic voltage/frequency scaling (DVFS)

Power consumption of CMOS circuits (ignoring leakage):

$$P = \alpha \, C_L \, V_{dd}^2 \, s \text{ with}$$

α : switching activity

C_L : load capacitance

V_{dd} : supply voltage

s : clock frequency

Delay for CMOS circuits:

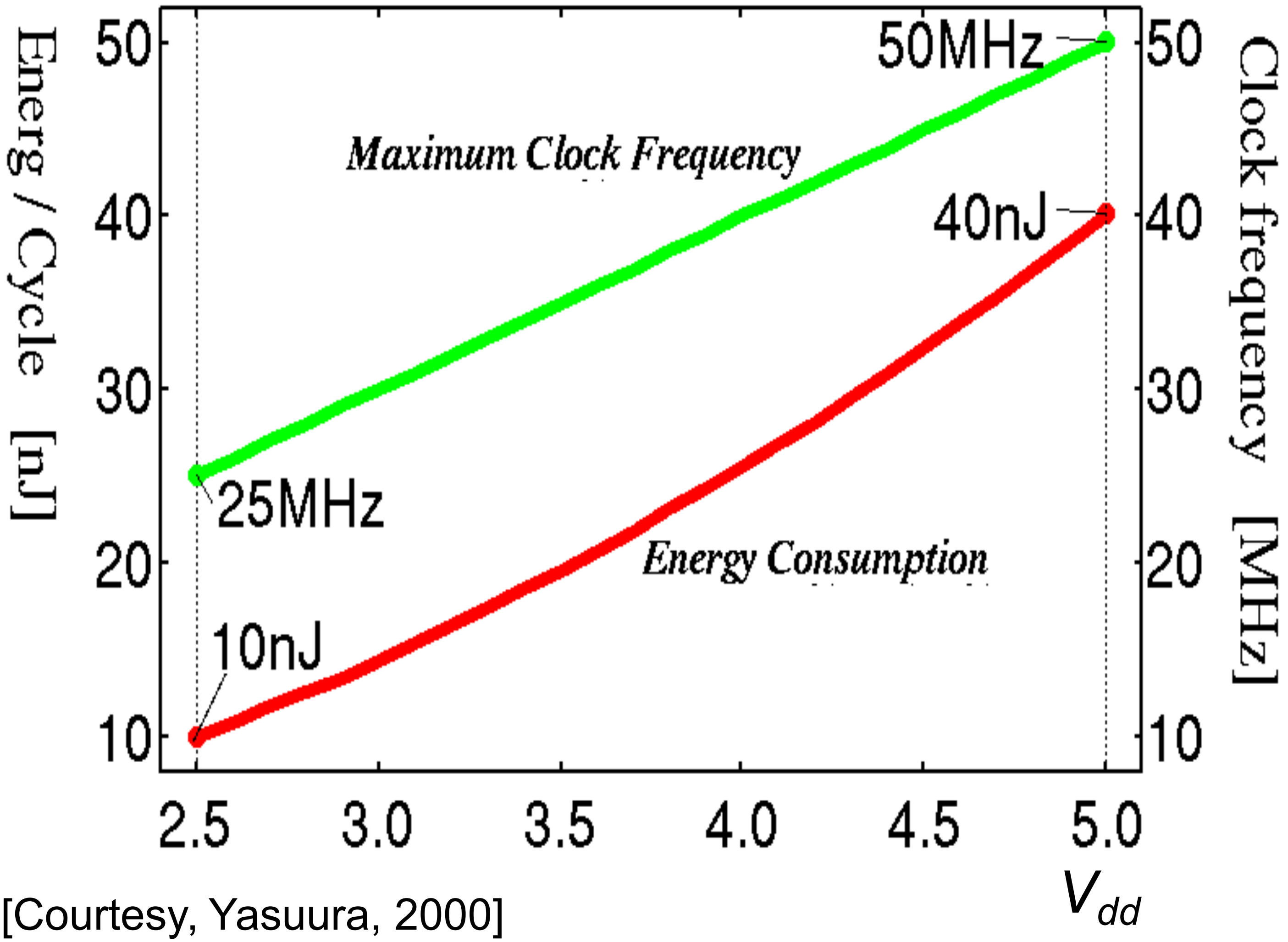
$$\tau = k \, C_L \frac{V_{dd}}{(V_{dd} - V_t)^2} \text{ with}$$

V_t : threshhold voltage

(V_t < than V_{dd})

☞Decreasing V_{dd} and frequency together reduces P cubically, while the run-time of algorithms is only linearly increased

Voltage/Frequency scaling: Example



[Courtesy, Yasuura, 2000]

Abstract Power Model

CMOS-core Power Model

$$P(s) = P_{\text{dynamic}}(s) + P_{\text{static}}$$

Considering that:

$$P_{\text{dynamic}}(s) = C_{\text{eff}} V_{dd}^2 s$$

$$s \propto \frac{(V_{dd} - V_t)^2}{V_{dd}}$$

We can approximate to:

$$P(s) = \alpha s^\gamma + \beta$$

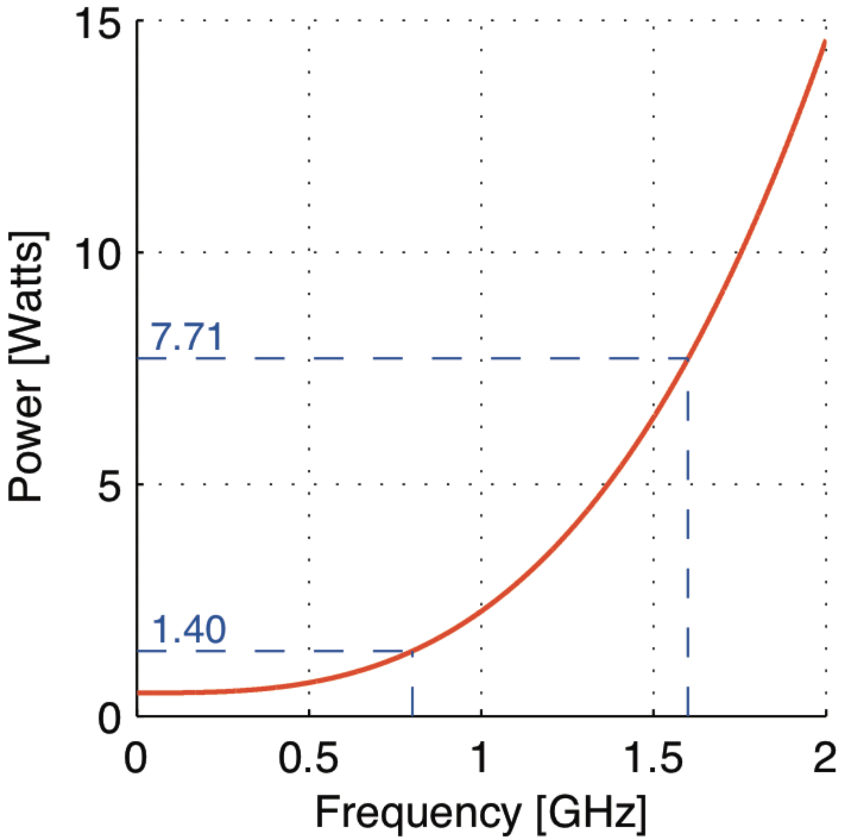


Figure: $\alpha = 1.76 \frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$ and $\beta = 0.5$ Watts

Abstract Energy Model

Energy Consumption

$$E(s) = (\alpha s^{\gamma} + \beta)\frac{\Delta c}{s}$$

Critical Frequency:

$$s_{\text{crit}} = \sqrt[\gamma]{\frac{\beta}{(\gamma - 1)\,\alpha}}$$

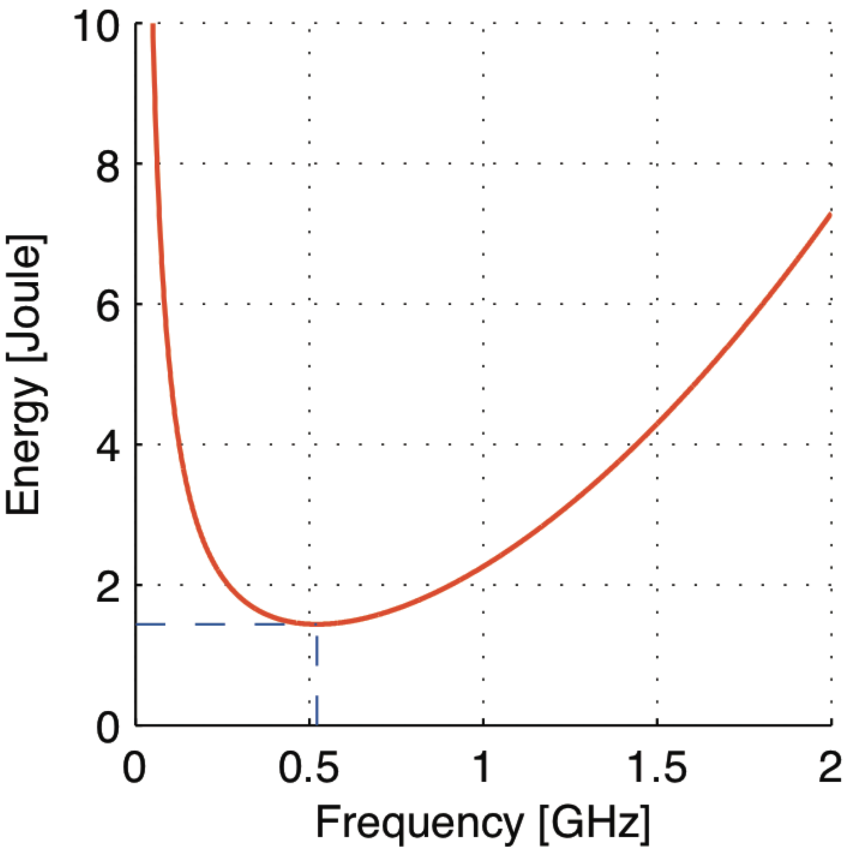
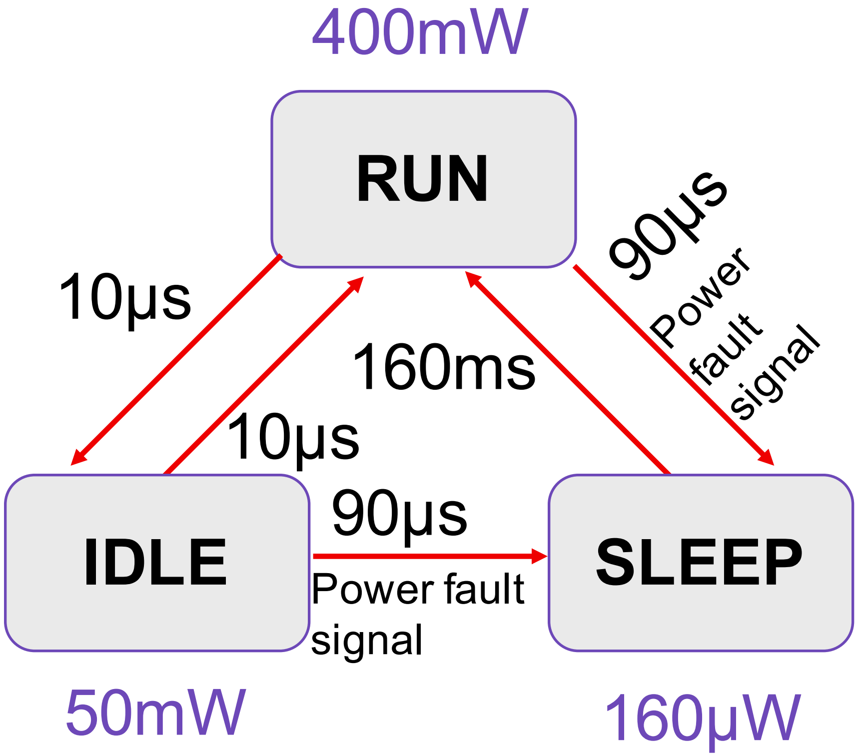


