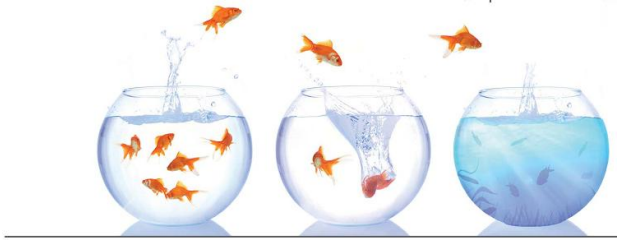
LIRMM



Systèmes multiprocesseurs embarqués adaptatifs

Gilles SASSATELLI

SEmba 2010

Autrans, October 18, 2010

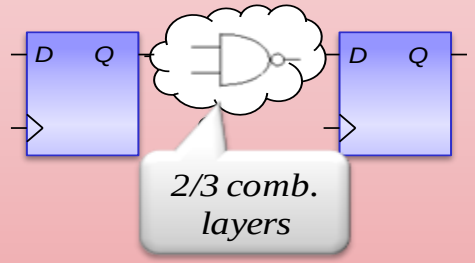
Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier
LIRMM
cnrs advancing the frontiers
UNIVERSITÉ MONTPELLIER 2 SCIENCES ET TECHNIQUES

Further increasing performance?

- Shrinking is not allowing this anymore
  - Shedding light from General Purpose computing…

Remember that good old fellow Pentium IV ? **Intel Netbust architecture Family**

- Initial plans:
  - Phase1: 20 stages (Willamette and Northwood cores), target: 3.4 GHz.
  - Phase 2: 31 stages (Prescott and Cedar Mill cores), target: 5 GHz
  - Phase 3: 45 stages (Tejas core), to reach over 7 GHz. (**never happened**)



Performance… not the only challenge

- Hitting ILP wall, the Power wall, what are the options?
  - ILP reached a plateau, frequency cannot be scaled further up…
  - …how do we utilize the additional transistors Moore's delivers every 18 month?

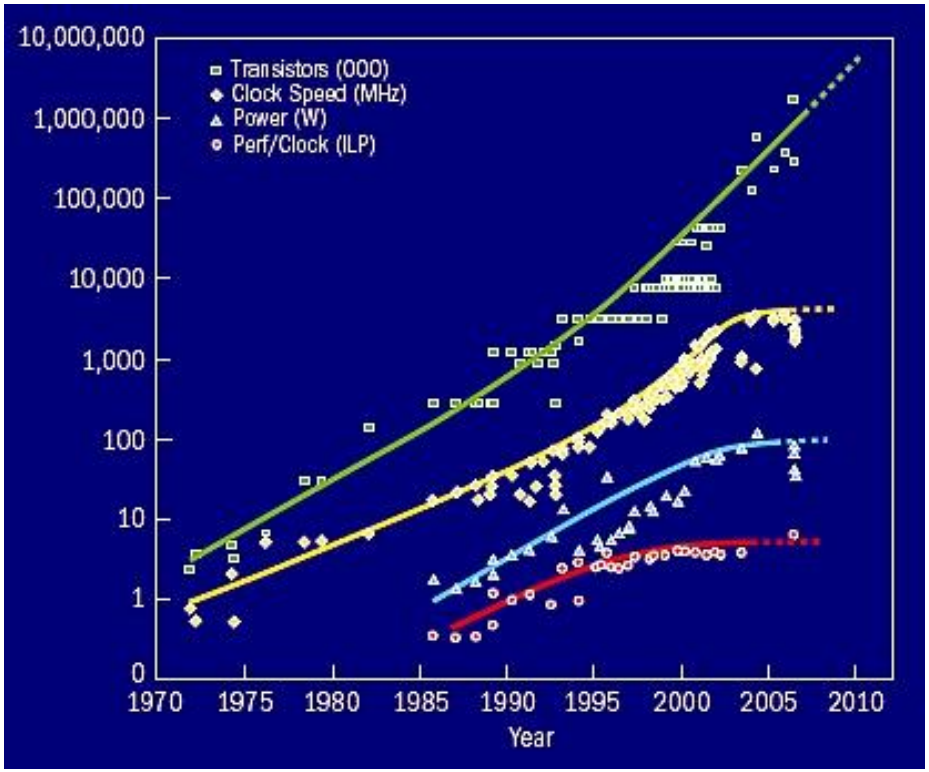
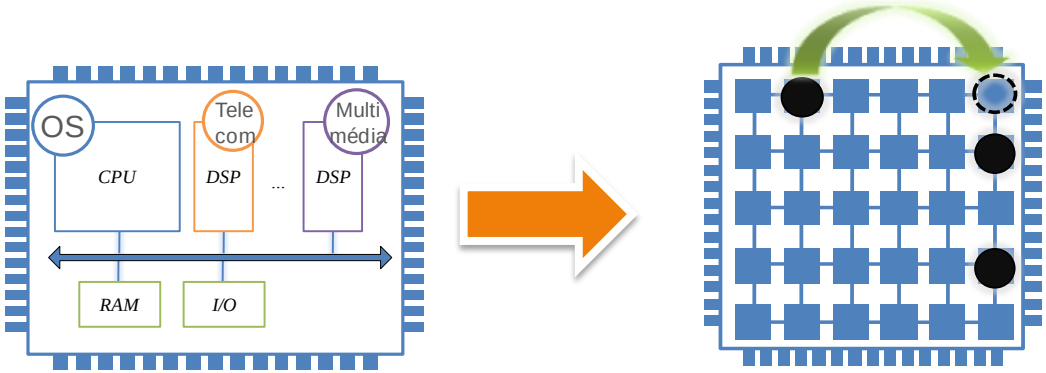


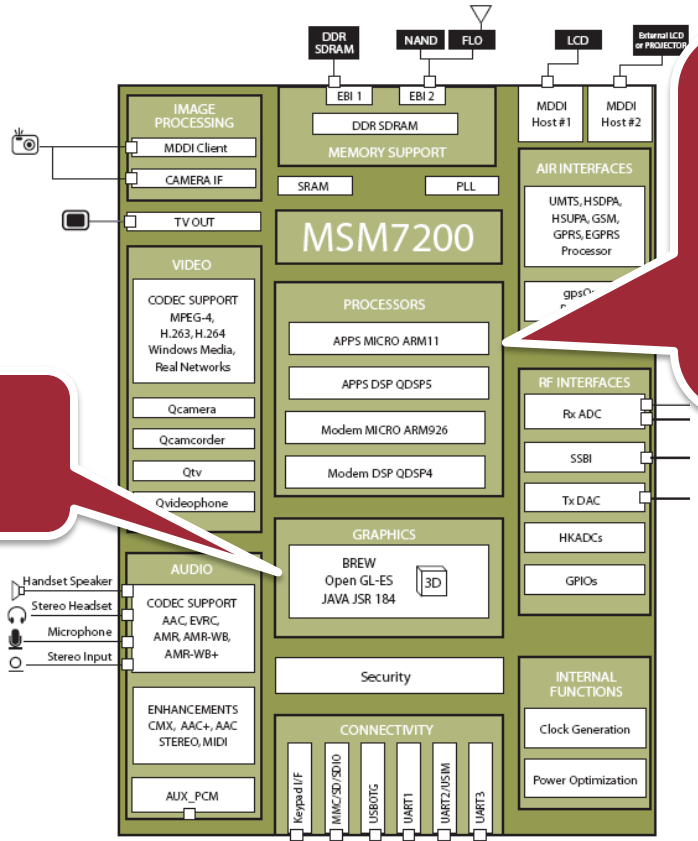
Illustration: A. Tovey Source: D. Patterson, UC–Berkeley

0- PRINCIPLES:
Homogeneous, Distributed & Adaptive



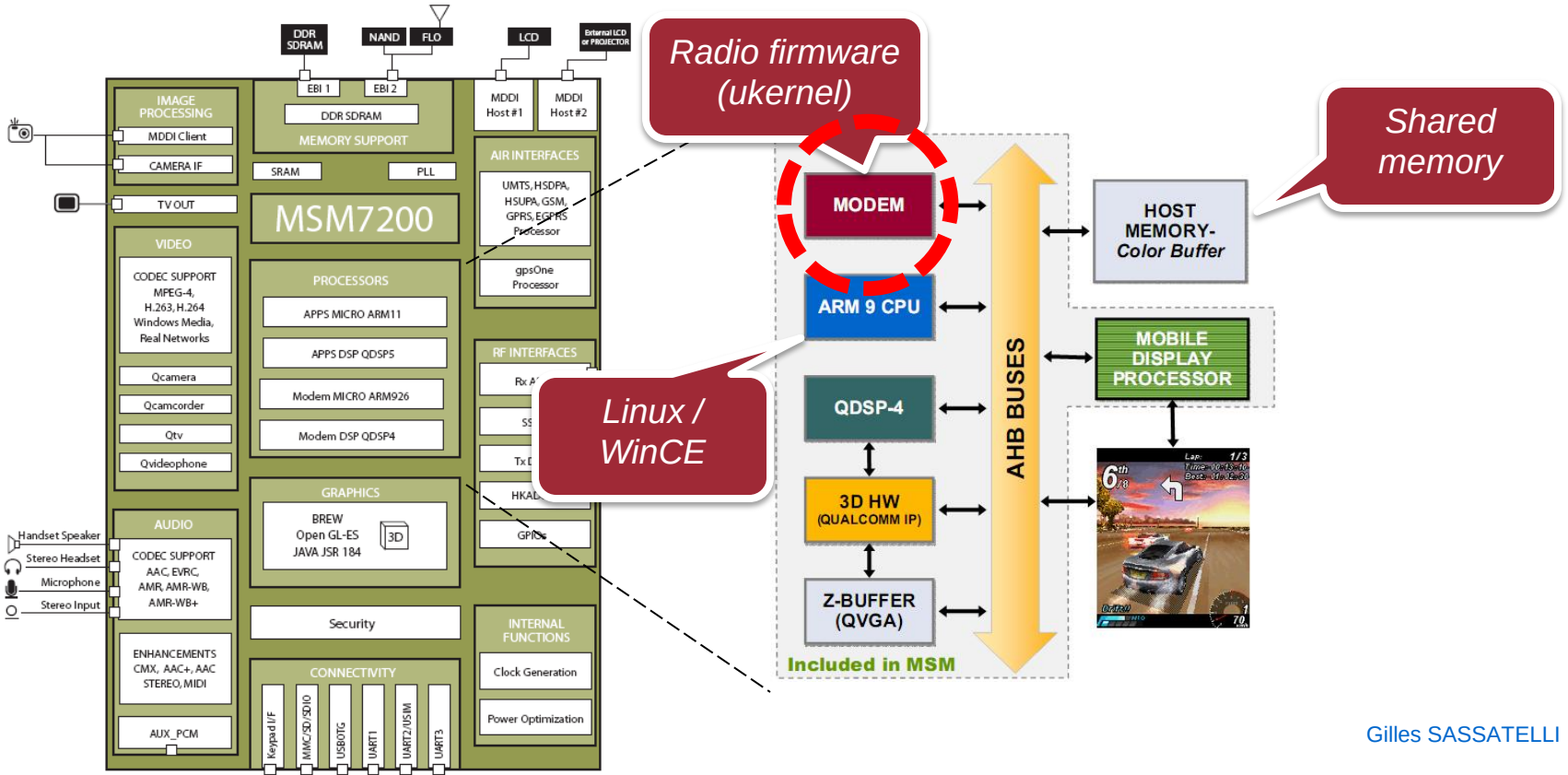
Shared memory / message passing

- Example in the Embedded domain: Qualcomm MSM7200
  - MPSoC + specialized ISPs + HW accelerators (1, 2, 3)
  - 4 cores heterogeneous / shared memory design + a number of accelerators / IFs
  - Packs up each and every possible accelerator,
  - Customer pays royalties for using optional features (3D acceleration, USB host, etc.)



