

HIPEAC2021 TUTORIAL: EMULATING THE POWER CONTROLLER OF HPC POWER MANAGEMENT SYSTEMS

Andrea Bartolini, Giovanni Bambini

a.bartolini@unibo.it, giovanni.bambini2@unibo.it

OUTLINE

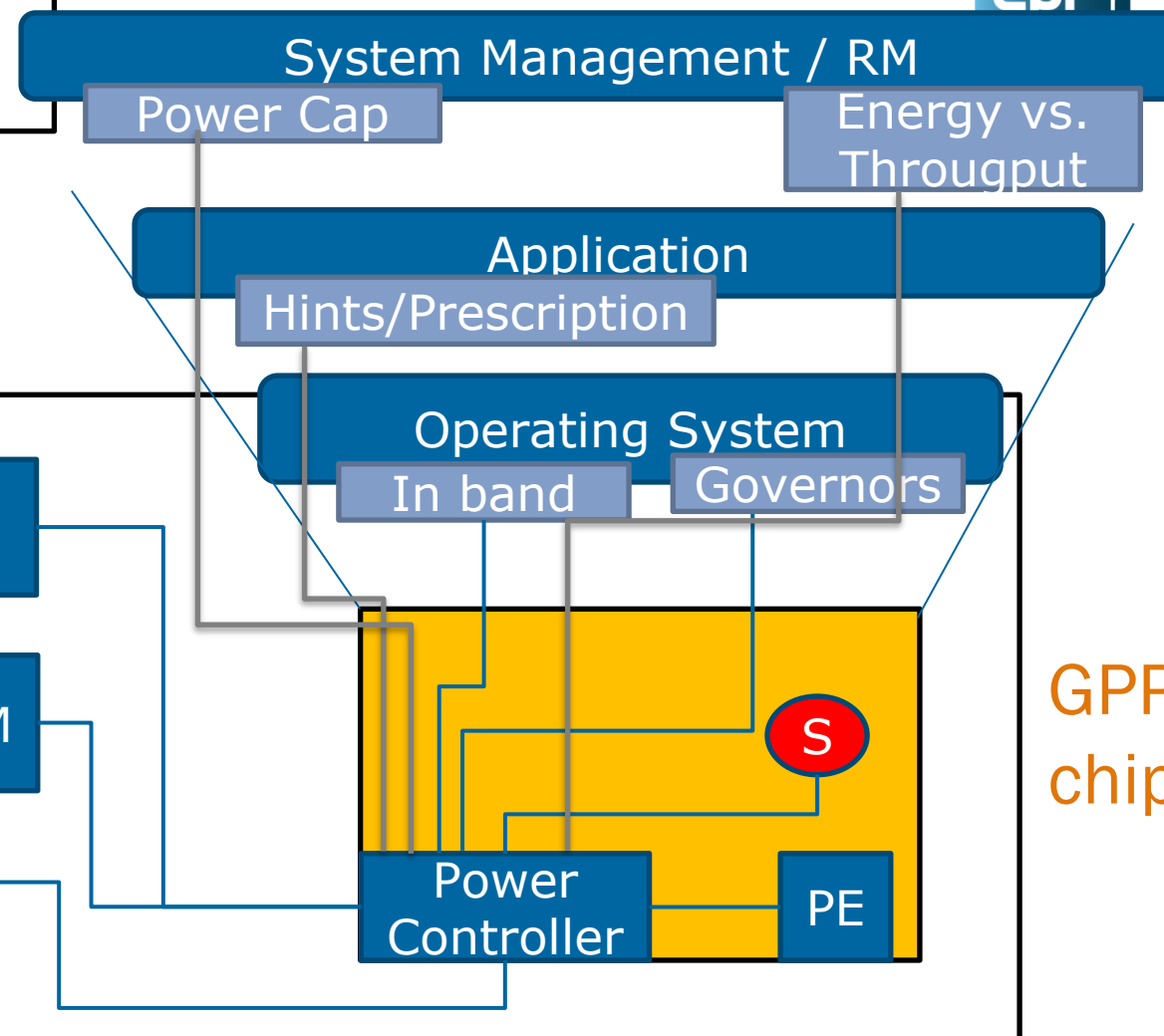


- The power management in HPC
- The power controller subsystem
- A PULP-based open-source power controller
- Software architecture
- Hardware-in-the-loop design methodology

- Out-of-band – zero overhead telemetry
- Node Pcap – Max perf @ $P_{node} < P_{max}$
- RAS – error and conditions reporting

Throughput $\Rightarrow F > F_{max}$ @ $T, P < P_{max}$

- cpufreq/ cpuidle
- Based on O.S. metrics
- Slow & often unused



RAS
Node Power Cap
Out of band

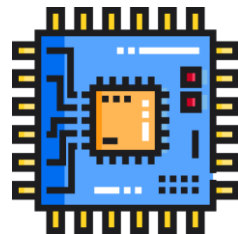
GPP
chip

Power controller subsystem

PROCESSORS in HPC ENVIRONMENT

- Increasing amount of heterogeneous cores
- Increasing design complexity (e.g. multi-chiplet)
- Increasing energy efficiency and performance demand

So Processors integrate
HW dedicated Controller



PCS
(Power Controller
Subsystem)

Aimed to:

- prevent Thermal Hazard,
- ensure TDP Power Budget
- increase Energy Efficiency and overall performance

Has to:

- interface with several on/off-chip sensors, power management interfaces, and actuators;
- perform complex computational tasks;
- support a large number of processing elements;
- interface with the OS and the board management controller (BMC)

Requires:

- Advanced Controller and Advanced Policies
 - Computationally Intensive tasks
 - More sophisticated software (RTOS)
 - Decrease the discrete-time output update interval
- and to achieve that:
- more powerful Microcontrollers