Figure: $\alpha = 1.76\frac{\text{Watts}}{\text{GHz}^3}$, $\gamma = 3$, $\beta = 0.5$ Watts and $\Delta c = 10^9$ cycles

Dynamic power management (DPM)

Example: STRONGARM SA1100

RUN: operational
IDLE: a SW routine may stop the CPU when not in use, while monitoring interrupts
SLEEP: Shutdown of on-chip activity



Low voltage, parallel operation more efficient than high voltage, sequential operation

Basic equations

Power: $P \sim V_{DD}^2\ s$,

Maximum clock frequency: $s \sim V_{DD}$,

Energy to run a program: $E = P \times t$, with: t = runtime

Time to run a program: $t \sim 1/s$

Changes due to parallel processing, with *M* operations per clock:

Clock frequency reduced to: $s' = s\ /\ M$,

Voltage can be reduced to: $V_{DD}' = V_{DD}\ /\ M$,

Power for parallel processing: $P^\circ = P\ /\ M^3$ per operation,

Power for β operations per clock: $P' = M \times P^\circ = P\ /\ M^2$,

Time to run a program is still: $t' = t$,

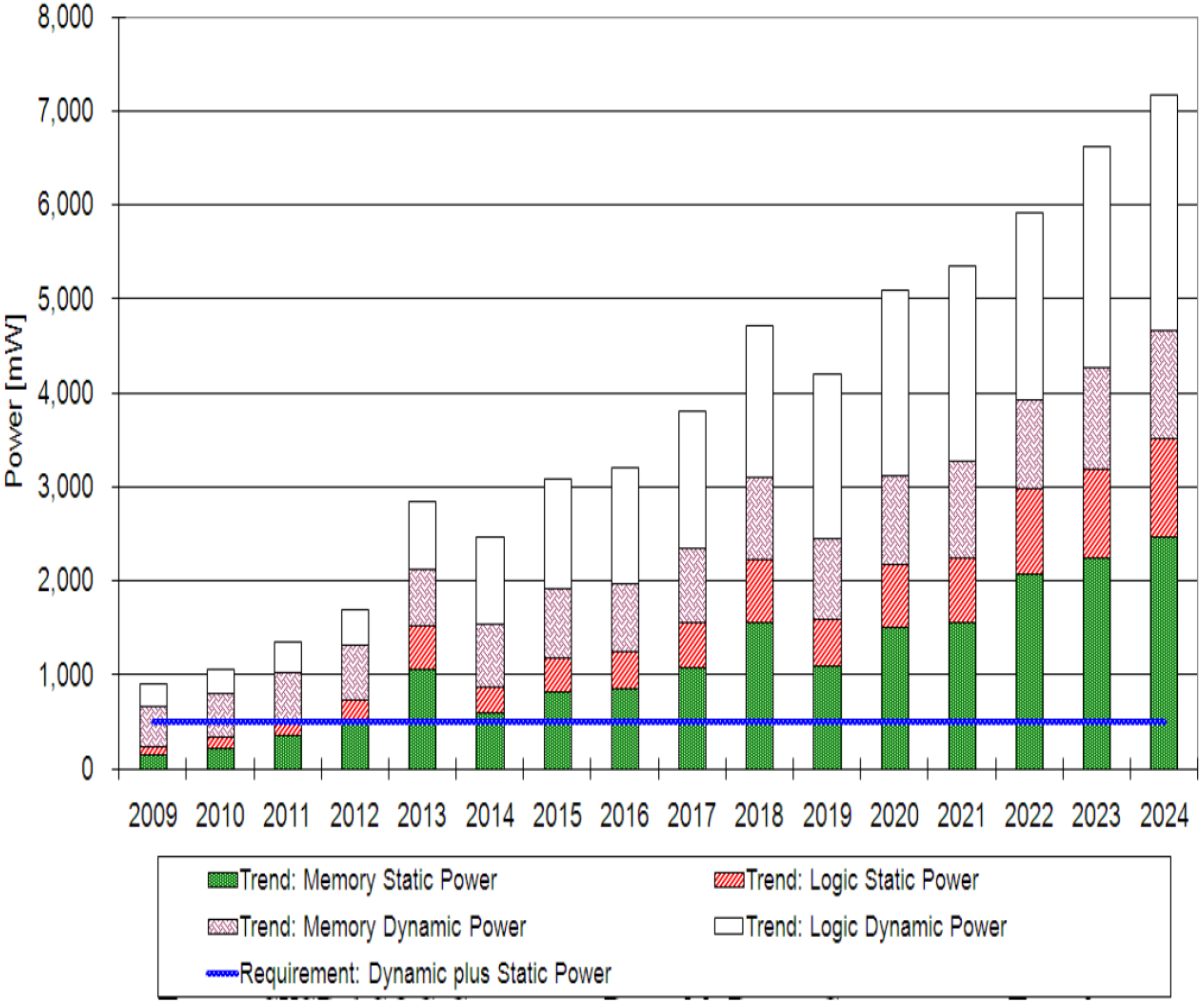
Energy required to run program: $E' = P' \times t = E\ /\ M^2$

☞Argument in favour of voltage scaling, and parallel processing

Rough approxi-mations!

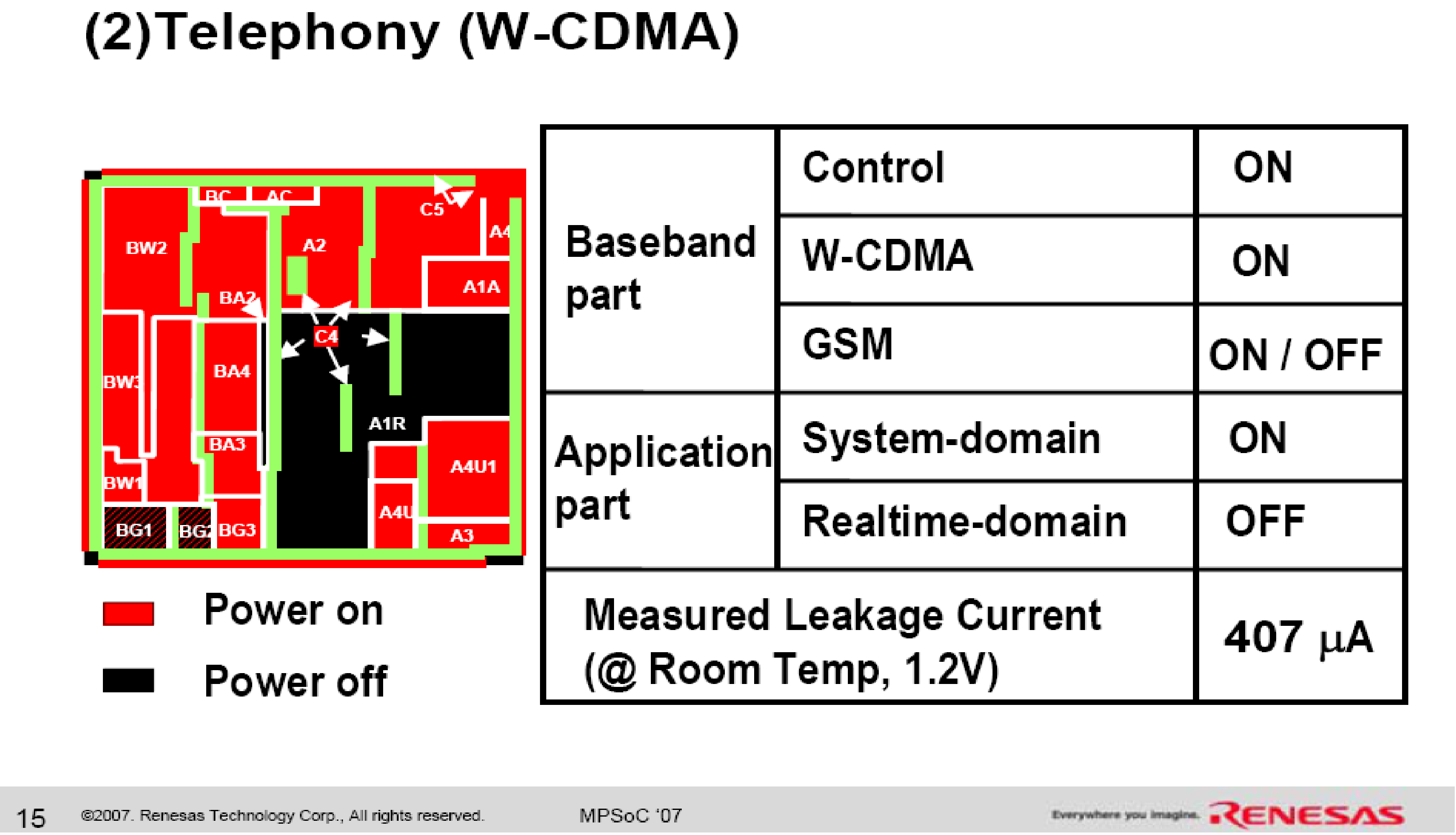
Where is the energy consumed? Target for the mobile phones

- According to *International Technology Roadmap for Semiconductors* (ITRS), 2010 update, [www.itrs.net]
- Current trends ☞ violation of 0.5-1 W constraint for small mobiles; large mobiles: ~ 7 W



© ITRS, 2010

Energy-efficient architectures: Heterogeneous processors



(2)Telephony (W-CDMA)

