ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Multitherman
EU FP7 ERC Project
(GA n. 291125)

PHIDIAS
EU FP7 Project
(GA n. 318013)

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Energy management policy for ultra-low power devices

Andrea Bartolini, PhD

Università di Bologna & ETHZ

NiPS Summer School Perugia 2014

TECHNISCHE UNIVERSITÄT KAISERSLAUTERN

UNIVERSITY OF CALIFORNIA · SAN DIEGO

BOSTON UNIVERSITY

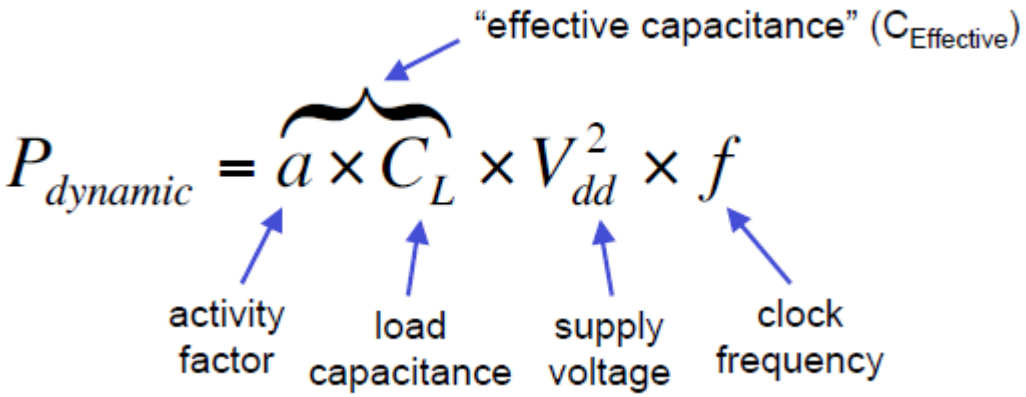
Energy Management – key concept

1. Computational devices are designed for the peak performance
2. But mostly are used in average condition
3. Energy management policies adapt with different granularities and with different mean the system performance and power to the user requested service level
4. As the number of resources increase the same functionality can be achieved with different HW composition. The energy management selects the one that preserve performance while minimizing the energy consumption

Outline

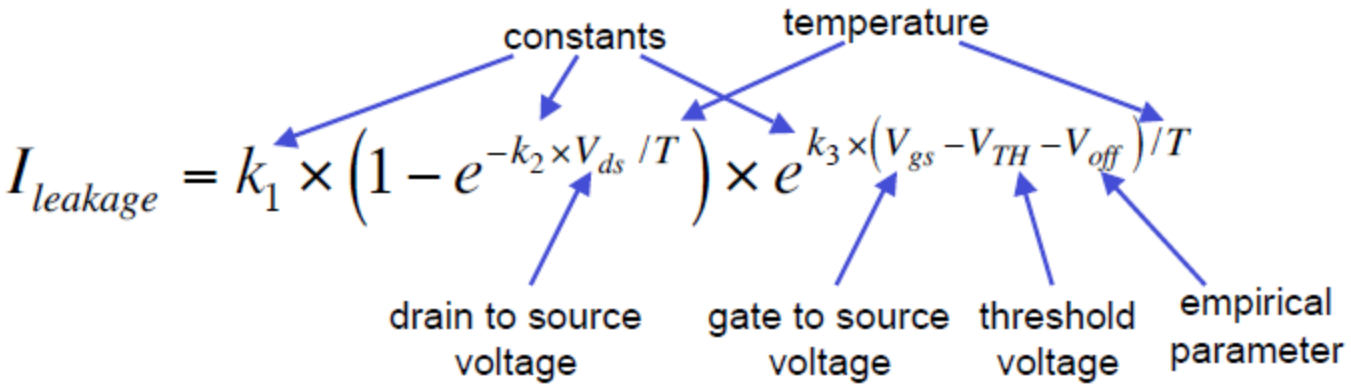
- **Power in digital systems**
- Energy Management
- Monitoring
- Task Allocation
- Reconfiguration

Dynamic Power



- **Linear ↓ with ↓ $C_{Effective}$**
- **Linear ↓ with ↓ f**
- **Quadratic ↓ with ↓ V_{dd}**
- **Cubic ↓ with ↓ both V_{dd} and f**

Sub-threshold Leakage Current



- **Exponential ↓ with ↓ V_{ds}**
- **Exponential ↓ with ↑ V_{TH}**
- **Exponential ↓ with ↓ T**

Alpha-Power Thermal Model

Delay:

$$D_p = \frac{C_{out} V_{dd}}{I_{ON}} = \frac{C_{out} V_{dd}}{\mu(T)[V_{dd} - V_{th}(T)]^{\alpha}}$$

Carrier Mobility:

$$\mu(T) = \mu(T_o)\left(\frac{T_o}{T}\right)^m$$

Threshold Voltage:

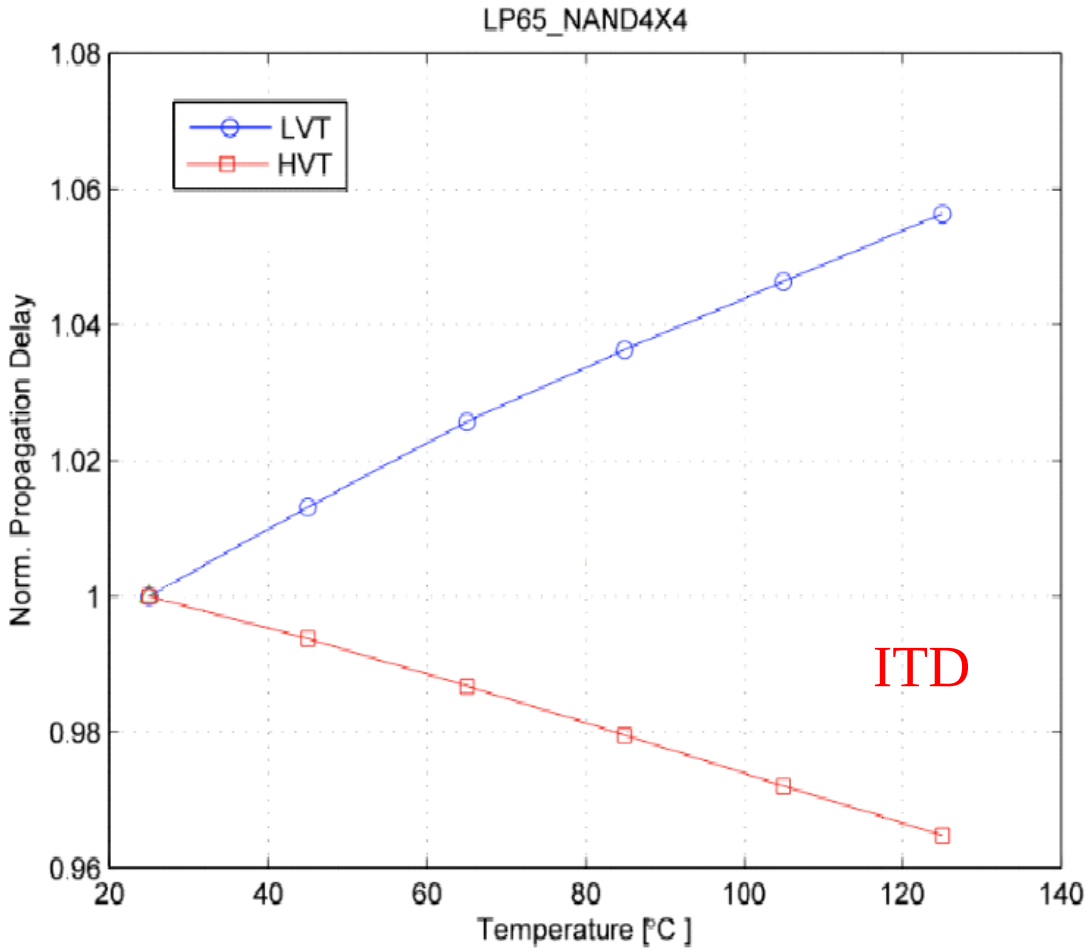
$$V_{th} = V_{th}(T_0) - k(T - T_0)$$

$T \uparrow$	$\mu \downarrow$	$V_{th} \downarrow$

Delay Trend

- For wires, the resistivity is linearly dependent from T
  - Delay increases as T increases
- For LVT cells ($V_{dd} >> V_{th}$)
  - μ dominates w.r.t. V_{th}
  - Delay Increases as T increases
- For HVT cells ($V_{dd} \approx V_{th}$)
  - V_{th} dominates w.r.t μ
  - Delay decreases as T increases, Indirect Temperature Dependence (ITD)

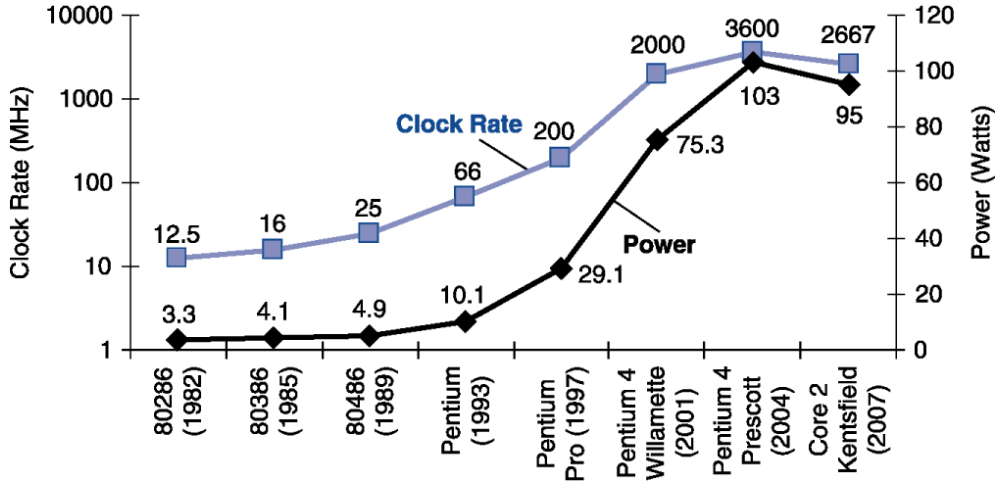
Thermal Behavior of CMOS gates

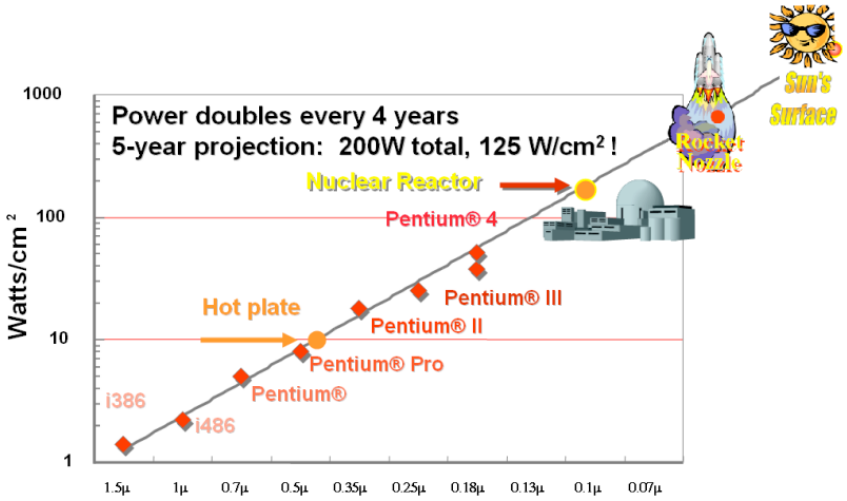


Power Wall

Here is a Clue to the Problem

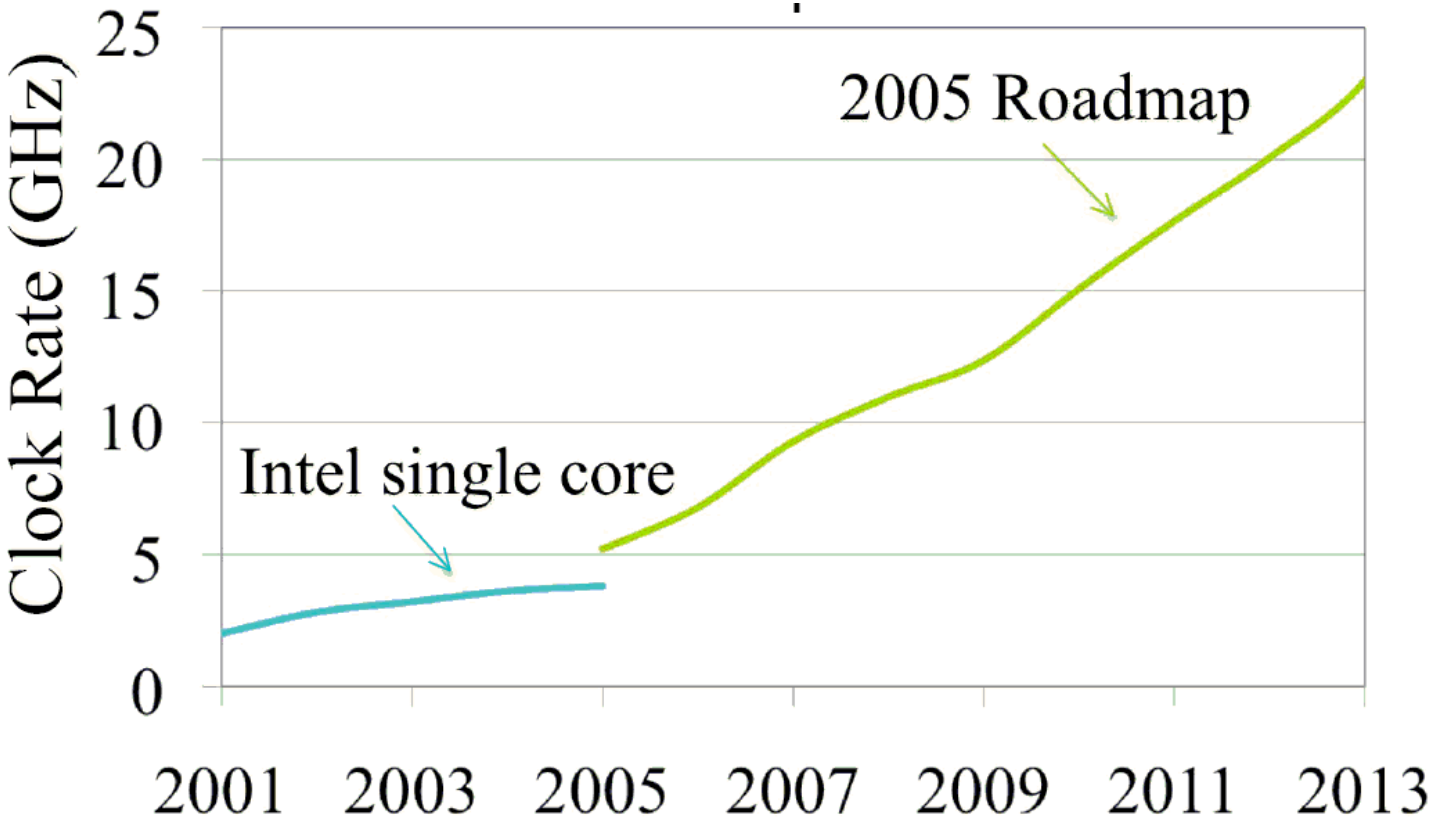
The problem is now called “the Power Wall”. It is illustrated in this figure, taken from Patterson & Hennessy.





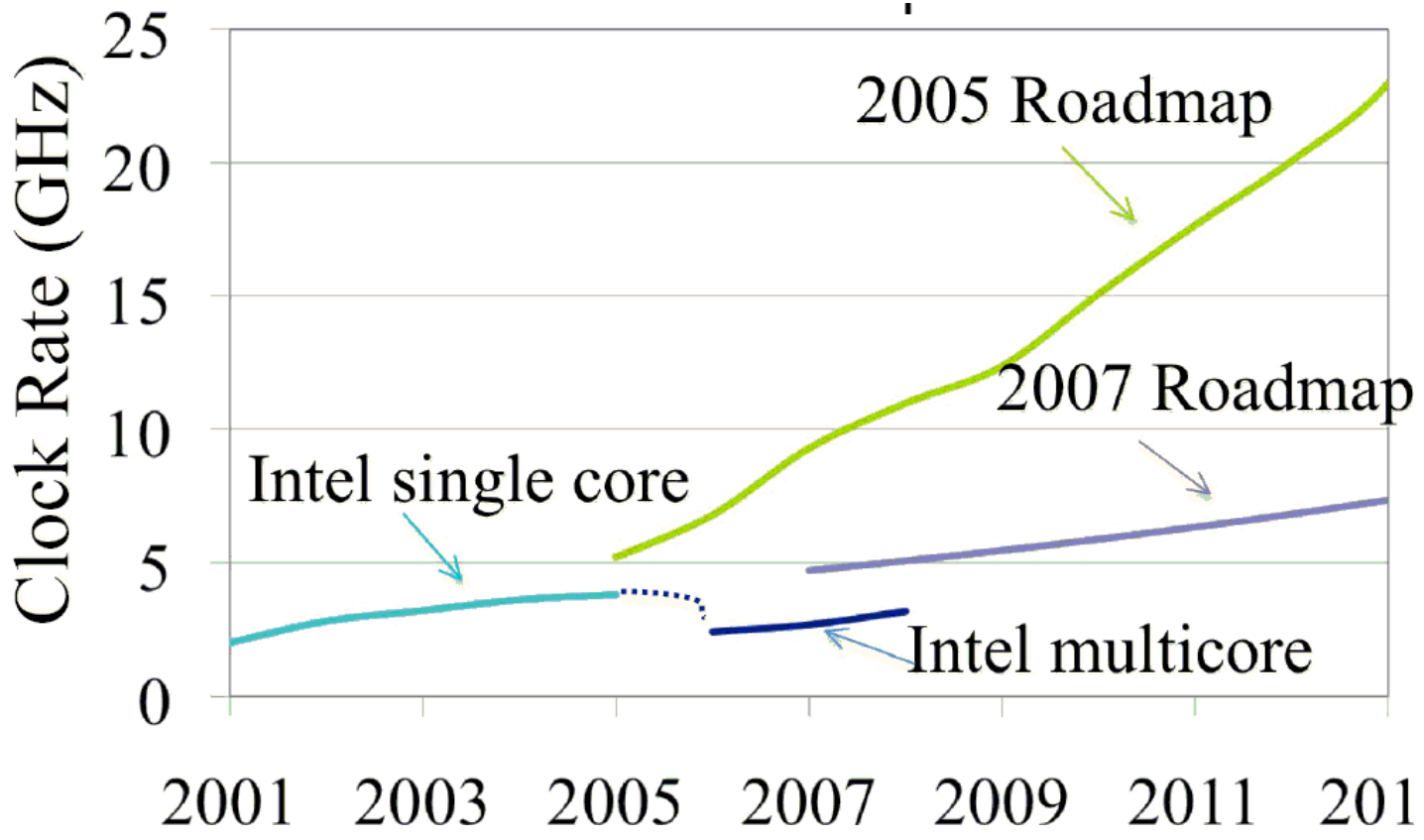
- The design goal for the late 1990’s and early 2000’s was to drive the clock rate up. This was done by adding more transistors to a smaller chip.
- Unfortunately, this increased the power dissipation of the CPU chip beyond the capacity of inexpensive cooling techniques

Roadmap for CPU Clock Speed: Around 2005



Here is the result of the best thought in 2005. By 2015, the clock speed of the top “hot chip” would be in the 12 – 15 GHz range.

The CPU Clock Speed Roadmap (A Few Revisions Later)

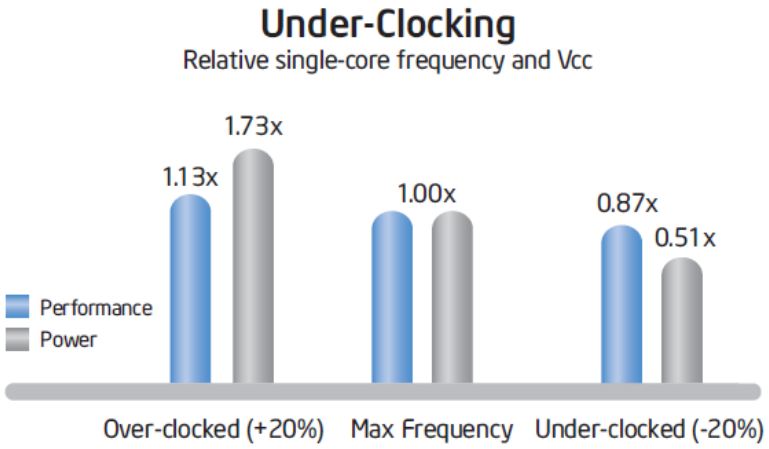


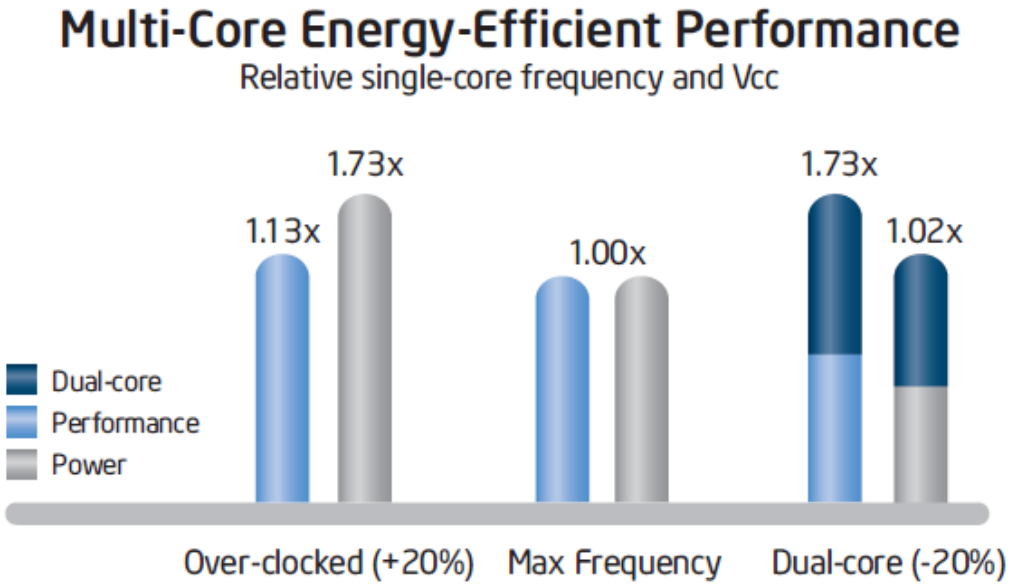
This reflects the practical experience gained with dense chips that were literally "hot"; they radiated considerable thermal power and were difficult to cool.
Law of Physics: All electrical power consumed is eventually radiated as heat.