POWER MANAGEMENT SOA & REQUIREMENTS

	Intel	IBM	ARM	AMD	Cray	Fujitsu
Monitor (Domain,Granularity)	S, M, A, T 1ms	N, S, M, A, T, U 500us , 10ms aggregation 16ms for T & U, 100ms aggregation	S, M, T 1-10KHz with SCP	N, S, M, A, T 1 sec (C), 1ms (G)	N, S, M, A, N OOB (100ms)	N, S, C, M 1ms (N), ~ns - model based (C)
Control (Domain,Granularity)	S, M RAPL 1ms (in-band), DVFS 500us	N, S, M, A 10-100ms	S, M 1-10KHz (100ms to 1s)	N, S, M, A ~secs	N, S, M, A DVFS, RAPL, min-max range, 10- 30s at job launch	S, C, M, DVFS , Decode Width, HBM2 B/W
Interfaces, Tools, etc	RAPL MSRS, msr-safe, libmsr, PAPI, likwid <i>Source PowerStack 19</i>	OpenBMC , amester, Memory Map	ACPI, SCP (sys ctrl proc), IPA (intelligent allocator), PAPI	Likwid, PAPI, Memory Map	CapMC, PAPI, Cray BMC interfaces	Power API, PAPI

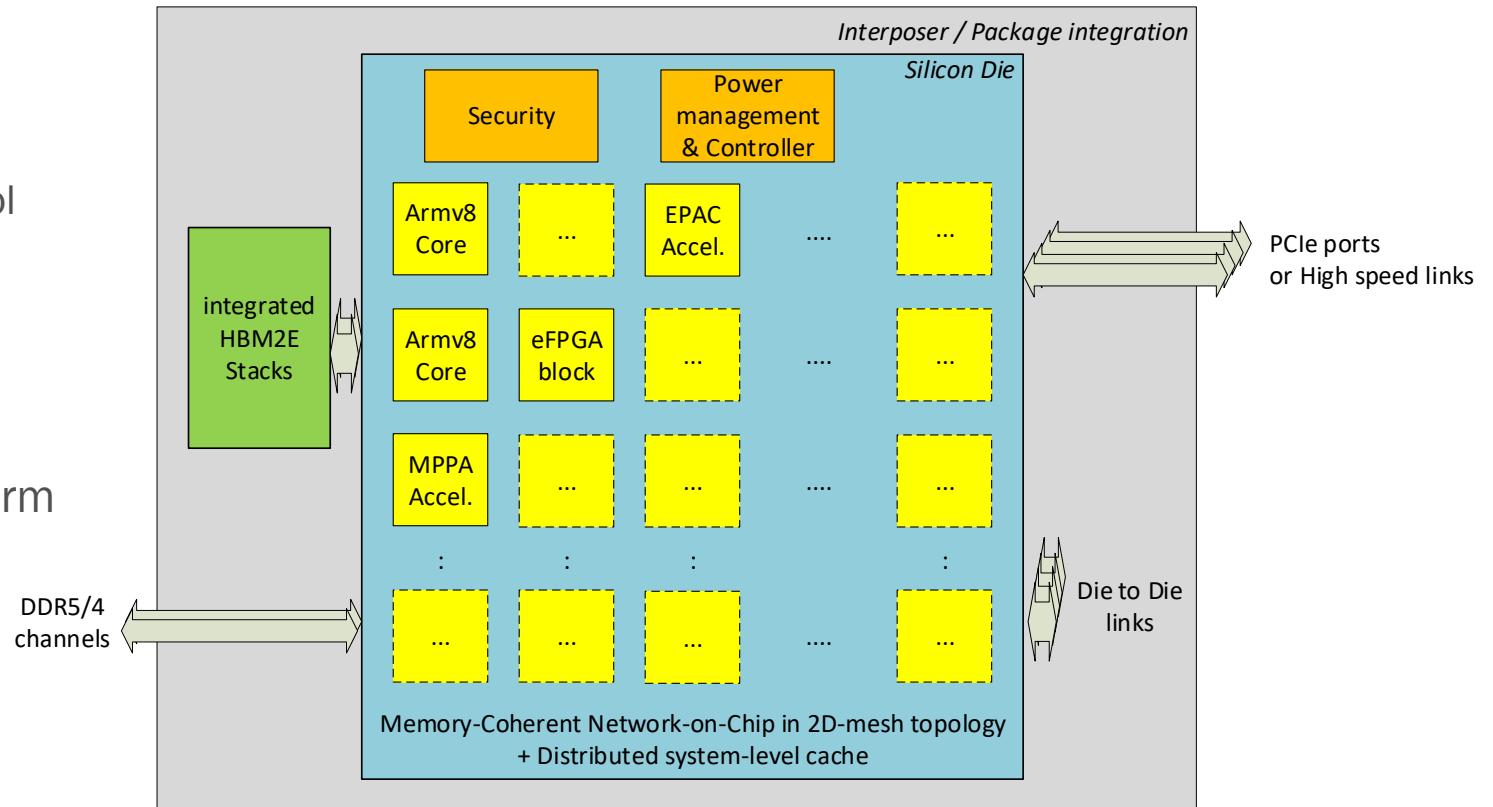
Socket (S), Core (C), Memory (M), Accelerator (G), Node (N), Utilization (U), Temperature (T)

EPI power management design targets:

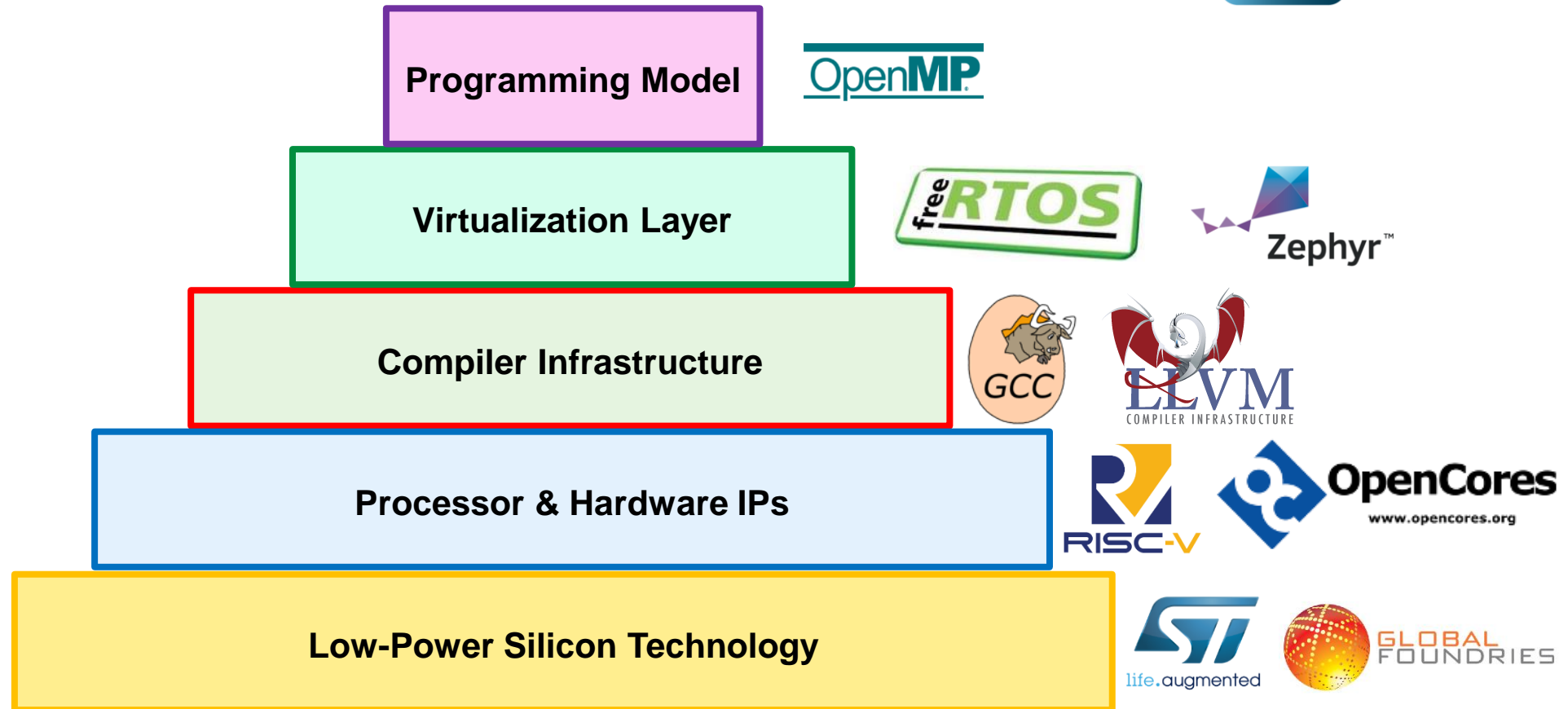
- Support for fine grain power monitoring, and control
- A higher performance power controller capable of supporting advanced power control algorithms.

GENERAL ARCHITECTURE

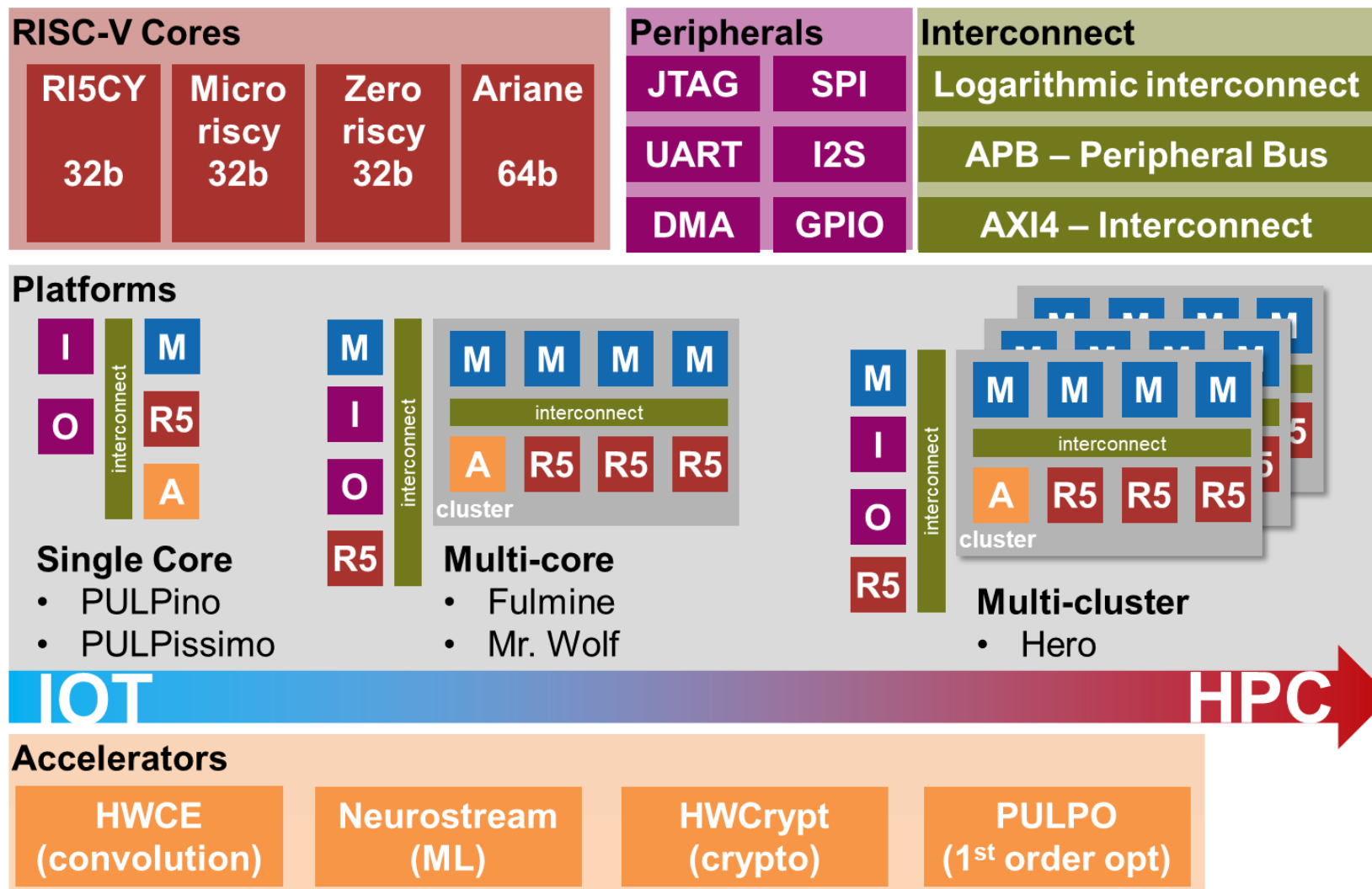
- Top level infrastructures
 - Power management & controller
 - Dedicated power management and control network
 - Security
- EPI Power Management Subsystem
- RISC-V ISA, Derived from the PULP platform
- Parallel processor w. DSP extensions
- Open-Source Design