Shared memory / message passing

- Example in the Embedded domain: Qualcomm MSM7200
  - MPSoC + specialized ISPs + HW accelerators (1, 2, 3)
  - 4 cores heterogeneous / shared memory design + a number of accelerators / IFs
  - Packs up each and every possible accelerator,
  - Customer pays royalties for using optional features (3D acceleration, USB host, etc.)

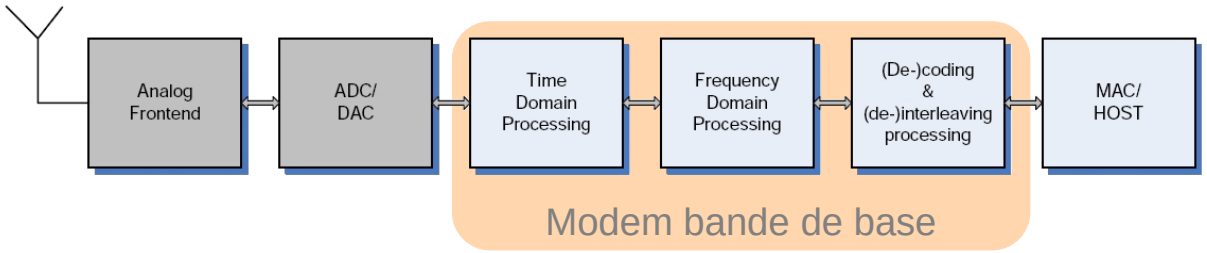


Scaling performance: the modem

LTE: Long Term Evolution

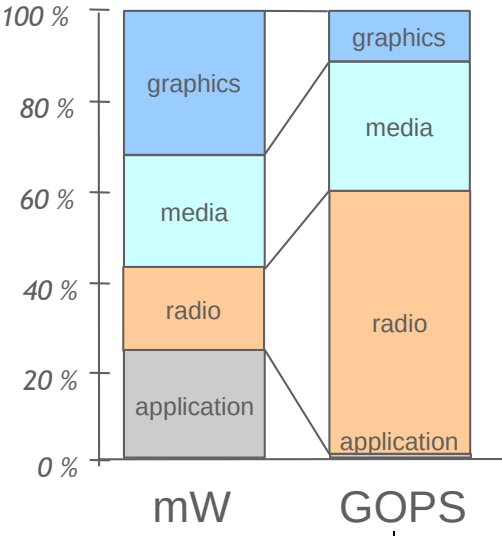
- Baseband complexity from GSM to 4G

Standard	2G GSM	2.5G GPRS	3G UMTS	3.5G HSPA	4G LTE
Débit (Mb/s)	0.01	0.1	1	10	100
Complexité (MOPS)	5	50	500	5000	50000
Consommation (mW)	100	200	300	400	500
Noeud Technologique	130nm	90nm	65nm	45nm	32nm



Radio Baseband

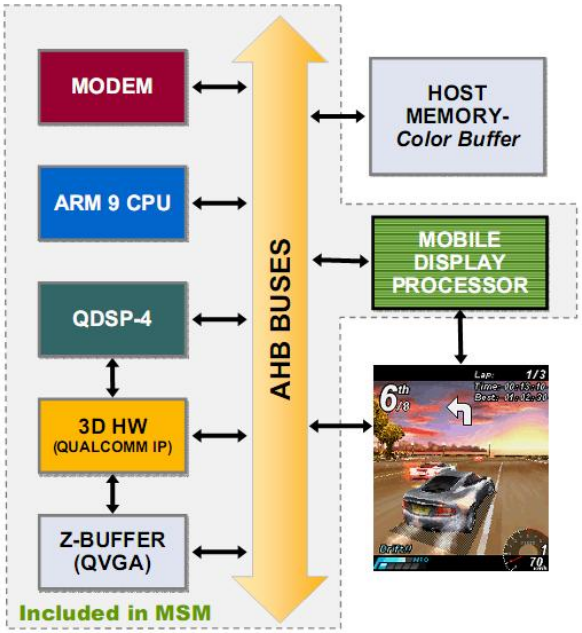
- *50 GOPS*
- *500 mW*



Scaling – what's at stake?

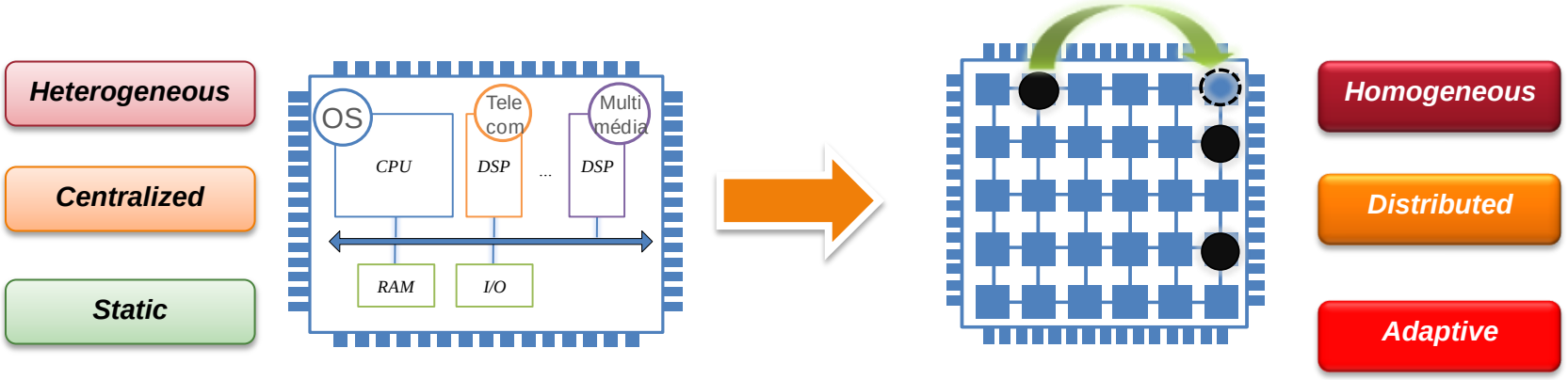
- Does this template scale?
  - Heterogeneity makes it difficult to increase processor count (TTM, progr.)
  - And yields to decreased overall silicon usage (ex: 2D/3D seldom used)

Heterogeneous ***Static*** ***Centralized***

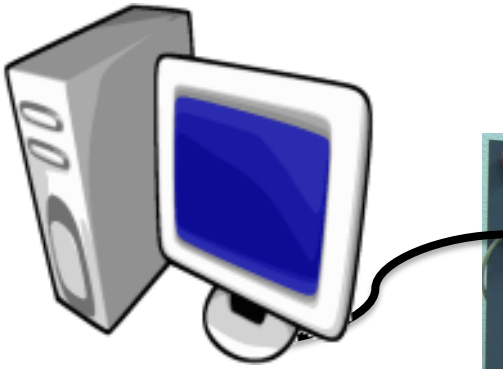


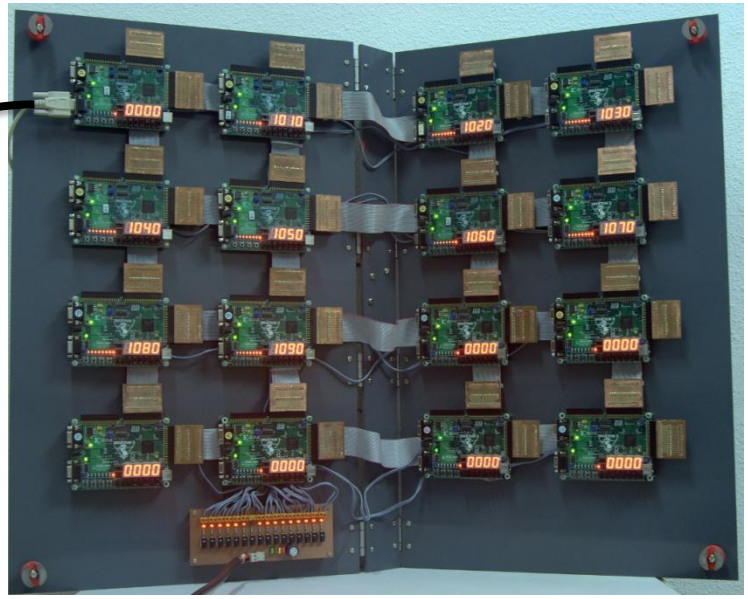
Scaling?

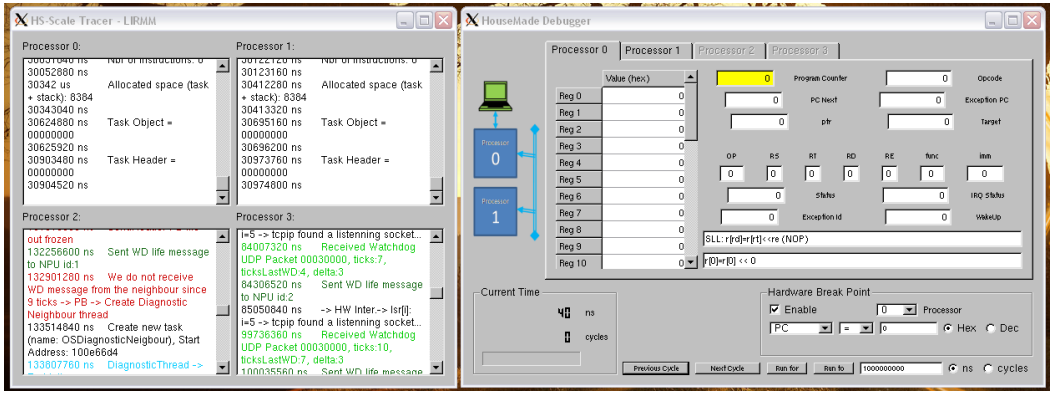
- 180 U-turn: Homogeneous, distributed & adaptive systems