The MultiCore Approach

Multiple cores on the same chip

- Simpler
- Slower
- Less power demanding





Outline

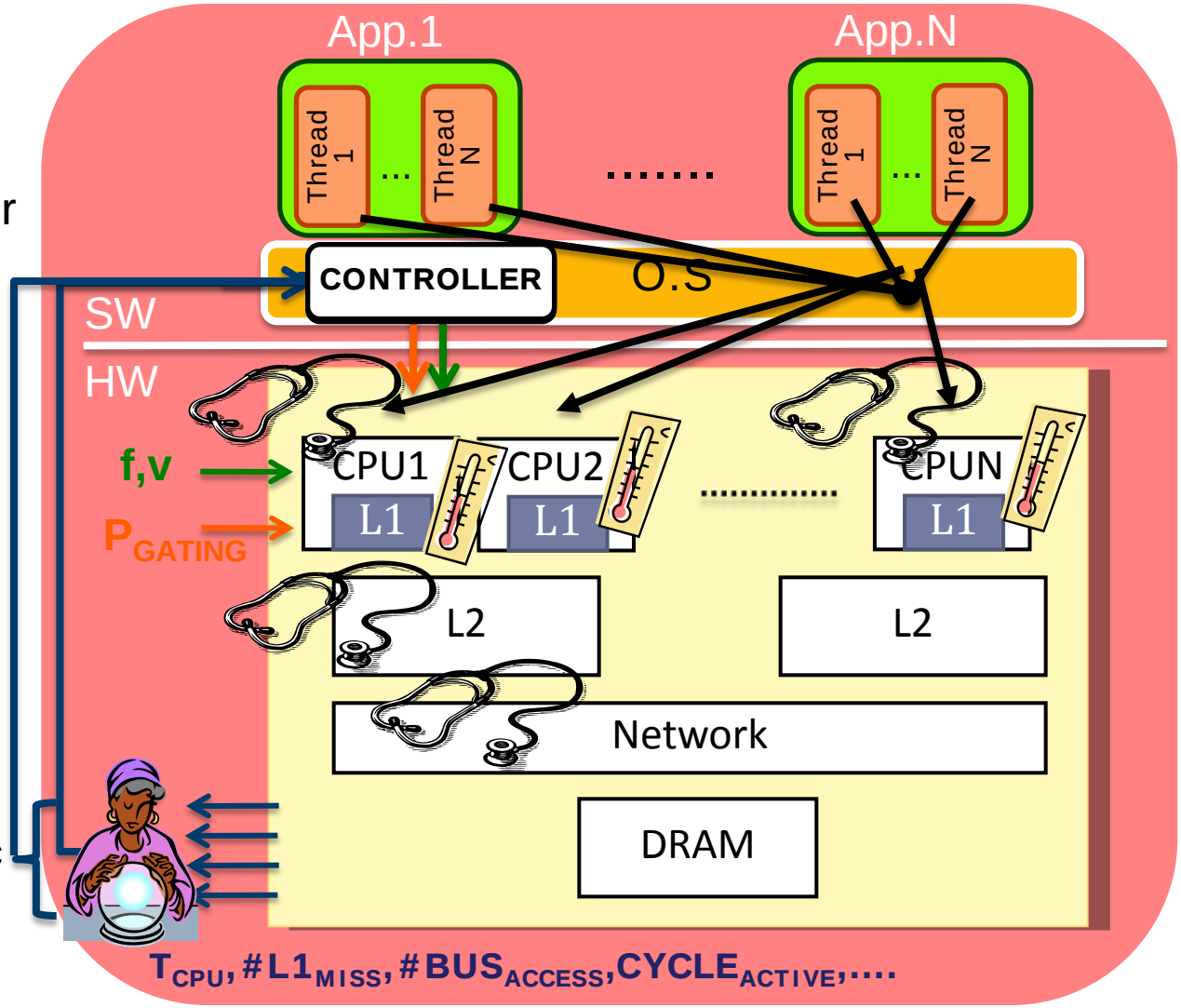
- Power in digital systems
- **Energy Management**
- Monitoring
- Task Allocation
- Reconfiguration

Attacking Dynamic Power

- **Dynamic Voltage and Frequency Scaling (DVFS)**
  - **Reduce voltage, frequency, or both**
  - **Exploit slack in application execution**
  - ***Cubic* dynamic power savings**
- **Reduce effective switching capacitance**
  - **Exploit idle or underutilized hardware resources**
  - **Match hardware resources to application behavior**
  - ***Linear* dynamic power savings**
  - ***Complementary* to DVFS**

General Architecture

- System
- Sensors
  - Performance counter - PMU
  - Core temperature
- Actuator - Knobs
  - ACPI states
    - P-State → DVFS
    - C-State → P_{GATING}
  - Task allocation
- Controller
  - Reactive
    - Threshold/Heuristic
    - Controller theory
  - Proactive
    - Predictors



State-of-the-art

Dynamic Voltage Frequency Scaling –Two approach in litterature:

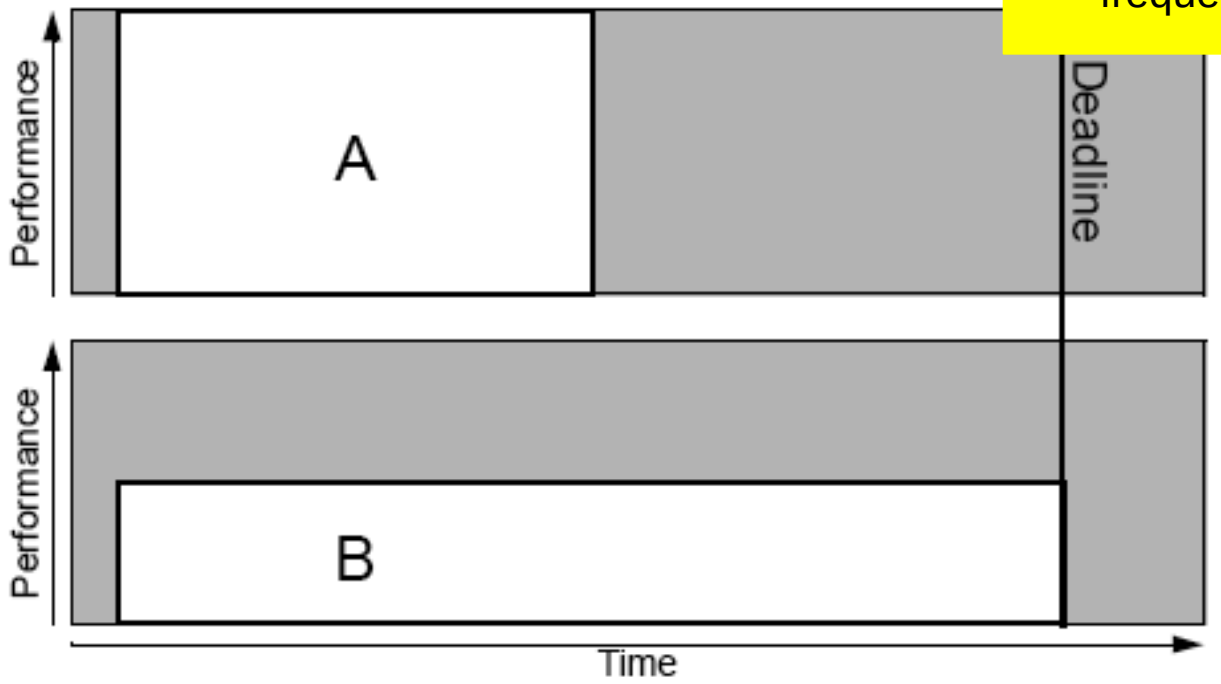
- Power Budgetting and Capping[1]:
  - *Use built-in power meters as inputs to close-loop feedback controllers for constraining the power consumption to a given budget by reducing the cores clock frequencies.*
  - Pros : Avoid power overshots;
  - Contra : Requires power meters, Significant performance penalties;
    - Capping mostly cpu-bound task, more sensitive to performance penalties
- Energy Minimizzation [1][2][3][4][5]:
  - Identifing program phases
  - Scale the frequency to reduce the stall cycle slack and idleness.
  - Low performance overhead

[1] *Z. Wang et al. Feedback Control Algorithms for Power Management of Servers* 2008.
[2] *K. Flautner et al. Vertigo: automatic performance-setting for Linux 2002.*
[3] *G. Dhiman, Dynamic voltage frequency scaling for multi-tasking systems using online learning, 2007.*
[4] *W.Y. Liang Memory-aware dynamic voltage and frequency prediction for portable devices 2008*
[5] *A.Bartolini Thermal and energy management of high-performance multicores 2013*

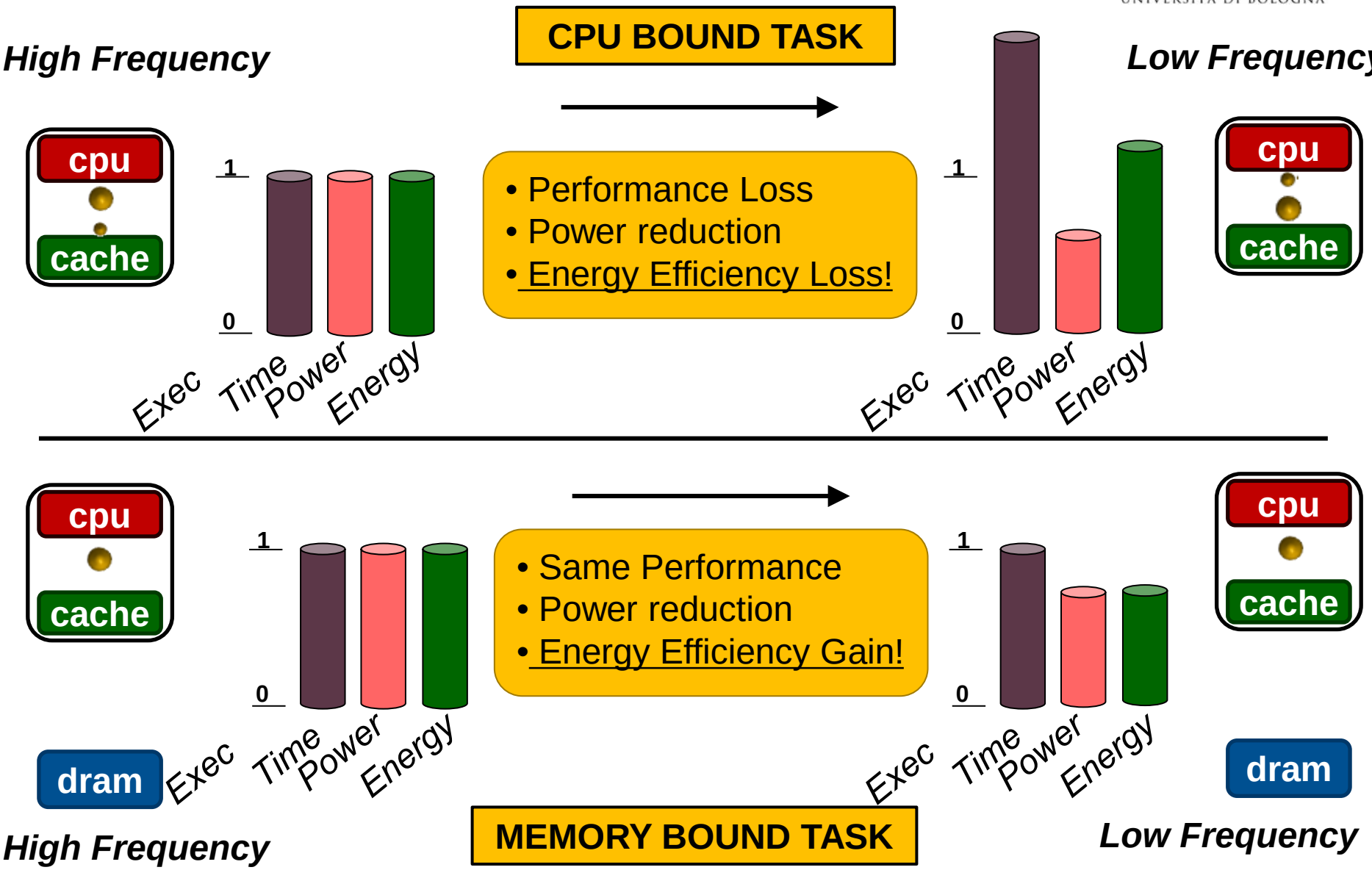
DVFS – with deadline or “on-demand governor”

Key idea: Exploit *slack* by scaling V & f to run evenly across a time quantum

Linux on-demand governor:
frequency ~ cpu-load

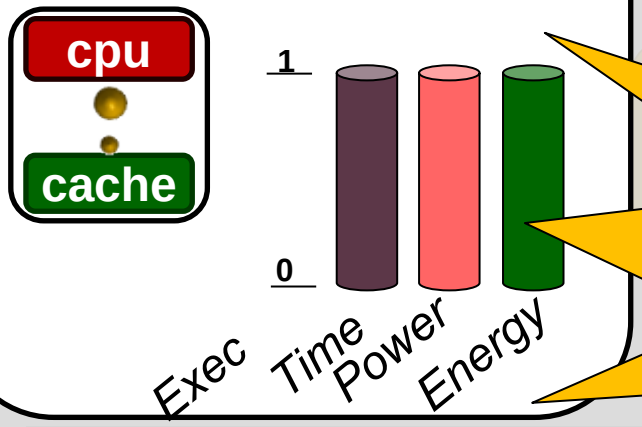


DVFS – Memory slack



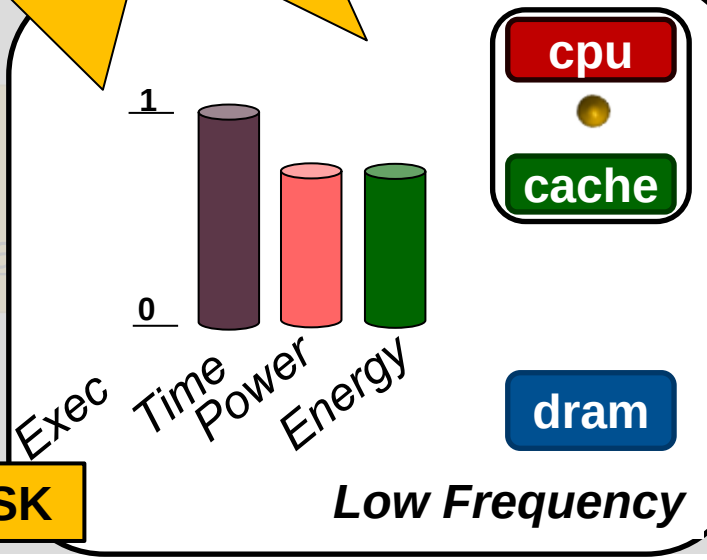
CPU BOUND TASK

High Frequency



Memory Slack - Aware

- **Power Saving**
- **No performance Loss**
- **Higher Energy Efficiency**

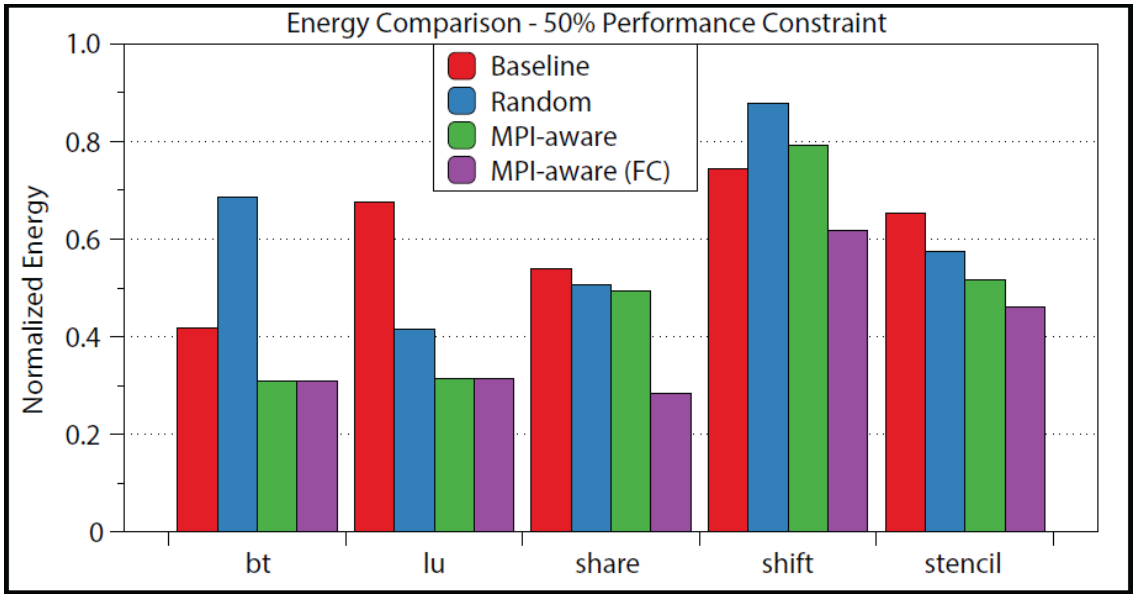


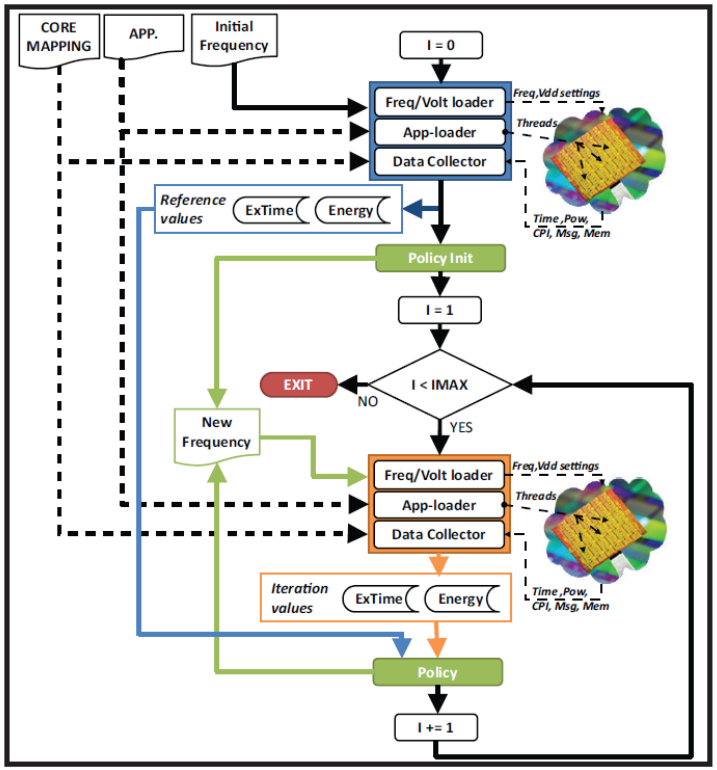
Low Frequency

MEMORY BOUND TASK

MP-aware DVFS

- Multi-node system - exploit communication slack
  - Tested on the intel single cloud computer
  - Iterative search the minimum energy
  - Four searching policy
    - MP agnostic





A.Bartolini «Quantifying the Impact of Frequency Scaling on the Energy Efficiency of the SCC» 2012

Near Threshold Computing

Moore's Law Twilight Era

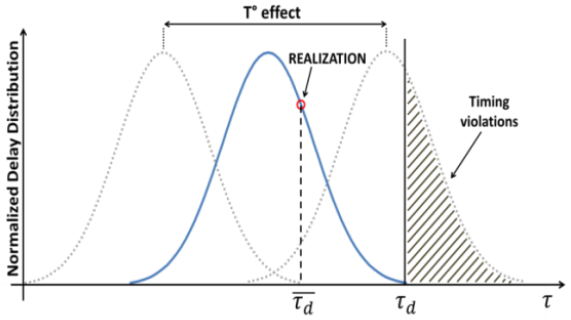
NEAR THRESHOLD COMPUTING

FUNCTIONAL FAILURE ?

65nm SRAM cell @ NTC
failure rate : 5 OM higher

memory V_{dd} ↑

VARIABILITY ?



ENERGY EFFICIENCY

10x improvement!

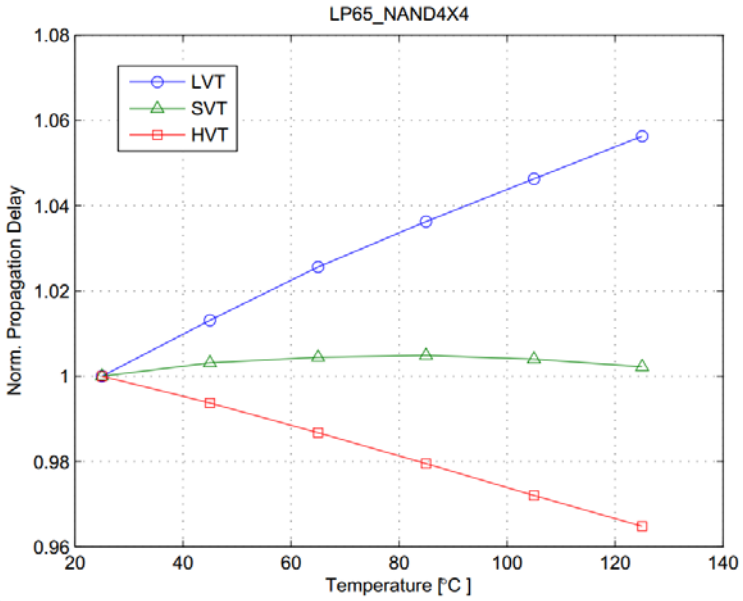
PERFORMANCE LOSS ?

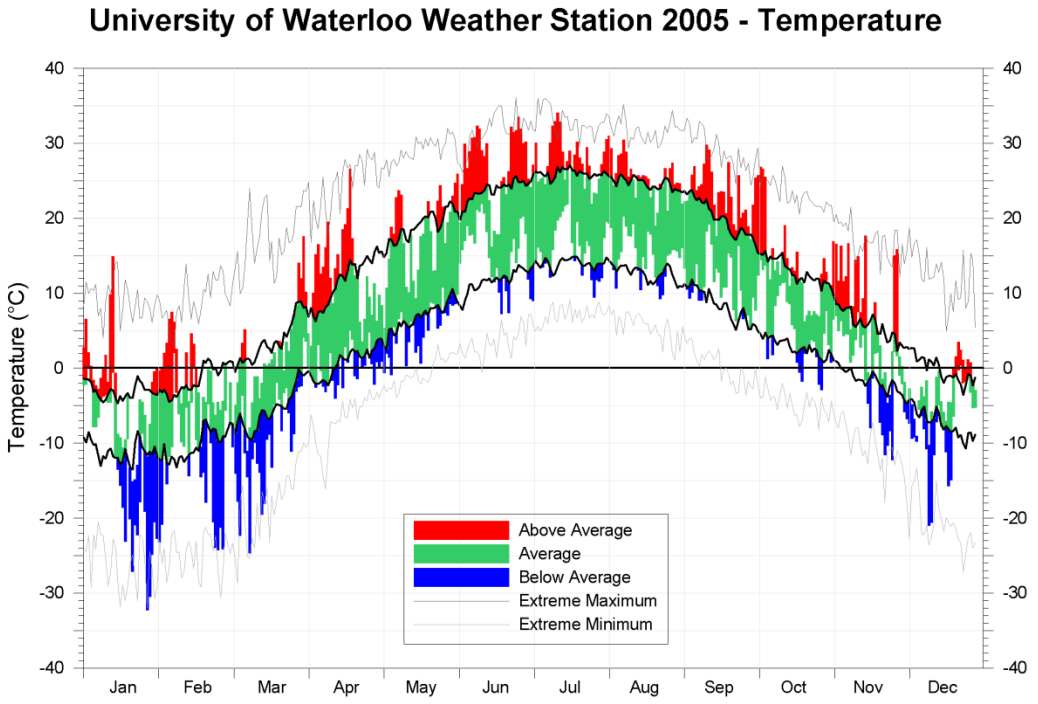
multi-core parallel architectures

NTC pushes the architecture towards a topology in which several processing elements communicate with each other through a fast shared L1 memory system

Variability and Ambient Temperature

- Critical path delay is affected by temperature [C10]
- **ULP devices** in **NTC** are intrinsically safe from self-heating
  - Die temperature is hot-spot free → $\mathbf{T_{system} \approx T_{amb}}$
- Ambient temperature has **strong variations**
  - daily/seasonal fluctuations
  - indoor/outdoor transitions



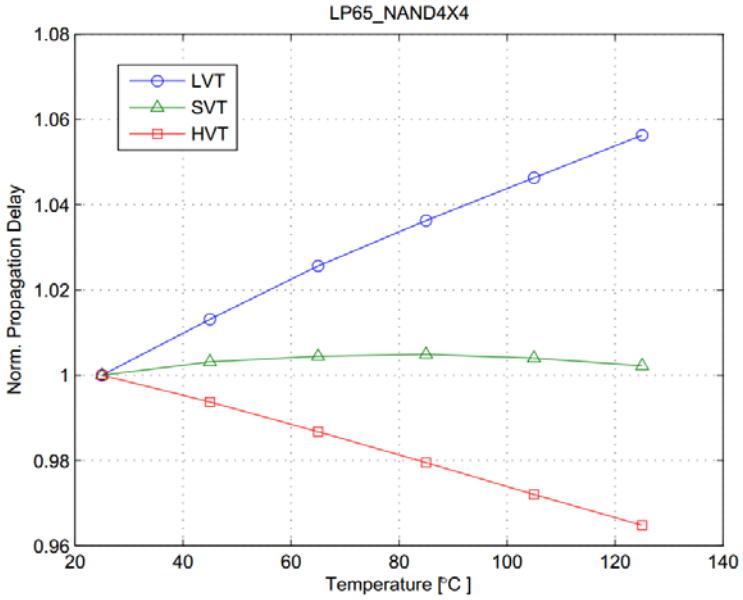


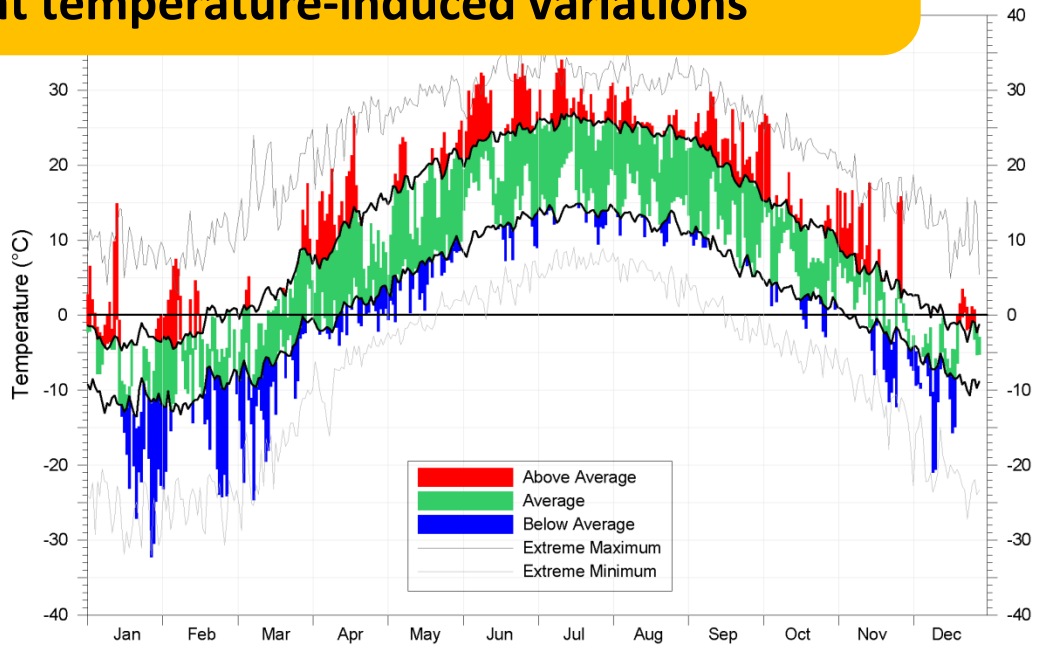
[C10] Calimera et al., *Reducing Leakage Power by Accounting for Temperature Inversion Dependence in Dual-Vt Synthesized Circuits*

Variability and Ambient Temperature

- Critical path delay is affected by temperature [C10]
- **ULP devices** in **NTC** are intrinsically safe from self-heating
  - Die temperature is hot-spot free → $T_{system} \approx T_{amb}$
- **A**

Performance Variability can not be effectively addressed
Only by adopting static solutions requiring reactive runtime solutions
To compensate ambient temperature-induced variations

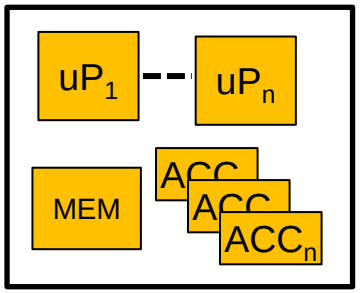




[C10] Calimera et al., *Reducing Leakage Power by Accounting for Temperature Inversion Dependence in Dual-Vt Synthesized Circuits*