PULP OPEN-SOURCE HARDWARE AND SOFTWARE STACK



<https://github.com/pulp-platform>



Silicon and Open Hardware fuel PULP success

-



Contributions from many groups

- Similar to Apache/BSD, adapted specifically for Hardware
- Allows you to:
 - Use
 - Modify
 - Make products and sell themwithout restrictions.
- Note the difference to **GPL**
 - Systems that include PULP do not have to be open source (Copyright not Copyleft)
 - They can be released commercially
 - LGPL may not work as you think for HW



WHY A PULP-BASED POWER CONTROLLER?

2) A more powerful SoC than SoA competitors

Architecture	Harvard	Harvard
ISA Support	Armv7-M	RISC-V (RV32IMFC)
Pipeline	6-stage superscalar + branch prediction	4-stage No superscalar. No branch prediction
DSP Extensions	Single cycle 16/32-bit MAC. Single -cycle dual 16-bit MAC. 8/16-bit SIMD arithmetic	Single cycle 16/32-bit MAC. Single cycle dual 16-bit MAC. 8/16-bit SIMD arithmetic
Floating-Point Unit	Optional single and double precision floating point unit	Optional 8, 16, 32 or 64 bit FPU. Optional half, single and double precision FPU
	IEEE 754 compliant	IEEE 754 compliant
Interconnect	64-bit AMBA4 AXI, AHB peripheral port	64-bit and 32-bit AXI
Interrupts	Non-maskable Interrupt (NMI) + 1 to 240 physical interrupts	1 to #M (#M number at will)
Dynamic Power	33 μ W/MHz	28.68 μ W/MHz
Floorplan Area	0.067mm ² @40nm (hypothesis)	0.077mm ² @65nm

TABLE II
RISC-V vs ARM CORTEX M7 (CONT.)

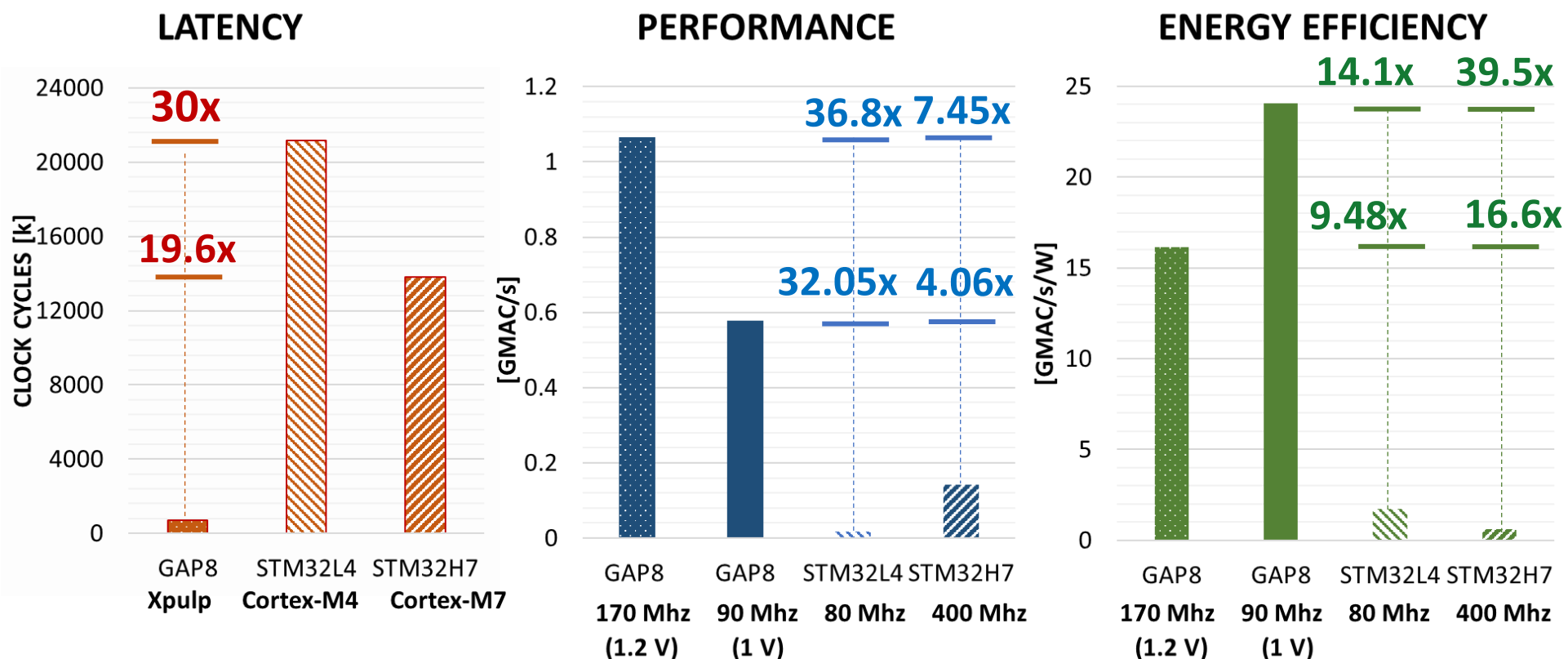
Processor	RI5CY	ARM Cortex M4	ARM Cortex M7
Max frequency (65nm)	560 MHz	n.a.	n.a.
Area (kgates)	40 @ 180 MHz, 51 @ 560 MHz	65	156
Power [uW/MHz] (65nm)	6.7@180 MHz, 24.9@560 MHz	23.7	63.8
CoreMark/MHz	3.19	3.4	5

TABLE I
RISCV vs ARM CORTEX M4,M7

[ICECS19] A. Bartolini et al., "A PULP-based Parallel Power Controller for Future Exascale Systems,"

WHY A PULP-BASED POWER CONTROLLER?