🟥 Power on

⬛ Power off

Baseband part	Control	ON
	W-CDMA	ON
	GSM	ON / OFF
Application part	System-domain	ON
	Realtime-domain	OFF
Measured Leakage Current (@ Room Temp, 1.2V)		407 μA

15 ©2007. Renesas Technology Corp., All rights reserved. MPSoC '07 Everywhere you imagine. RENESAS

http://www.mpsoc-forum.org/2007/slides/Hattori.pdf

ARM‘s big.LITTLE as an example

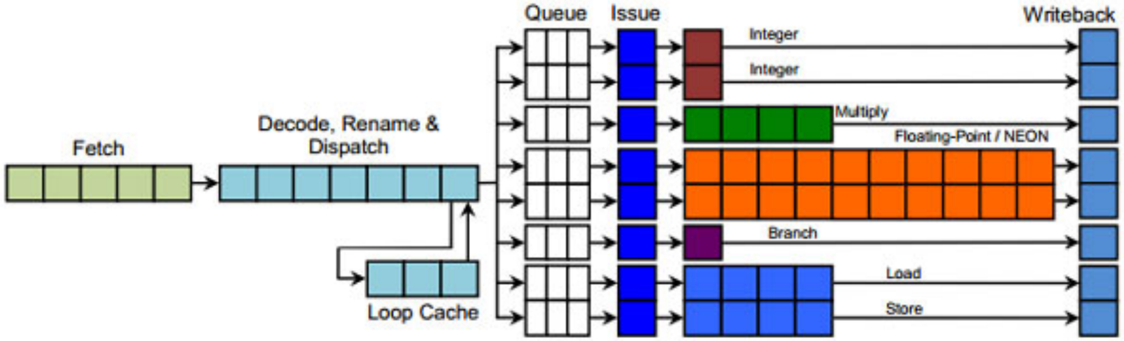


Figure 2 Cortex-A15 Pipeline

Used in Samsung S4

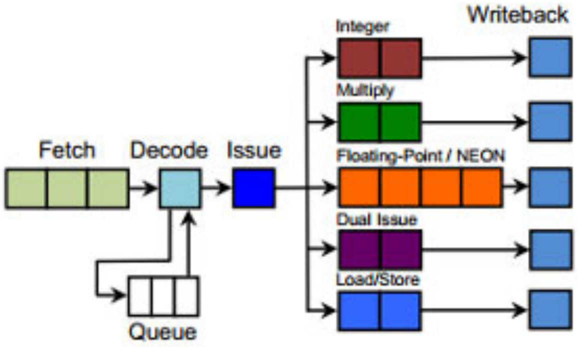
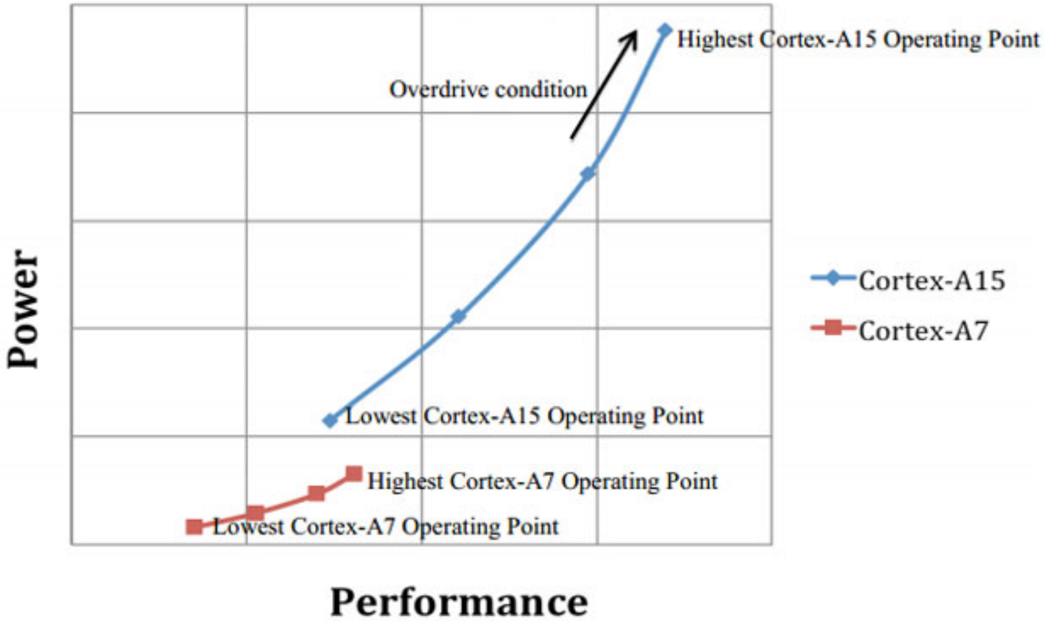


Figure 1 Cortex-A7 Pipeline



© ARM, 2013

Embedded System Hardware
- Power and Temperature Issues – (Sections 5.4/5.5)

Jian-Jia Chen
(Slides are based on
Peter Marwedel)
Informatik 12
TU Dortmund
Germany

2019年 11 月 12 日

Thermal Modeling: A Single Power Source

- Thermal conduction
  - Fourier's Law of Cooling: the temperate change is proportional to the different of the chip and the ambient temperature (or the heat sink temperature)
    - If the chip is hotter, the temperature change drops more
    - If the chip is cooler, the temperature change drops less
  - Heating generation is proportional to the power consumption
    - If the power consumption is larger, the temperature change increases more
    - If the power consumption is smaller, the temperature change increases less
  - *Therefore,* $T'(t) = uP(t) - v(T(t)-T_{amb})$
    - $T(t)$ is the temperature of the power source at time t
    - $P(t)$ is the power consumption of the power source at time t
    - T_{amb} is the ambient temperature. I will simple use it as 0. Why?
    - u and v are both hardware-dependent constants.

Solving Ordinary Differential Equation (ODE): $T'(t) = uP(t) - vT(t)$

It is a standard linear ODE, where u and v are constants:

$$d\frac{T(t)e^{vt}}{dt} = e^{vt}d\frac{T(t)}{dt} + T(t)\cdot ve^{vt} = e^{vt}(uP(t) - vT(t)) + T(t)\cdot ve^{vt} = e^{vt}uP(t).$$

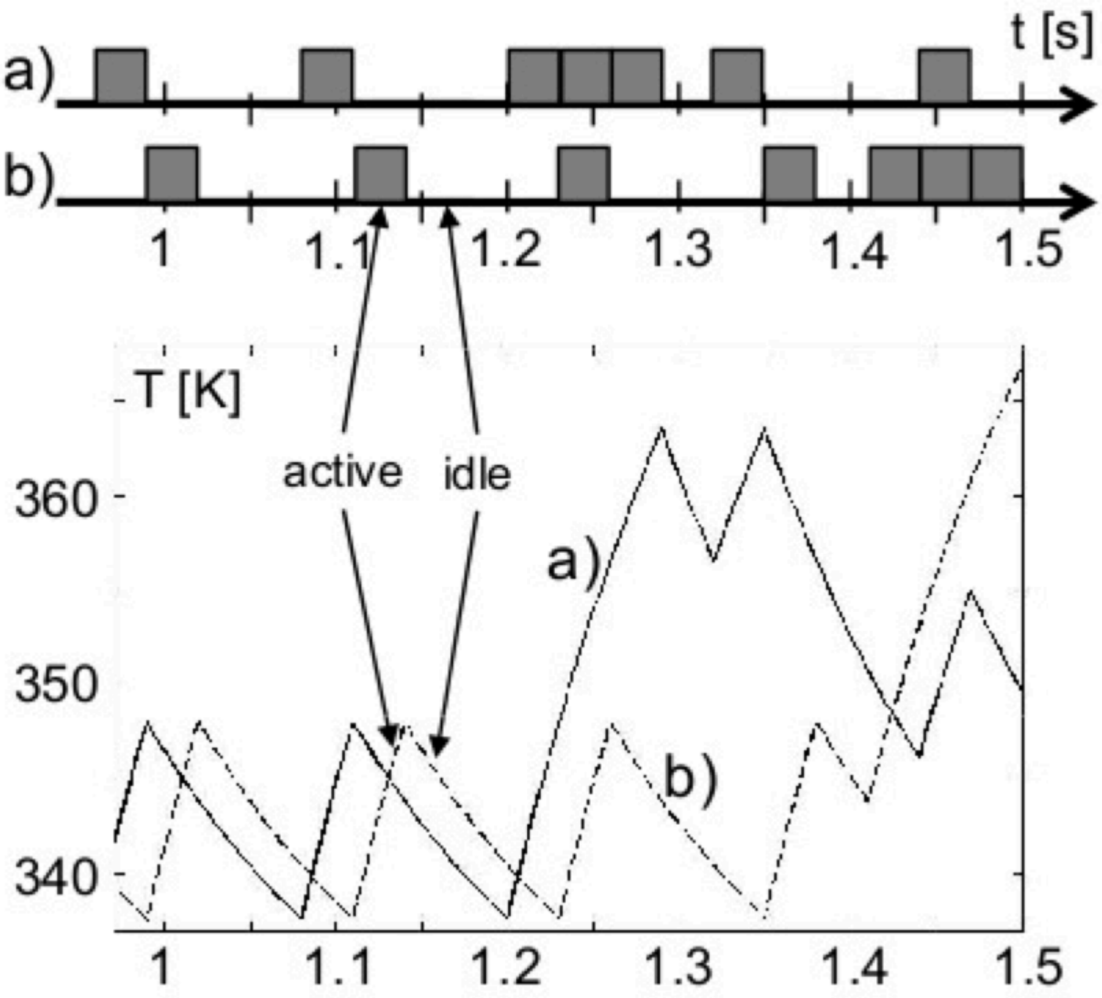
$$\int_{t_0}^{t} d\frac{T(t)e^{vt}}{dt} = \int_{t_0}^{t} e^{vt}uP(t) \Rightarrow T(t)e^{vt} - T(t_0)e^{vt_0)} = \int_{t_0}^{t} e^{vx}uP(x)dx$$

$$\Rightarrow T(t) - T(t_0)e^{-v(t-t_0)} = \int_{t_0}^{t} e^{v(x-t)}uP(x)dx$$

$$\Rightarrow T(t) = T(t_0)e^{-v(t-t_0)} + \int_{t_0}^{t} e^{v(x-t)}uP(x)dx$$

- The temperature effect at time t_0 decreases exponentially by $T(t_0)e^{-v(t-t_0)}$.
- The power consumption effect at time x decreases exponentially by $T(t_0)e^{v(x-t)}$, since $v > 0$ and $x - t \leq 0$ for $x \leq t$.