1- AN EXPLORATORY MPSoC: HS-SCALE







HS-scale

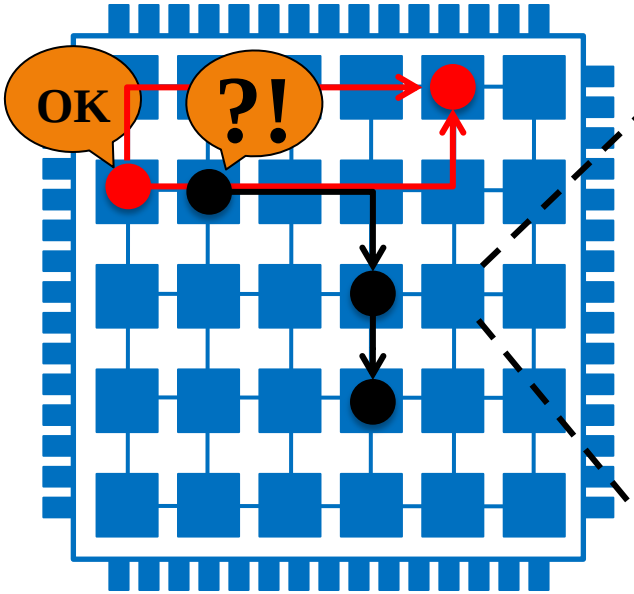
- Concepts:
  - A grid of tiny networked (Hermes XY NoC) PEs
  - PE made of simplest RISC + RAM + µK + NI → Private memories, message passing
  - Distributed decision-making, in SW

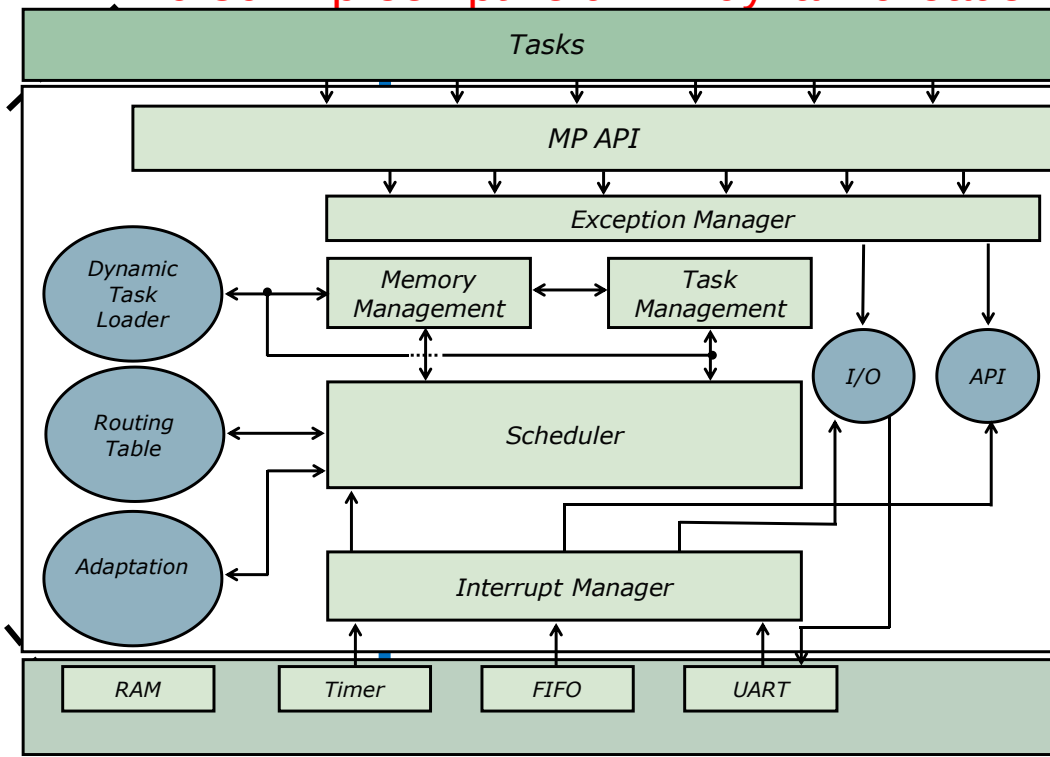
Homogeneous

Distributed

Adaptive

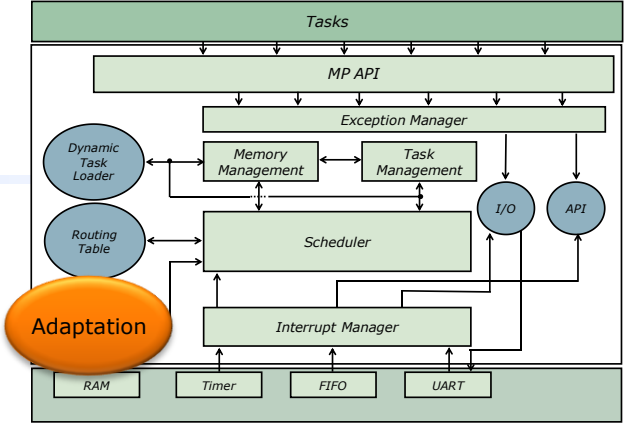
20-30kB preemptive uK w. dynamic loader



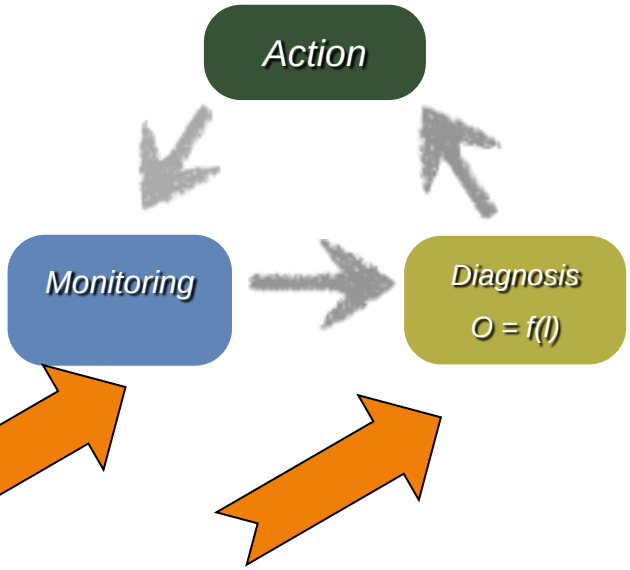


Adaptation

Adaptive

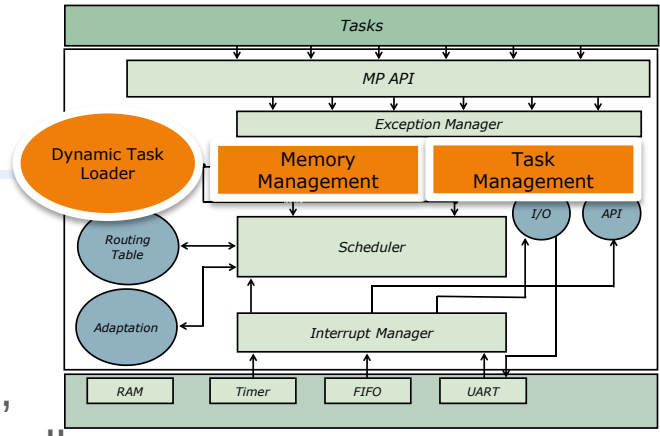


- What are the main levers?
  - Voltage/Frequency scaling (perf/power)
  - Route reconfiguration
  - Rescheduling
  - Task migration
  - Algorithmic reconfigurations
  - …



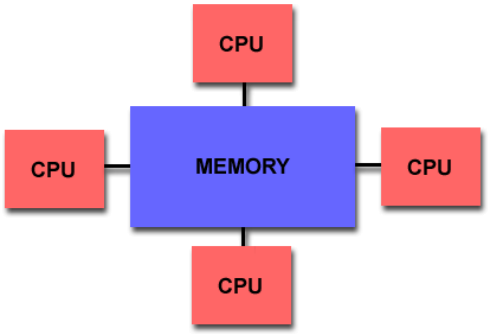
- And what about decision making?
  - Scenario-driven
  - monitoring-driven
  - Iterative
  - Heuristic
  - …

Task migration

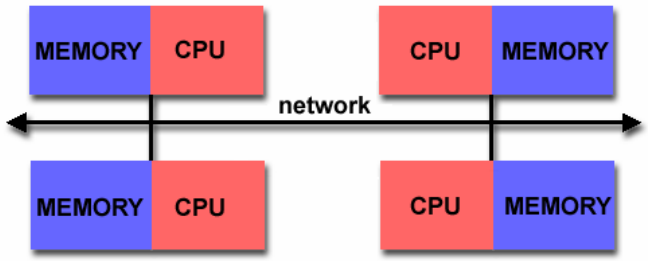


- Task migration incurs a cost..
  - Contrarily to shared / coherent memory MPSoCs, task code and context have to be migrated physically..
  - Usually requires MMU HW / dynamic loader in SW

Shared memory



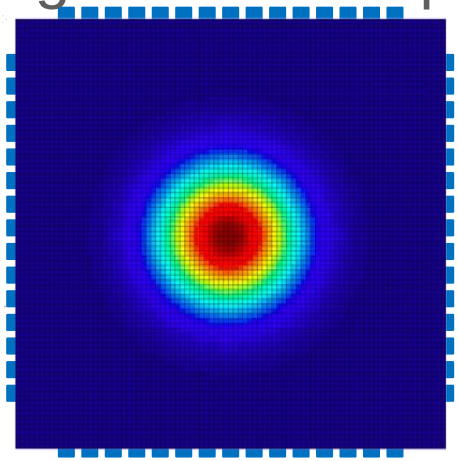
Private memories



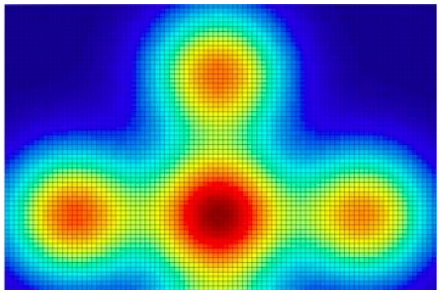
Experiment

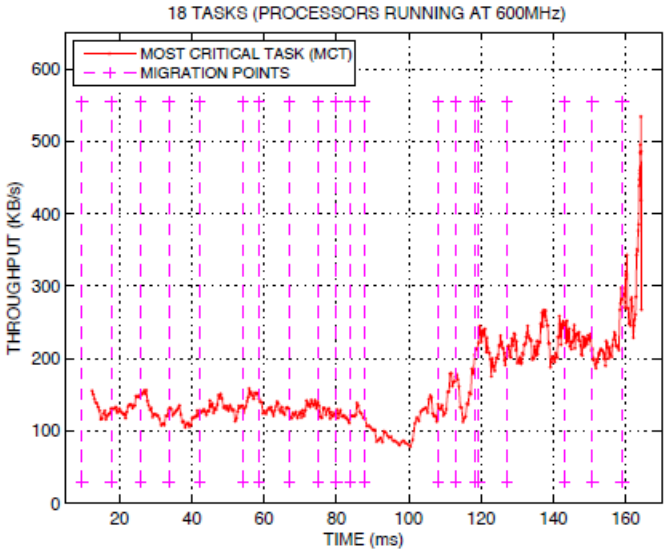
Adaptive

- Neigbor load balancing « hot / cold potatoe»