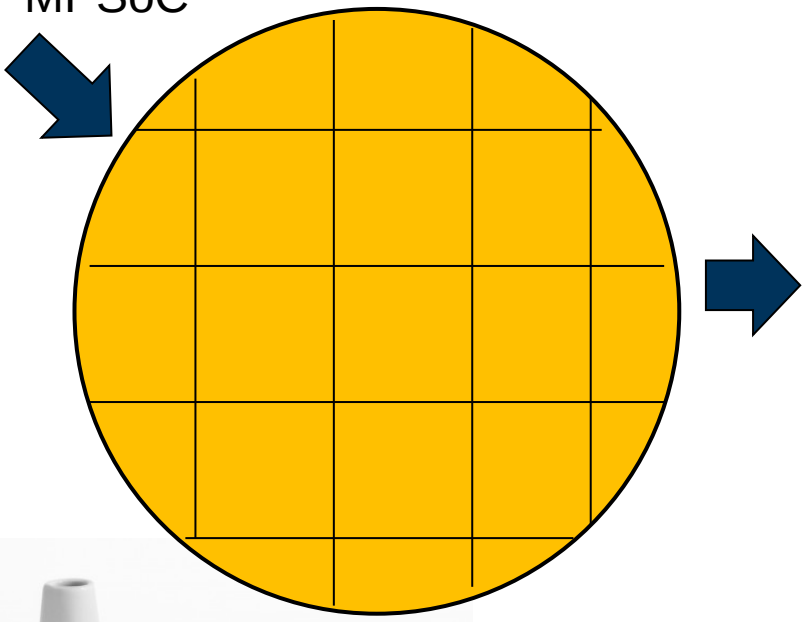
Variability

@ design

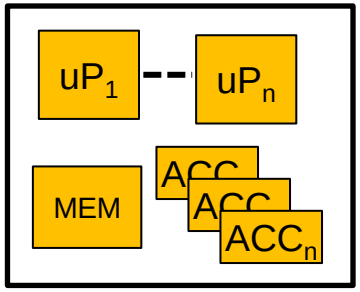


MPSoC

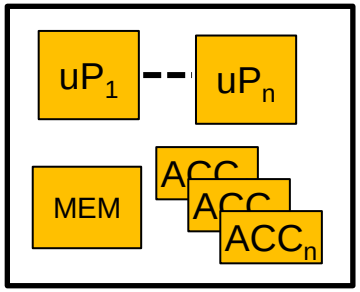
- Ideally



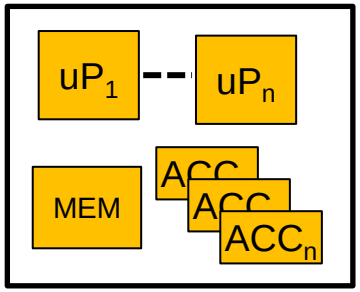
@ post-Silicon



MPSoC#1



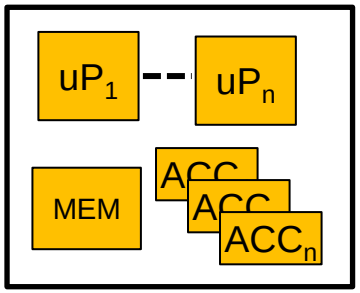
MPSoC#2



MPSoC#N

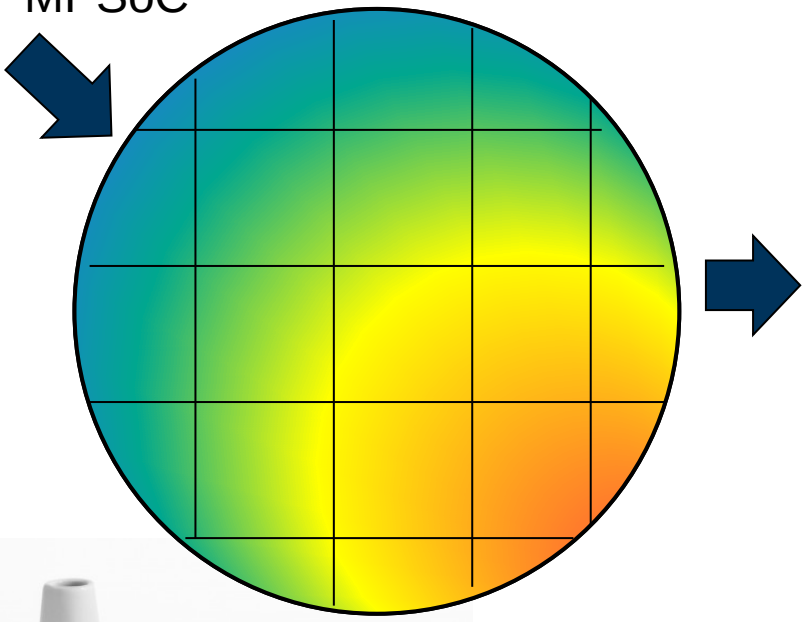
Variability

@ design

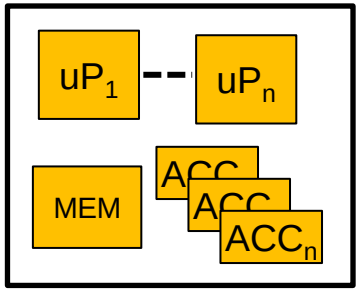


MPSoC

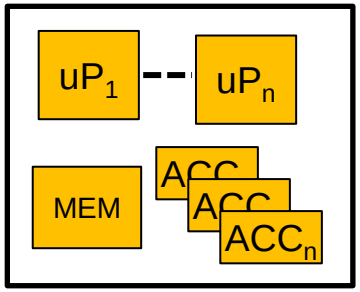
- Ideally
- Silicon Variability



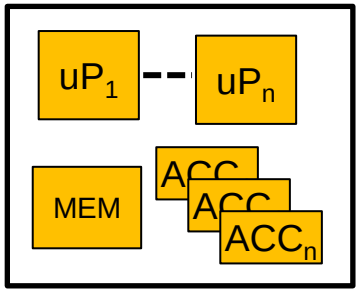
@ post-Silicon



MPSoC#1



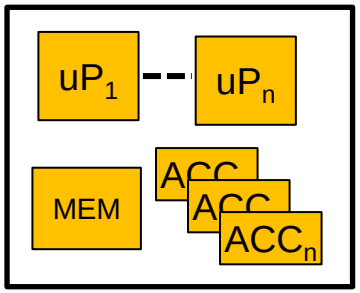
MPSoC#2



MPSoC#N

Variability

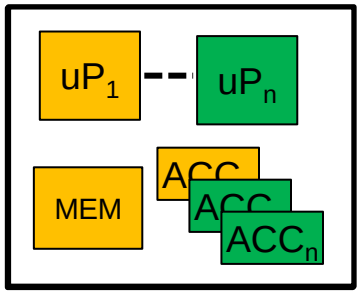
@ design



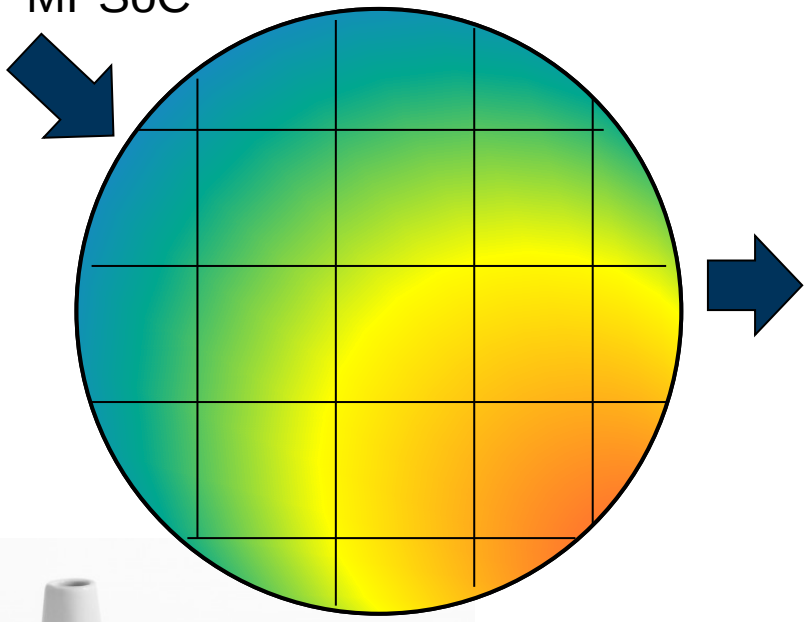
MPSoC

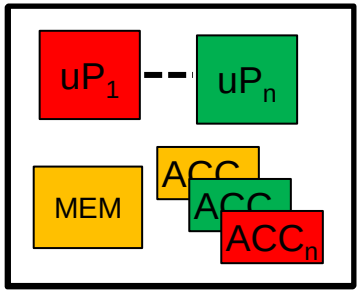
- Ideally
- Silicon Variability
- PVT Variation

@ post-Silicon

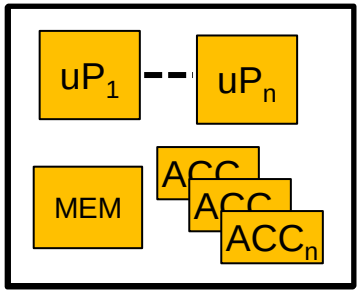


MPSoC#1





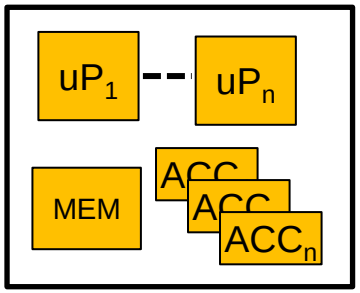
MPSoC#2



MPSoC#N

Variability

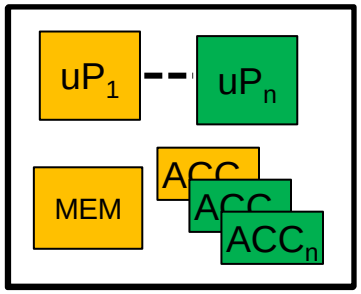
@ design



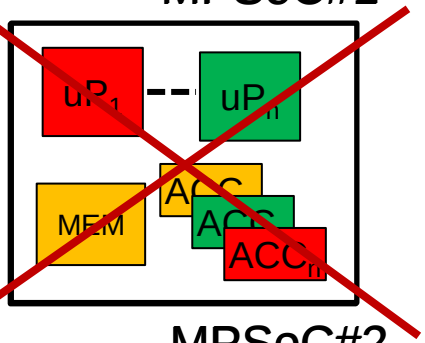
MPSoC

- Ideally
- Silicon Variability
- PVT Variation
- Yeld Loss

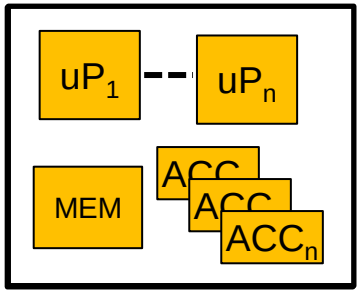
@ post-Silicon



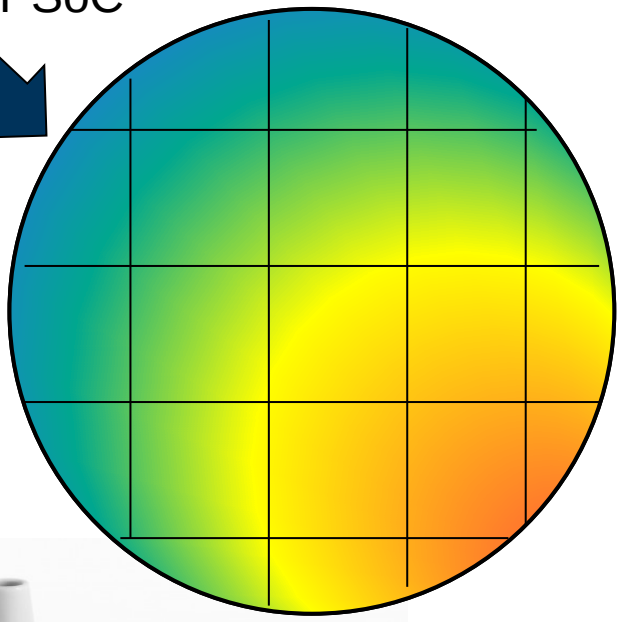
MPSoC#1



MPSoC#2

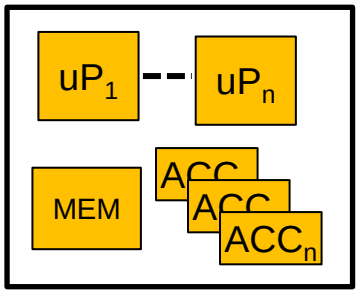


MPSoC#N



Variability

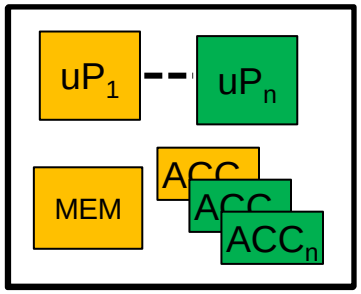
@ design



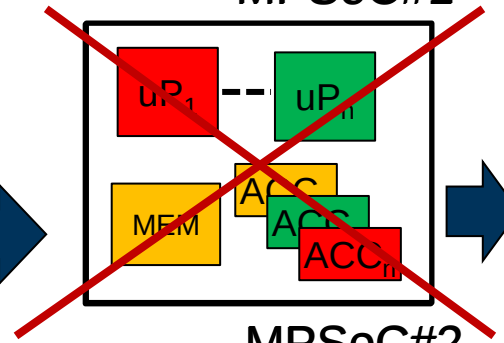
MPSoC

- Ideally
- Silicon Variability
- PVT Variation
- Yeld Loss

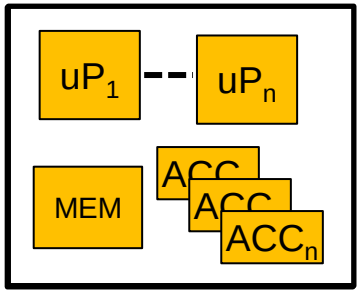
@ post-Silicon



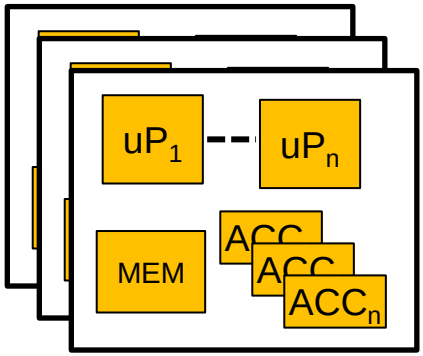
MPSoC#1

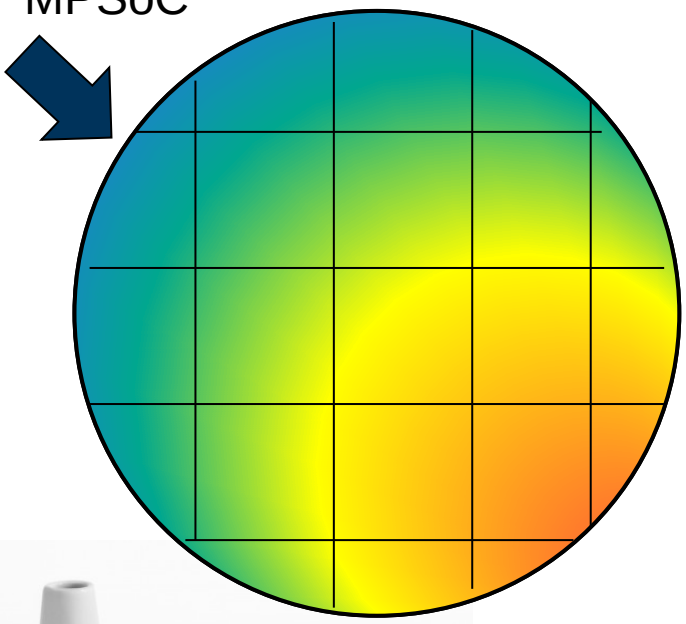


MPSoC#2



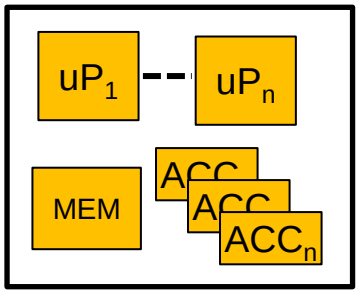
MPSoC#N





Variability

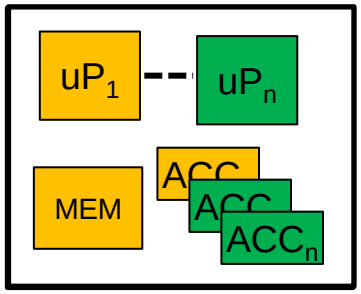
@ design



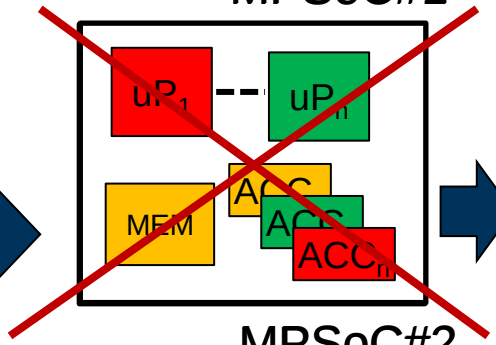
MPSoC

- Ideally
- Silicon Variability
- PVT Variation
- Yeld Loss
- Performance Loss

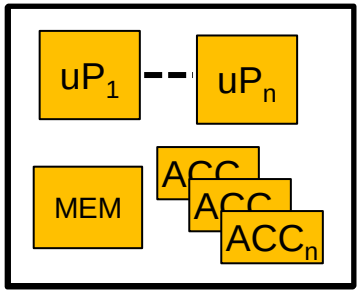
@ post-Silicon



MPSoC#1

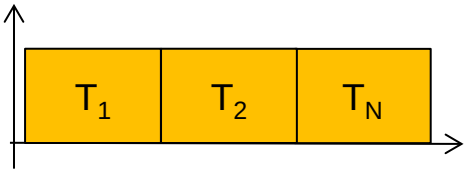


MPSoC#2



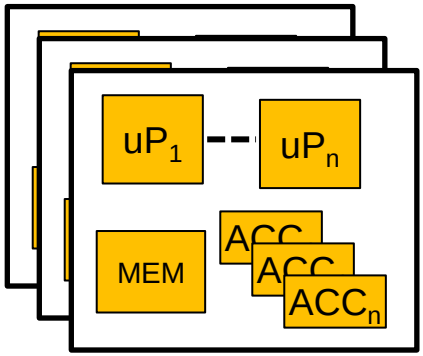
MPSoC#N

@ run-time



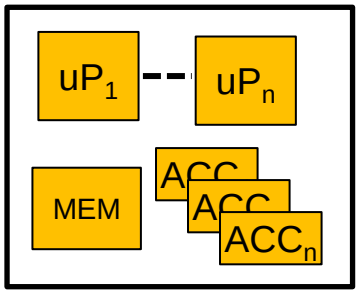
SW

HW



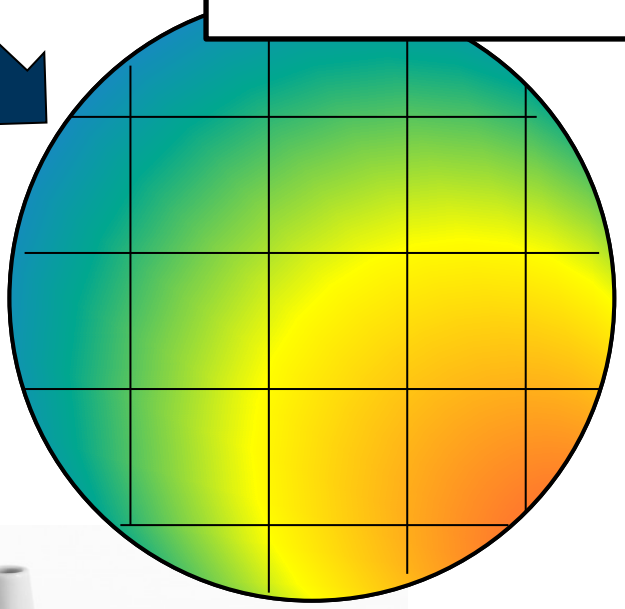
Variability

@ design

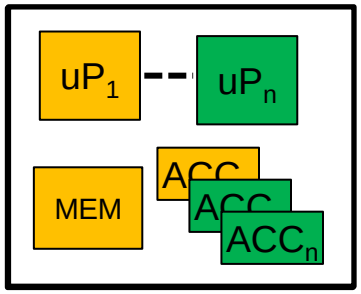


MPSoC

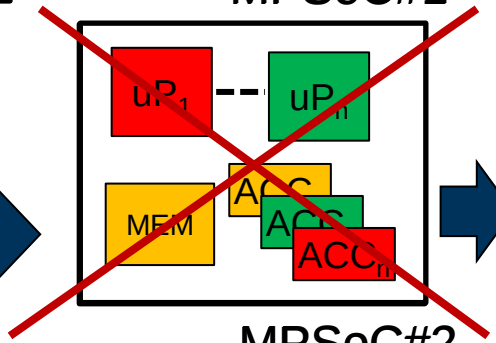
- Ideally
- Silicon Variability
- PVT Variation
- Yeld Loss
- Performance Loss
- Different QoS



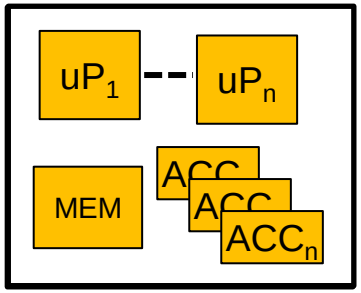
@ post-Silicon



MPSoC#1

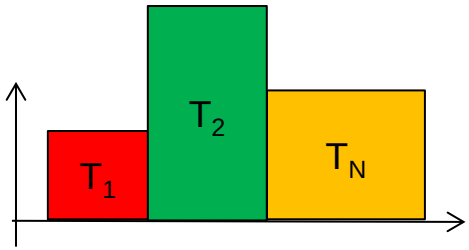


MPSoC#2



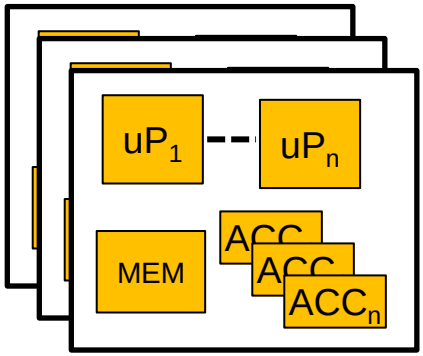
MPSoC#N

@ run-time



SW

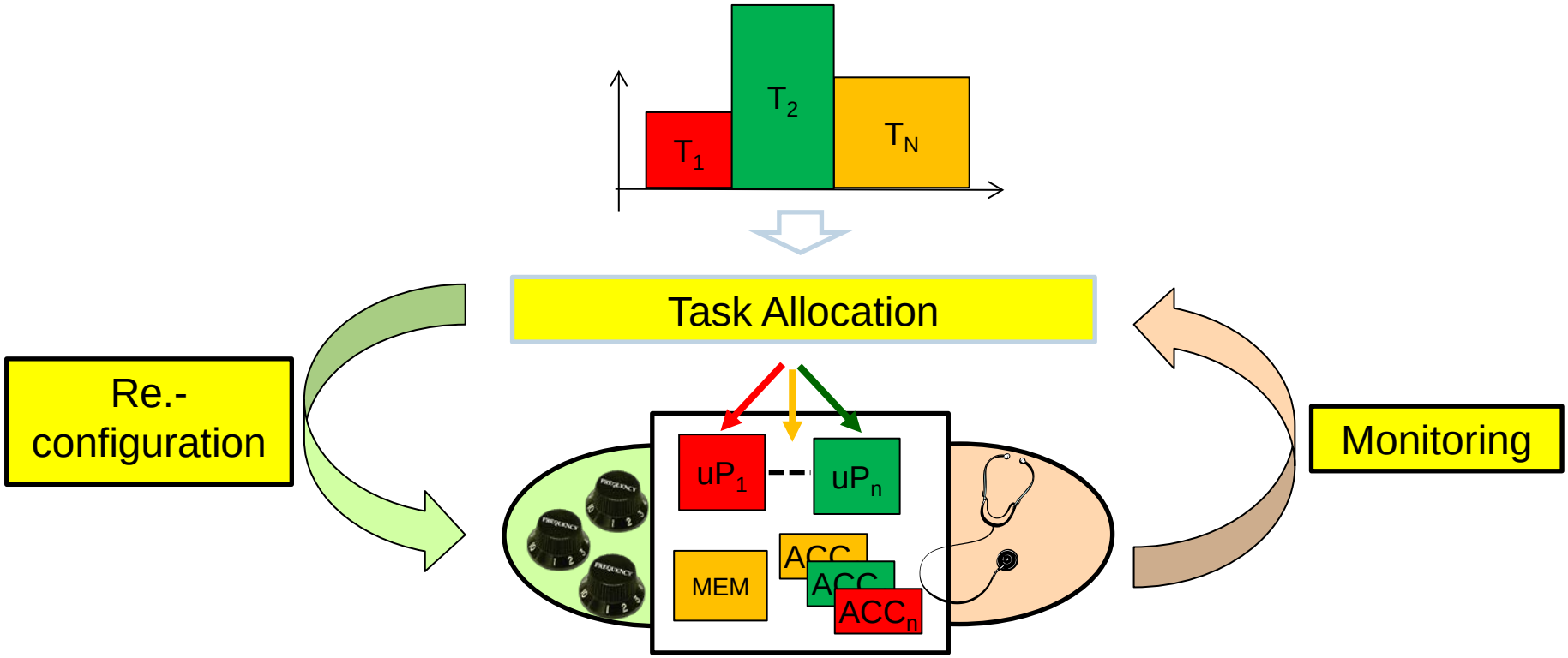
HW



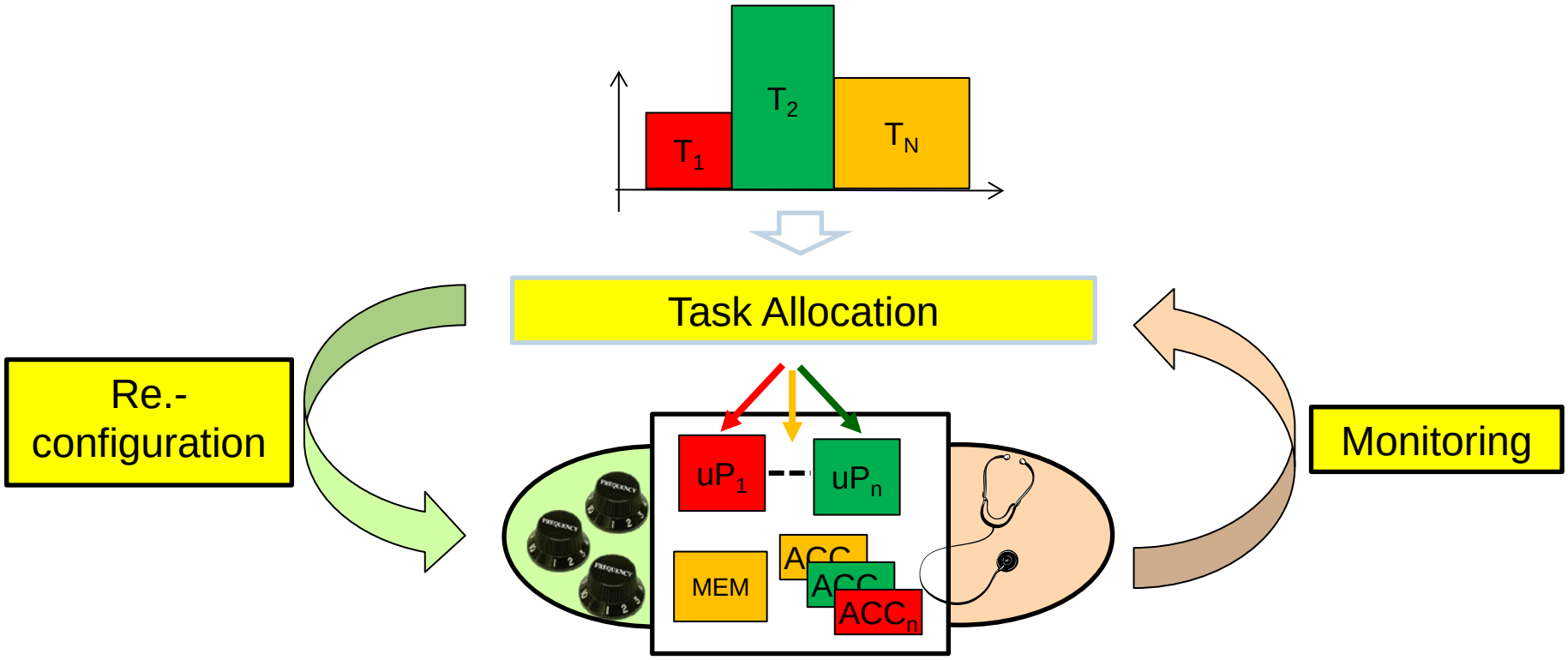
Power Management in ULP

System-level management of variability for energy-efficient multi-core processors