Different Traces versus Temperature



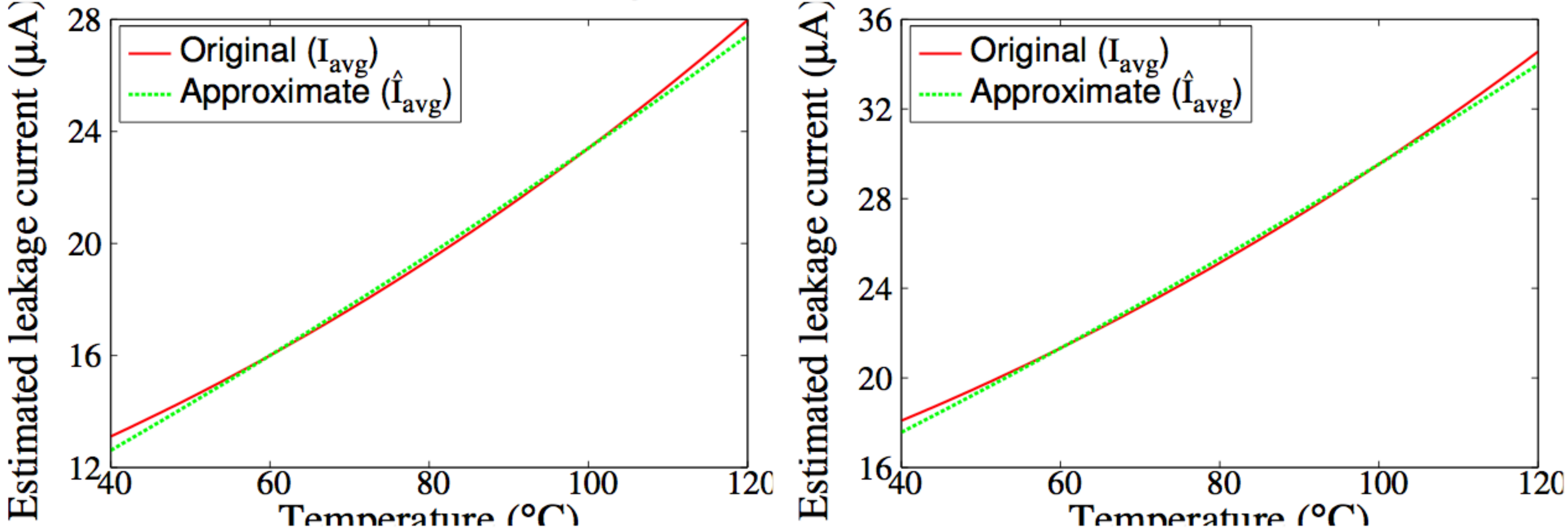
© Thiele et al. DATE 2011

Thermal-Dependent Leakage Power Consumption

$$I_{avg}(T, V_{dd}) = I(T_0, V_0)\left(AT^2 e^{\left(\frac{q_1 \cdot V_{dd} + q_2}{T}\right)} + Be^{(\gamma \cdot V_{dd} + \delta)}\right),$$

However, the term $e^{(1/T)}$ dose not provide significant role in the accuracy. It is possible to use a simpler formula to formulate the leakage current.

$$\hat{I}_{avg}(T) = \hat{A}T^2 + \hat{B},$$

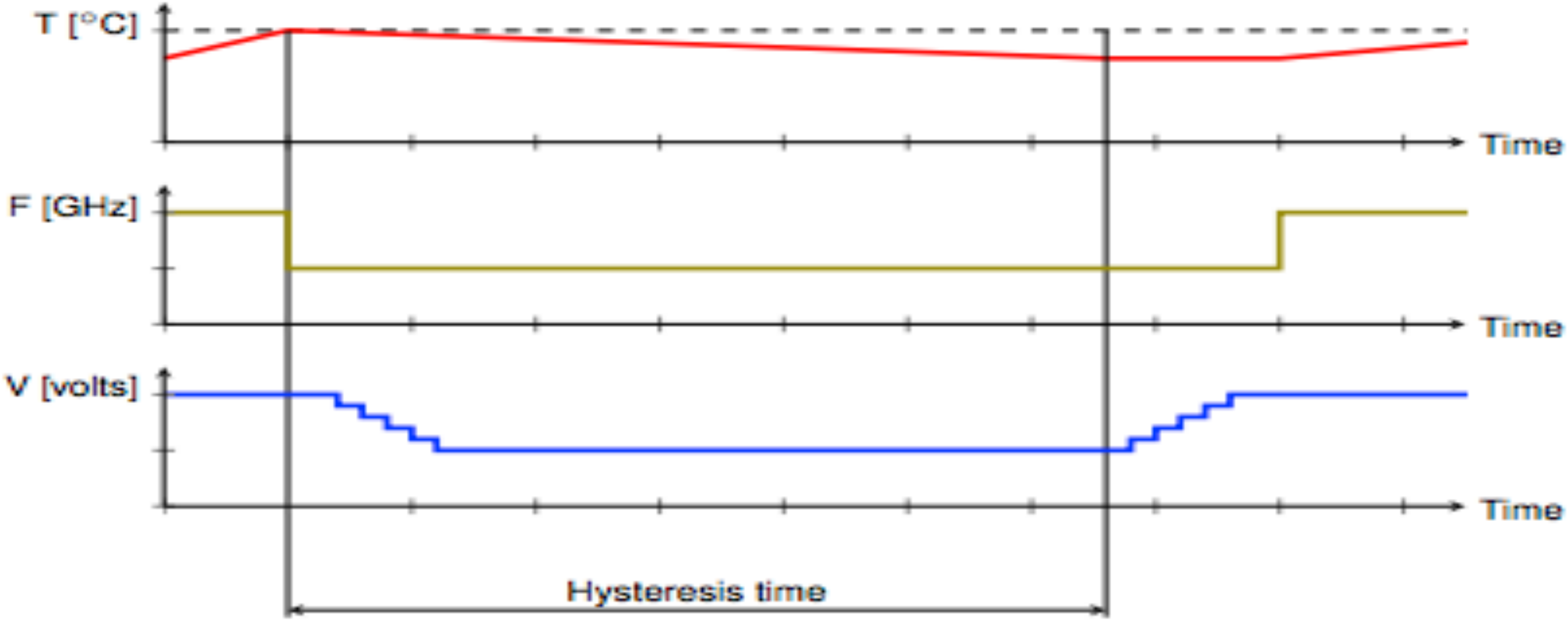


(a) $V_{dd} = 0.95\text{V}$ (b) $V_{dd} = 1.05\text{V}$

Chuan-Yue Yang, Jian-Jia Chen, Lothar Thiele, Tei-Wei Kuo: Energy-efficient real-time task scheduling with temperature-dependent leakage. DATE 2010: 9-14

Dynamic Thermal Management (DTM)

- Avoid possible over heating
  - DVFS
  - DPM



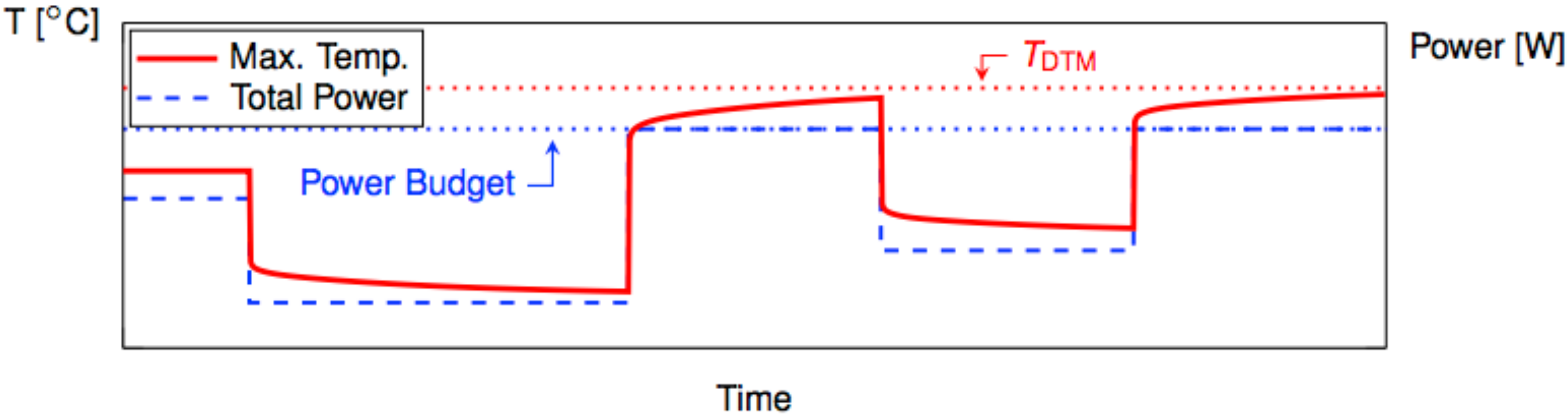
Thermal Networks (only for your reference here)

- Thermal models of applications depend on neighbouring cores.
  - A resistance-capacitance (RC) thermal network is widely used
    - A set of first order differential equations
  - Steady states (the equilibrium temperatures if the power does not change)
    - Simple linear algebra
  - Transient states (temperature profile in time)
    - Approximate the solution by using fourth-order *Runge- Kutta* numerical method [HotSpot, Huang et al. 2009]
    - Exact solution by using matrix exponential (many approximations are available) methods [MatEx, Pagani et al. to be published in DATE 2015]

[P.-Y. Huang and Y.-M. Lee, "Full-chip thermal analysis for the early design stage via generalized integral transforms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 5, pp. 613–626, May 2009. ;]
[Santiago Pagani, Muhammad Shafique, Jian-Jia Chen and Jörg HenkelMatEx: Efficient Transient and Peak Temperature Computation for Compact Thermal Modelsin 18th Design, Automation & Test in Europe (DATE) 2015 ;]

Power Budget / Power Constraint

- Abstraction: Not deal directly with temperature.
- Generally, a power budget (for thermal safety) is a single value:
  - For each core (per-core).
  - For the entire chip (per-chip).