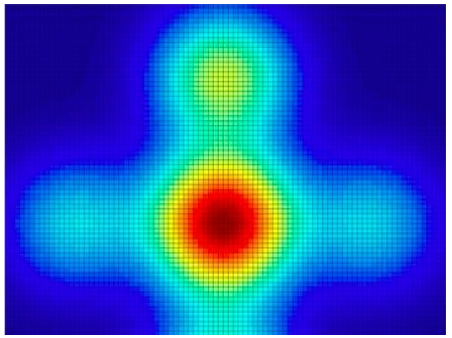


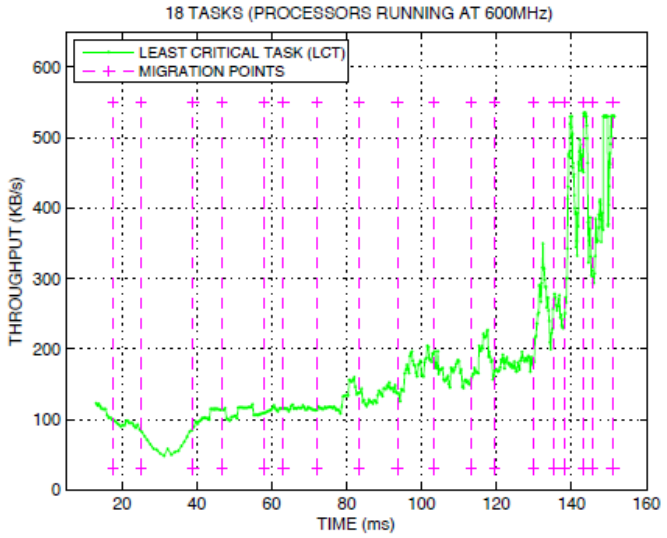
Most Critical Task





Least Critical Task

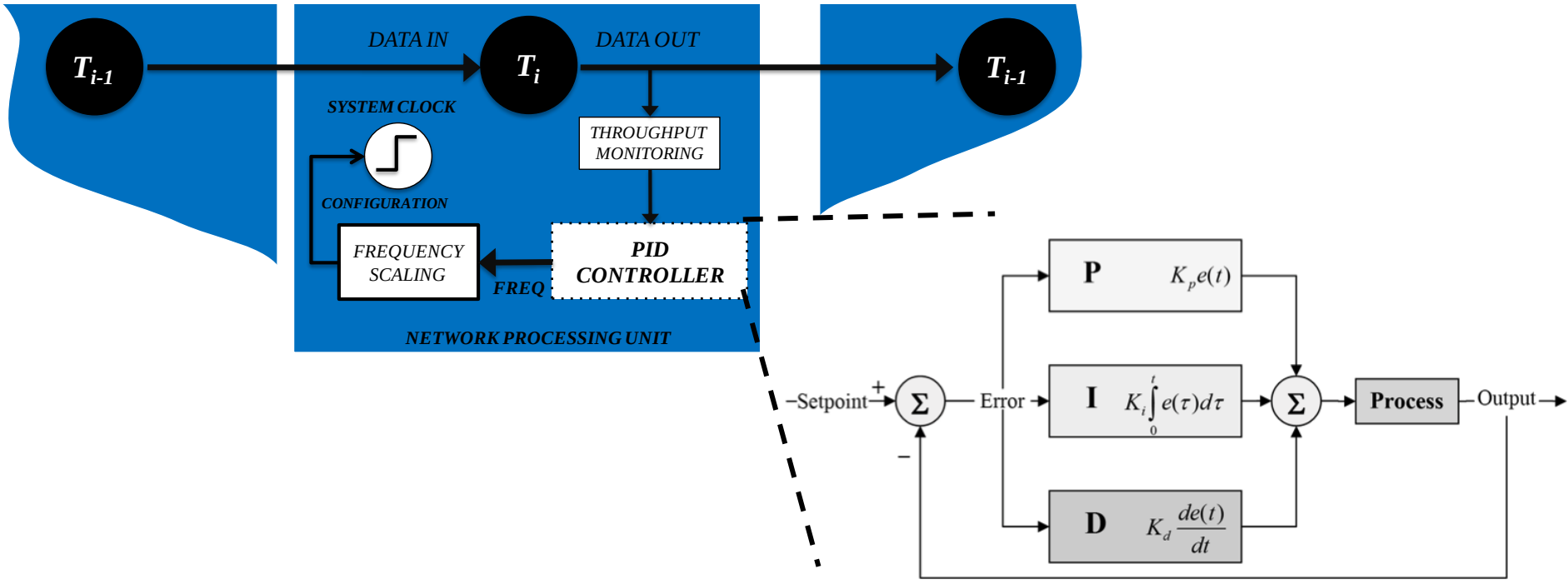




Example decision making strategy

Adaptive

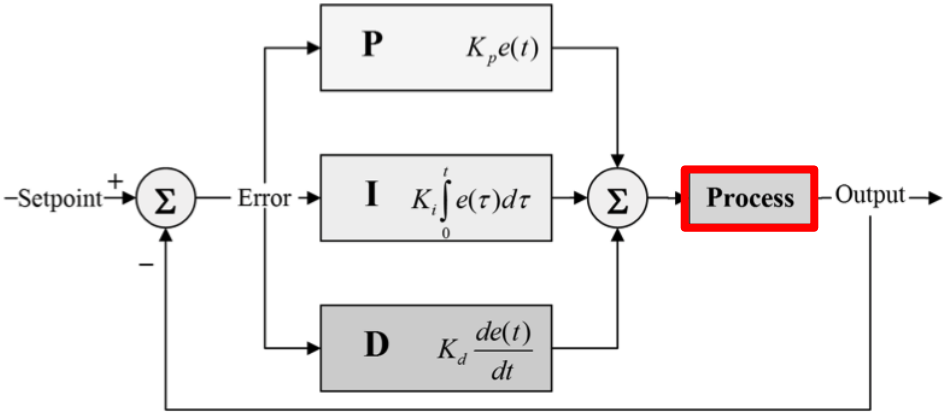
- Use of well-known feedback control-loop techniques..
  - Frequency scaling for ‘boosting’ performance whenever necessary (task migration) and revert to the lowest possible otherwise (saving power)
  - All of this at task-level, overlaid network of PID controllers