- **Monitoring - Workload, HW run-time characteristics status**
- **Re-configuration – Tune Power/Resiliency/Speed to user need**
- **Task Allocation – Optimal resource usage**

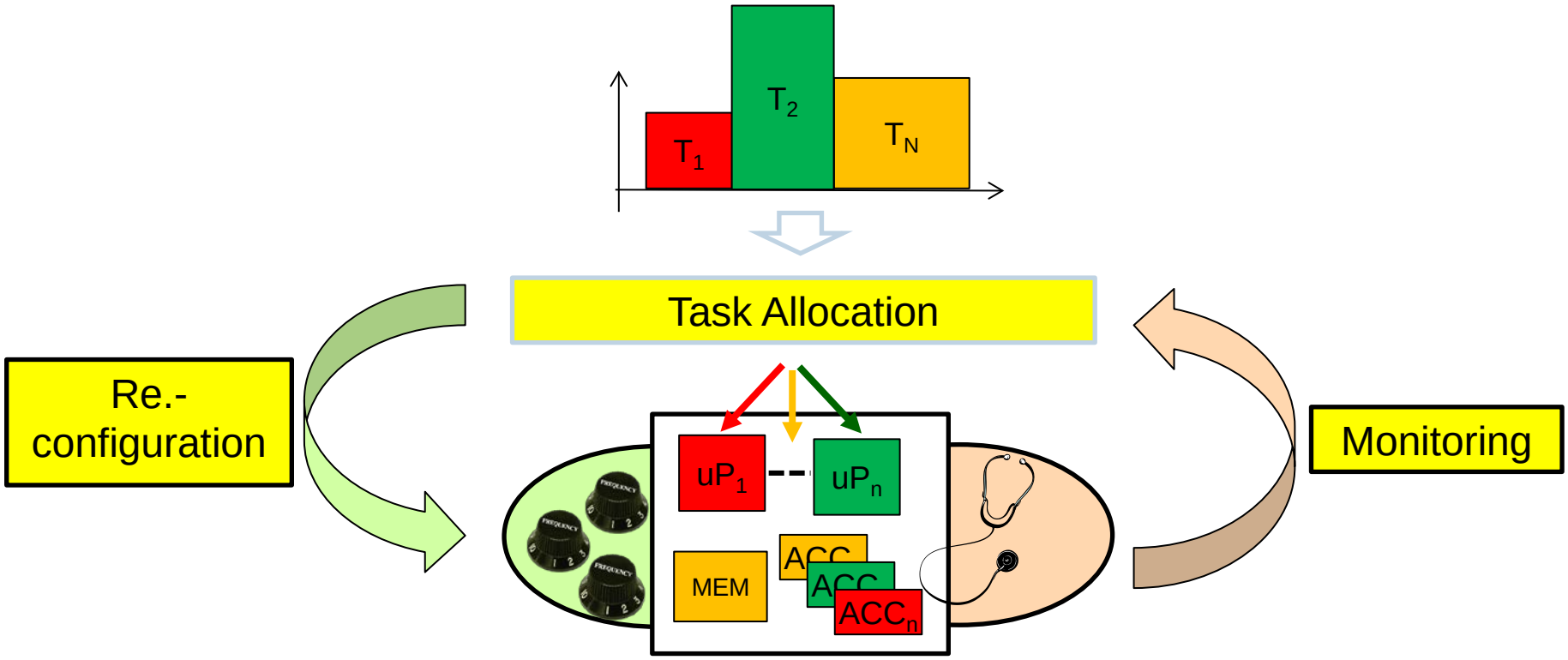


i.e. ....


T1
T2
TN
Task Allocation
Re.-
configuration
Monitoring
uP1
uPn
MEM
ACC
ACC
ACCn

i.e. ….

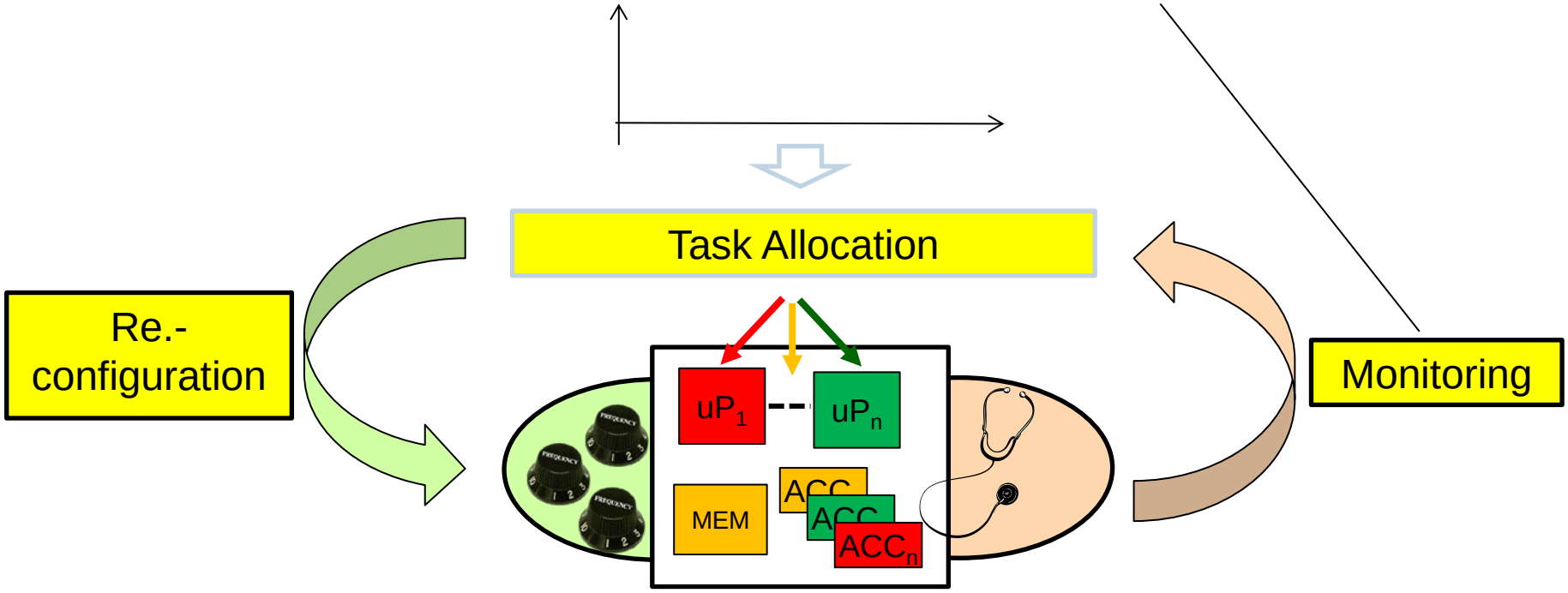
Monitoring…



i.e. ….

Monitoring…

Task / Resource	uP1 (Time/Error)	uPn (Time/Error)	Acc (Time/Error)
T1	1s / 10%	3s / 70%	0.5s / 0%
T2	3s / 40%	5s / 40%	
TN	2s / 10%	6s / 60%	0.5s / 0%

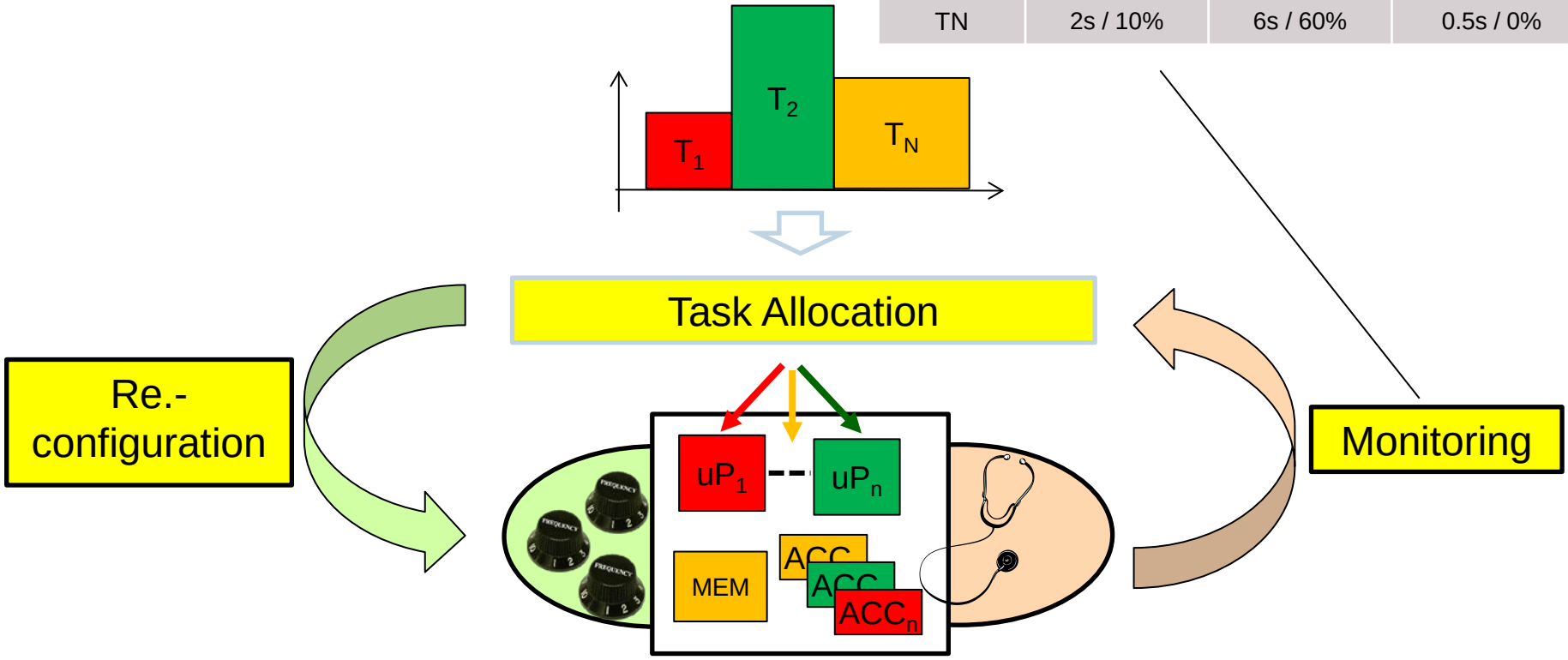


i.e. ….

Monitoring…

Allocation…

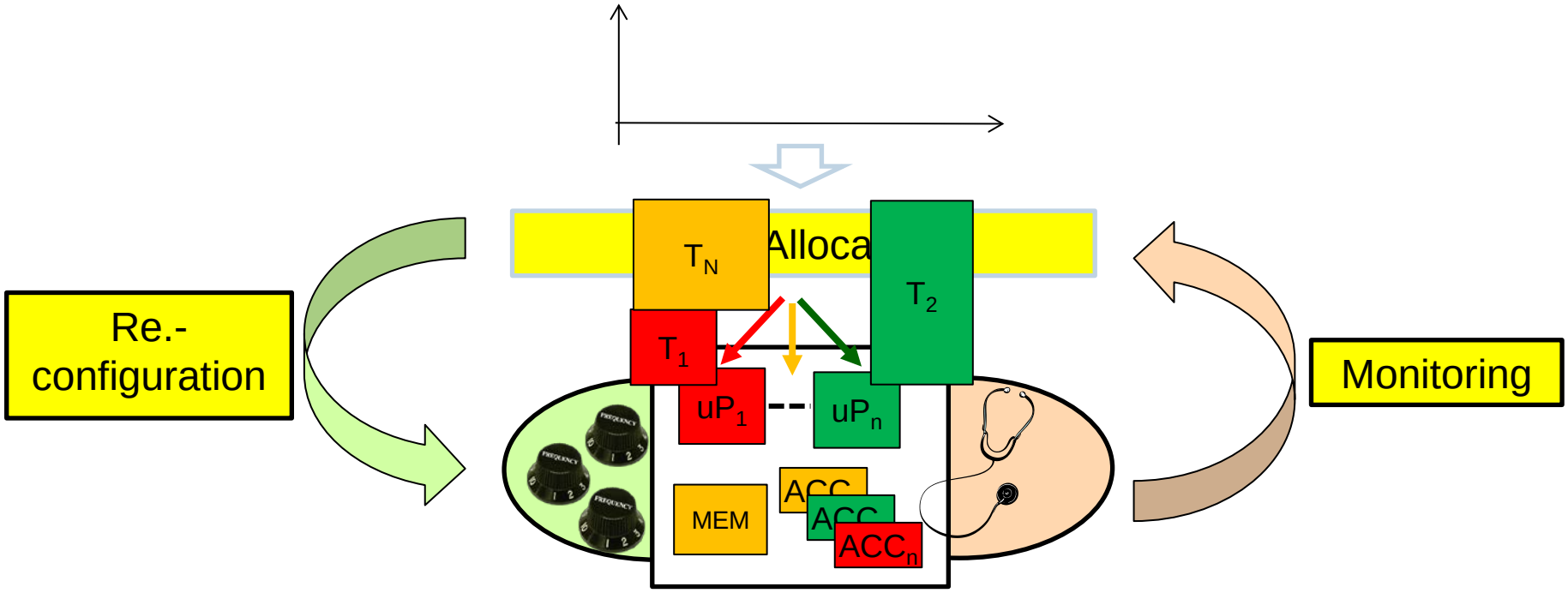
Task / Resource	uP1 (Time/Error)	uPn (Time/Error)	Acc (Time/Error)
T1	1s / 10%	3s / 70%	0.5s / 0%
T2	3s / 40%	5s / 40%	
TN	2s / 10%	6s / 60%	0.5s / 0%



i.e. ….

Monitoring…

Allocation…

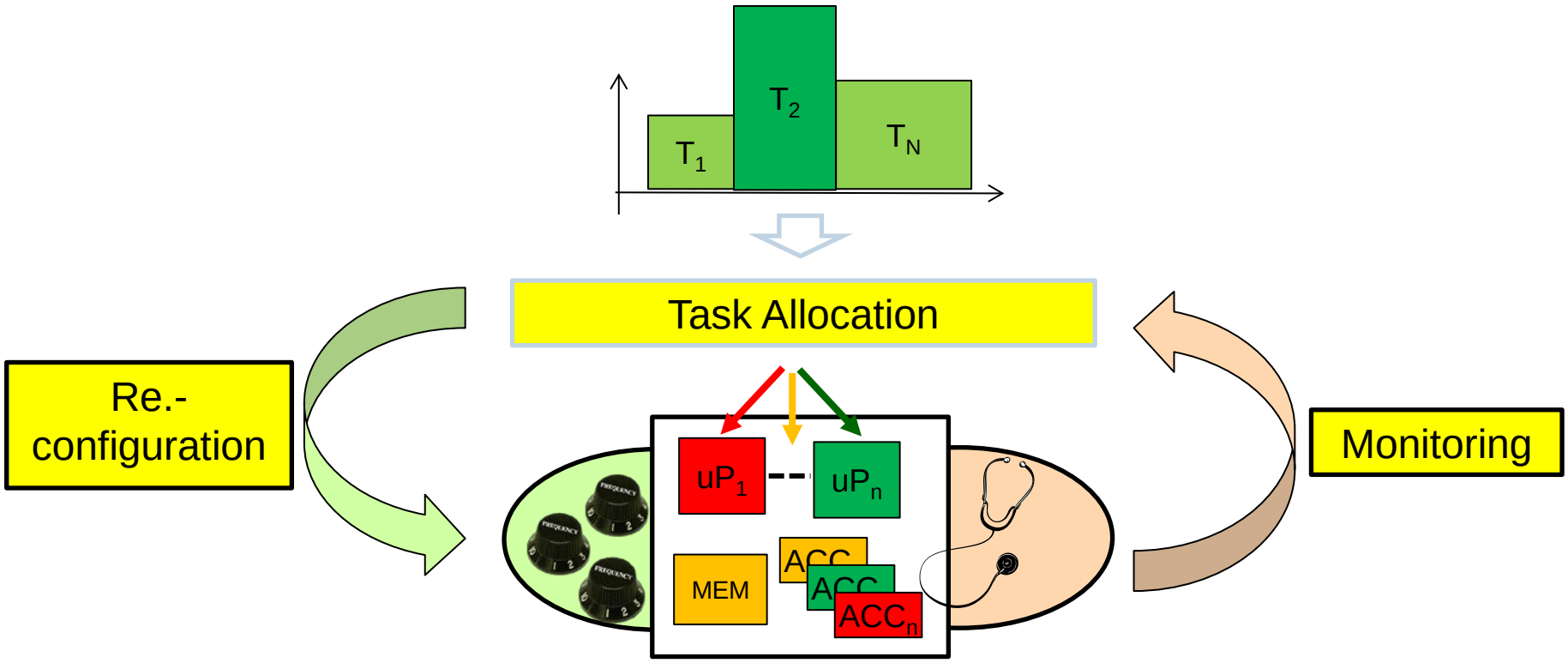


i.e. ….

Monitoring…

Allocation…

Re-configuration…

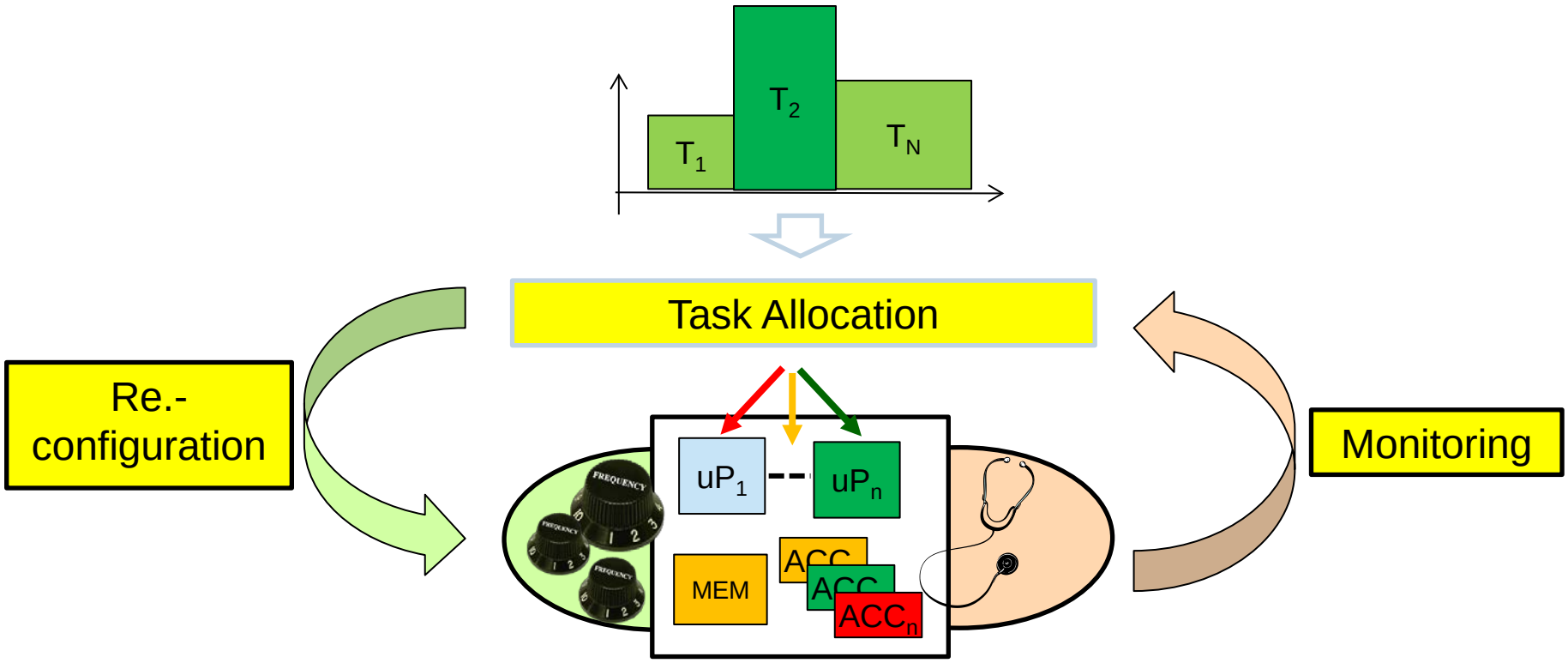


i.e. ….

Monitoring…

Allocation…

Re-configuration…

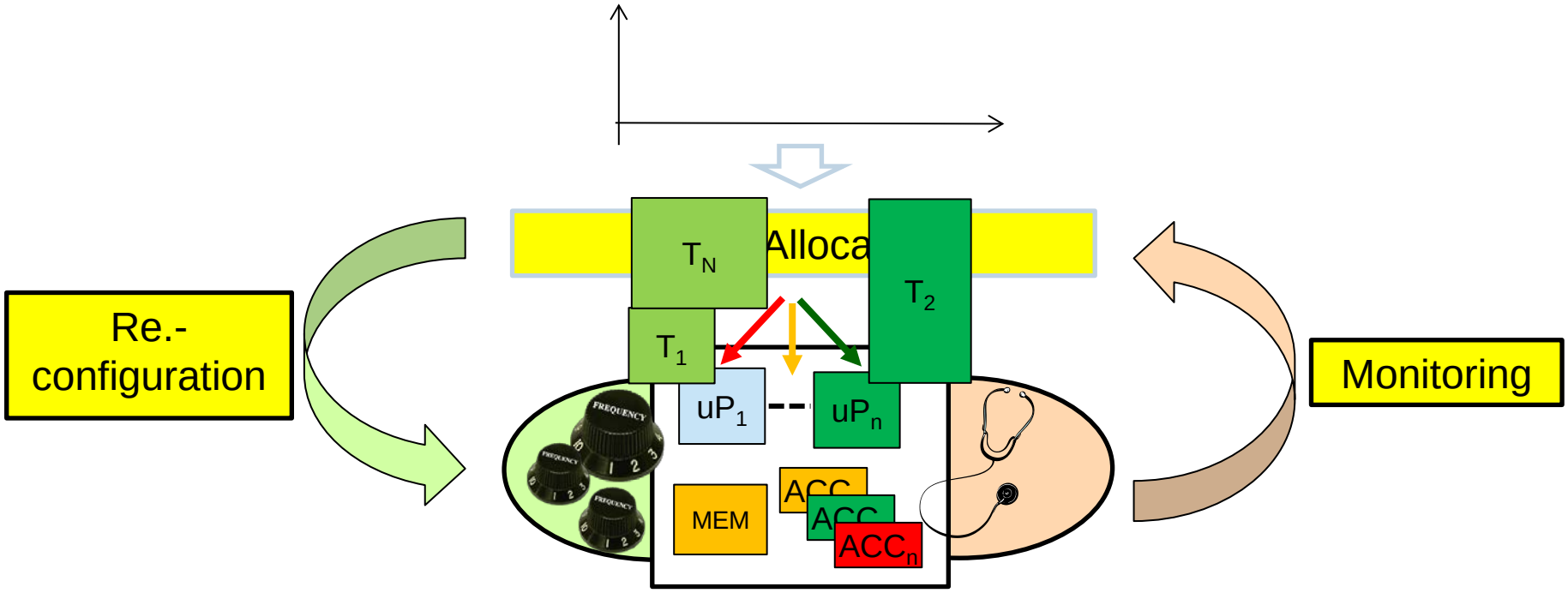


i.e. ….