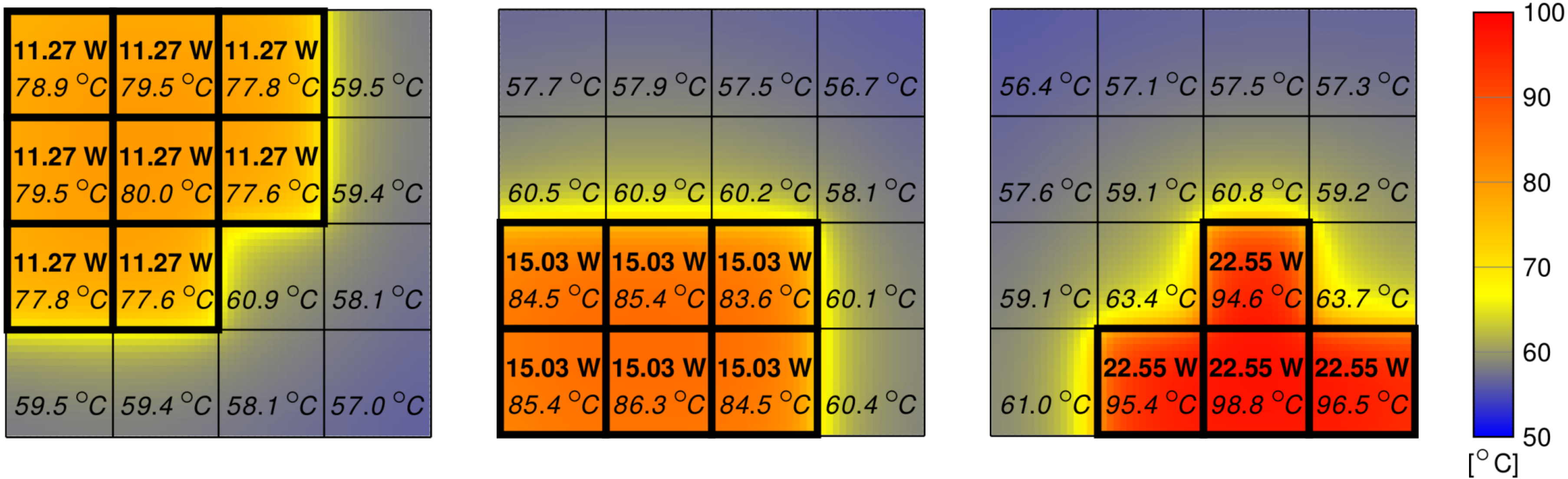


TDP

According to Wikipedia, “**The thermal design power (TDP),** sometimes called thermal design point, is the maximum amount of heat generated by a computer chip or component (often the CPU or GPU) that the cooling system in a computer is designed to dissipate in typical operation. Rather than specifying CPU’s real power dissipation, TDP serves as the nominal value for designing **CPU cooling** systems. ”

Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm^2
Threshold temperature: 80° C
Power budget: 90 W



11.27 W 78.9 °C	11.27 W 79.5 °C	11.27 W 77.8 °C	59.5 °C
11.27 W 79.5 °C	11.27 W 80.0 °C	11.27 W 77.6 °C	59.4 °C
11.27 W 77.8 °C	11.27 W 77.6 °C	60.9 °C	58.1 °C
59.5 °C	59.4 °C	58.1 °C	57.0 °C

Highest Temperature: 80.0° C

8 active cores

57.7 °C	57.9 °C	57.5 °C	56.7 °C
60.5 °C	60.9 °C	60.2 °C	58.1 °C
15.03 W 84.5 °C	15.03 W 85.4 °C	15.03 W 83.6 °C	60.1 °C
15.03 W 85.4 °C	15.03 W 86.3 °C	15.03 W 84.5 °C	60.4 °C

Highest Temperature: 86.3° C

6 active cores

56.4 °C	57.1 °C	57.5 °C	57.3 °C
57.6 °C	59.1 °C	60.8 °C	59.2 °C
59.1 °C	63.4 °C	22.55 W 94.6 °C	63.7 °C
61.0 °C	22.55 W 95.4 °C	22.55 W 98.8 °C	22.55 W 96.5 °C

Highest Temperature: 98.8C

4 active cores

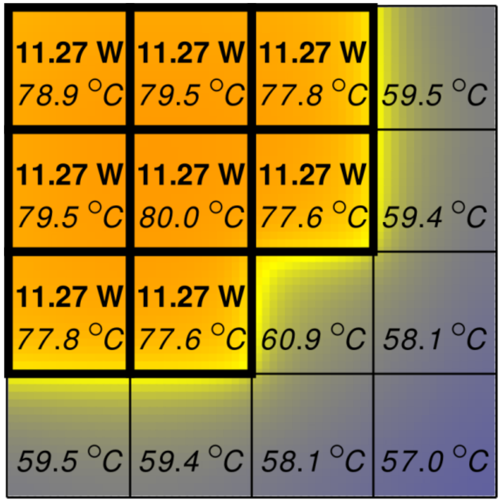
Pagani et al.,
CODES+ISSS 2014

Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm^2

Threshold temperature: 80° C

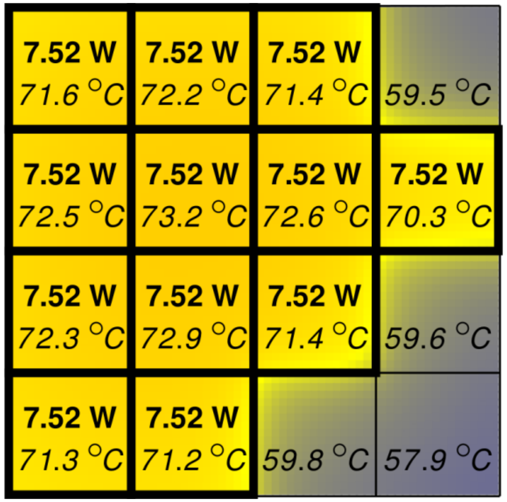
Power budget: 90 W



11.27 W *78.9 °C*	**11.27 W** *79.5 °C*	**11.27 W** *77.8 °C*	*59.5 °C*
11.27 W *79.5 °C*	**11.27 W** *80.0 °C*	**11.27 W** *77.6 °C*	*59.4 °C*
11.27 W *77.8 °C*	**11.27 W** *77.6 °C*	*60.9 °C*	*58.1 °C*
59.5 °C	*59.4 °C*	*58.1 °C*	*57.0 °C*

Highest Temperature: 80.0° C

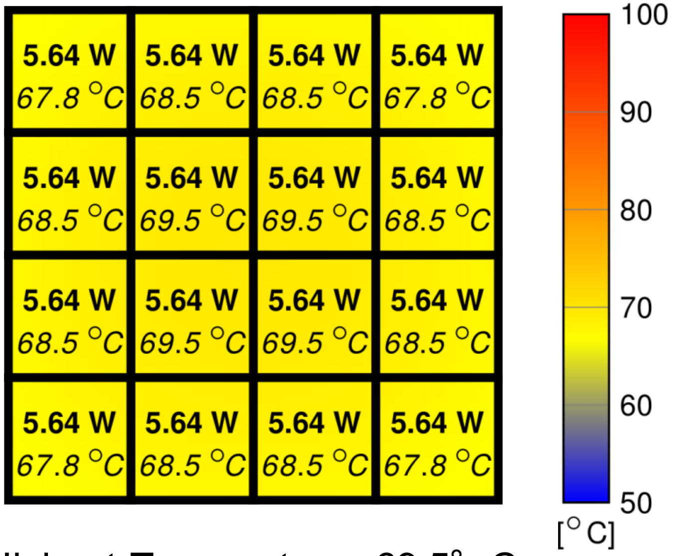
8 active cores



7.52 W *71.6 °C*	**7.52 W** *72.2 °C*	**7.52 W** *71.4 °C*	*59.5 °C*
7.52 W *72.5 °C*	**7.52 W** *73.2 °C*	**7.52 W** *72.6 °C*	**7.52 W** *70.3 °C*
7.52 W *72.3 °C*	**7.52 W** *72.9 °C*	**7.52 W** *71.4 °C*	*59.6 °C*
7.52 W *71.3 °C*	**7.52 W** *71.2 °C*	*59.8 °C*	*57.9 °C*

Highest Temperature: 73.2° C

12 active cores



5.64 W *67.8 °C*	**5.64 W** *68.5 °C*	**5.64 W** *68.5 °C*	**5.64 W** *67.8 °C*
5.64 W *68.5 °C*	**5.64 W** *69.5 °C*	**5.64 W** *69.5 °C*	**5.64 W** *68.5 °C*
5.64 W *68.5 °C*	**5.64 W** *69.5 °C*	**5.64 W** *69.5 °C*	**5.64 W** *68.5 °C*
5.64 W *67.8 °C*	**5.64 W** *68.5 °C*	**5.64 W** *68.5 °C*	**5.64 W** *67.8 °C*

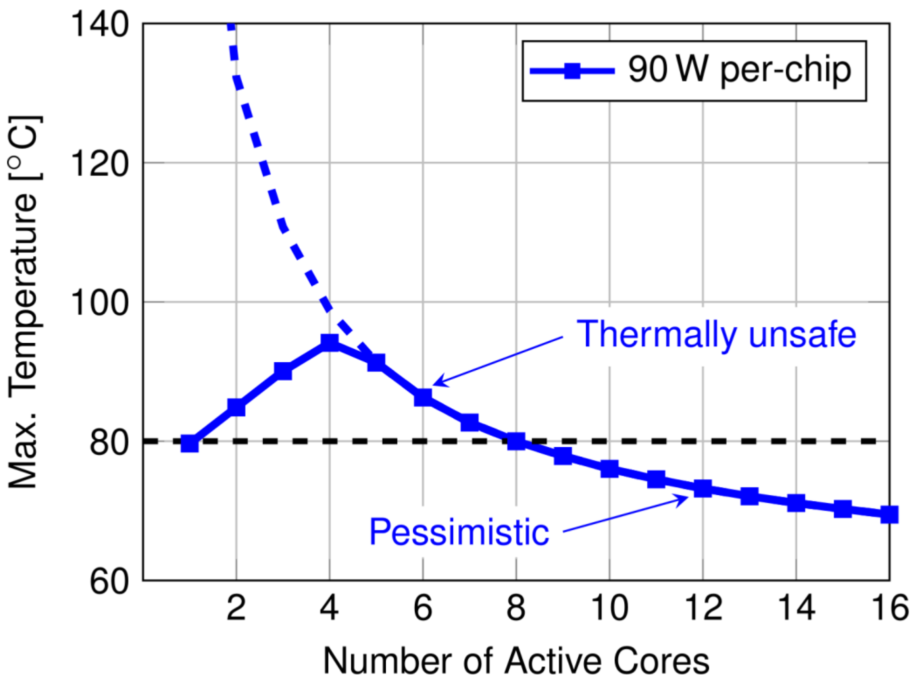
Highest Temperature: 69.5° C

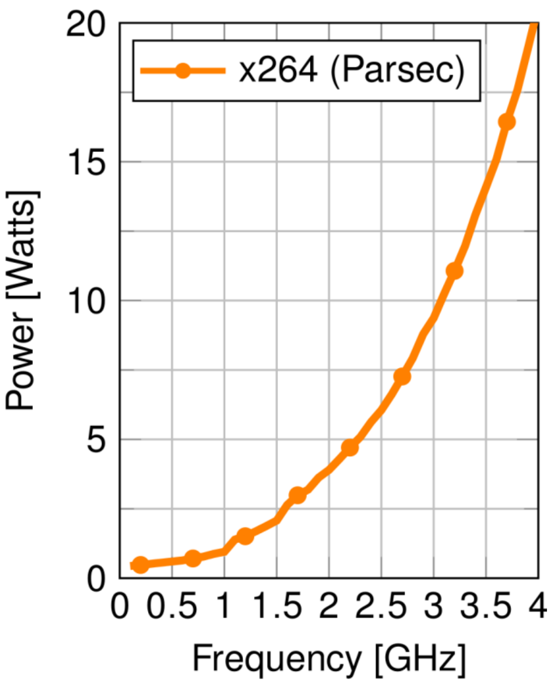
16 active cores

Pagani et al.,
CODES+ISSS 2014

Problem with Per-Chip / Per-Core Power Budgets

16 cores with area 5.3 mm^2
Threshold temperature for DTM: 80° C
Power budget: 90 W