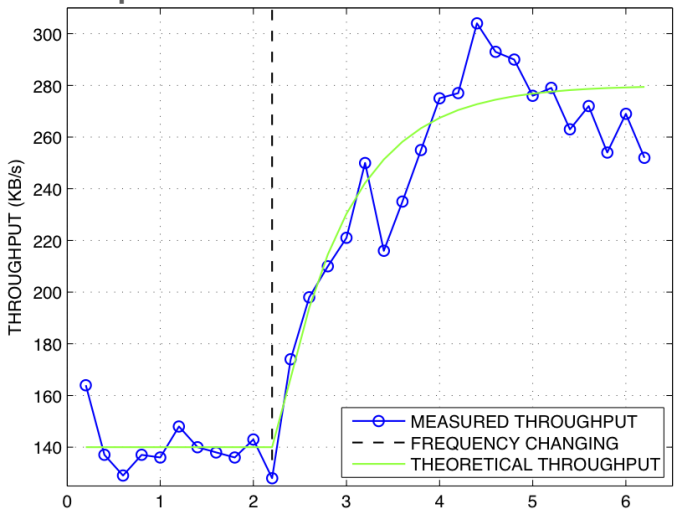


Use of PID controllers

Adaptive

- Strategy:
  - 1- model the process when exposed to a step

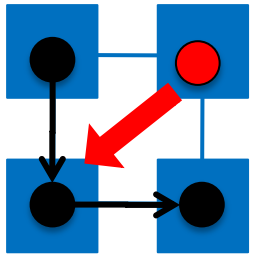


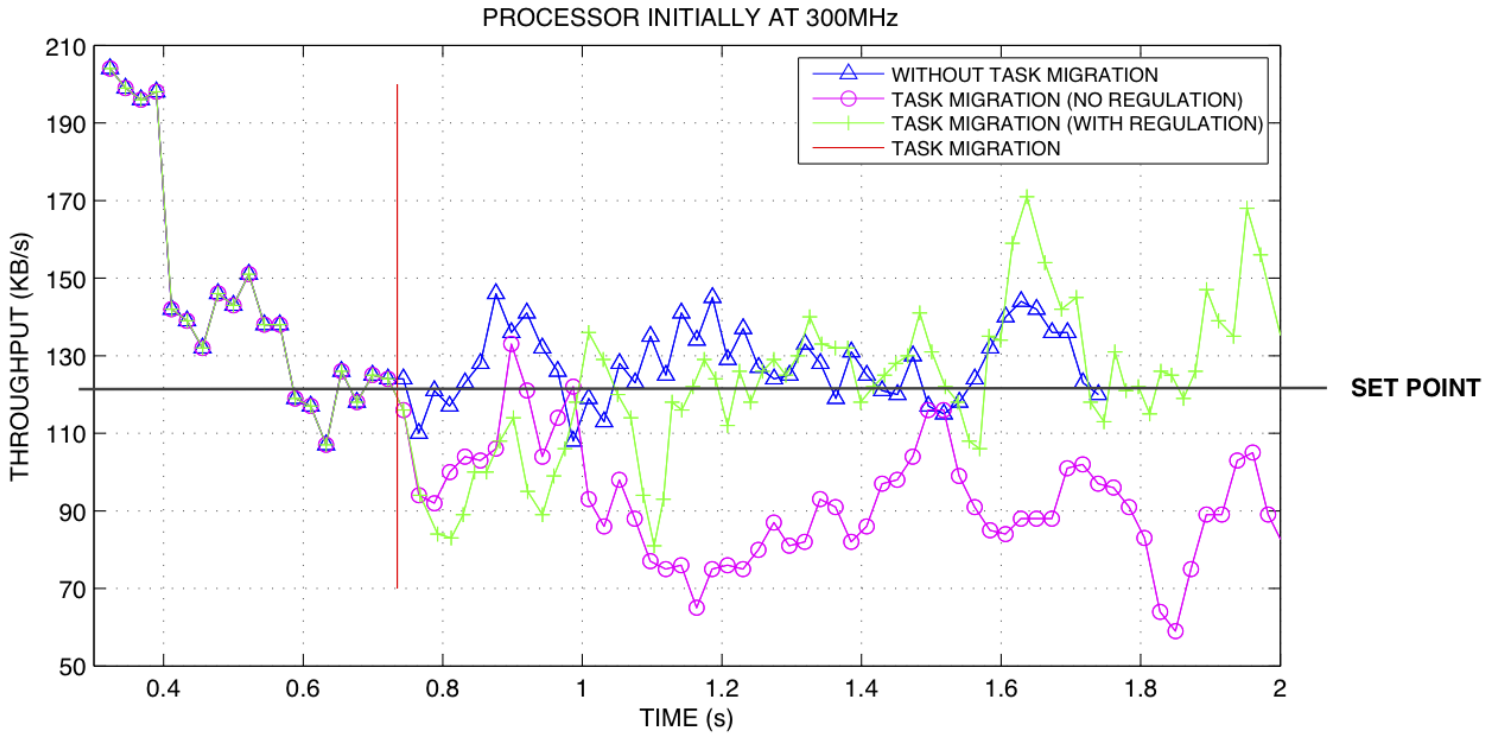


Use of PID controllers

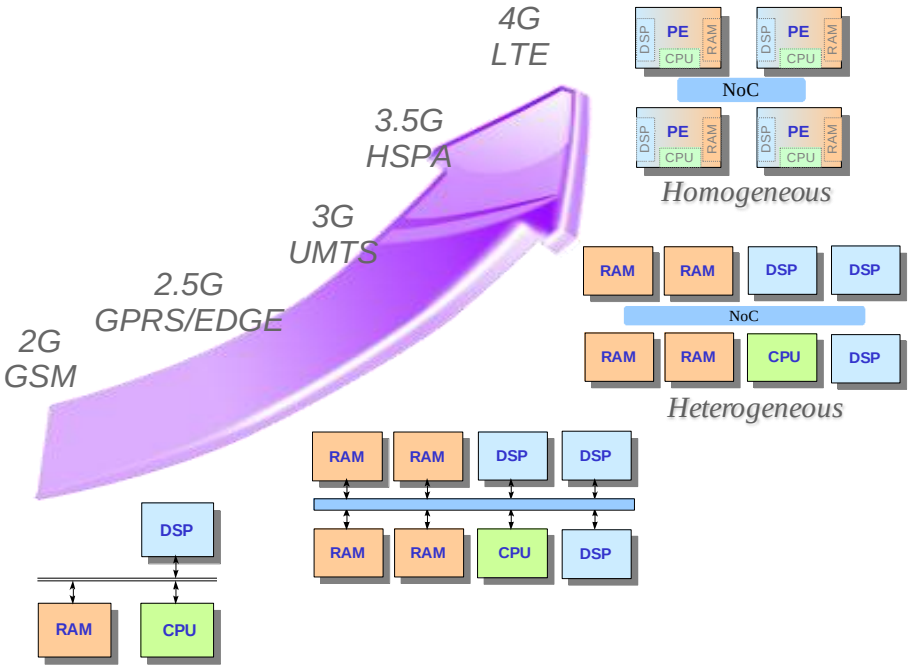
Adaptive

- And then see how the system behaves when undergoing perturbations
  - Audio/video decoder (MJPEG / ADPCM): 7 tasks
  - Here an incoming task (task migration)



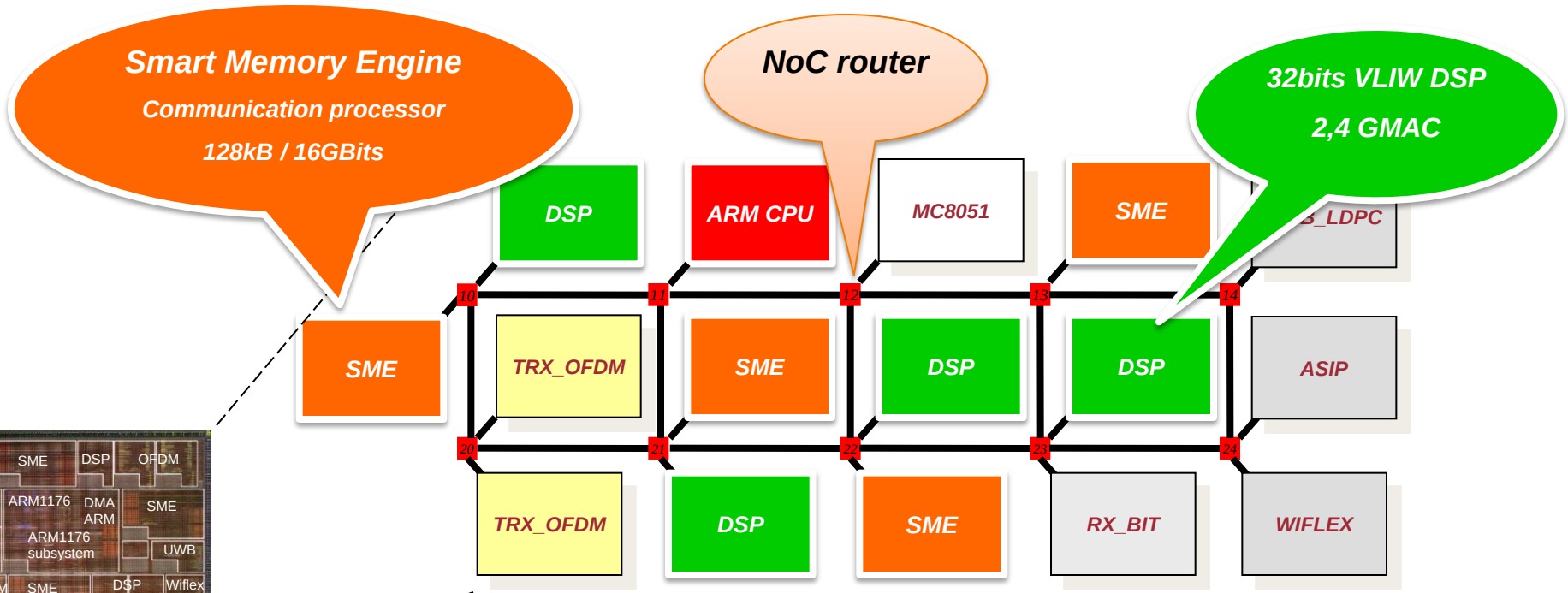


II- CASE STUDY:
GENEPY, a MPSoC for baseband processing



State of the art

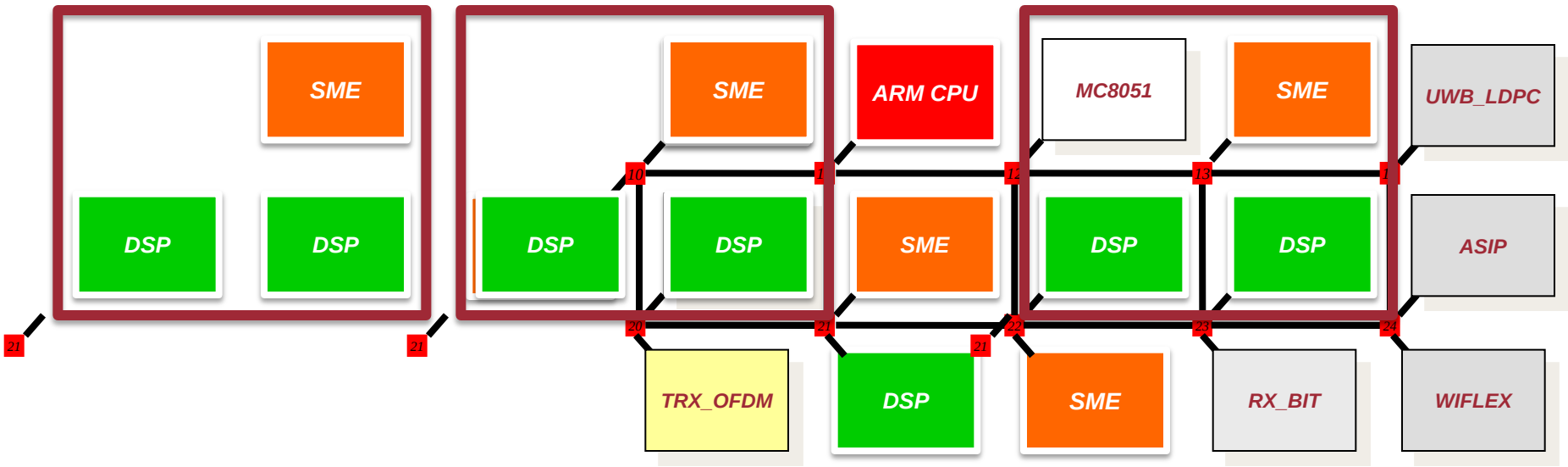
- MAGALI
  - Tuned for 3GPP-LTE telecom, 65nm ST CMOS process
  - Powerful energy efficient solution



Going homogeneous

- Keep the ingredients, change the recipe:
  - Define a tile that embeds 2 DSPs and a SME
  - Keep the ARM CPU for control purposes
  - And instantiate as many tiles as necessary

Homogeneous

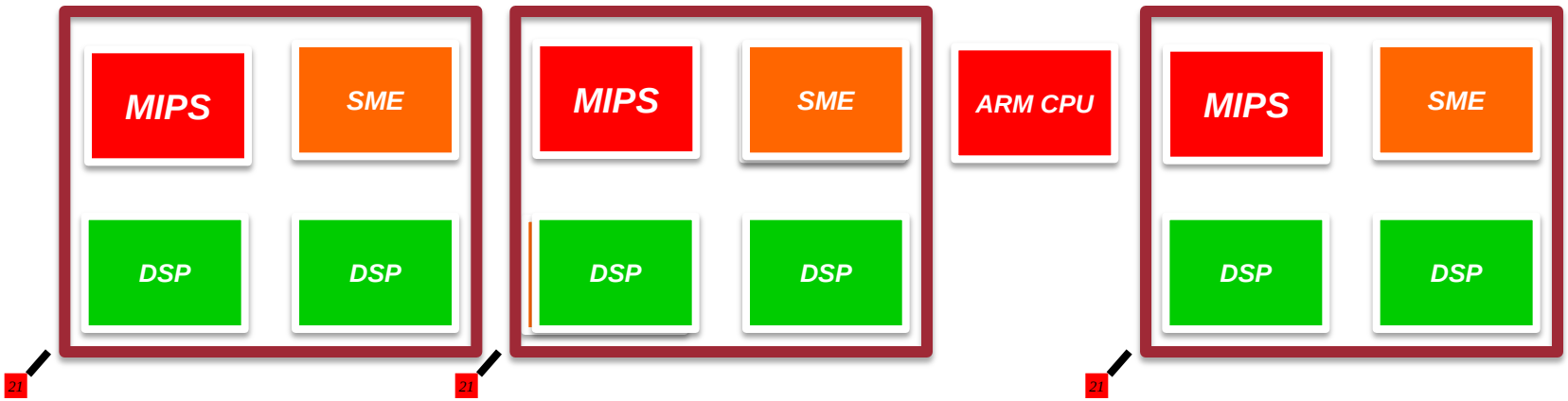


Going homogeneous, distributed control

- Distributed control?
  - Go a little further, remove the ARM and use a local control CPU