Monitoring…

Allocation…

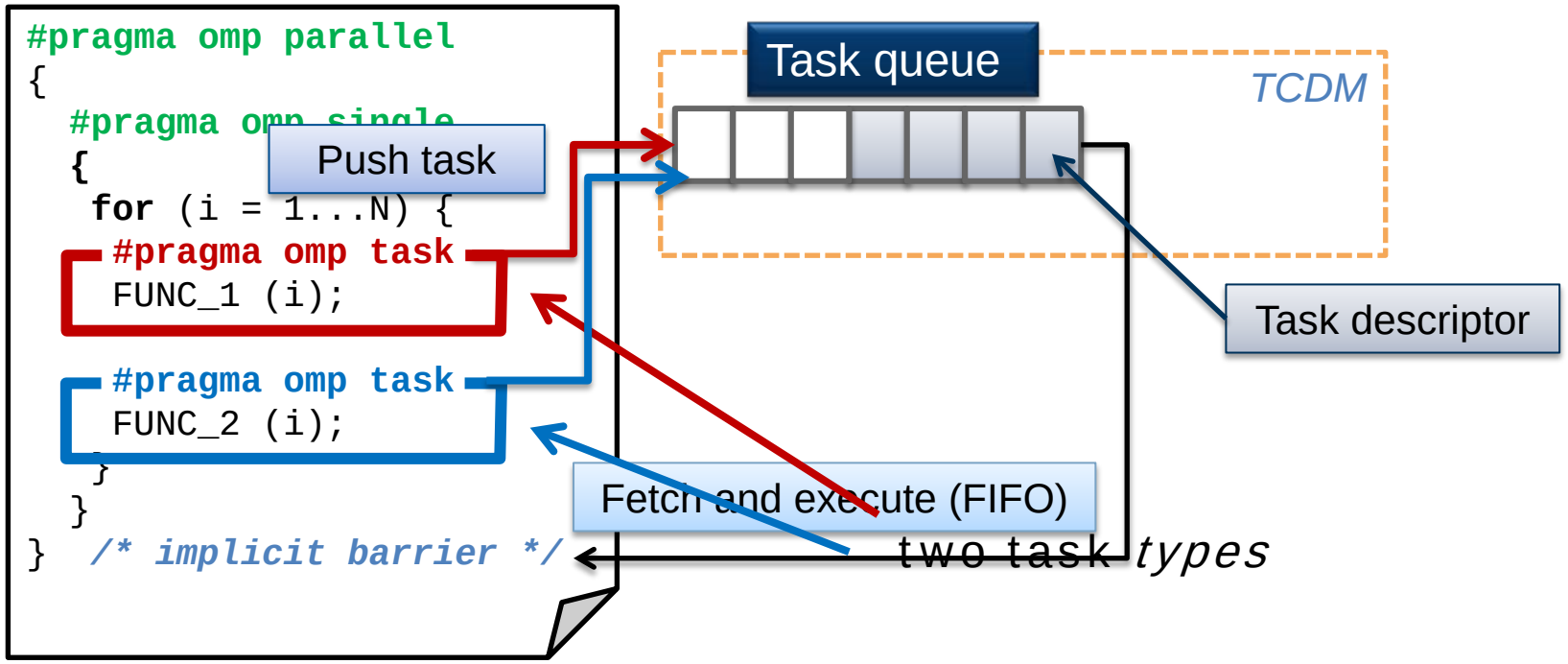
Re-configuration…



Outline

- Power in digital systems
- Energy Management
- **Monitoring**
- Task Allocation
- Reconfiguration

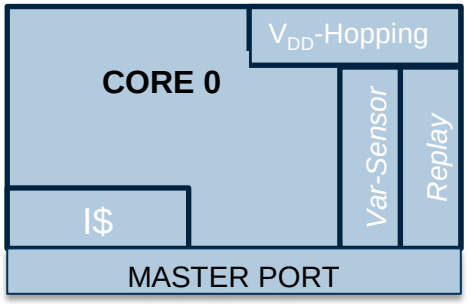
Monitoring – online profiling *[Rahimi 2013 DATE]*



- Task descriptors created upon encountering a **`task`** directive
- Task fetched by any core encountering a barrier

- **`task`** directives identify given **portions of code (tasks)**
- A task ***type*** is defined for every occurrence of the **`task`** directive in the program

Architecture

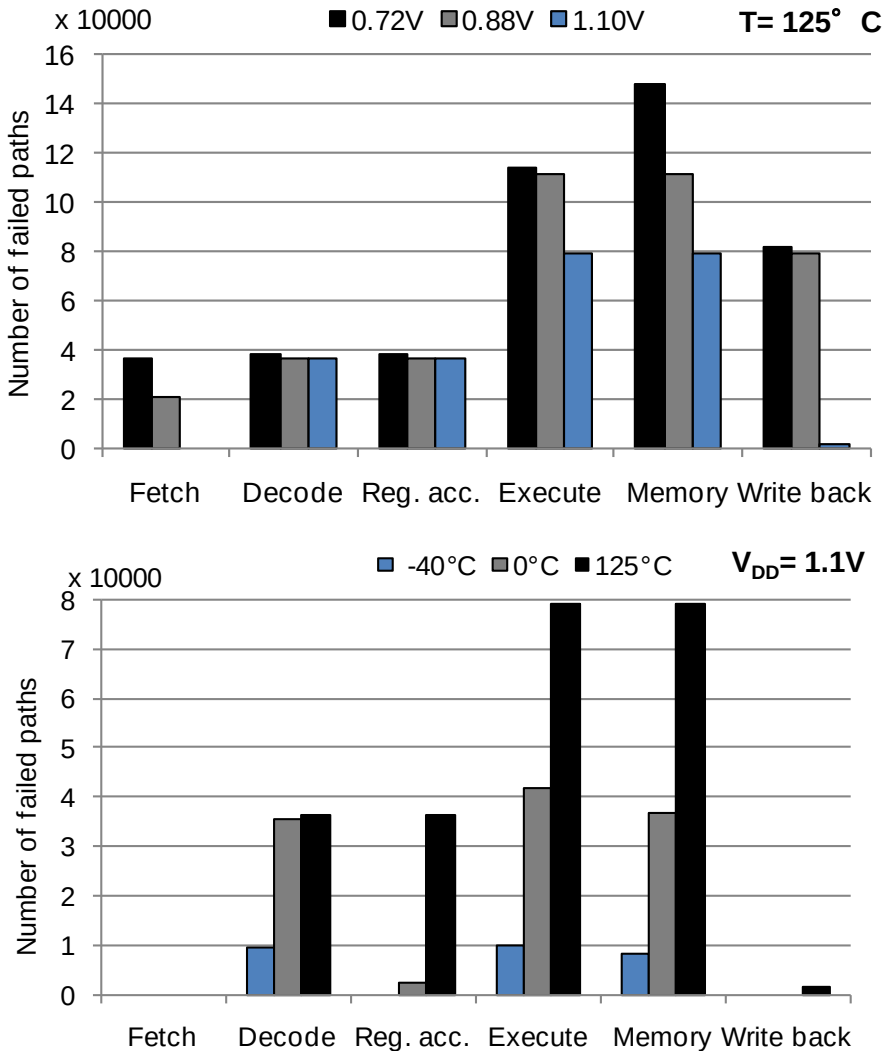


- Every core is equipped with:
  - Error sensing (EDS [1])
    - detect any timing error due to dynamic delay variation
  - Error recovery (*Multiple-issue replay* mechanism [2])
    - to recover the errant instruction without changing the clock frequency

[1] K.A. Bowman, et al., "Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance," IEEE Journal of Solid-State Circuits, 44(1): 49-63, 2009.
[2] K.A. Bowman, et al., "A 45 nm Resilient Microprocessor Core for Dynamic Variation Tolerance," IEEE Journal of Solid-State Circuits, 46(1): 194-208, Jan. 2011.
[3] S. Miermont, P. Vivet, M. Renaudin ,"A power supply selector for energy- and area-efficient local dynamic voltage scaling," Proc. *PATMOS,* pp. 556-565, 2007.

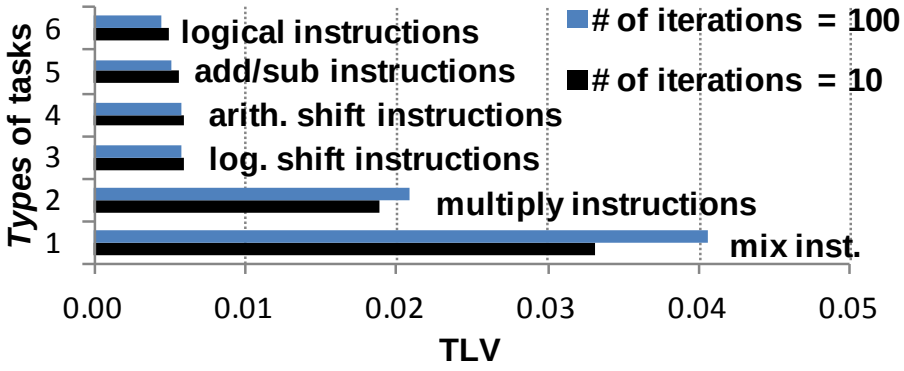
Delay Variability Among Pipeline Stages

- We analyze the effect of a full range of operating conditions (temp. range -40C–125C, volt. range .72V–1.1V) on the delay a LEON processor in 65nm TSMC.
- The *execute* and *memory* parts are very sensitive to voltage and temperature variations, and also exhibit a large number of critical paths in comparison to the rest of processor.
- Similarly, we anticipate that the instructions that significantly exercise the *execute* and *memory* stages are likely to be more vulnerable to voltage and temperature variations→ Instruction-level Vulnerability (ILV)

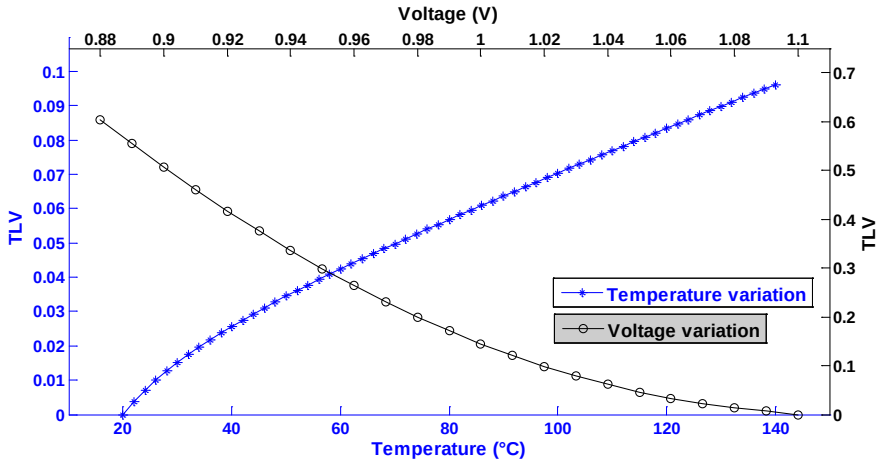


Inter- and Intra-Corner TLV

- **TLV across various type of tasks:** TLV of each *type* of tasks is different (up to 9×) even within the fixed operating condition in a core$_i$ (as observed in ILV).



- **Inter-corner TLV (across different operating conditions)**
  - The average TLV of the six types of tasks is an increasing function of temperature.
  - In contrast, decreasing the voltage from the nominal point of 1.1V increases TLV.



Variation-tolerant OpenMP tasking

- **Online TLV characterization**
  - TLV table: LUT containing TLV for every core and task *type*
  - Kept in L1. Parallel inspection from multiple cores
- Each core collects TLV information in parallel
  - Distributed scheduler
  - LUT updated at every task execution

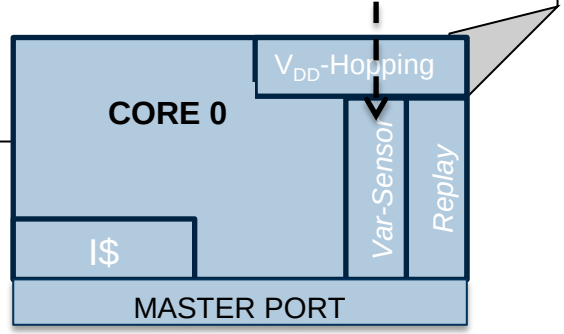
```
void handle_tasks () {
 while (HAVE_TASKS) {  // Task scheduling loop
  task_desc_t *t = EXTRACT_TASK ();
  if (t) {
   float Otlv = tlv_read_task_metadata (core_id);
   /* Reset counter for this core */
   tlv_reset_task_metadata (core_id);
   /* EXEC! */
   t->task_fn (t->task_data);
   /* We executed. Fetch TLV ...*/
   float tlv = tlv_read_task_metadata (core_id);
   /* Update TLV. Average new and old value */
   tlv_table_write(t->task_type_id,
                   core_id, (tlv-Otlv)/2);
  }
 }
}
```

TLV-table

cores

task types	C0	C1	C2
T0	0.0211	**0.11**	-
T1	0.891	-	0.000005

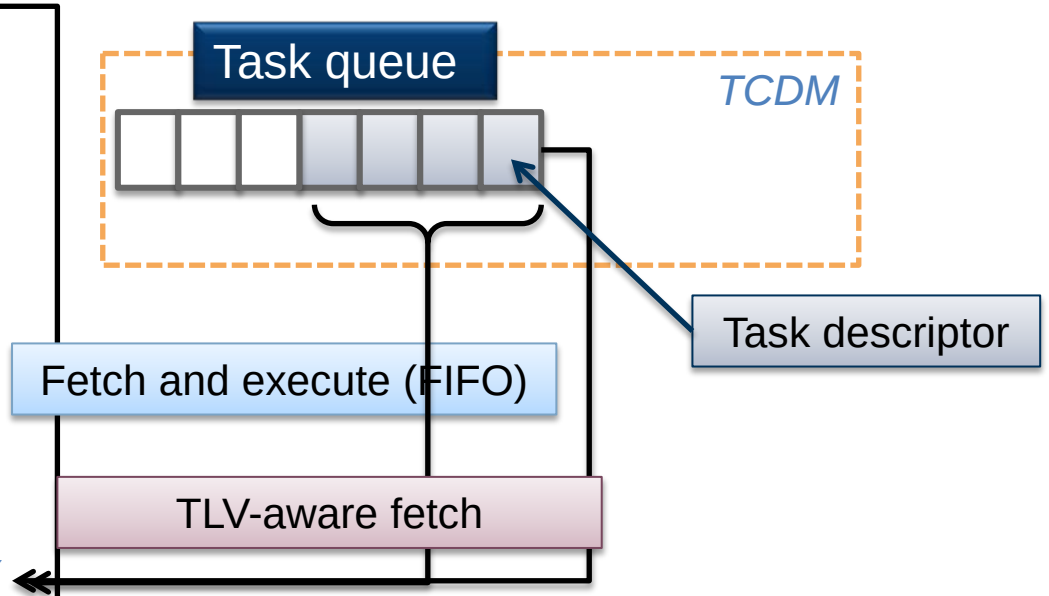
TCDM



TLV-aware extensions


```
#pragma omp parallel
{
  #pragma omp single
  {
   for (i = 1...N) {
    #pragma omp task
    FUNC_1 (i);

    #pragma omp task
    FUNC_2 (i);
   }
  }
}  /* implicit barrier */
```



- Variation-tolerant OpenMP scheduler
  - Proactive scheduling. Idle processors trying to fetch a task check if their TLV for the task is under a certain threshold to minimize number of errant instructions (and costly replay cycles)
  - limited number of rejects for a given tasks, to avoid starvation

Outline

- Power in digital systems
- Energy Management
- Monitoring
- **Task Allocation**
- Reconfiguration

Task Allocation

[Paterna TC 2012]

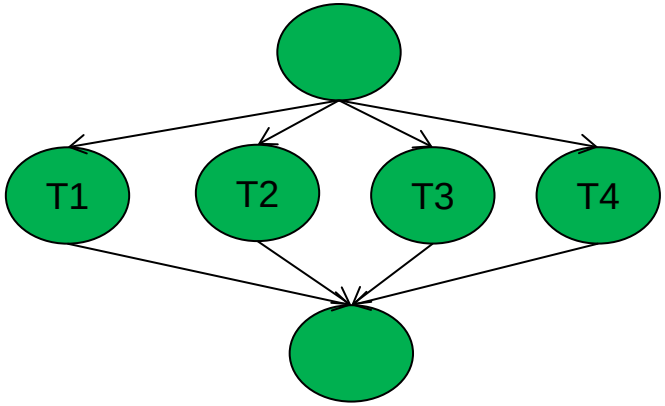
GOAL: find a **task allocation** technique for **variability-affected multicore** platform which **minimizes the energy consumption** while meeting a performance constraint

- Proposes a task allocation technique which provides a near-optimal solution
- The technique can be applied at run-time
- It is based on a 2-step problem formulation

FORMULATION HYPOTHESIS

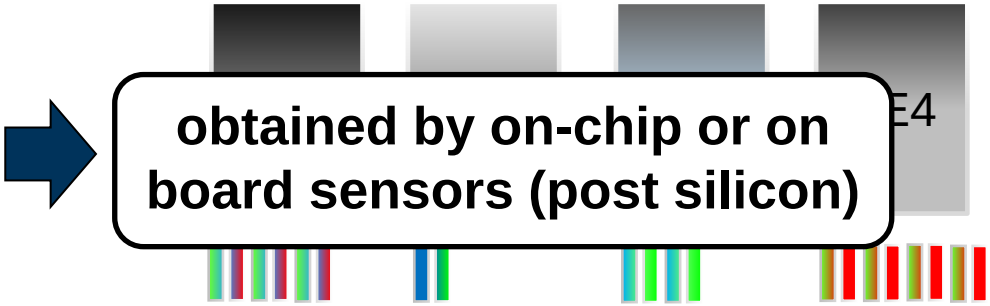
WORKLOAD:

1. The tasks are independent and synchronized on a barrier
2. The length of the tasks in terms of instructions is known



PLATFORM:

1. PE can be ***active*** or ***idle***
2. For each PE are known its own
   - Clock frequency
   - Power consumption in active
   - Power consumption in idle



LP+BP problem formulation

1° step, **LP formulation**:

- disregards the task granularity but only considers the total number of instr.
- provides the cycle budget of each core (how many instr. a core can execute)

INPUT

- Variability
- Time Constraint
- Total task lengths

$$E_{tot} = \sum_{i=1}^{N_{cores}} \left[\frac{P_{Ai}}{f_{cki}} C_{Ai} + \frac{P_{Ii}}{f_{cki}} C_{Ii} \right]$$

$$\begin{cases} \frac{C_{Ai}}{f_{cki}} + \frac{C_{Ii}}{f_{cki}} \leq T \\ \sum_{i=1}^{N_{cores}} C_{Ai} = \sum_{j=1}^{M_{tasks}} C_j \end{cases}$$

OUTPUT:

Activity cycles budget

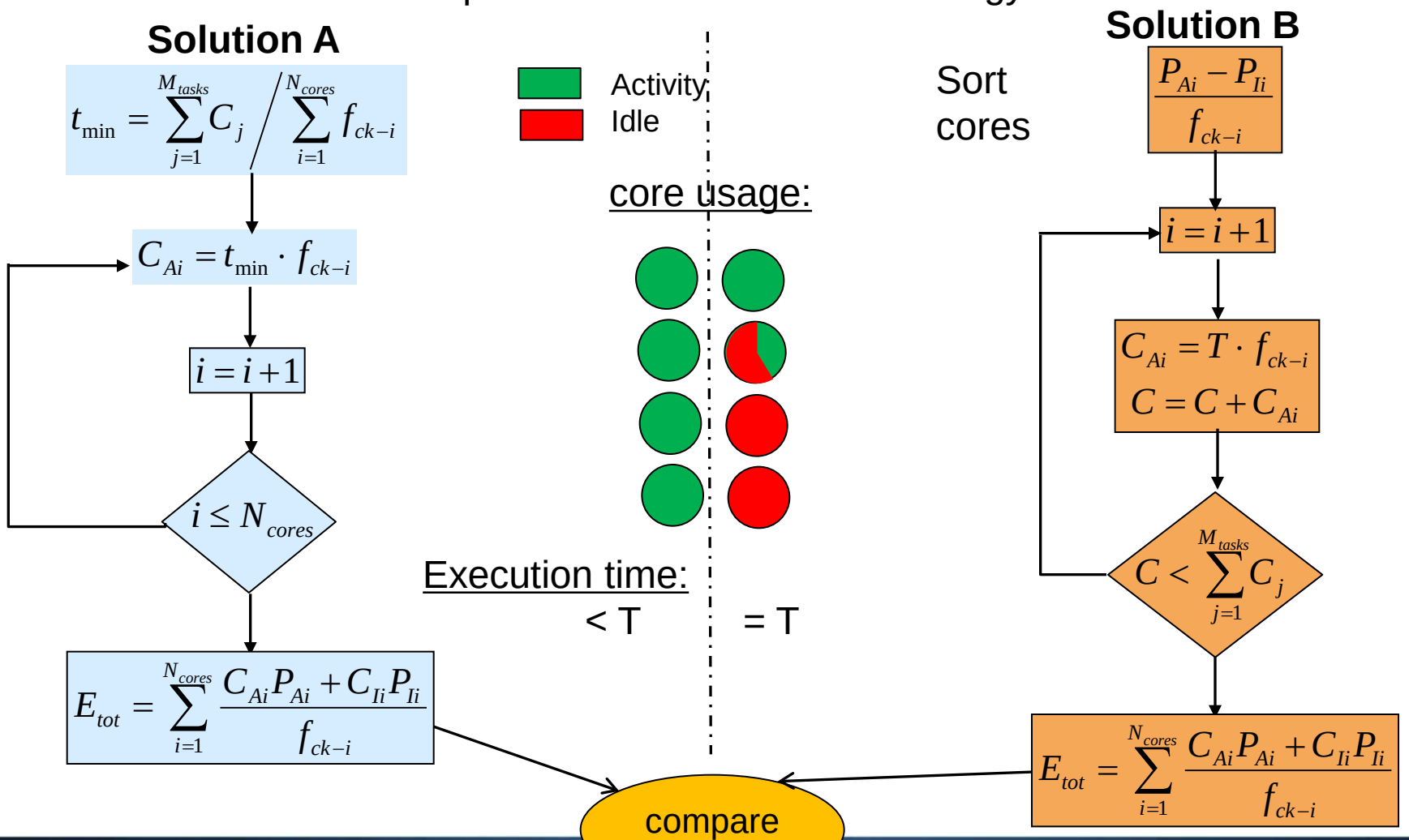
Idle cycles

Key properties! It can be **solved** through a **closed form**
No time-demanding Algorithm (i.e. Simplex)

Closed Form Solution

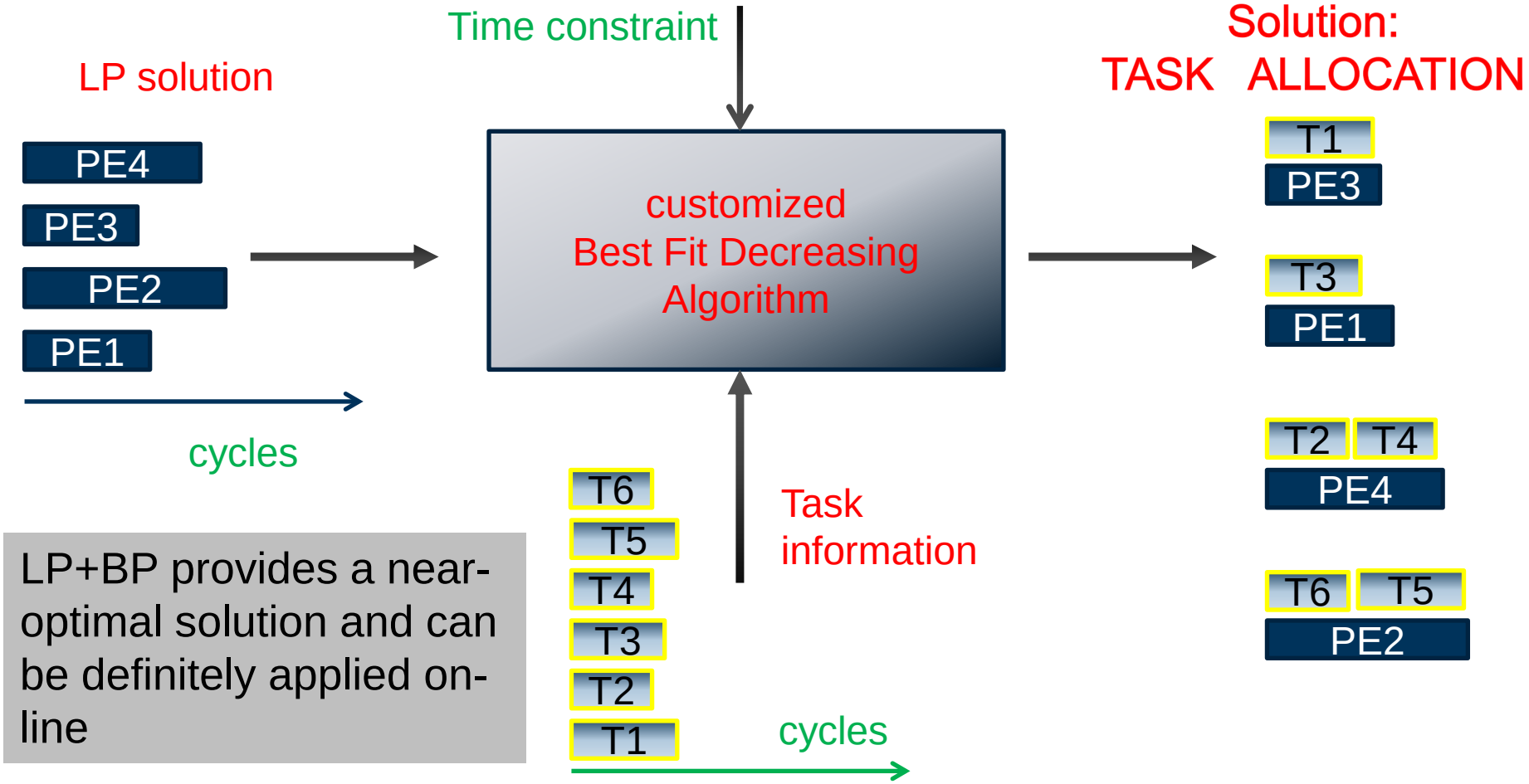
Proven to be optimal!!!
[2012 Paterna TC]