3) Already at the heart of a wide set of smart applications edge-AI and DSP



[ICECS19] A. Garofalo et al. PULP-NN: a Computing Library for Quantized Neural Network Inference at the Edge on RISC-V Based Parallel Ultra Low Power Clusters

PULP-based power controller



OPEN-SOURCE HARDWARE

The hardware implementation is open-source and based on the PULP (Parallel Ultra Low Power) design, with RISC-V cores and a multicore accelerator.



OPEN-SOURCE SOFTWARE

The firmware and the control algorithm are fully open-source.



SIMPLER FIRMWARE DESIGN

The firmware is deployed over an RTOS, and thus is more flexible, scalable and portable, upgradable and modular, and capable to execute multiple functions “simultaneously” by exploiting pre-emptive scheduling.

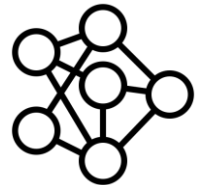
ADVANCED CONTROL ALGORITHM

A new cascade control for advanced power and thermal capping, that gives priority to more demanding cores and can enforce binding constraints.



ADDITIONAL FEATURES

The advanced hardware and the flexibility introduced by RTOS allow the implementation of additional and advanced features, as Model Adaptation and Machine Learning algorithms.



CONTROL STRUCTURE

POWER DISPATCHING ALGORITHM

The power dispatching is an independent structure block that computes the operating point to enforce the given power budget. It privileges more demanding cores. To improve the performance can receive formula's parameters from model adaptation or machine learning algorithms.

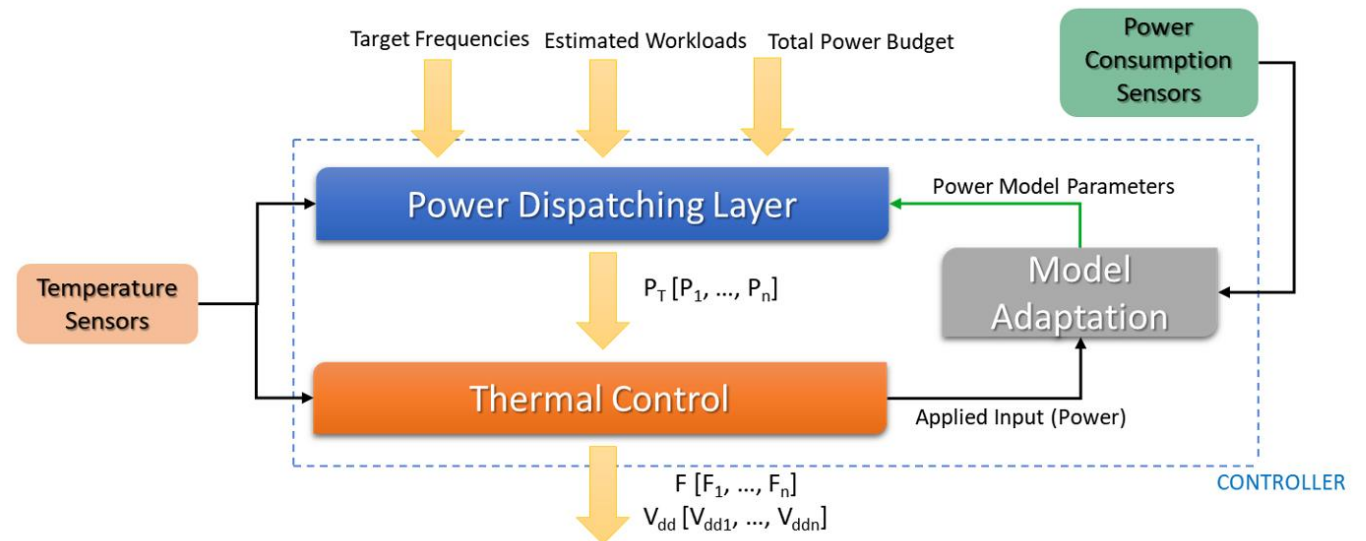
MODEL ADAPTATION

A Model Adaptation technique can be deployed to improve the performance of the Power Dispatching Algorithm, while a machine learning block can guess the type of workload to be executed to find a better operating point.