Homogeneous

Distributed

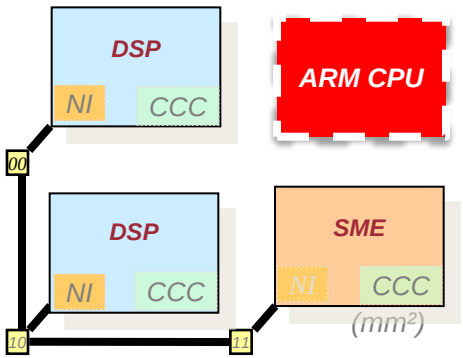


Area

ST65nm - 400MHz

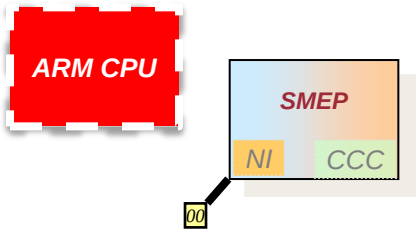
Homogeneous | **Homogeneous** | **Distributed**

Centralized control



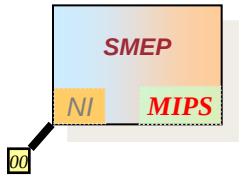
		(mm²)
Routeur	:	0,159
DSP +NI	:	0,483
SME+NI	:	1,323
TOTAL	:	2,768

Semi-distributed control



		(mm²)
Routeur	:	0,159
2 DSPs	:	0,792
SME	:	1,226
Mover	:	0,033
NI + CCC	:	0,126
TOTAL	:	2,303

Fully distributed control



		(mm²)
Routeur	:	0,159
2 DSPs	:	0,816
SME	:	1,216
Mover	:	0,033
NI + MIPS	:	0,201
TOTAL	:	2,392

- 17%

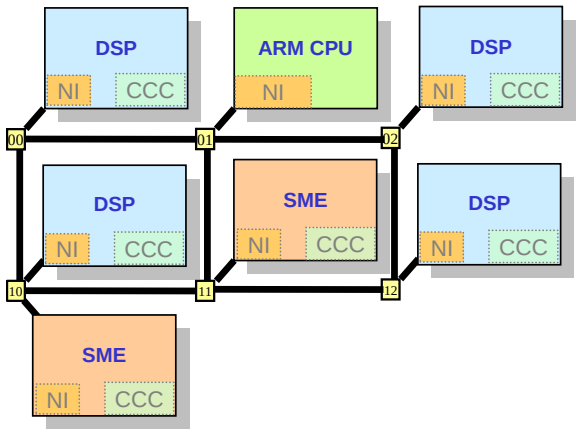
- 14%

Performance

- 400MHz – 1 frame computation

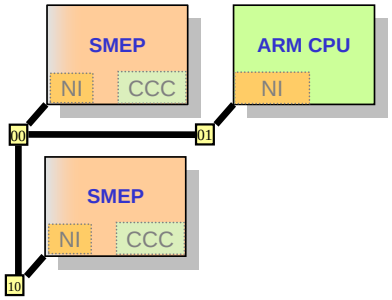
Homogeneous | **Homogeneous** | **Distributed**

Centralized control



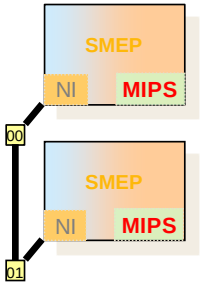
Processing time : 551 µs

Semi-distributed control



Processing time : 531 µs

Fully distributed control



Processing time : 530 µs

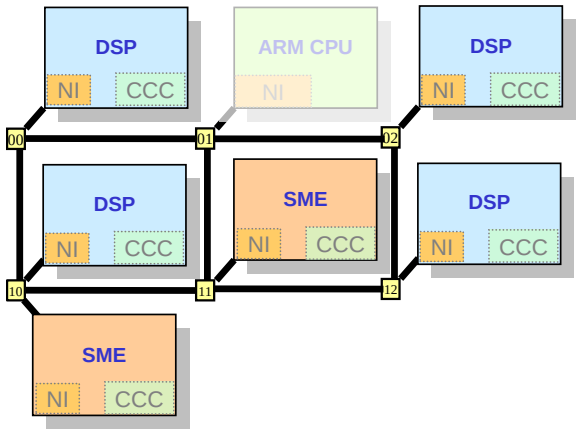
NO PERFORMANCE OVERHEAD

Power

- Post P&R extraction (w/o ARM)

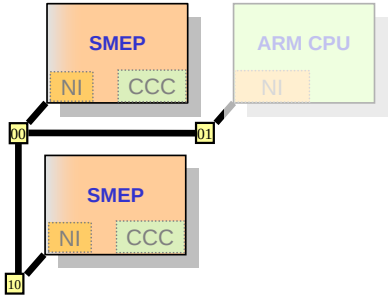
Homogeneous | **Homogeneous** | **Distributed**

Centralized control



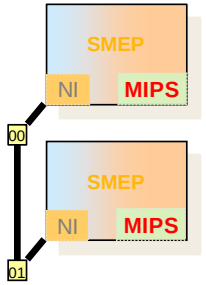
Consommation : 234 mW

Semi-distributed control



Consommation : 209 mW

Fully distributed control



Consommation : 192 mW

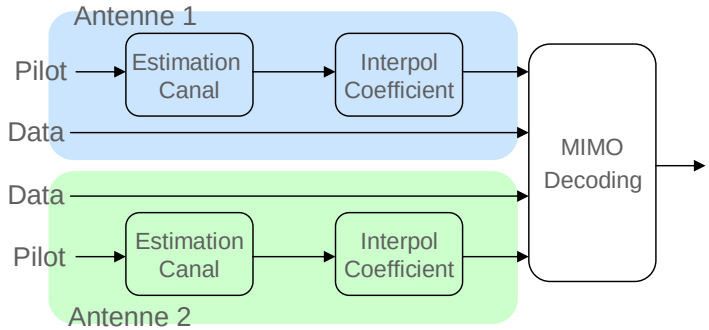
- 10%

- 18%

Task mapping

Homogeneous **Distributed**

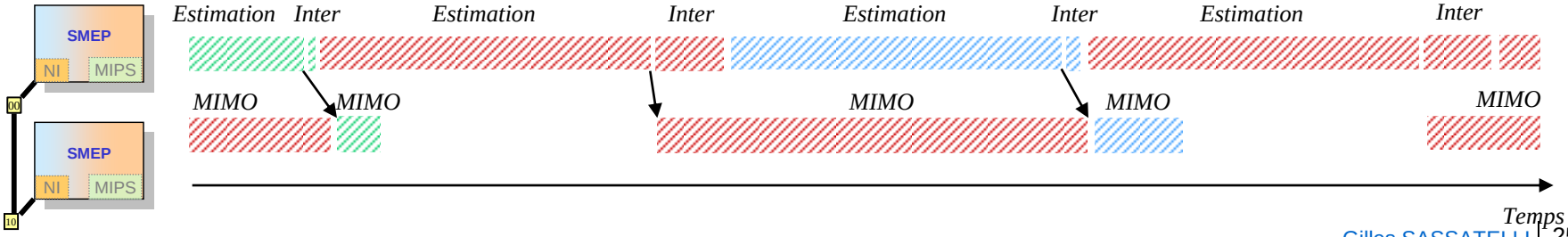
- 3GPP-LTE features adaptive modulation
  - Frames of multiple modes have to be supported,
  - With the associated processing overhead:



	Mode 1	Mode 3	Mode 5
Estimation	91000 cycles	91000 cycles	91000 cycles
Interpolation	1750 cycles	3500 cycles	17500 cycles
MIMO decoding	12000 cycles	24000 cycles	120000 cycles

x 2 x 10

Static mapping



And adaptive mapping

Homogeneous | Distributed | Adaptive

Up to another 7% power savings

- 3GPP-LTE features adaptive modulation
  - Frames of multiple modes have to be supported,
  - With the associated processing overhead:

	Mode 1	Mode 3	Mode 5
Estimation	91000 cycles	91000 cycles	91000 cycles
Interpolation	1750 cycles	3500 cycles	17500 cycles
MIMO decoding	12000 cycles	24000 cycles	120000 cycles

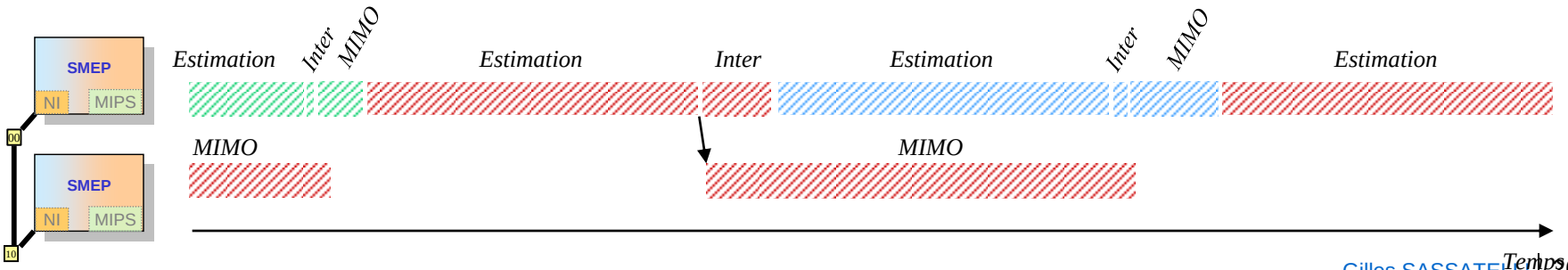
x 2

x 10

Adaptive mapping of MIMO decoding :

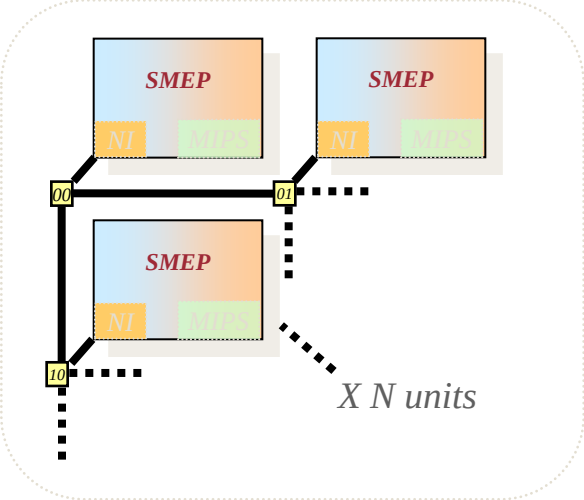
- *Mode 1,3 : SMEP 1*
- *Mode 5 : SMEP 2*