- Two candidate solutions (i.e. meet the time constraint)
- Take the one which makes the platform to consume less energy



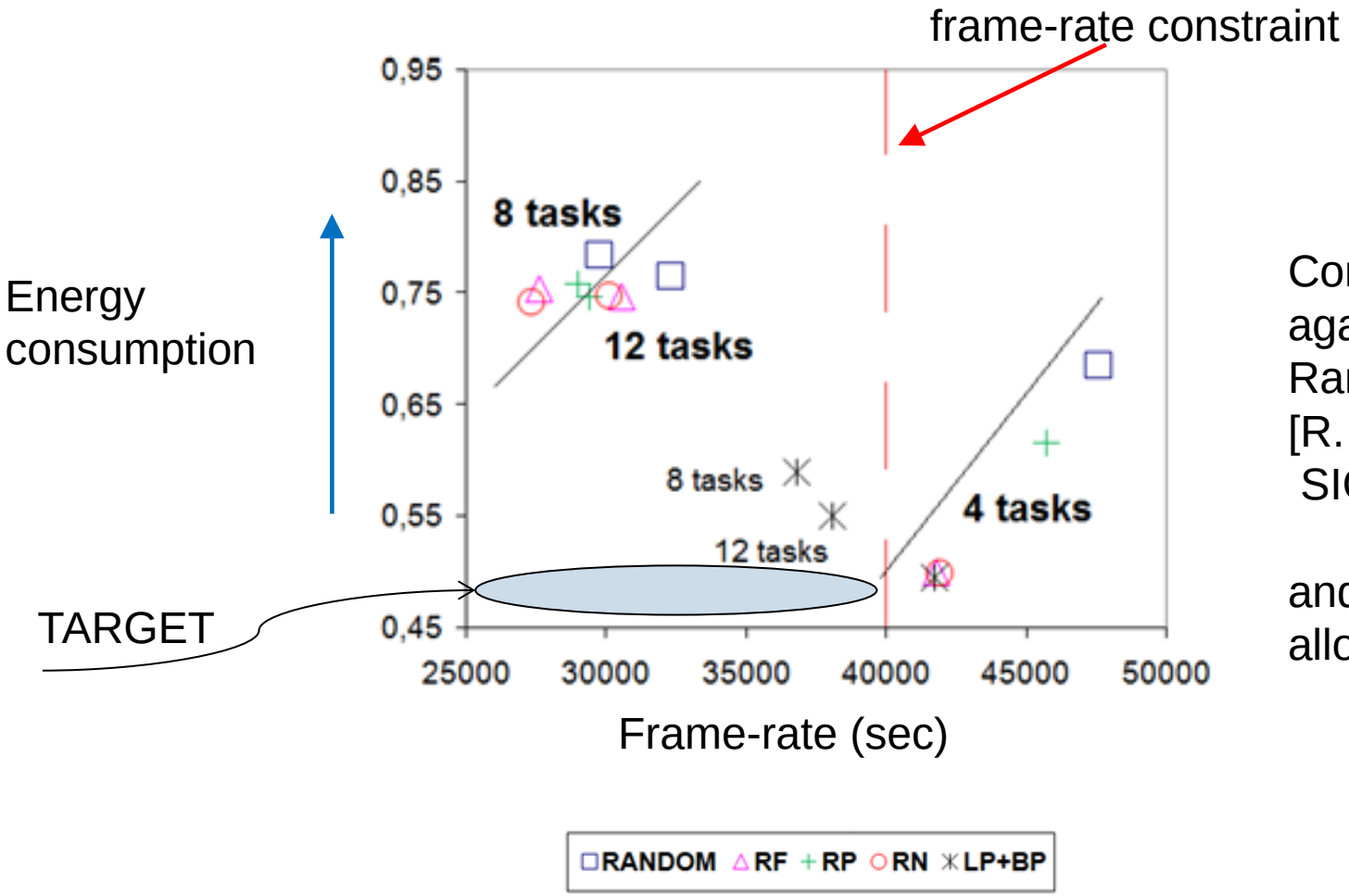
LP+BP problem formulation

2° step, **Bin Packing formulation**: it considers the task granularity
it fits the tasks into the cycle budgets



LP+BP provides a near-optimal solution and can be definitely applied on-line

MPEG2: Energy vs. Frame-rate



Comparison against Rank techniques [R. Teodorescu SIGARCH, '08]

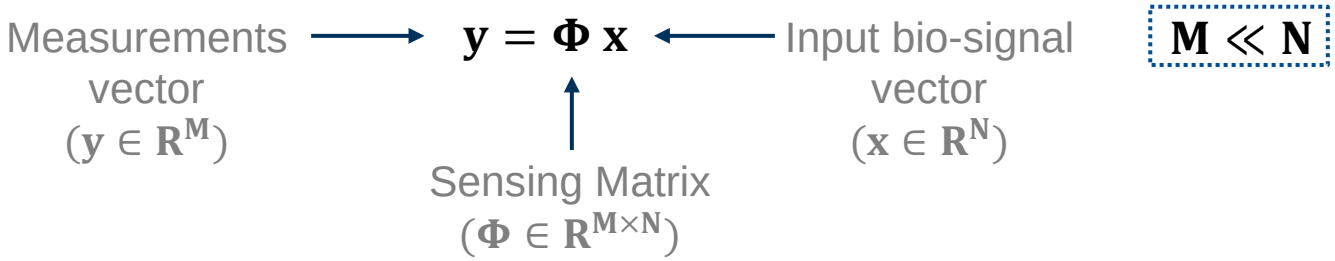
and a random allocation

Outline

- Power in digital systems
- Energy Management
- Monitoring
- Task Allocation
- **Reconfiguration**

Compressed Sensing *[Bortolotti 2014 Date]*

- emerging paradigm for signal acquisition/compression [Donoho06]
- enables sub-Nyquist sampling rate for sparse signals
- reduces the amount of samples required in processing and storage



- Sparsity of $\mathbf{x}$ with constraints on $\mathbf{\Phi} \rightarrow$ convex optimization problem
- ECG: $\mathbf{x}$ is not sparse in time domain but is sparse in a wavelet domain
  - SoA: low-power embedded ECG monitoring [Mamaghanian11]

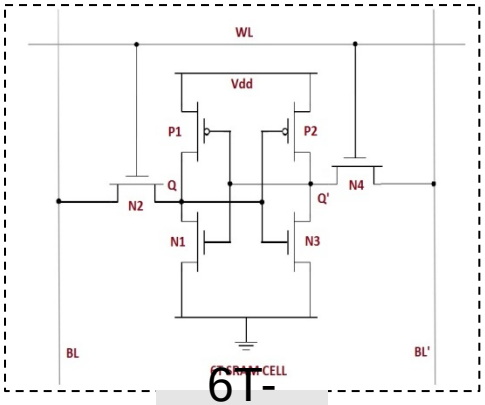
Multi-core Architecture

- medical grade monitoring requires **multi-channel** signal analysis
- multi-channel processing for several biosignals (ECG, EMG,…) is often **embarrassingly parallel**
- multi-core architectures enable more aggressive voltage-frequency scaling than single-core solutions

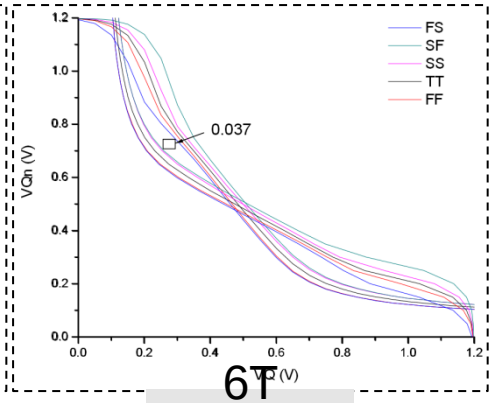
- **SoA** [Dogan12, Munir13]
  - proved their efficiency compared to single-core solutions
    - at ultra-low workload requirements **leakage dominates**
    - aggressive voltage scaling cannot be applied due to **reliability issues** for the memories

Voltage Scaling and Memory

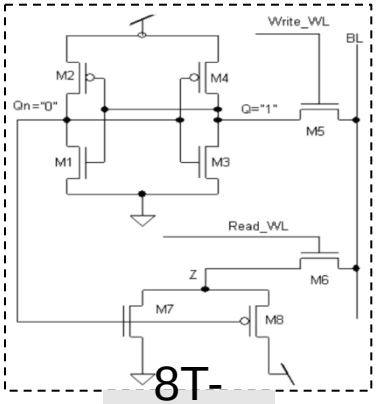
- **Voltage Scaling** is an effective technique to improve energy efficiency
- Embedded SRAM memory
  - **Energy bottleneck** at low-voltage
  - classic **6T** cell structure has **reliability issues**



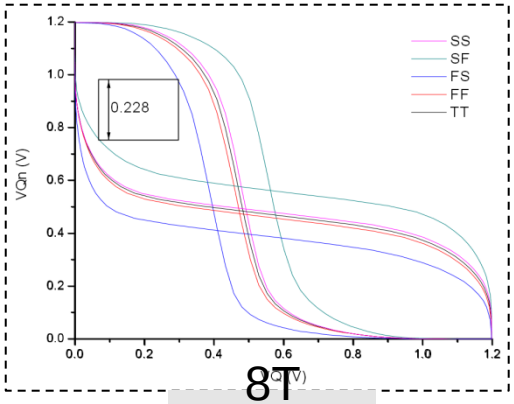
6T-cell



6T SNM



8T-cell

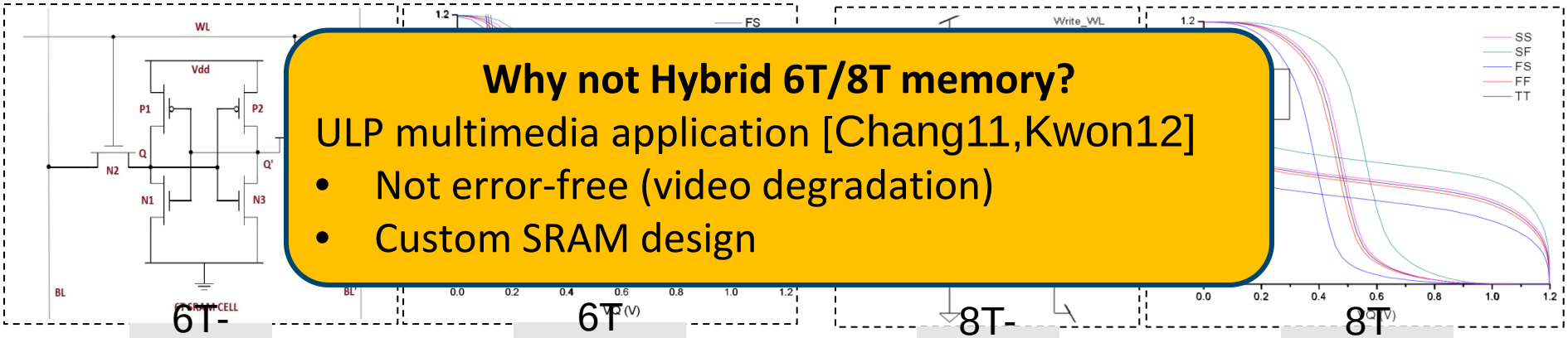


8T SNM

- **Alternative SRAM** designs (8T, 10T, SCMEM...)
  - more robust
  - area overhead

Voltage Scaling and Memory

- **Voltage Scaling** is an effective technique to improve energy efficiency
- Embedded SRAM memory
  - **Energy bottleneck** at low-voltage
  - classic **6T** cell structure has **reliability issues**



6T-cell | 6T SNM | 8T-cell | 8T SNM

Why not Hybrid 6T/8T memory?

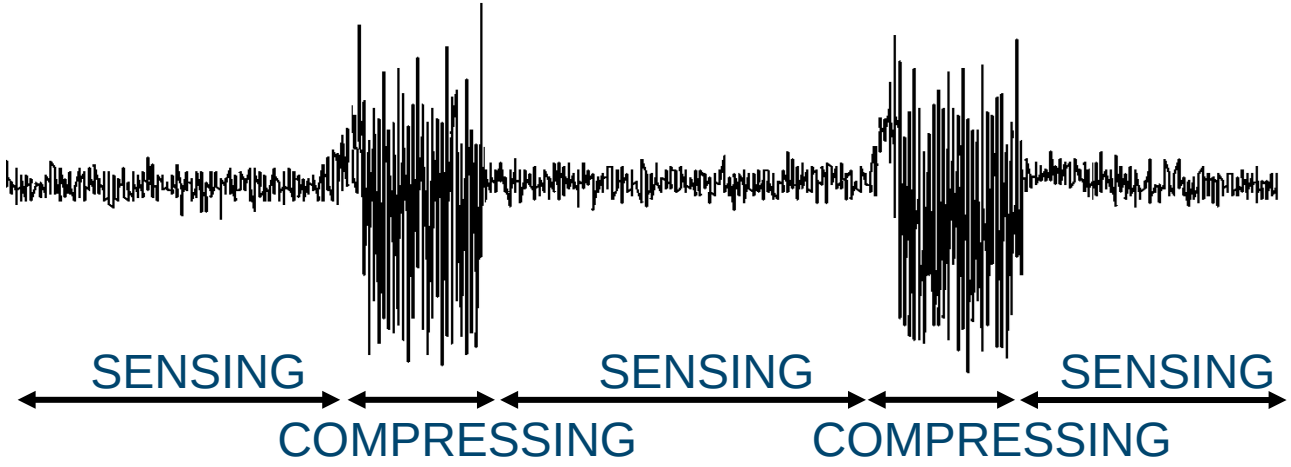
ULP multimedia application [Chang11,Kwon12]

- Not error-free (video degradation)
- Custom SRAM design

- **Alternative SRAM** designs (8T, 10T, SCMEM...)
  - more robust
  - area overhead

Idea

SYSTEM MEMORY FOOTPRINT AND WORKLOAD



- **SENSING** : *low* memory footprint, *low* workload

  LOW-POWER PHASE (8T active, 6T state retentive)

- **COMPRESSING** : *high* memory footprint, *high* workload

  HIGH-PERFORMANCE PHASE (6T/8T active)

- A typical biosignal digital node spends most of the time in sensing and a small portion of time in compression

  Example : 256 samples/1sec window → $T_{compressing} \approx 4ms$

Idea

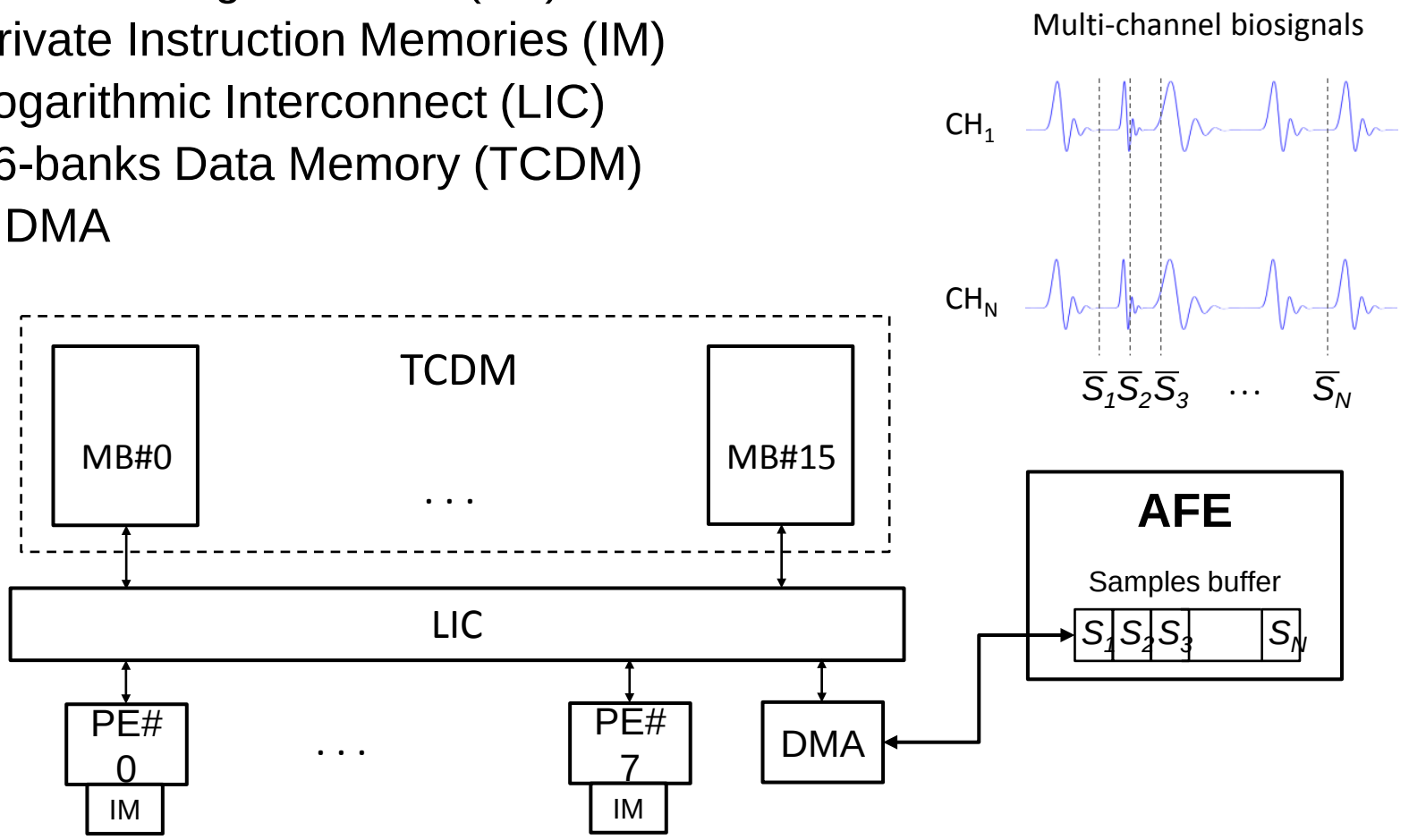
SYSTEM MEMORY FOOTPRINT AND WORKLOAD

IDEA

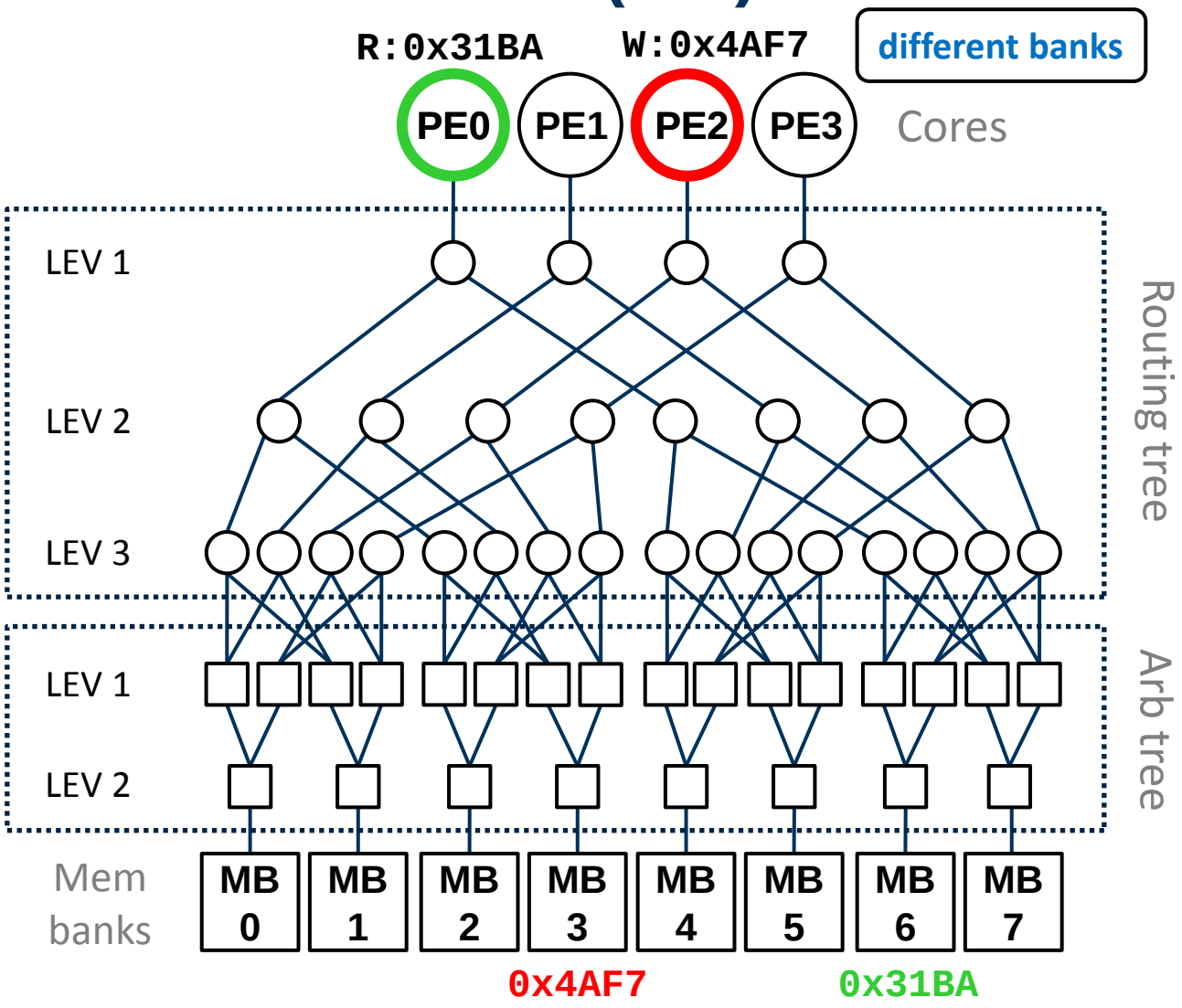
- a **hybrid memory** architecture in a **single voltage domain** for ULP multicore biomedical processors is proposed
- **6T/8T-banks** enable **aggressive voltage scaling** during phases with **low memory usage and low computational requirements**
- significant **energy savings** compared to a 6T-only architecture
- **very-low area overhead** compared to a 6T-only architecture and considerably lower than a 8T-only architecture

Baseline Architecture

- Architecture for multi-channel biomedical CS
  - 8 Processing Elements (PE)
  - Private Instruction Memories (IM)
  - Logarithmic Interconnect (LIC)
  - 16-banks Data Memory (TCDM)
  - 1 DMA

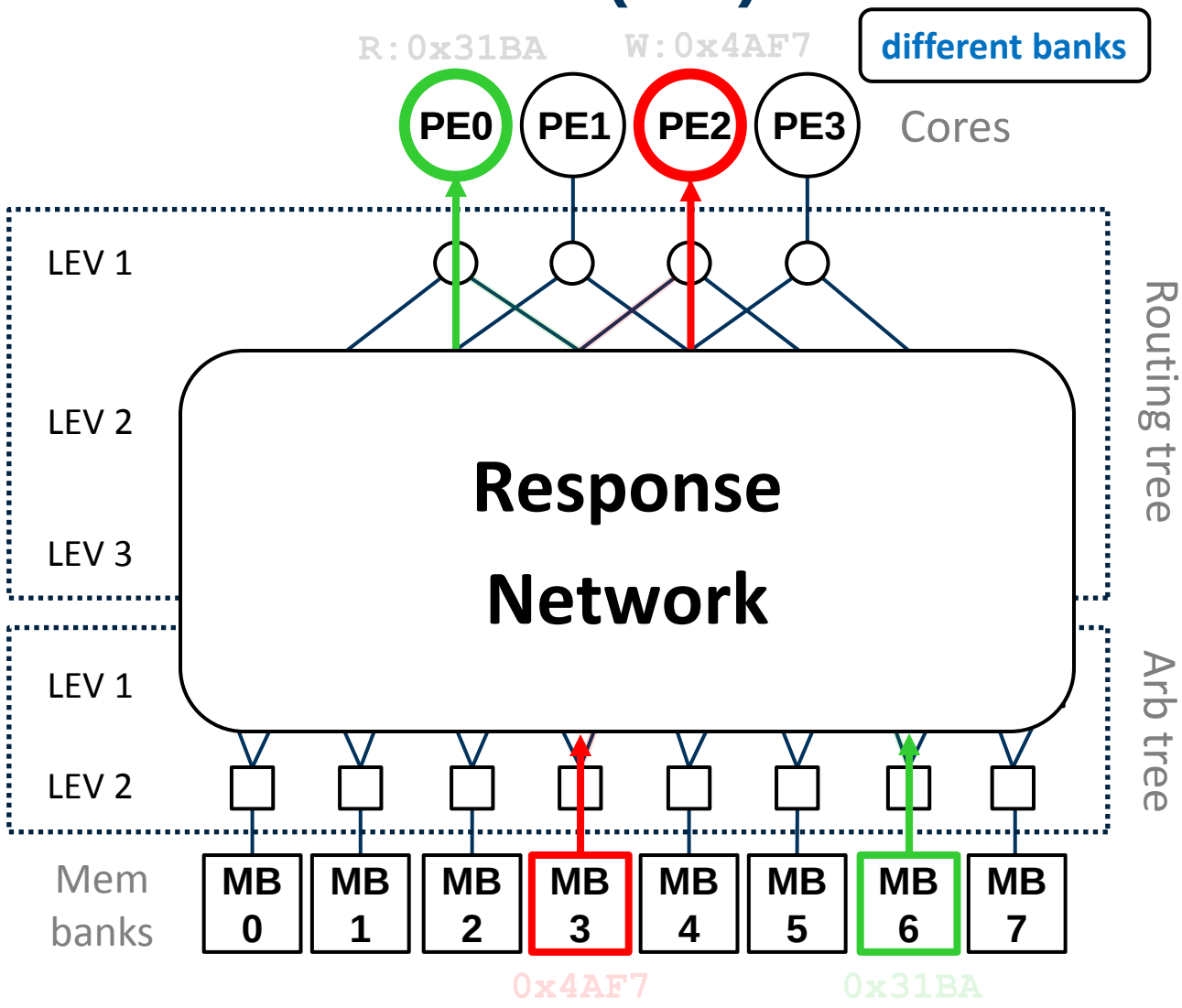


Logarithmic Interconnect (LIC)