Pagani et al., CODES+ISSS 2014

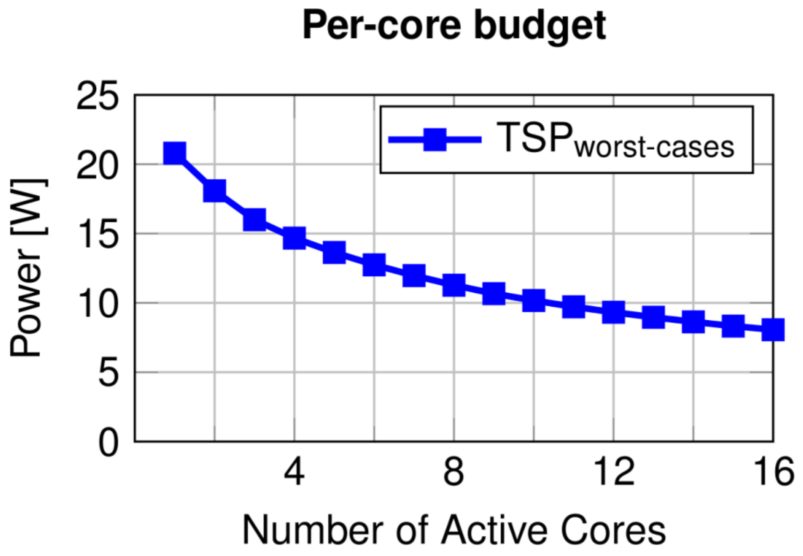
Thermal Safe Power (TSP): Power Budget depending on # of activated cores

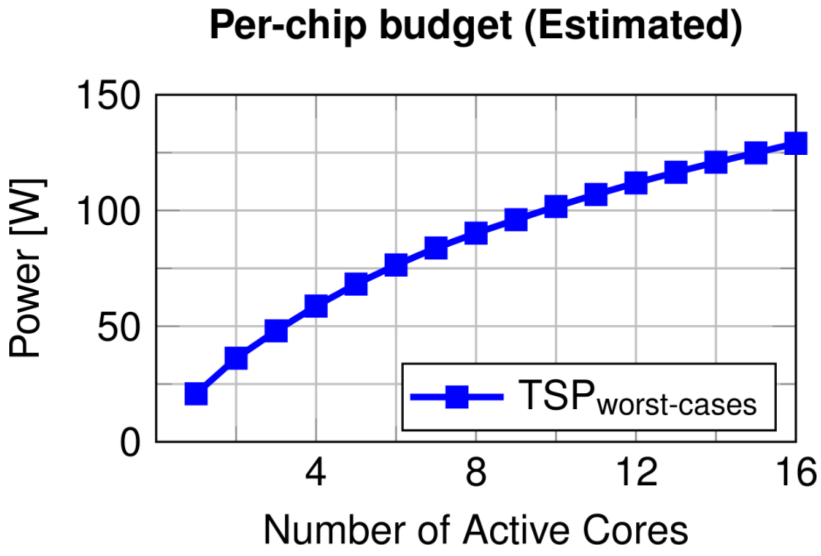
Power budget depends on the number of active cores

Safe for **any** '*m*' active cores => Abstract mapping decisions

TSP table:

Active Cores	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
TSP per-core [W]	20.79	18.08	16.00	14.67	13.64	12.74	11.97	11.27	10.67	10.17	9.72	9.33	8.96	8.63	8.33	8.06





Pagani et al., CODES+ISSS 2014

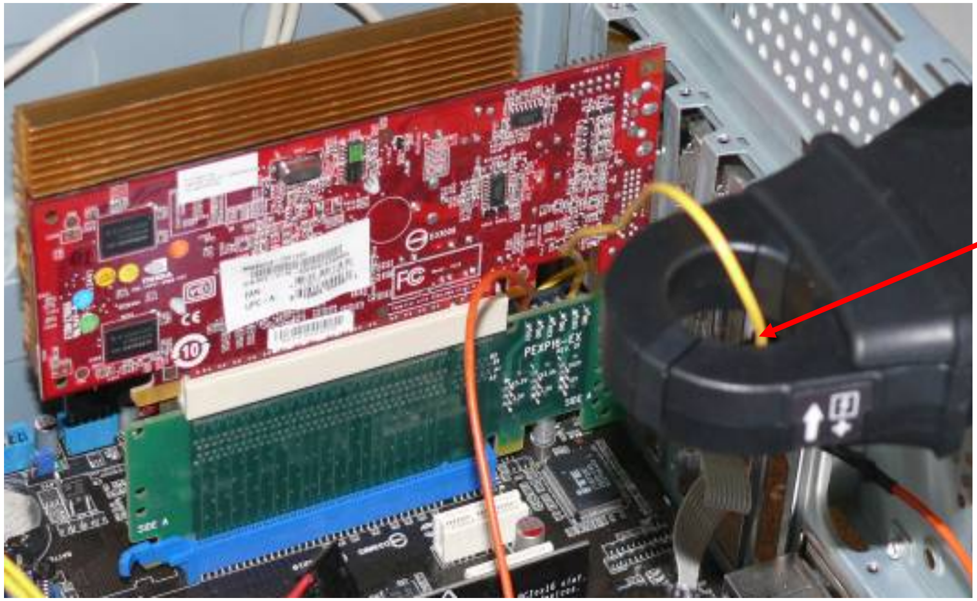
Energy and power models

How to obtain power models:

1. Measurements on real hardware
   - potentially very precise, applies only to HW at hand

2. Models
   - can be used for unavailable HW, can be very imprecise
   - typically requires tuning against measurements (parameter fitting, e.g. least squares, machine learning)

In some cases, current clamps can be used



current clamp

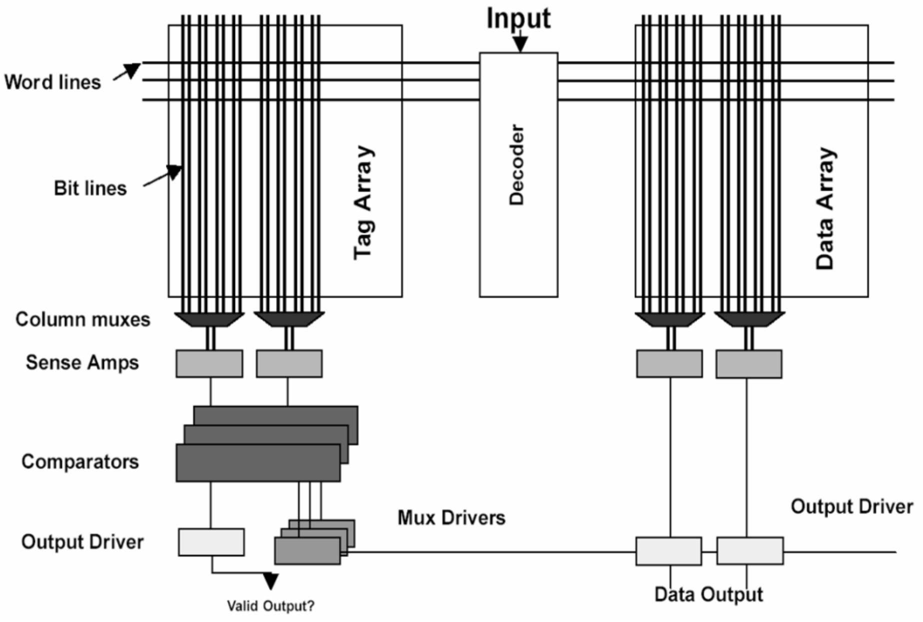
C. Timm, A. Gelenberg, P. Marwedel, F. Weichert: Energy Considerations within the Integration of General Purpose GPUs in Embedded Systems. Intern. Conf. on Advances in Distributed and Parallel Computing, 2010

C. Timm, F. Weichert, P. Marwedel, H. Müller: Design Space Exploration Towards a Realtime and Energy-Aware GPGPU-based Analysis of Biosensor Data. Computer Science - Research and Development, ENA-HPC, 2011

Analysis of memories: CACTI

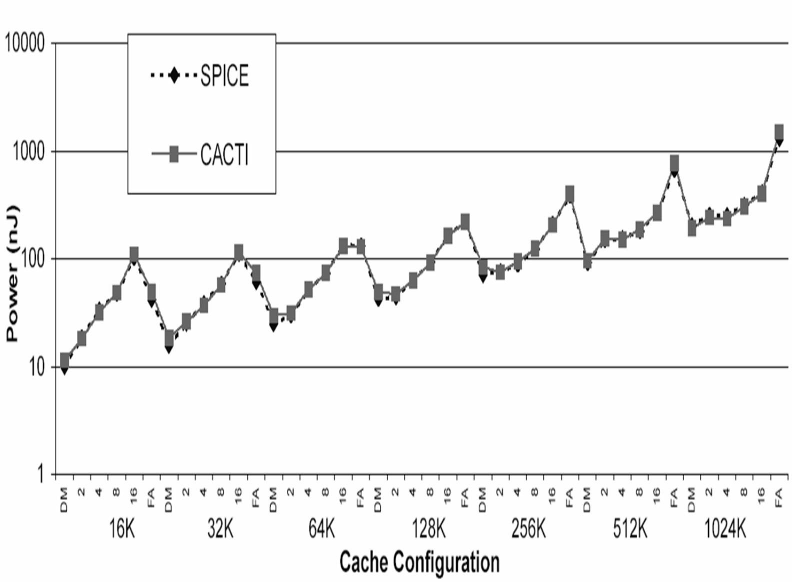
Initially designed for caches, but extended for various memories