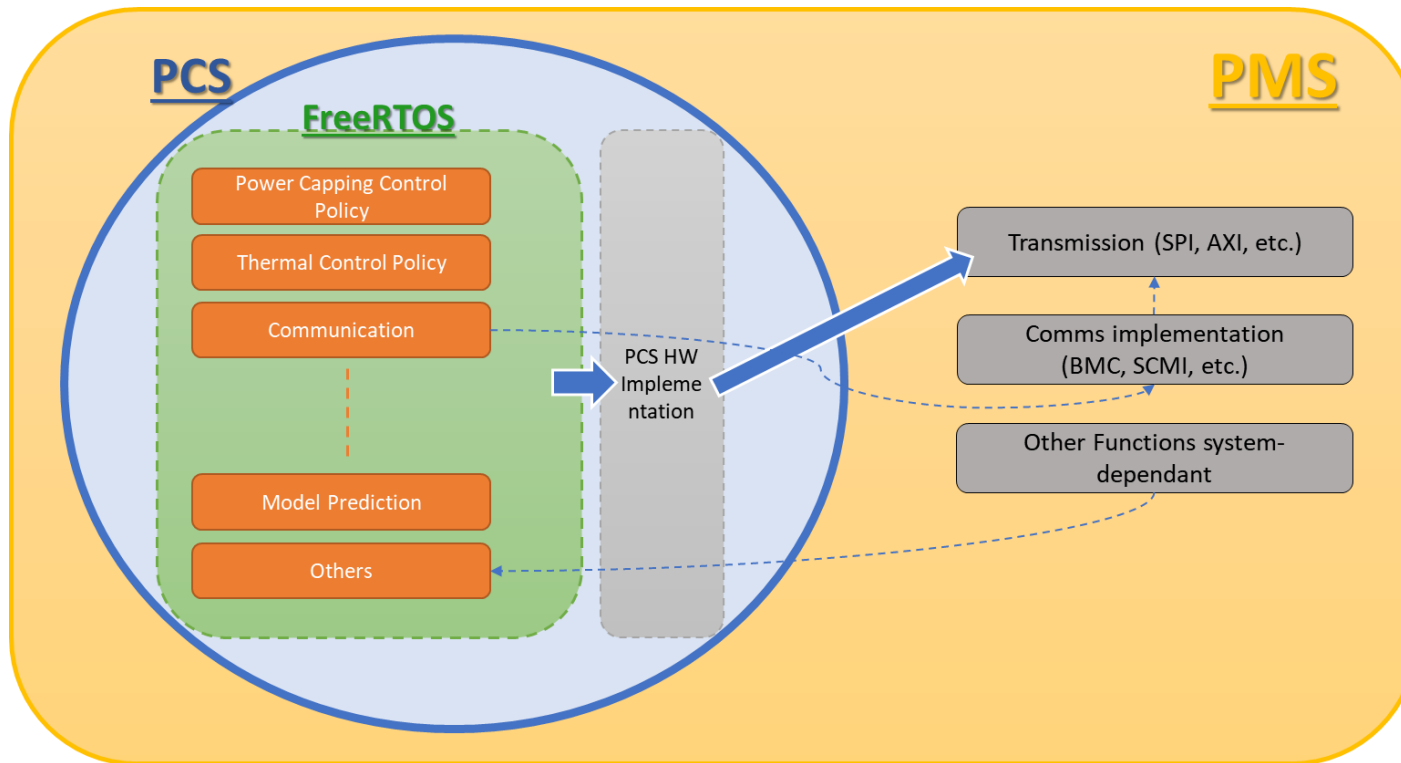
	Periodicity
Power D.A.	1ms
Thermal Capping	500us
Model Adaptation	1ms

THERMAL CAPPING

The thermal capping is entrusted to independent PIs (one for each core). Their task is to reduce the operating point received by the Power Dispatching Algorithm if the temperature is approaching the configured limit.



Firmware Structure



PCS:

- Open-Source
- System-Independent
- Further divided into two parts: one part is (almost) PCS-platform independent; the other depends on the PCS Hardware, and it is the one that has to be changed in case of a hw upgrade.
- Every function and feature is inside FreeRTOS and divided into Tasks
- The scheduling is Rate Monotonic (RMS) with the inclusion of aperiodic tasks

PMS:

- Is the PCS implementation of EPI
- Has all the transmission implementation
- Has all the other comms implementation
- Has other system-dependant functions that are called inside the PCS at a defined interval

PMS I/Os

Sensors and Cores

N Temperatures

N Core Workloads
(perf. counters)