[Rahimi11]

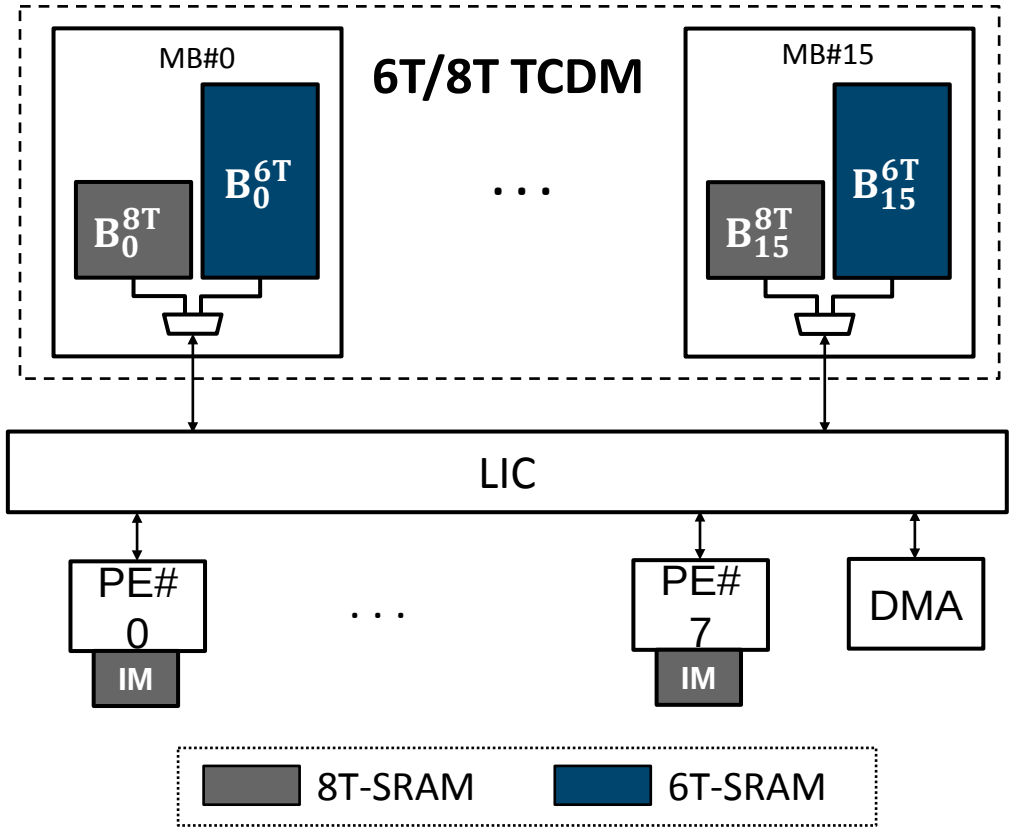
Logarithmic Interconnect (LIC)

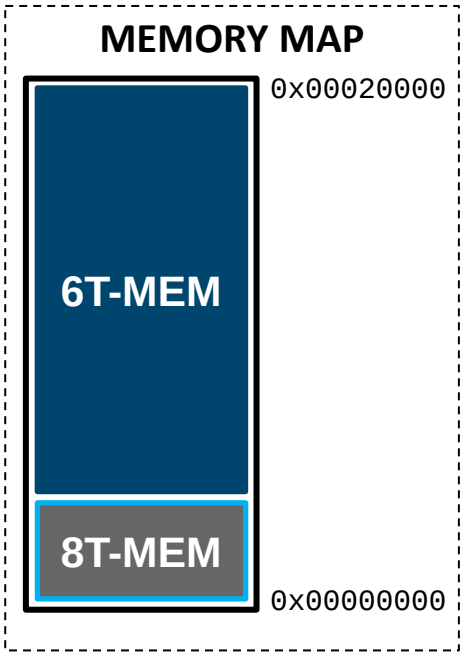


[Rahimi11]

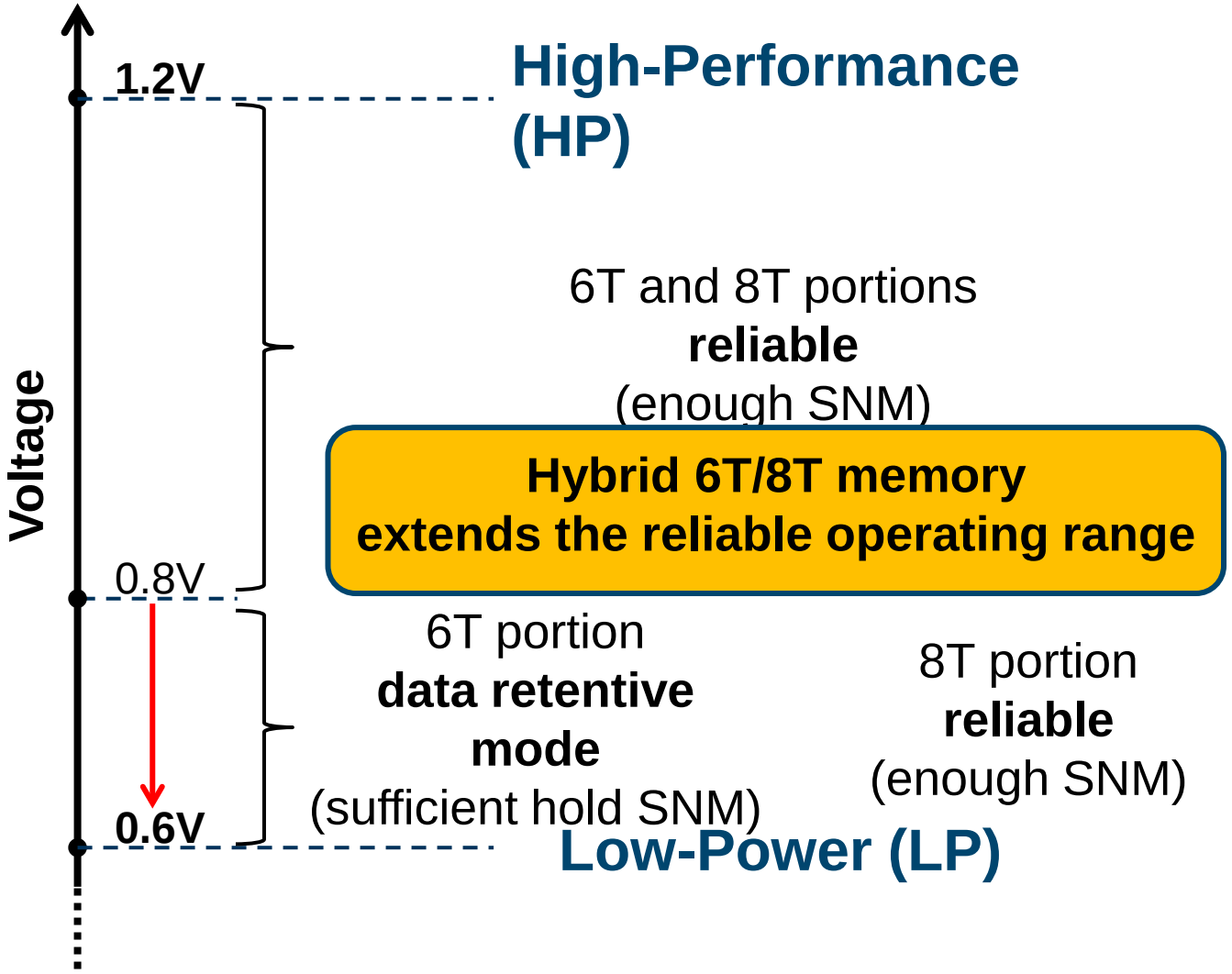
2 cycles

Hybrid Memory Architecture



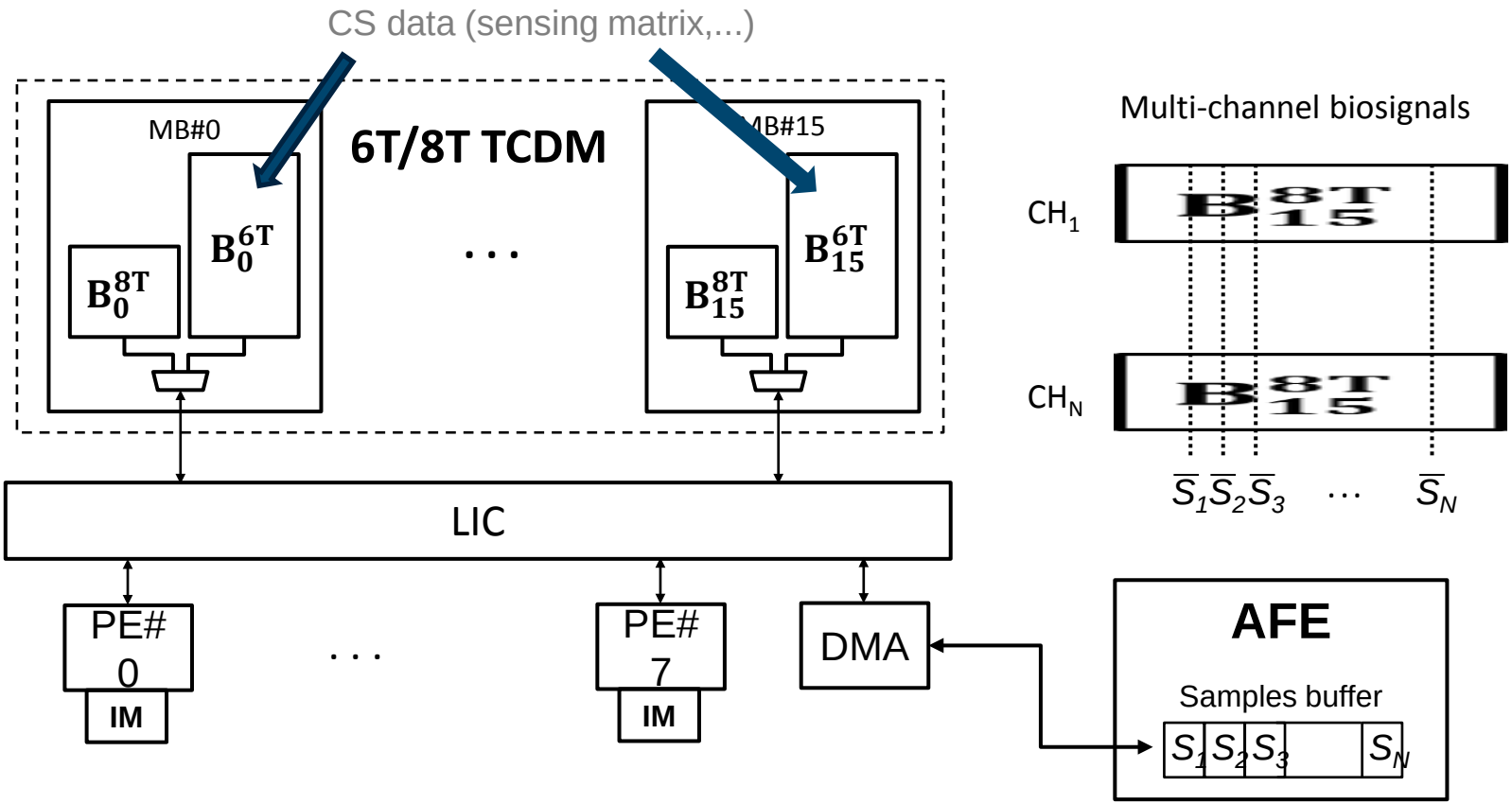


Hybrid Memory Architecture



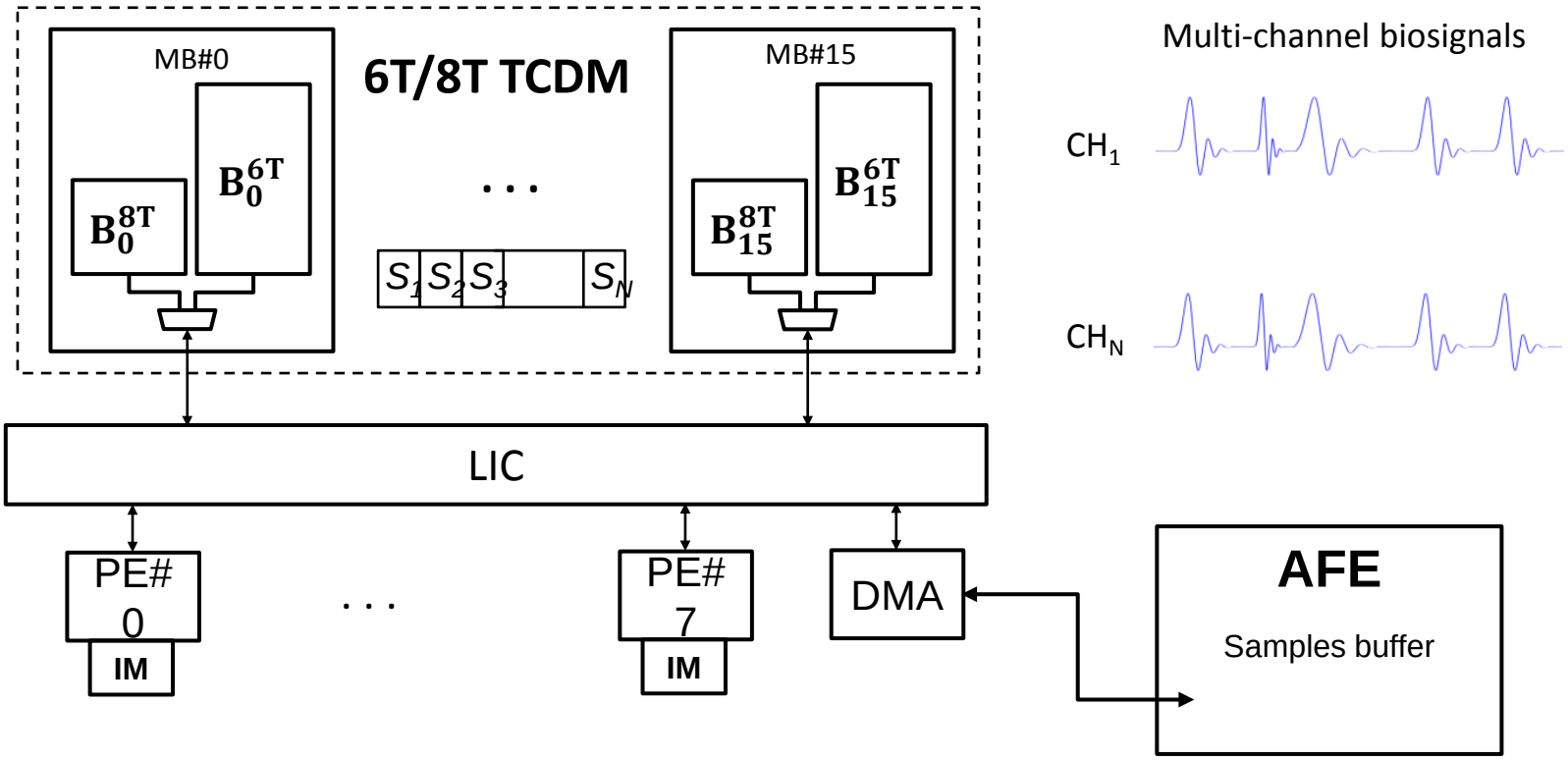
LP phase (sensing)

- PEs are idle (leakage) while DMA is moving samples from AFE to TCDM
- 6T banks are in data retentive state (only leakage)
- 8T banks are active and filled with *samples* by DMA



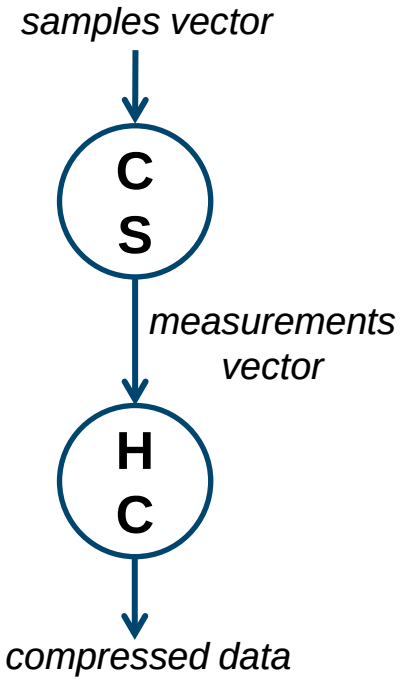
HP phase (compression)

- All PEs are active performing CS in parallel
- Both 8T (*samples*) and 6T banks (*stack, CS data*) are active
- DMA is inactive



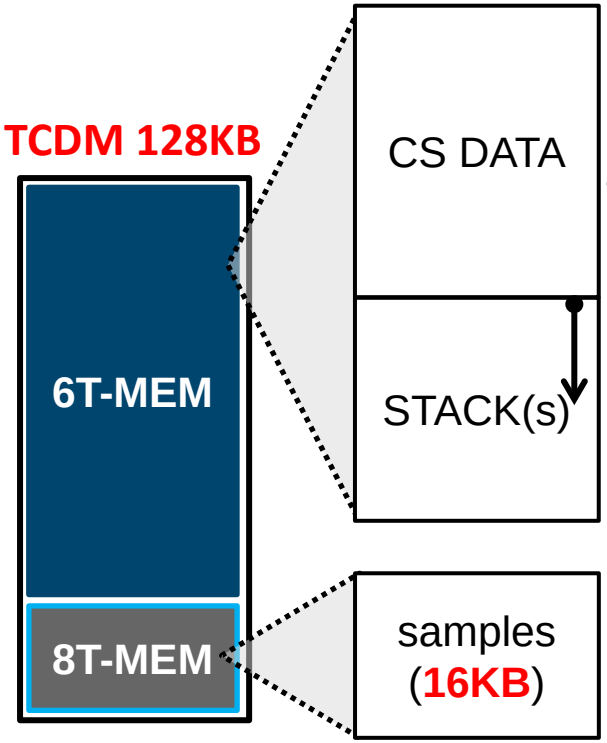
CS code analysis

- Multi-lead (8) ECG processing algorithm
  - bare-metal parallelization (core id)
- CS algorithm [Mamaghanian11]



static code analysis

MEMORY SIZING



Matrix Φ
Ouput y
3 LUTs
for CS and HC

1window
512 samples/lead
8 leads
32bits/sample

- Compiler `attributes` and linker script `SECTIONS` for static allocation

Simulation Infrastructure

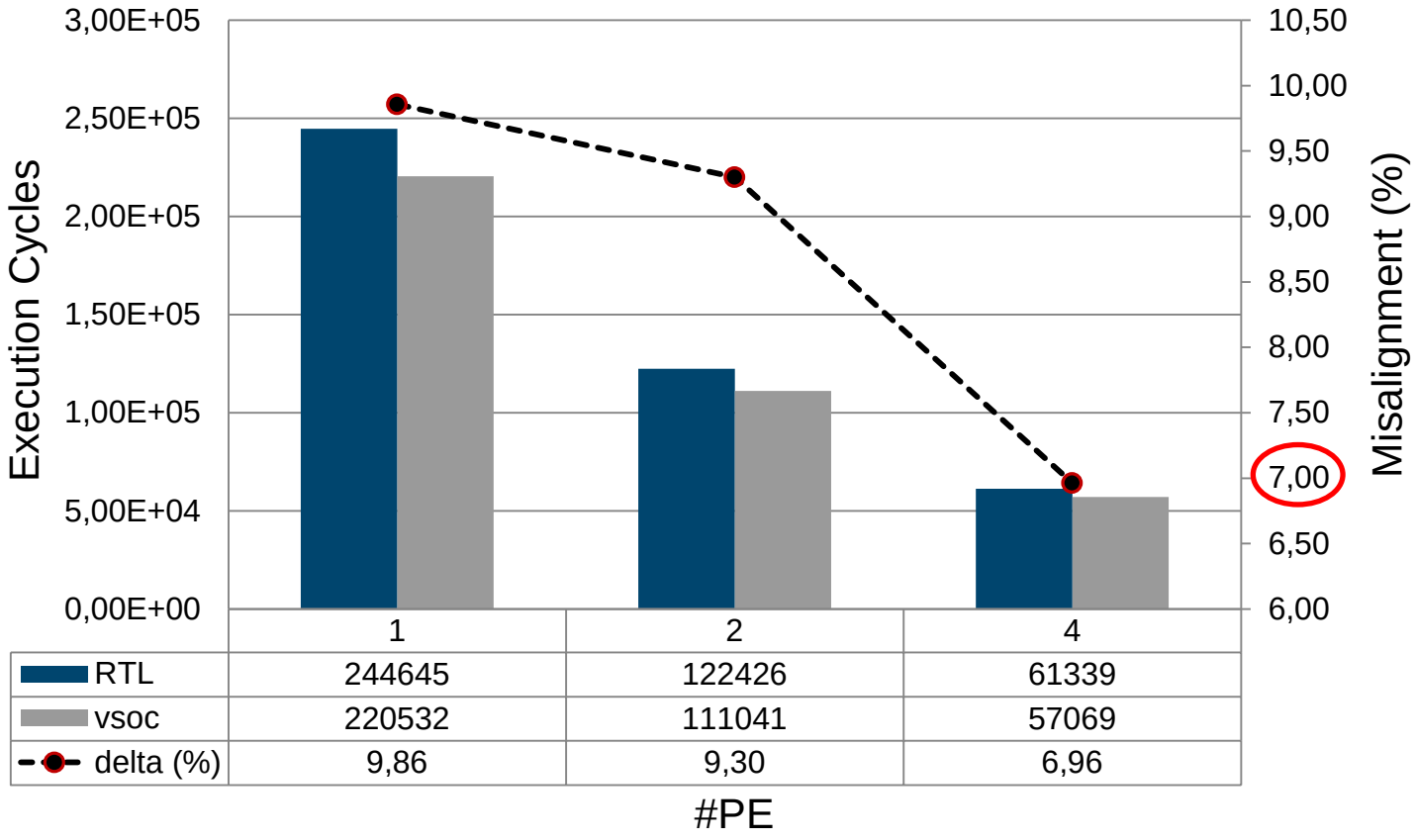
- **VSoC**: SystemC cycle-accurate virtual platform [Bortolotti13]
  - models 8x16 openRISC-based platform [Gautschi14]
  - `vsoc::power_module()` collects activity of each module

- **Power numbers**
  - 6T, 8T : LP 65nm commercial memory compiler
  - PE, DMA, LIC : scaled to 65nm from 28nm RTL platform

	DYNAMIC [µW/MHz]					
	6T-MEM		8T-MEM		PE	
	HP	LP	HP	LP	HP	LP
IDLE	2.20	0.54	2.32	0.56		
READ	11.79	2.87	12.04	2.93	68.76	16.74
WRITE	13.88	3.38	14.11	3.43		
	LEAKAGE [µW]					
	6T-MEM		8T-MEM		PE	
	HP	LP	HP	LP	HP	LP
-40 C	0.61	0.31	0.27	0.13	0.63	0.32
25 C	11.56	5.89	5.35	2.63	11.18	5.69
125 C	326.77	166.23	158.77	80.77	338.44	172.17

SystemC – RTL alignment

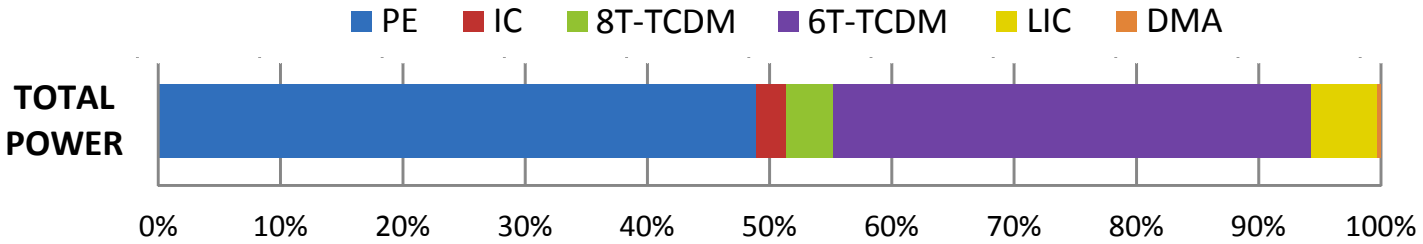
- Matrix Multiplication benchmark
  - Scalability analysis and RTL comparison