Adaptive mapping

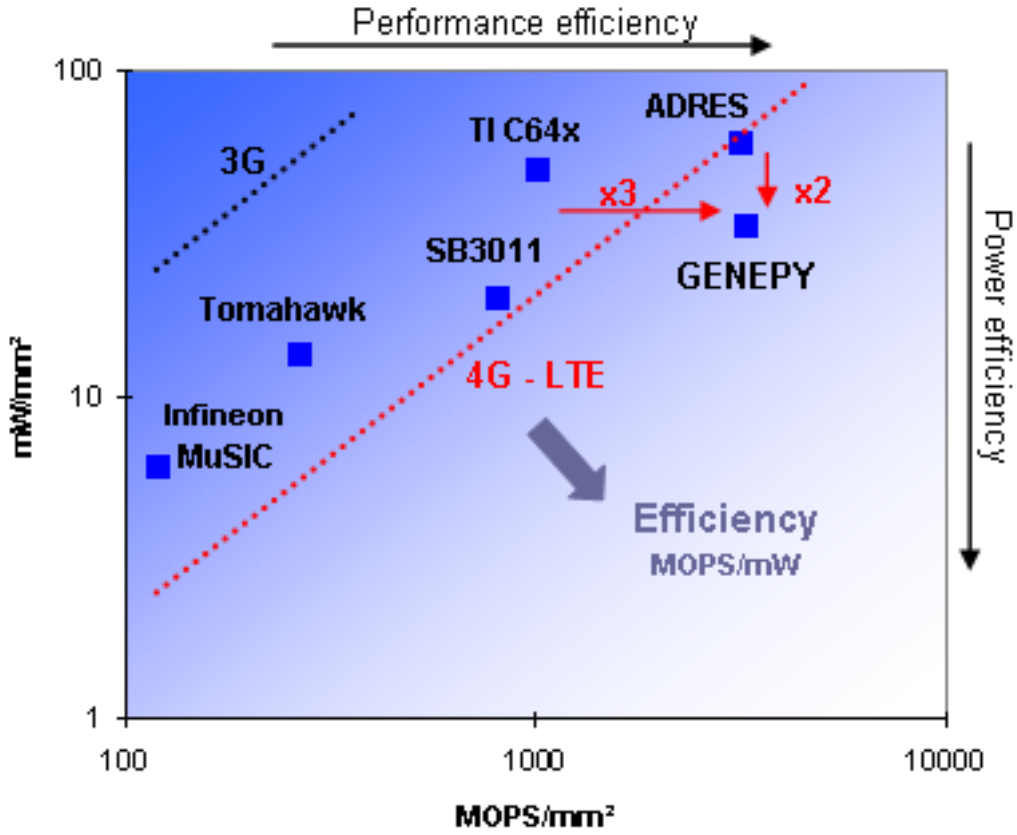


Vs. State of the art

- Compared to other MIMO capable devices



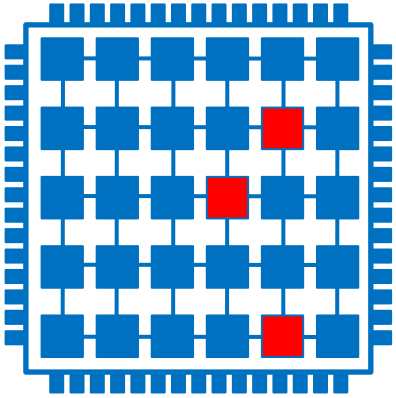
GENEPY platform

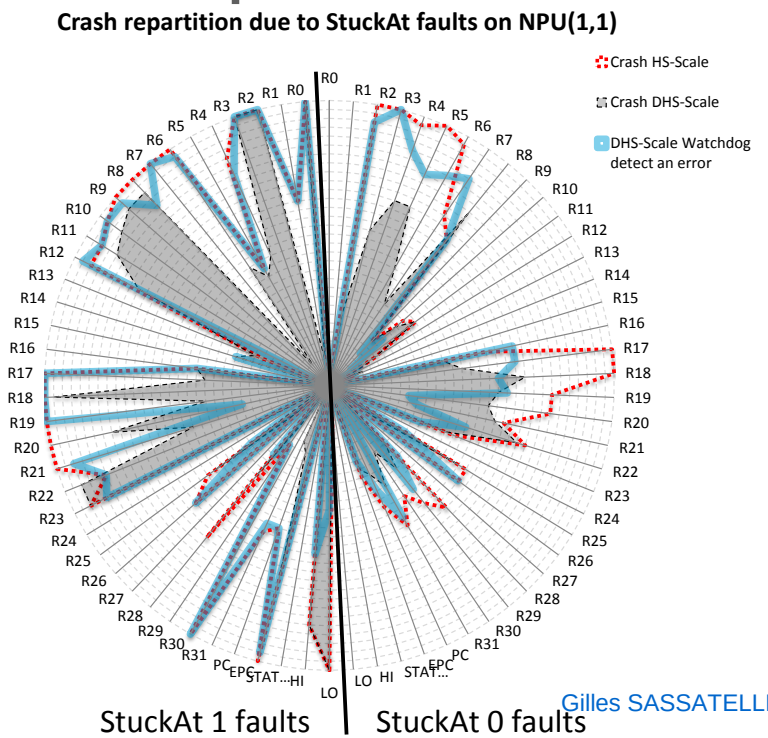


Results normalized against 65nm – 1.2V :

Infineon MuSiC, Tomahawk, SandBlaster SB3011, TI C64x, IMEC ADRES

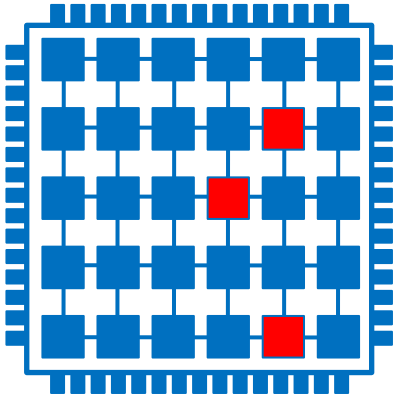
III- PUTTING HOMOGENEITY TO USE: DScale, a reliable exploration platform





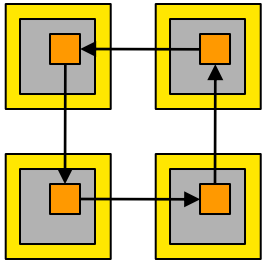
Last one: improving reliability

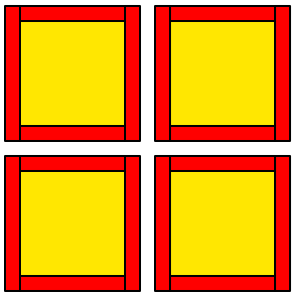
- Is there a means to exploit the natural redundancy of such systems?

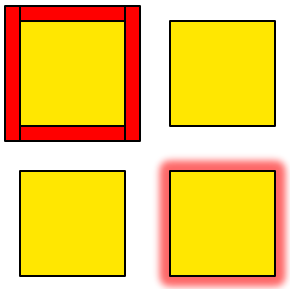


- Targeted Applications
  - Typical non life critical (smartphones, Pay TV, etc.)
- Performance constraints
  - Low area / power overhead
  - Transient loss of availability / functionality / data acceptable
- Target crash faults