Cache model used



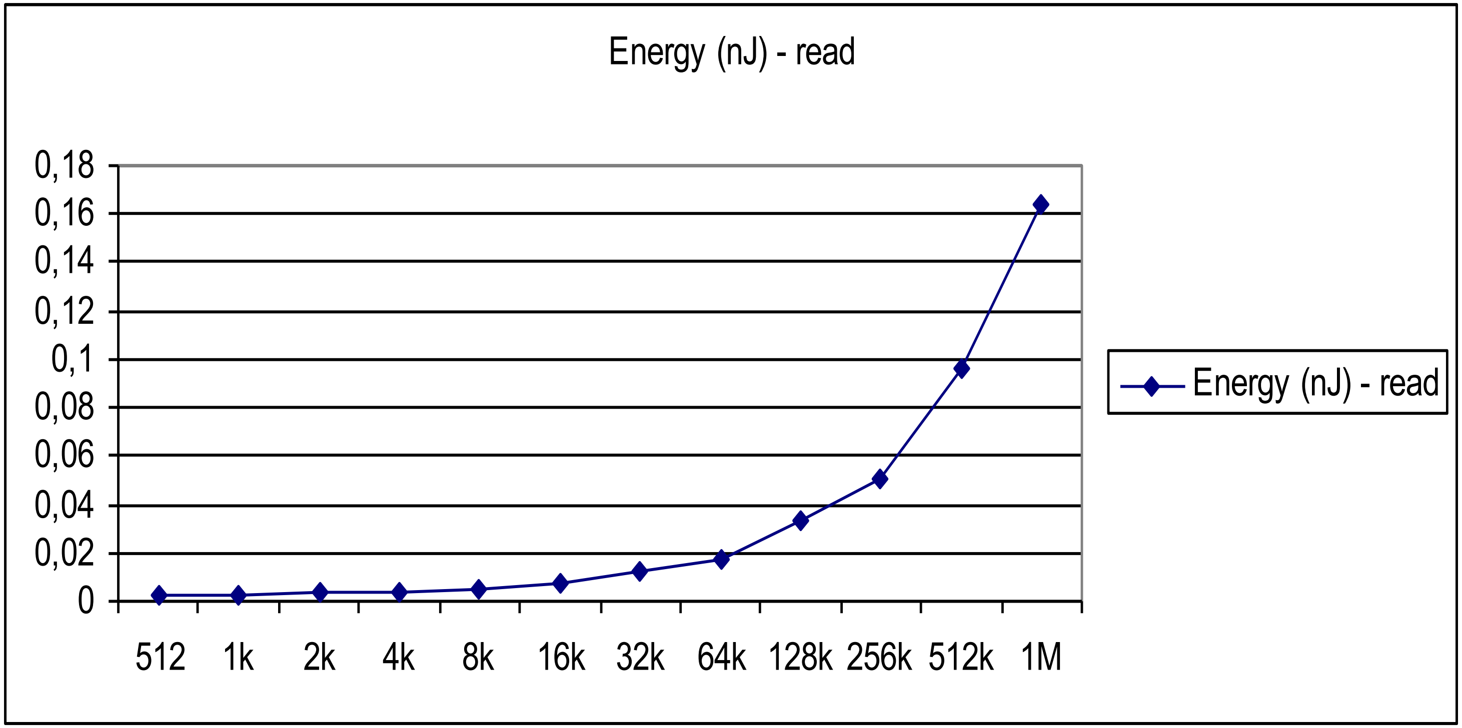
http://research.compaq.com/wrl/people/jouppi/CACTI.html

Comparison with SPICE



Energy consumption of memories

Example: CACTI / high performance Scratchpad (SRAM):



16 bit read; size in bytes; 65 nm technology

Source: Olivera Jovanovic, TU Dortmund, 2011

DRAM power

Complex DRAM models:

- http://www.micron.com/products/support/power-calc
- T. Vogelsang: Understanding the Energy Consumption of Dynamic Random Access Memories, Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture, pp. 363—374, http://dx.doi.org/10.1109/MICRO.2010.42

Analysis of processors

Various levels of detail:

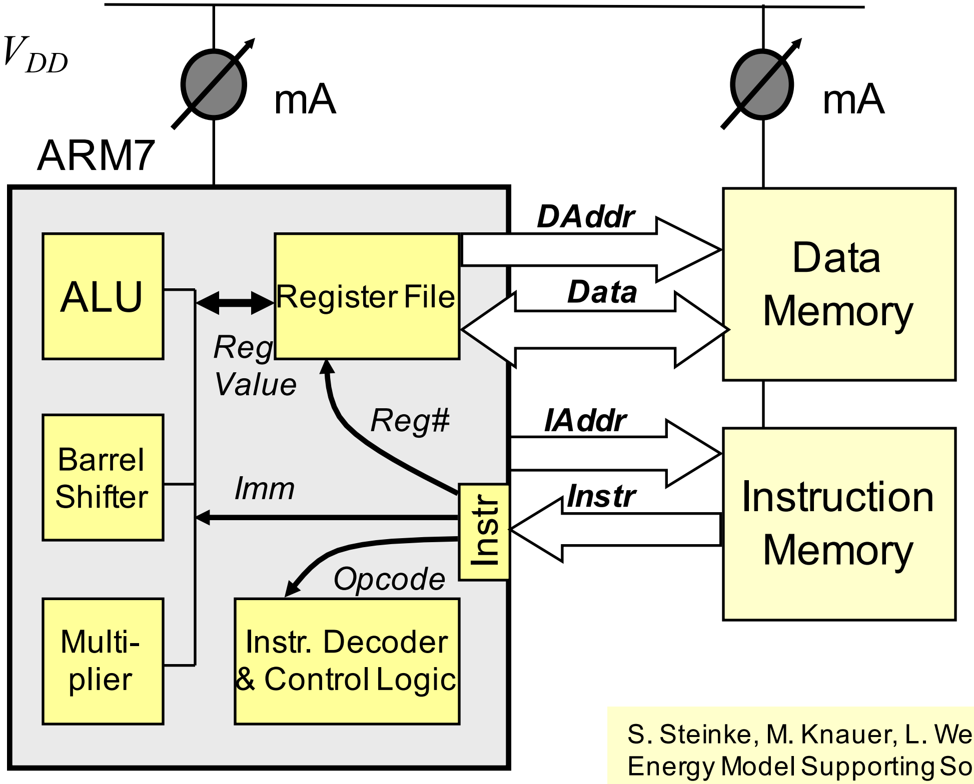
1. Tiwari (1994): effects of instructions and transitions between instructions (informal model)

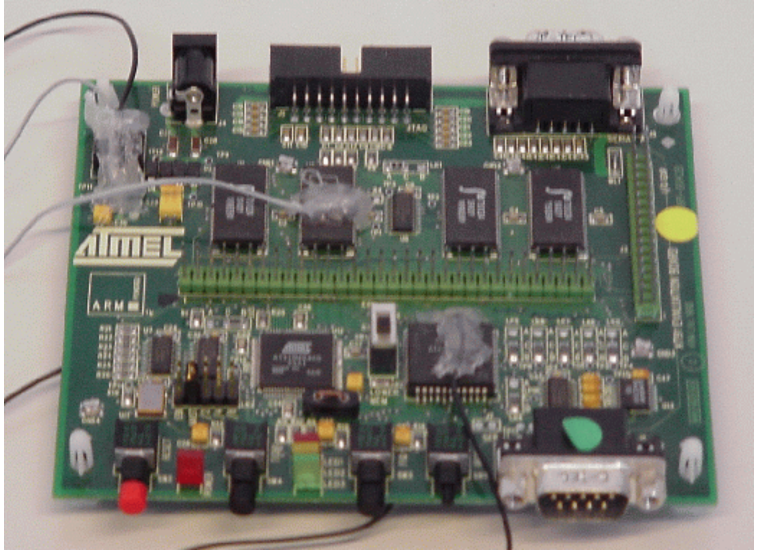
2. `Wattch` (2000): Estimation at the microarchitectural level, using a `simplescalar` model supposedly close to real HW.

Steinke's model (Level of detail between Tiwari and Wattch)

$$E_{total} = E_{cpu\_instr} + E_{cpu\_data} + E_{mem\_instr} + E_{mem\_data}$$

E.g.: ATMEL board with ARM7TDMI and ext. SRAM





S. Steinke, M. Knauer, L. Wehmeyer, P. Marwedel: An Accurate and Fine Grain Instruction-Level Energy Model Supporting Software Optimizations, Int. Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS), 2001

Example: Instruction dependent costs in the CPU

Cost for a sequence of m instructions

$$
\begin{aligned}
E_{cpu\_instr} = & \sum MinCostCPU(\mathbf{Opcode}_i) + FUCost(\mathbf{Instr}_{i-1}, \mathbf{Instr}_i) + \\
& \alpha_1 * \sum w(\mathbf{Imm}_{i,j}) + \beta_1 * \sum h(\mathbf{Imm}_{i-1,j}, \mathbf{Imm}_{i,j}) + \\
& \alpha_2 * \sum w(\mathbf{Reg}_{i,k}) + \beta_2 * \sum h(\mathbf{Reg}_{i-1,k}, \mathbf{Reg}_{i,k}) + \\
& \alpha_3 * \sum w(\mathbf{RegVal}_{i,k}) + \beta_3 * \sum h(\mathbf{RegVal}_{i-1,k}, \mathbf{RegVal}_{i,k}) + \\
& \alpha_4 * \sum w(\mathbf{IAddr}_i) + \beta_4 * \sum h(\mathbf{IAddr}_{i-1}, \mathbf{IAddr}_i)
\end{aligned}
$$