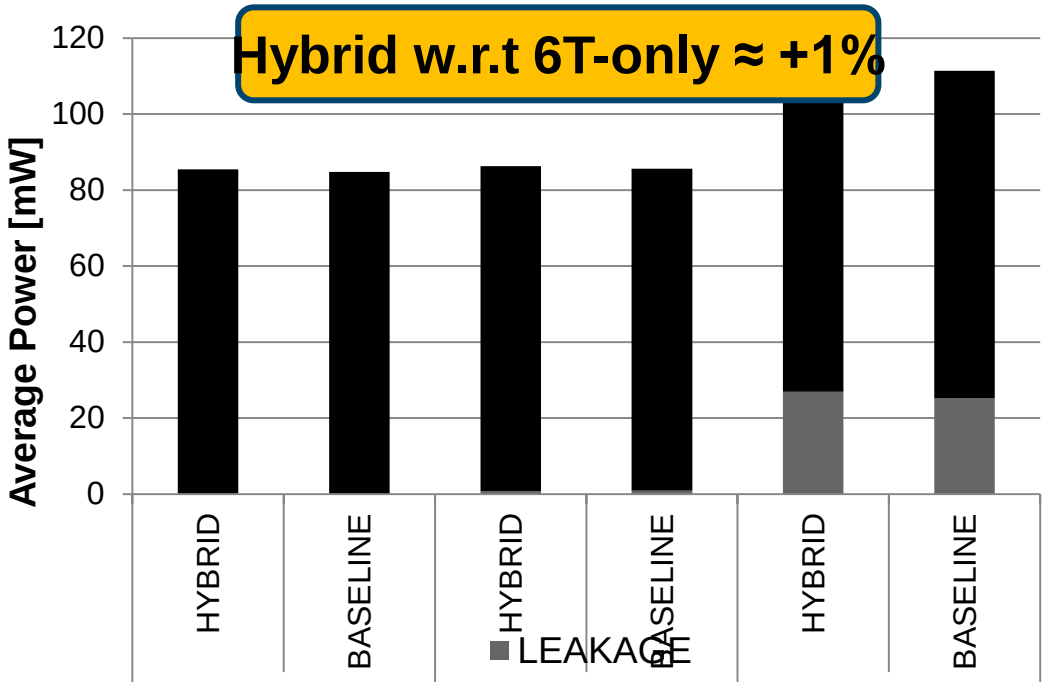
	1	2	4
RTL	244645	122426	61339
vsoc	220532	111041	57069
delta (%)	9,86	9,30	6,96

HP Phase @ (100MHz,1.2V)

- All PEs are active performing CS in parallel, DMA is idle
- Both 8T (*samples*) and 6T banks (*stack, CS data*) are active

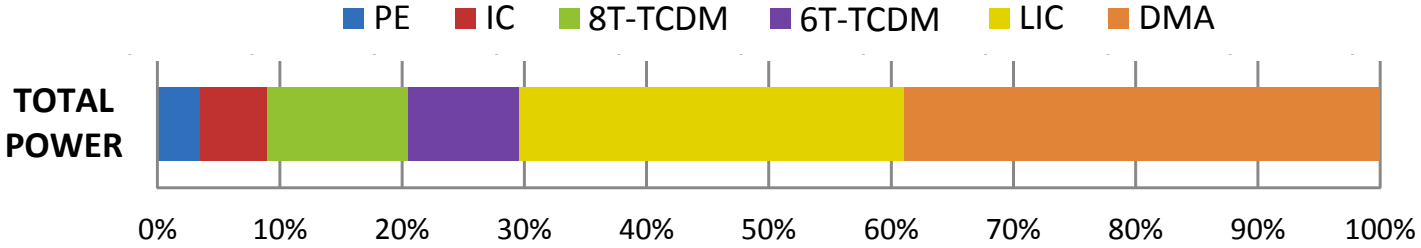


BASELINE
TCDM 128KB
@ 100MHz,1.2V

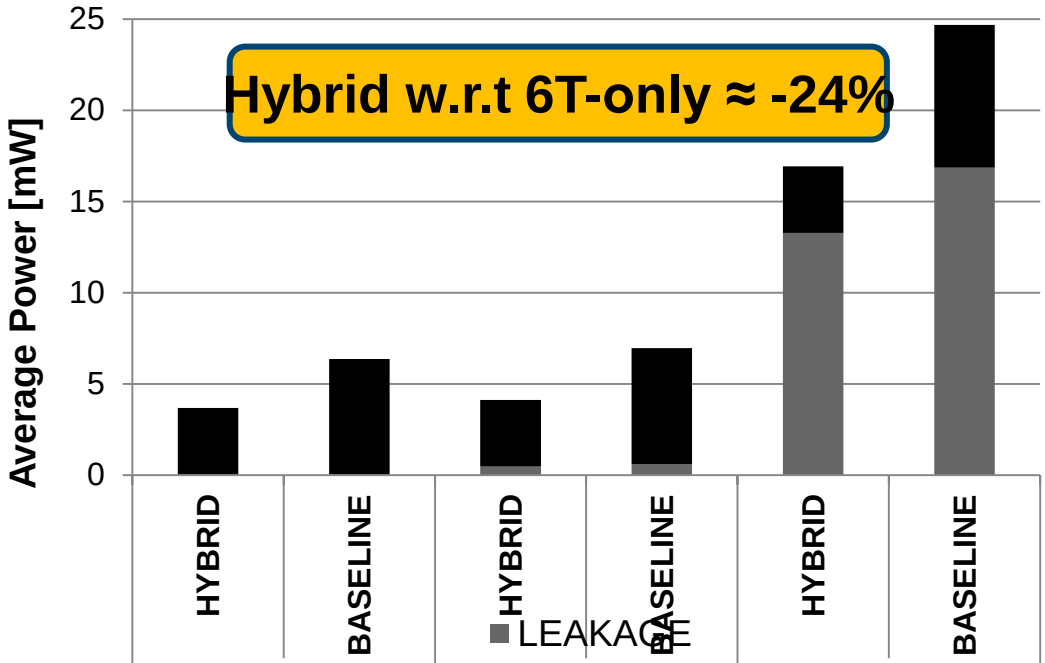


LP Phase @ (10MHz,0.6V)

- PEs are idle (leakage) while DMA is moving samples from AFE to 8T-TCDM
- 6T banks are in data retentive state, 8T banks are active

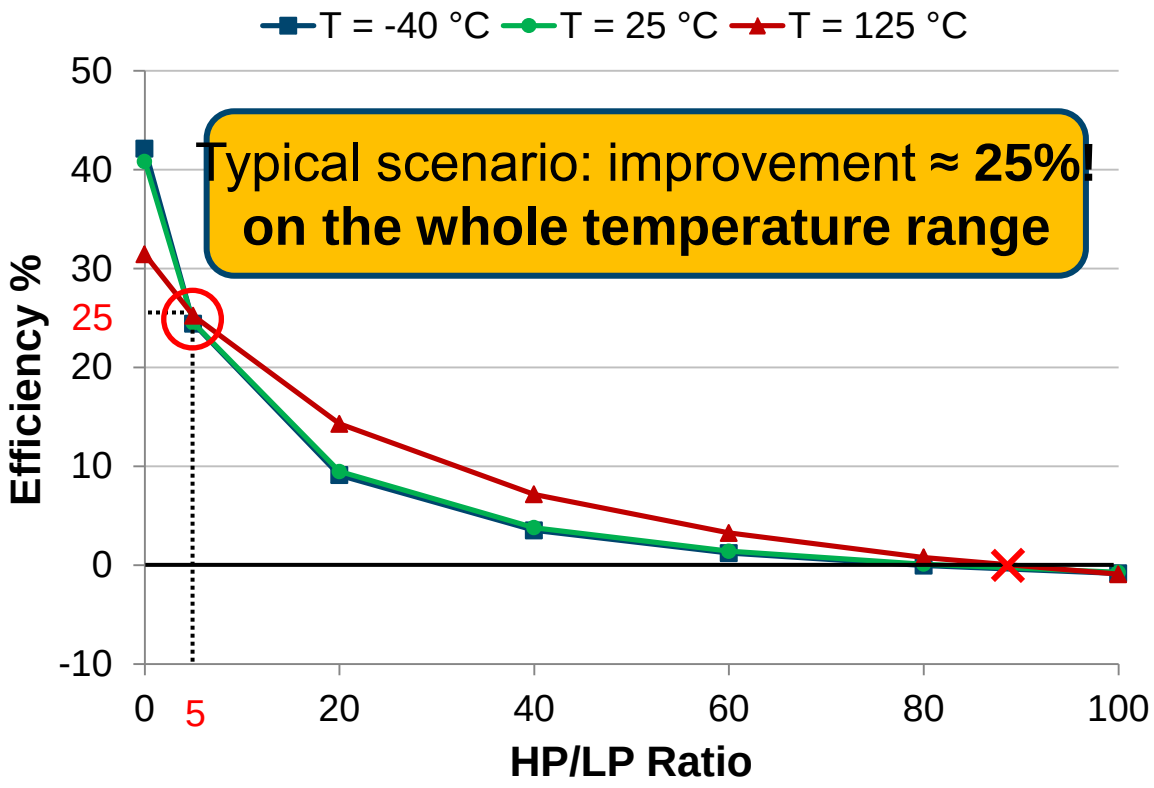


BASELINE
TCDM 16KB
@ 10MHz,0.8V



Energy Efficiency

- HP Phase: slightly higher power consumption
- LP Phase: much lower power consumption
- **Overall Efficiency** considering the amount of time in LP and HP phases



Area Overhead

- Estimation of **overhead w.r.t. 6T-only** (*iso-size* comparison)
- Overhead of extra-circuitry for the hybrid memory negligible

Element	Hybrid [mm²]	Baseline [mm²]
PEs	0.85408	0.85408
6T-TCDM	0.70652	0.80746
8T-TCDM	0.13323	-
6T IM	-	0.05047
8T IM	0.06662	-
DMA	0.09801	0.09801
LIC 8x16	0.23348	0.23348
TOTAL	**2.09194**	**2.04349**

Area Overhead

- Estimation of **overhead w.r.t. 6T-only** (*iso-size* comparison)
- Overhead of extra-circuitry for the hybrid memory negligible

Element	Hybrid [mm²]	Baseline [mm²]
PEs	0.85408	0.85408
6T-	0.70652	0.80746
TCDM		
6T IM	-	0.05047
8T IM	0.06662	-
DMA	0.09801	0.09801
LIC 8x16	0.23348	0.23348

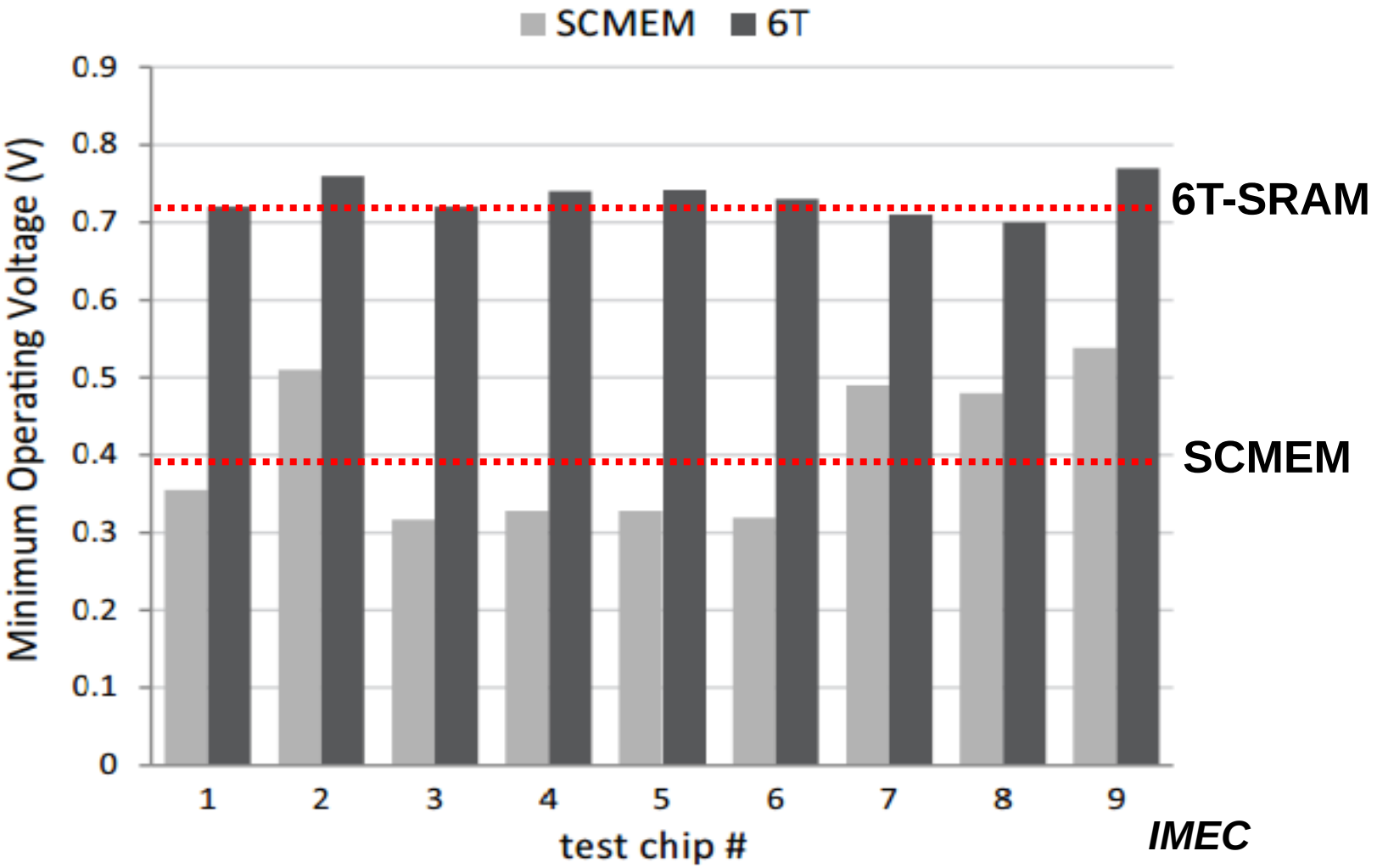
Overhead w.r.t **6T-only < 2%**

Overhead of an **8T-only** solution **≈ 14%**
Leakage would have a high impact!

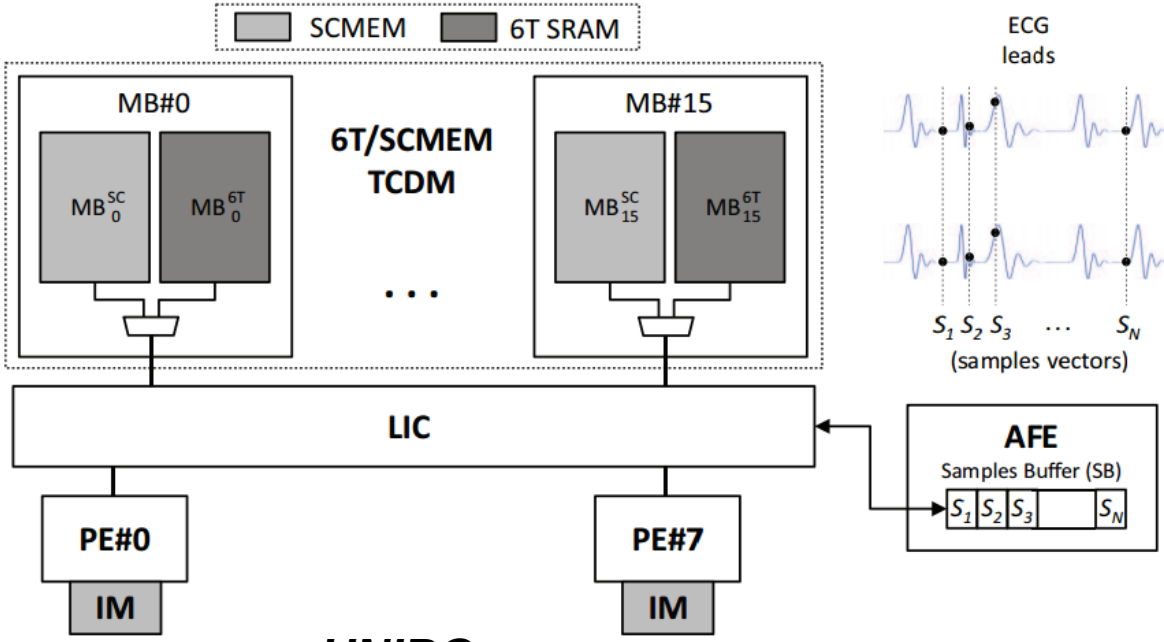
Outline

- Power in digital systems
- Energy Management
- Monitoring
- Task Allocation
- Reconfiguration
- **Approximation**

Approximate Motivation *[Bortolotti ISLPED 2014]*

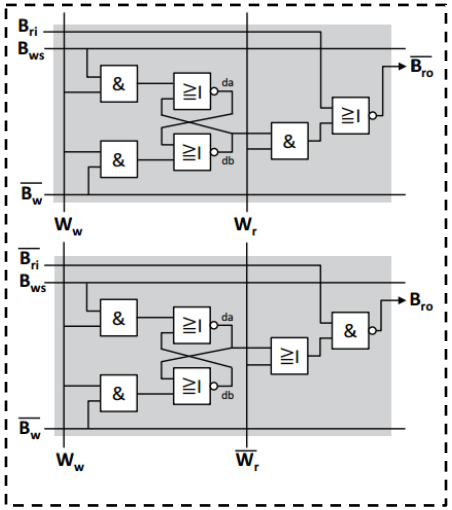


Hybrid Memory Architecture



UNIBO

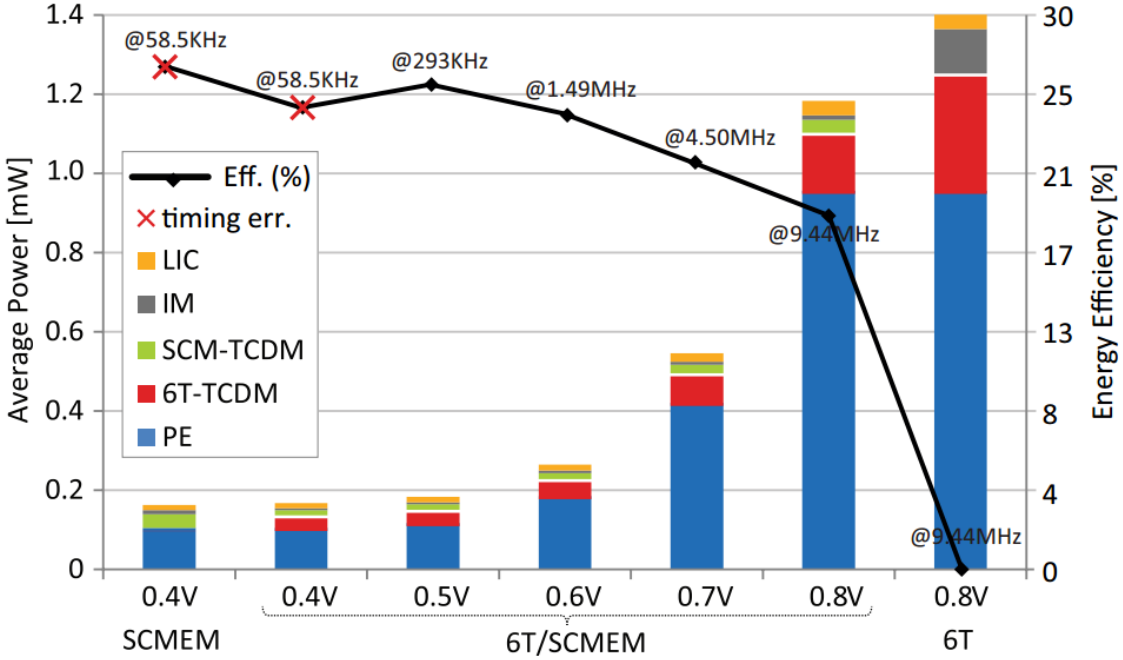
SCMEM: AOI/OAI gates



IMEC

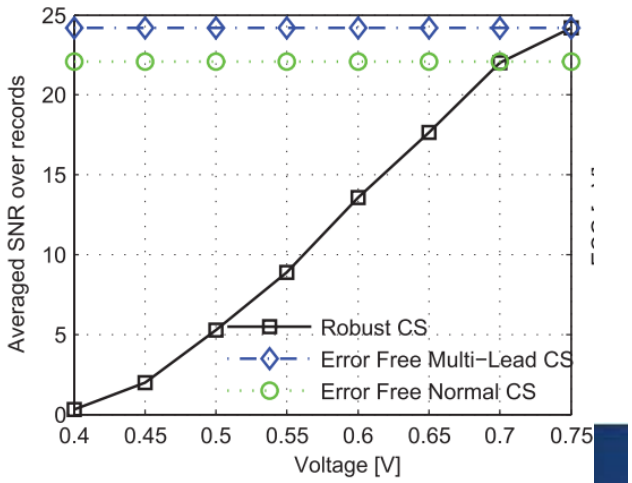
$$\min_{\tilde{\mathbf{A}},\tilde{\mathbf{E}}} \left\|\tilde{\mathbf{A}}\right\|_{1,2} + \lambda \left\|\tilde{\mathbf{E}}\right\|_1 \quad \text{s.t.:} \left\|(\boldsymbol{\Phi}+\tilde{\mathbf{E}})\boldsymbol{\Psi}\tilde{\boldsymbol{A}} - \mathbf{Y}\right\|_2 = 0$$

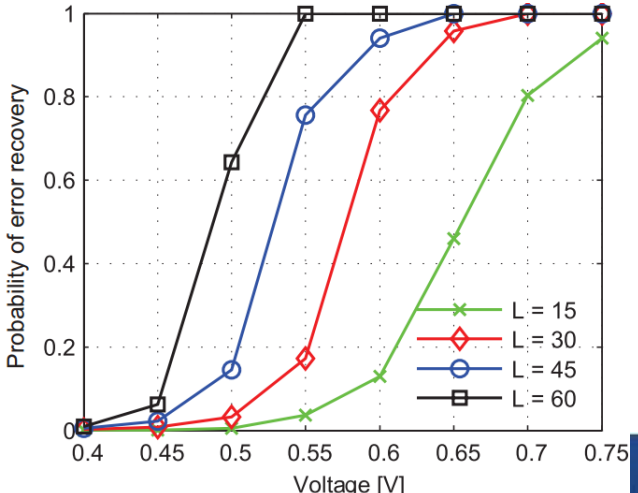
EPFL

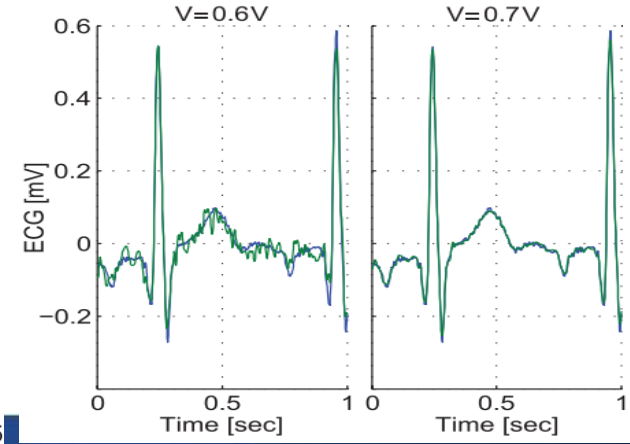


ELEMENT	6T/SCMEM [μm^2]	6T-ONLY [μm^2]	SCMEM-ONLY [μm^2]
PEs	323439	323439	323439
IM	132819	97960	132819
SCM TCDM	332048	-	597686
6T TCDM	195920	431968	-
TOT TCDM	527968	431968	597686
LIC 8x16	88420	88420	88420
TOTAL	1072646	941787	1142364

≃13%







Thank You!

Questions?

References #1

- **[Bartolini TPDS 13]** Bartolini, A; Cacciari, M.; Tilli, A; Benini, L., "*Thermal and Energy Management of High-Performance Multicores: Distributed and Self-Calibrating Model-Predictive Controller*," Parallel and Distributed Systems, IEEE Transactions on , vol.24, no.1, pp.170,183, Jan. 2013
- **[Bartolini DATE 12]** Bartolini, A; Sadri, M.; Furst, J.; Coskun, AK.; Benini, L., "*Quantifying the impact of frequency scaling on the energy efficiency of the single-chip cloud computer*," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2012 , vol., no., pp.181,186, 12-16 March 2012
- **[Paterna TC 2012]** Paterna, F.; Acquaviva, A; Caprara, A; Papariello, F.; Desoli, G.; Benini, L., "*Variability-Aware Task Allocation for Energy-Efficient Quality of Service Provisioning in Embedded Streaming Multimedia Applications*," Computers, IEEE Transactions on , vol.61, no.7, pp.939,953, July 2012.
- **[Rahimi DATE 13]** Rahimi, Abbas; Marongiu, Andrea; Burgio, Paolo; Gupta, Rajesh K.; Benini, Luca, "*Variation-tolerant OpenMP tasking on tightly-coupled processor clusters*," Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013 , vol., no., pp.541,546, 18-22 March 2013
- **[Bortolotti DATE 14]** D Bortolotti, A Bartolini, C Weis, D Rossi, L Benini, «*Hybrid memory architecture for voltage scaling in ultra-low power multi-core biomedical processors*» Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014
- **[Bortolotti ISLPED 14]** Bortolotti, Daniele, et al. "*Approximate Compressed Sensing: Ultra-Low Power Biosignal Processing via Aggressive Voltage Scaling on a Hybrid Memory Multi-core Processor.*" Proc. of 2014 IEEE International Symposium on Low Power Electronics and Design (ISLPED 2014). Vol. 1. No. EPFL-CONF-200128. IEEE/ACM Press, 2014.

References #2

- **[Gautschi14]** Gautschi M.et al., "*Customizing an Open Source Processor to Fit in an Ultra-Low Power Cluster with a Shared L1 Memory*", In: GLSVLSI 2014 (to appear).
- **[Bortolotti13]** Bortolotti D. et al., "*VirtualSoC: a Full-System Simulation Environment for Massively Parallel Heterogeneous System-on-Chip*", In: IPDPWS 2013