Implementing recovery







Fault Detection

System-wide RESET

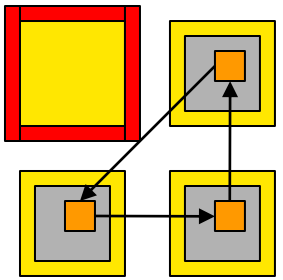
Error diagnostic upon reboot

Local defect isolation

or

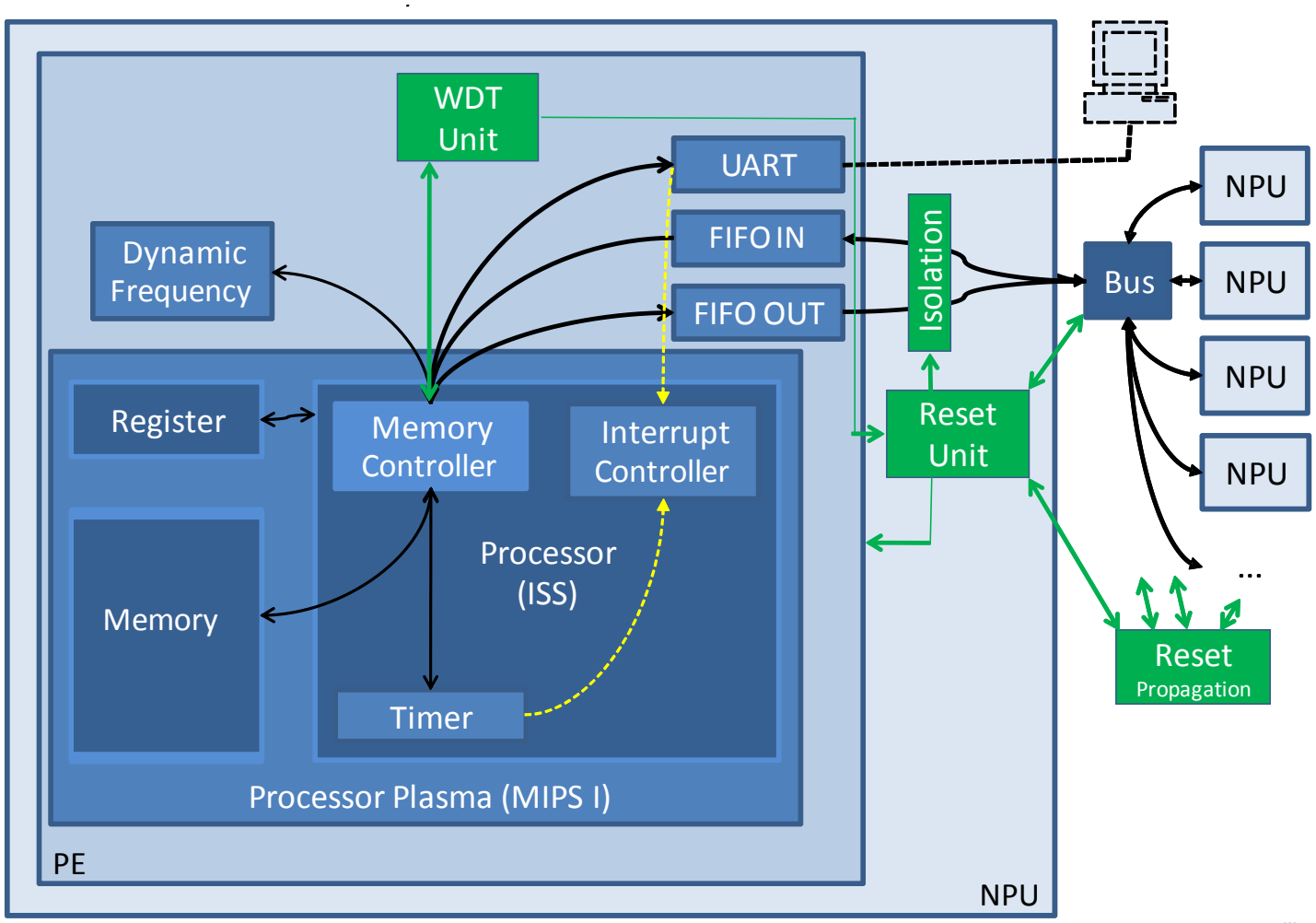
Local process compensation

Application remapping



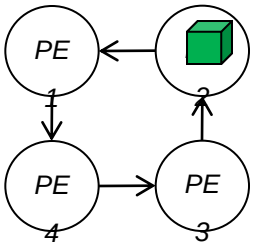
Implementing recovery

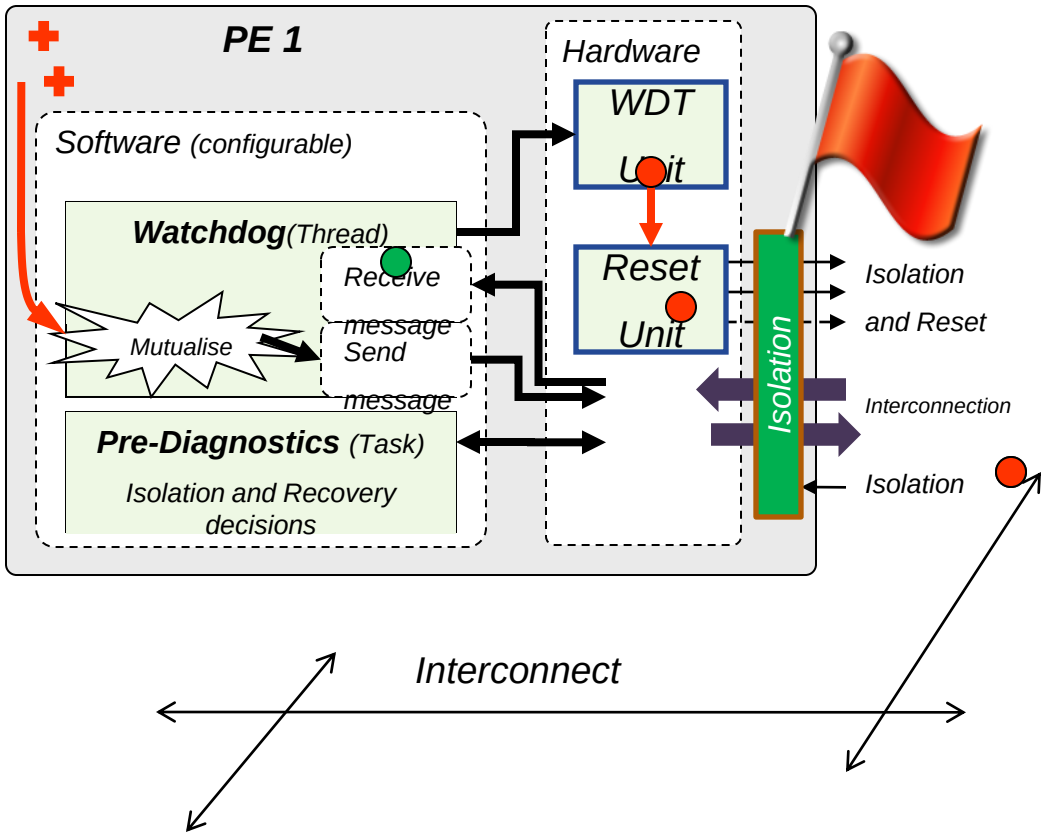
- Additional hardware

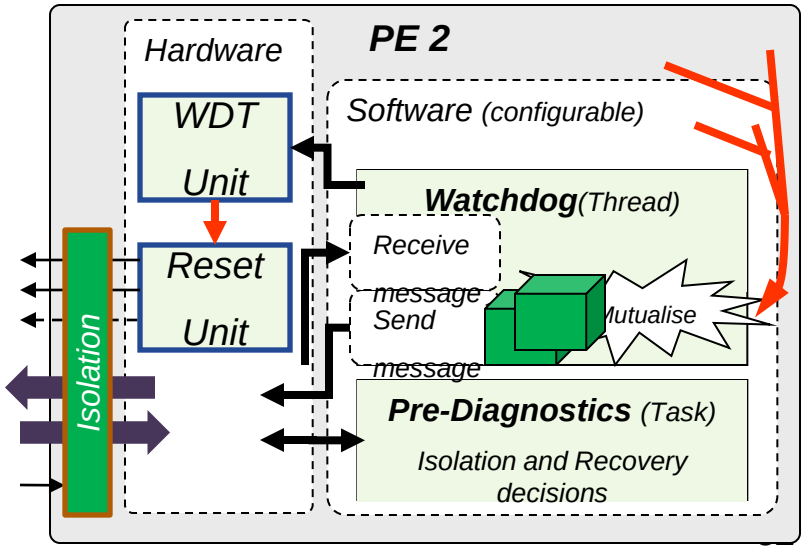


Implementing recovery

- Functionning

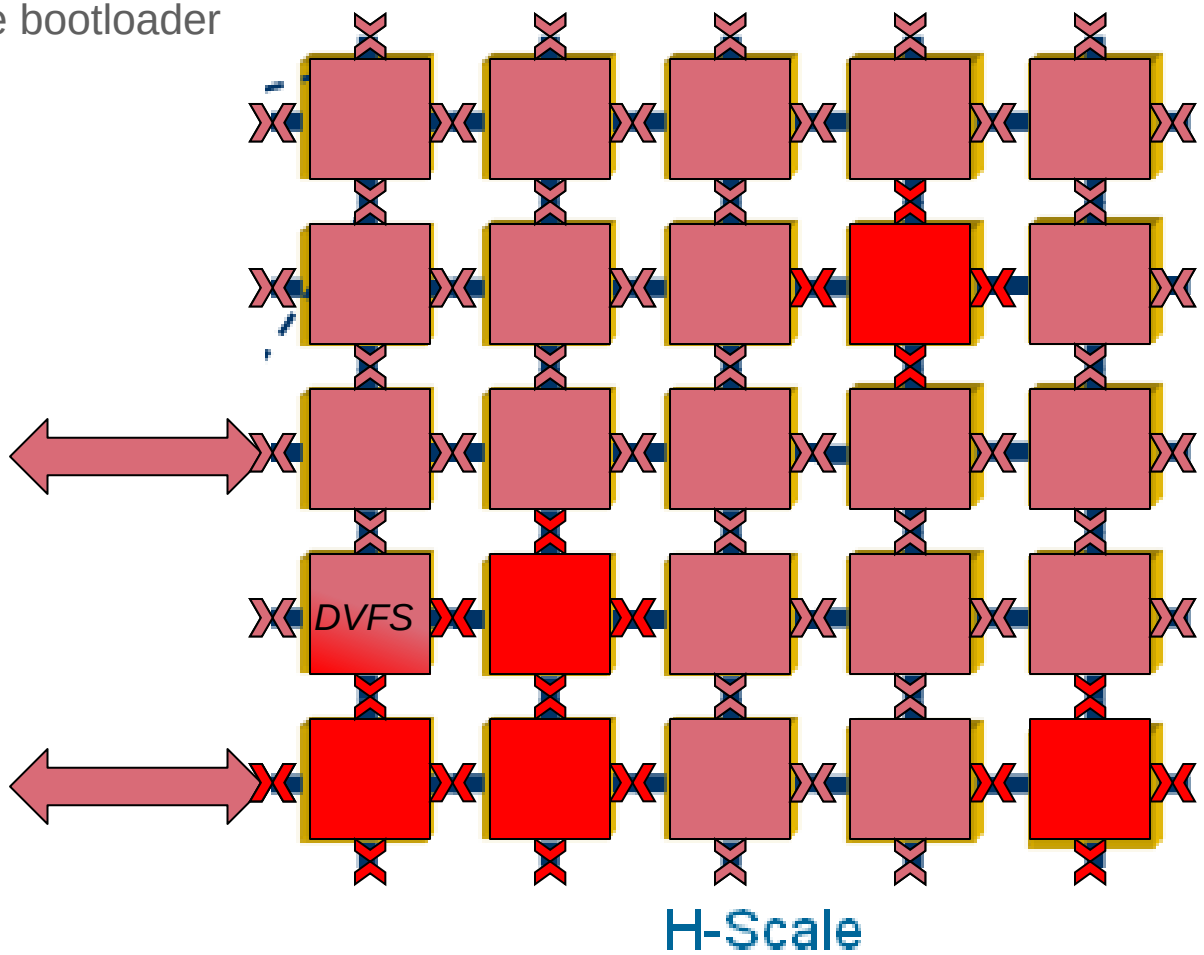






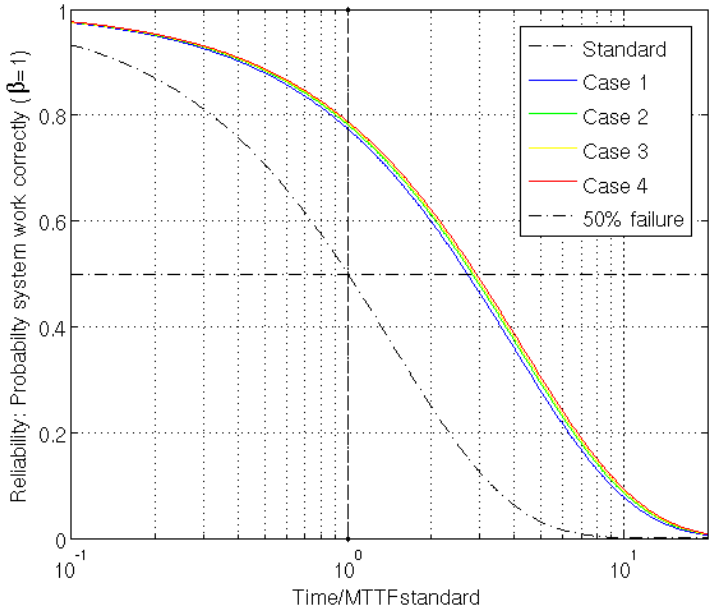
Implementing recovery

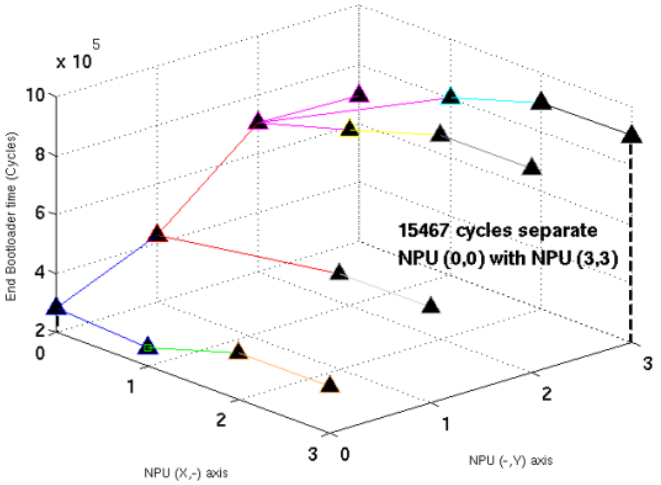
- Depending on error type and frequency, the distributed boot loader can:
  - Do fine tuning (DVFS)
  - Maintain the defective element isolated
  - Performed by the bootloader



How reliable?

- Accepting a performance overhead
  - Less PEs, others take over computation
  - MTTF improvements limited by #interface PEs (case 1-4) and interconnect





Conclusion

- Shifting part of the application handling at run-time (vs. design-time) is a promising alternative: Adaptation
- Combined with homogeneity, a means to fill the performance gap, to an extent. Further, allows for facilitated design, reliability strategies
- Finally control has to be distributed (hierarchical) so that performance scales