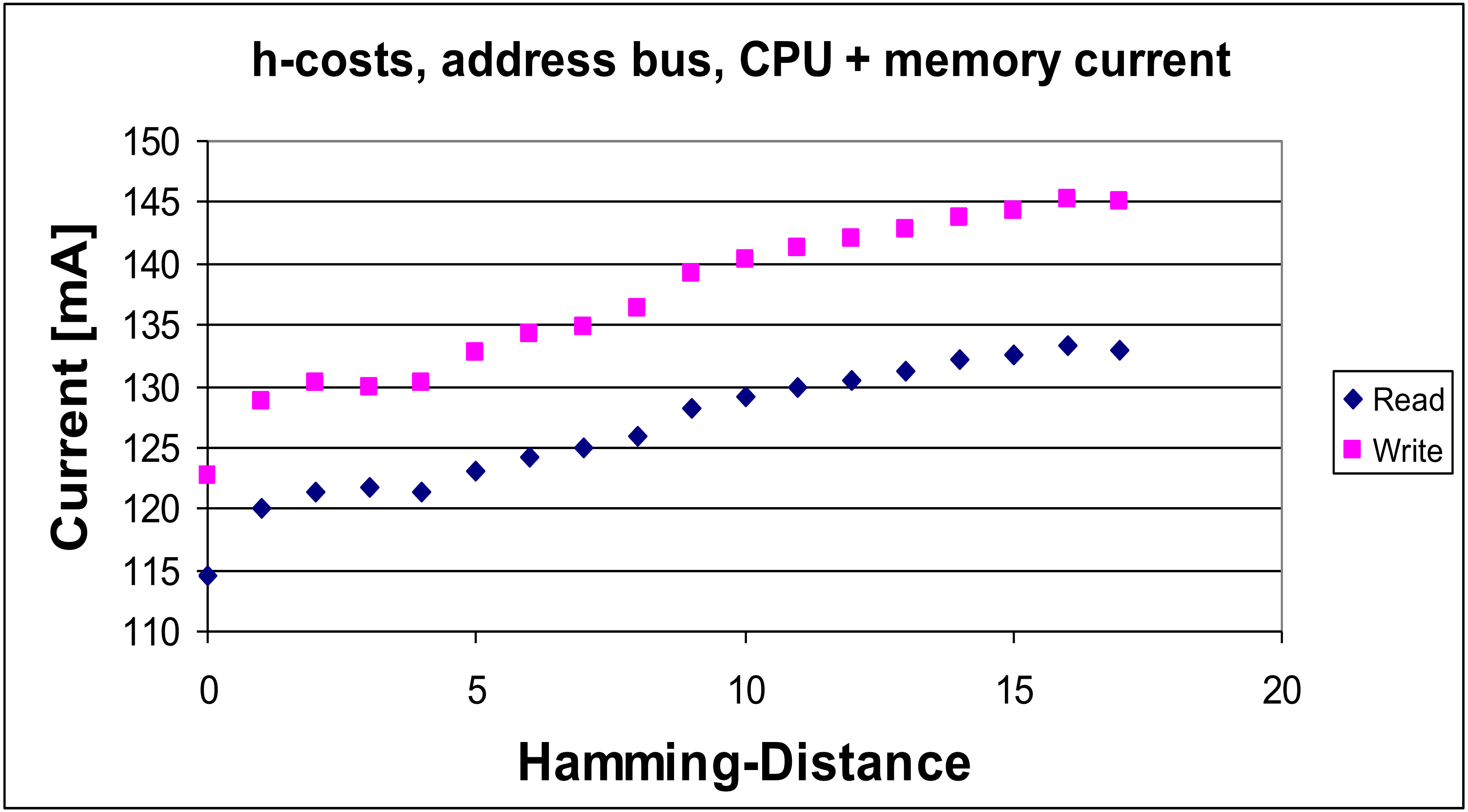
w: number of ones;

h: Hamming distance;

$FUCost$: cost of switching functional units;

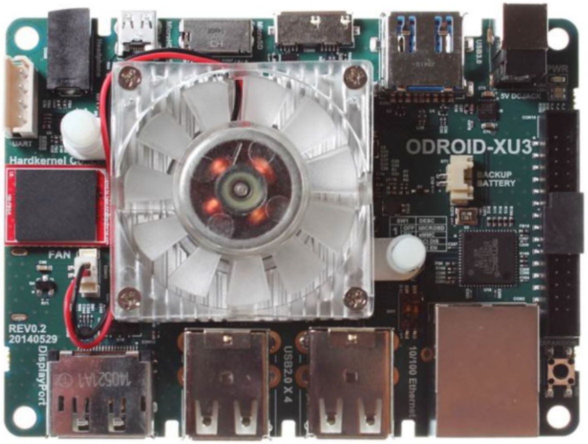
α, ß: determined through experiments.

Hamming Distance between adjacent addresses is playing major role

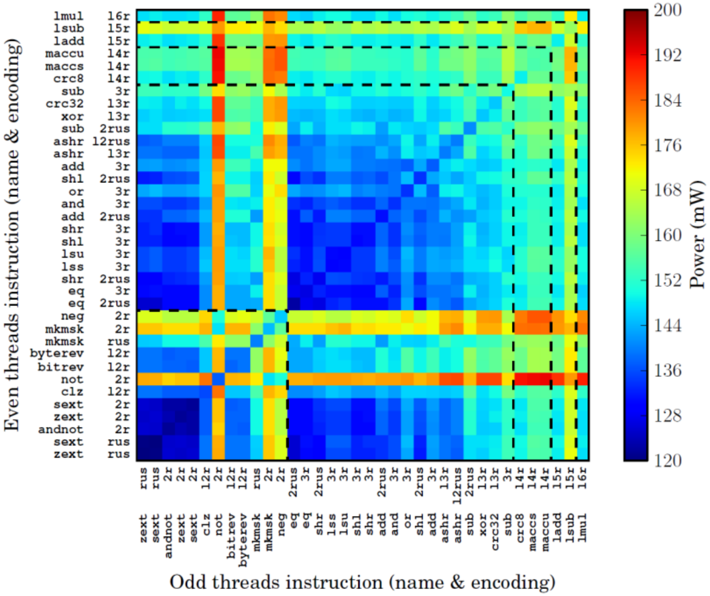


Analysis of applications

1. Analysis of single applications using real hardware like Odroid XU3, containing current sensors (e.g. Neugebauer (2015))

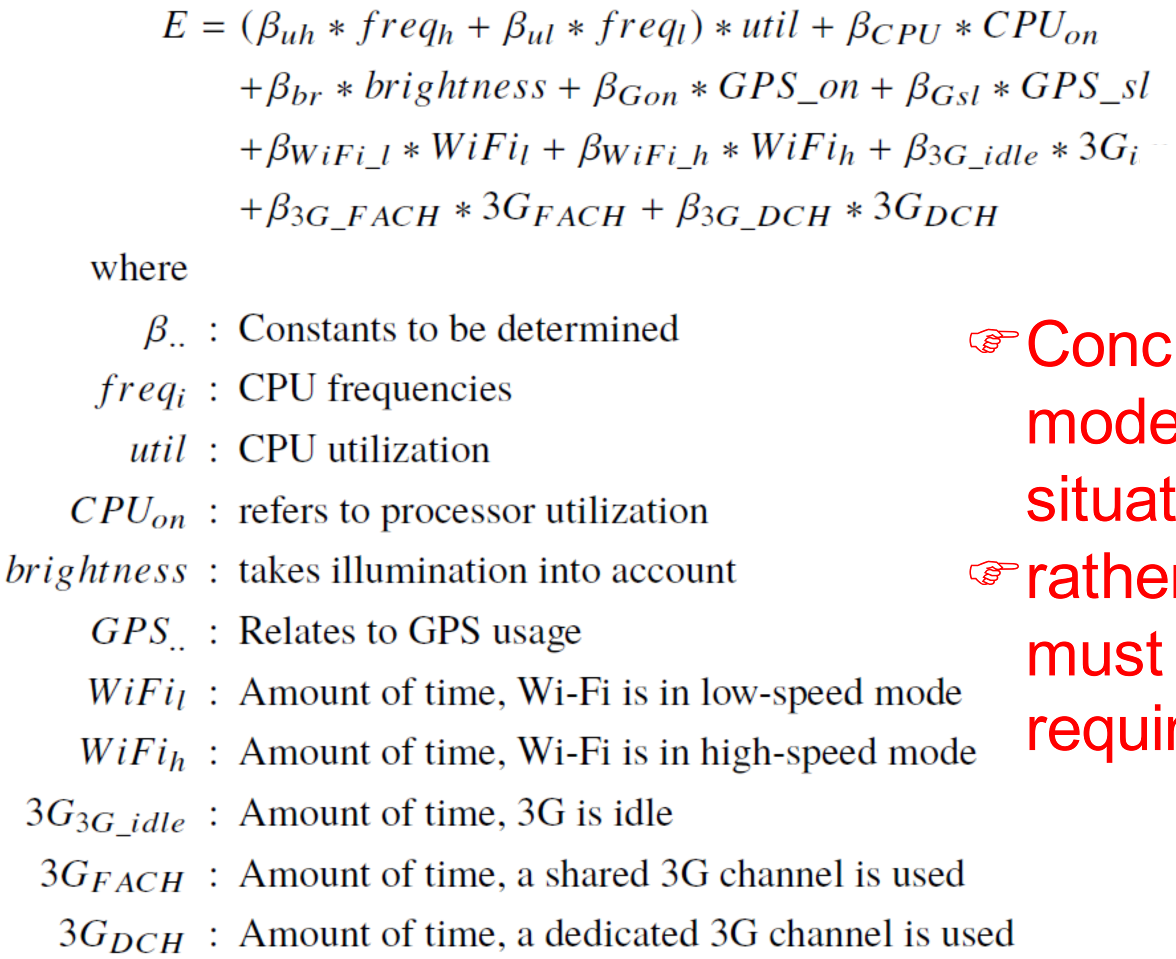


2. Analysis of multiple applications with multithreading for XMOS XS-1L (K. Eder et al.)



Complete applications

Android phone [Zhang, 2010]:



$$
\begin{aligned}
E = {} & (\beta_{uh} * freq_h + \beta_{ul} * freq_l) * util + \beta_{CPU} * CPU_{on} \\
& + \beta_{br} * brightness + \beta_{Gon} * GPS\_on + \beta_{Gsl} * GPS\_sl \\
& + \beta_{WiFi\_l} * WiFi_l + \beta_{WiFi\_h} * WiFi_h + \beta_{3G\_idle} * 3G_i \\
& + \beta_{3G\_FACH} * 3G_{FACH} + \beta_{3G\_DCH} * 3G_{DCH}
\end{aligned}
$$

where

$\beta_{..}$: Constants to be determined
$freq_i$: CPU frequencies
$util$: CPU utilization
CPU_{on} : refers to processor utilization
$brightness$: takes illumination into account
$GPS_{..}$: Relates to GPS usage
$WiFi_l$: Amount of time, Wi-Fi is in low-speed mode
$WiFi_h$: Amount of time, Wi-Fi is in high-speed mode
$3G_{3G\_idle}$: Amount of time, 3G is idle
$3G_{FACH}$: Amount of time, a shared 3G channel is used
$3G_{DCH}$: Amount of time, a dedicated 3G channel is used

- Conclusion: no “one model fits all purposes” situation;
- rather, level of detail must be adjusted to requirements

Evaluation of energy consumption: Challenges

- Energy consumption hardly predictable from the source code, due to difficult to predict impact of compiler & linker
- Small variations of the code can lead to large variations of the energy consumption
  - ex. notorious examples
  - Example: shifting code in memory by one byte

- Energy consumption must be predicted from executable code
- Energy consumption might even depend on HW instance

Summary

Hardware in a loop

- Sensors
- Discretization
- Information processing
  - Importance of energy/power efficiency
  - Special purpose HW very expensive
  - Energy/power efficiency of processors
  - Code size efficiency
  - Run-time efficiency
  - MPSoCs
- D/A converters
- Actuators