ALMA MATER STUDIORUM BOLOGNA

Automation Engineering M

DESIGN AND PROTOTYPING OF A FREERTOS-BASED POWER CONTROL FIRMWARE FOR HPC PROCESSORS IN GAP8

RELATORE

Chiar.mo Prof. Luca Benini

CANDIDATO

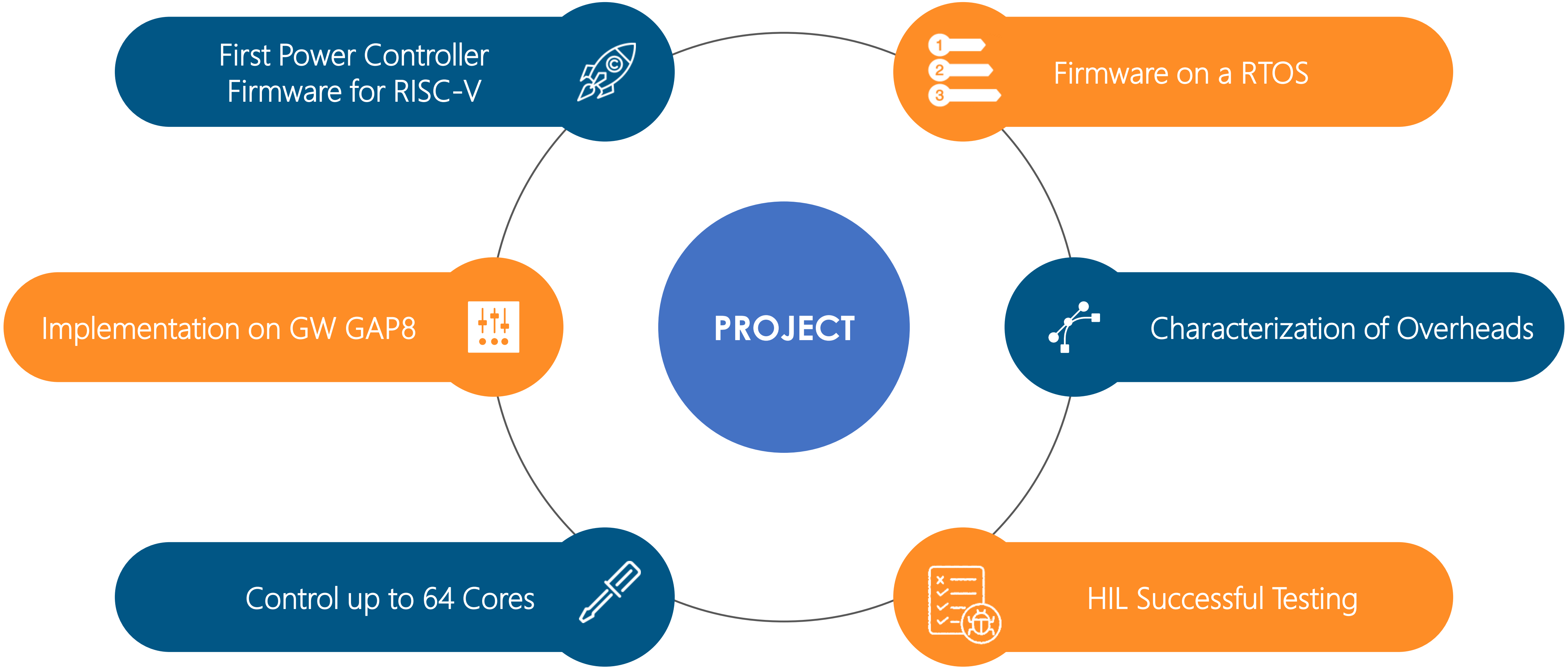
Giovanni Bambini

CORRELATORI

Prof. Andrea Bartolini
Prof. Christian Conconi
Dr. Ing. Simone Benatti

Contribution

Project Analysis



PROJECT

First Power Controller Firmware for RISC-V

Firmware on a RTOS

Implementation on GW GAP8