Board Power Budget

K Power Domains

Cores

K Voltage Domains

K Frequency
Domains

N Core Domains

Other Core
Functionalities

OS, BMC

N Target Frequencies

Board Power Budget

Binding Matrix

Configuration Tables

SMS

Initial Configuration

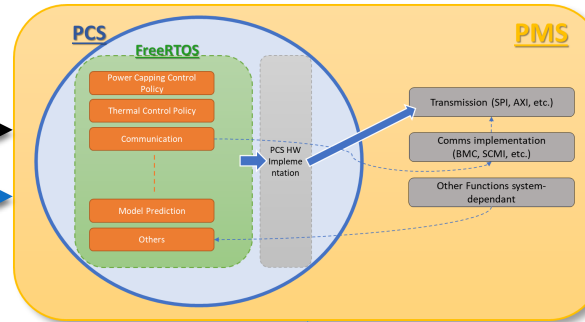
Boundaries and
Constraints

OS, BMC

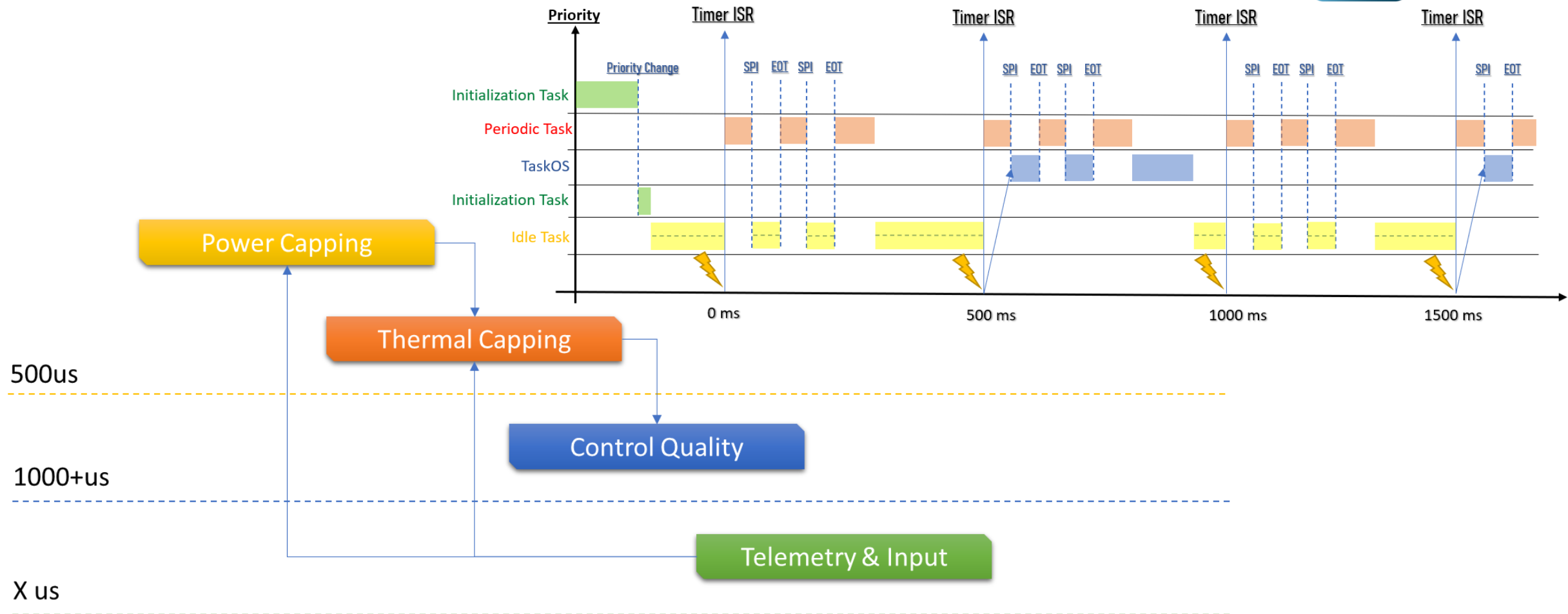
Telemetry

Errors and Infos

Performance Parameters



Control Tasks:



METHODS

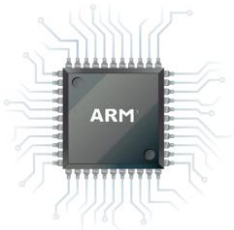
HARDWARE-IN-THE-LOOP EMULATION

We tested the design of the firmware in a hardware-in-the-loop test with a Xilinx zcu102 board consisting of 4 ARM A-53 cores and 2 ARM R5 microcontrollers, with an FPGA with the IP design of ControlPulp

TO MEASURE:

- RTOS overheads
- Performance of Power Dispatching Algorithm
- Performance of the overall Control

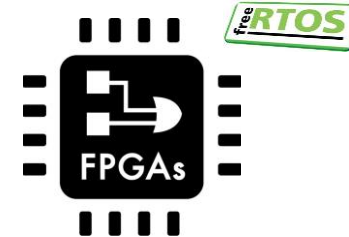
EPI Chip and Board Models



 XILINX



PMS Chip with Firmware



METHODS

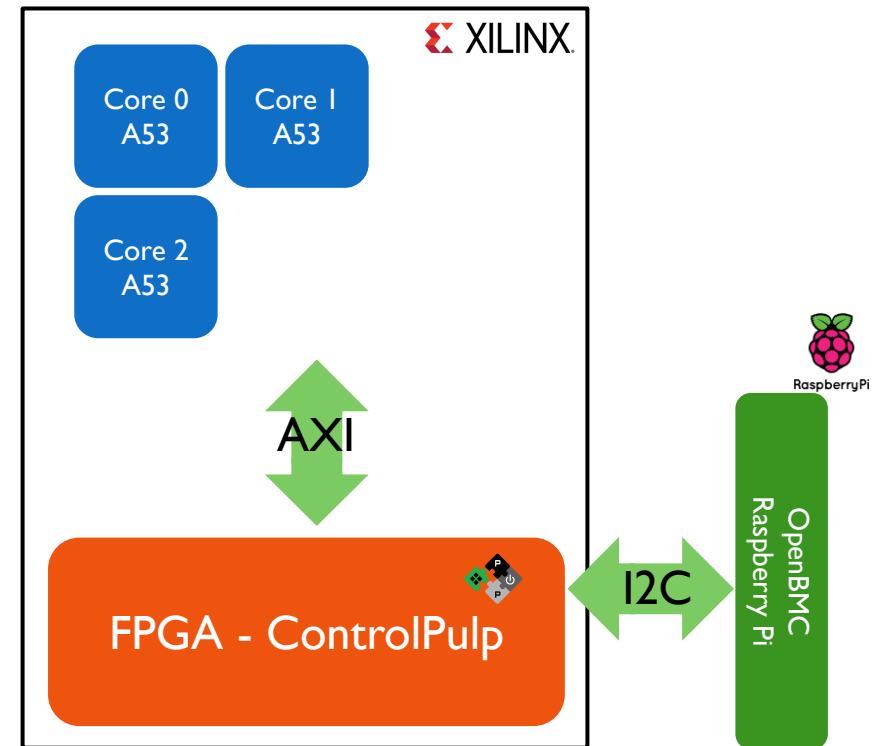
How we Modeled EPI chip

Core 0 (of the A53 cores) runs a Discrete-time State-Space **thermal simulation** for the N cores, preceded by an algebraic **Power simulation** computed for each core based of the power-variables input of the core quadrant, the throttled frequency and the type of workload the core is executing with a precision of 1 cycle. The simulation runs 1 step-per-us

Core 1 (of the A53 cores) runs a simple **OS model**, sending commands to the Pulp Controller such as Target Frequency, several power budgets, binding matrix, etc. and reading outputs such as telemetry, statistics, and data analysis. The definition of commands is 1ms

Core 2 (of the A53 cores) runs a basic **Governor model**, gathering all the data and sending them through Ethernet, while also reading from ethernet user commands to be passed to overwrite OS commands.

It is also in development the addition of openBMC implemented on a Raspberry Pi connected through I2C, And to connect all these different agents, the implementation of SCMI ARM protocol inside the Xilinx Board.



RESULTS

OVERHEADS:

Overheads wrt Total cycles in the Period (1ms)

(1ms interval)	Nominal	Worst Case
RTOS	5.95%	9.44%
RTOS + Locks	11.24%	15.86%

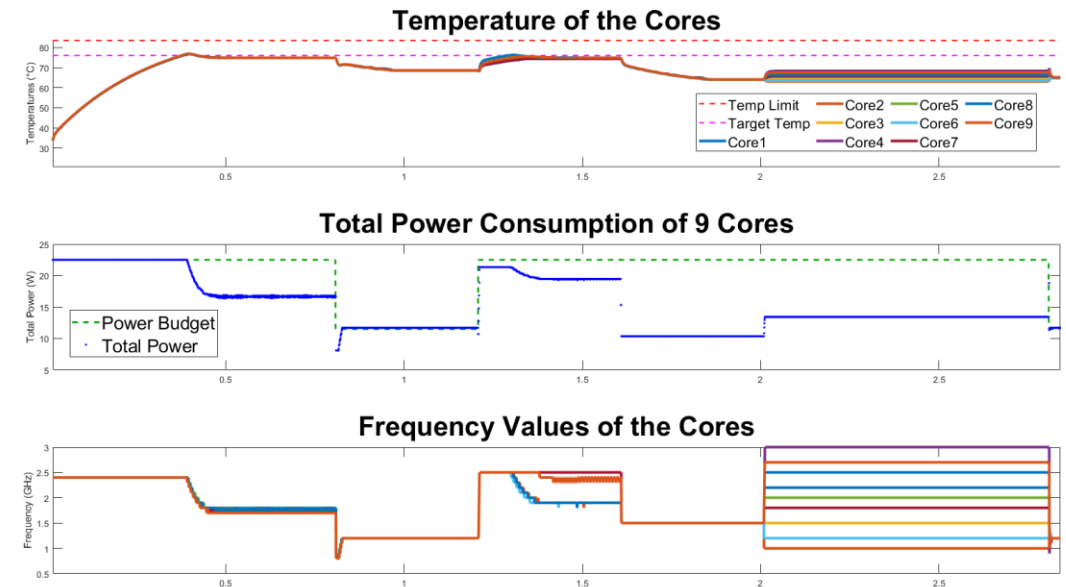
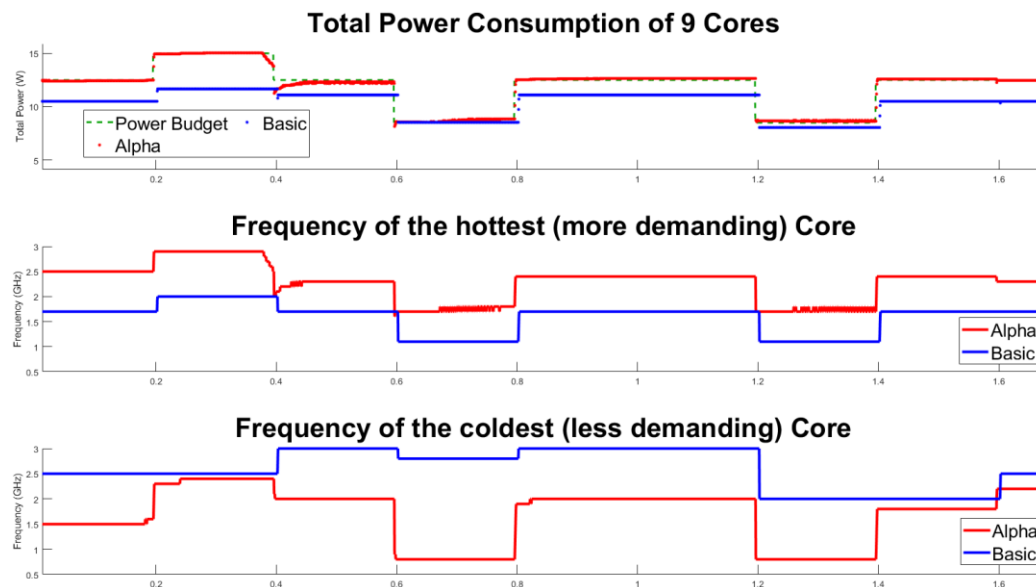
POWER DISPATCHING ALGORITHM

Compared to a naïve equal power dispatcher:

- Improved power budget utilization of 13.7%
- The demanding cores run at 700MHz average higher frequencies, up to 1.8GHz boost.

CONTROL PERFORMANCE

- Power budget is respected
- Thermal limit is respected with a maximum overshoot of 41ms and a 1°C higher over the lower thermal limit



Thanks for Your Attention

The Team: Giovanni Bambini, Robert Balas, Antonio Mastrandrea, Christian Conficoni, Andrea Tilli, Luca Benini, Simone Benatti, Andrea Bartolini

References:

Pulp: <https://pulp-platform.org/projectinfo.html>

Dei Unibo: <https://dei.unibo.it/it/index.html>

EPI project: <https://www.european-processor-initiative.eu/>

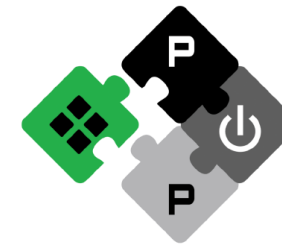
Readings:

“An Open-Source Scalable Thermal and Power Controller for HPC Processors” G. Bambini, R. Balas, C. Conficoni, A. Tilli, L. Benini, S. Benatti, A. Bartolini

“A pulp-based parallel power controller for future exascale systems” A. Bartolini, D. Rossi, A. Mastrandrea, C. Conficoni, S. Benatti, A. Tilli, and L. Benini

“Benefits in relaxing the power capping constraint” D. Cesarini, A. Bartolini, L. Benini

“Thermal and energy management of high-performance multicores: Distributed and selfcalibrating model-predictive controller” A. Bartolini, M. Cacciari, A. Tilli, and L. Benini





a.bartolini@unibo.it

Assistant Professor @ University of Bologna (UNIBO)

Giovanni Bambini

giovanni.bambini2@unibo.it

Research Assistant @ University of Bologna (UNIBO)



 www.european-processor-initiative.eu



 European Processor Initiative