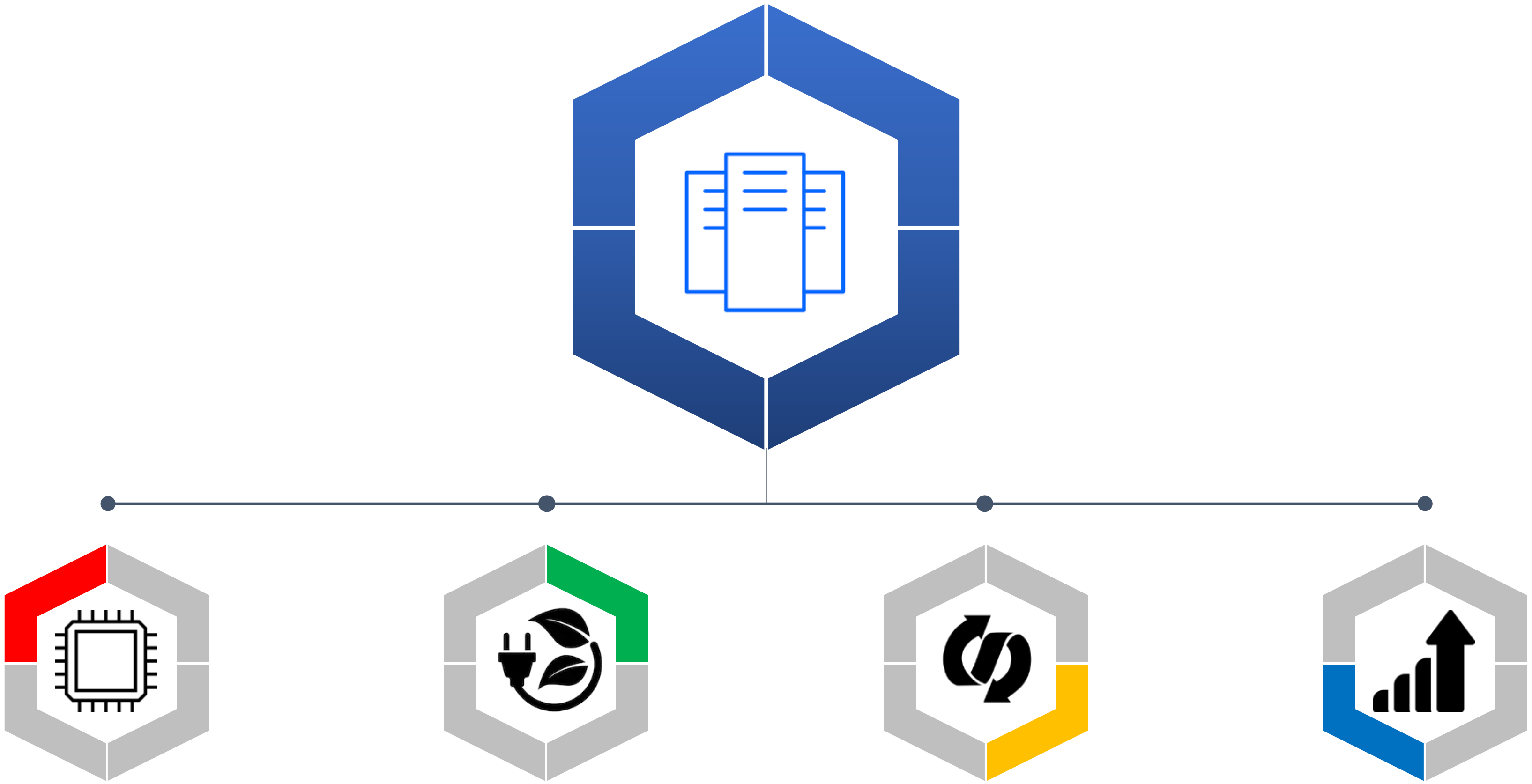
Characterization of Overheads

Control up to 64 Cores

HIL Successful Testing

HPC Problem

Introduction



High Power Density and Computing Performance

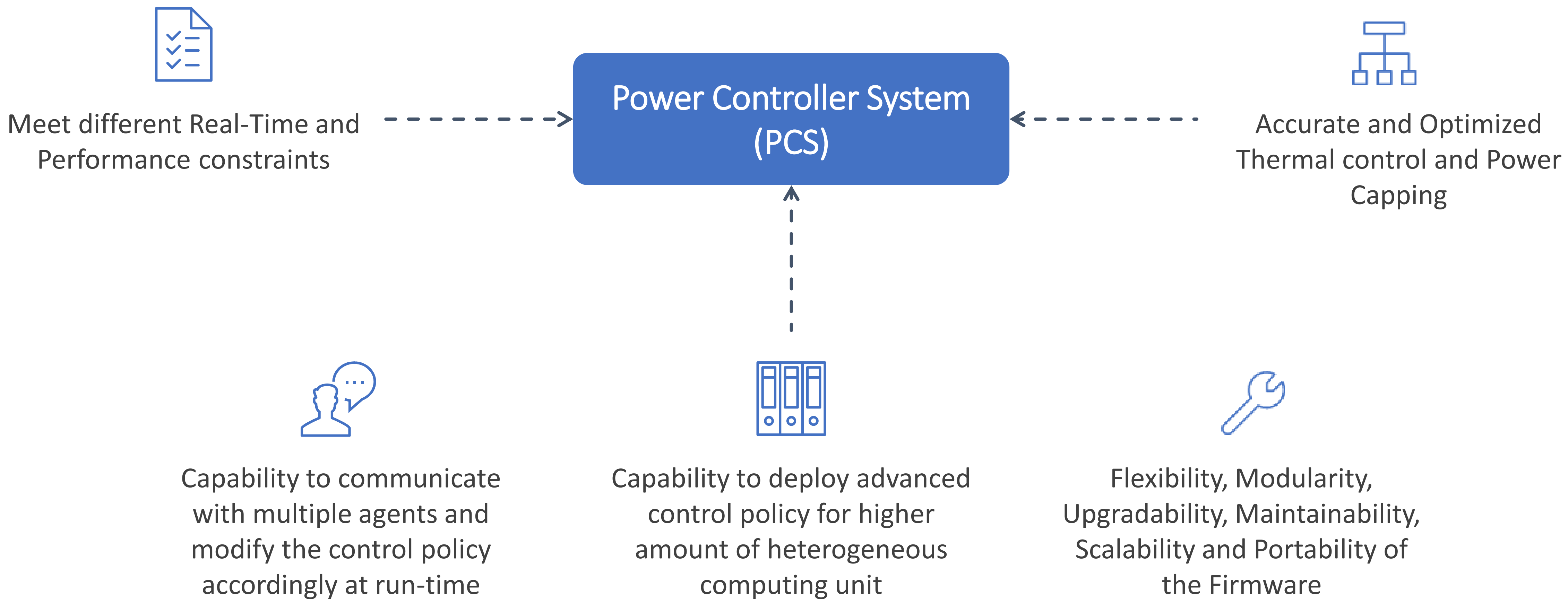
High Energy Efficiency

Dynamic and Flexible Thermal and Power Control

Increasing amount of heterogeneous Computation Units

Power Controller System Requirements

Introduction



Power Controller System (PCS)

Meet different Real-Time and Performance constraints

Accurate and Optimized Thermal control and Power Capping

Capability to communicate with multiple agents and modify the control policy accordingly at run-time

Capability to deploy advanced control policy for higher amount of heterogeneous computing unit

Flexibility, Modularity, Upgradability, Maintainability, Scalability and Portability of the Firmware

Control Solution

The structure

POWER DISPATCHING LAYER

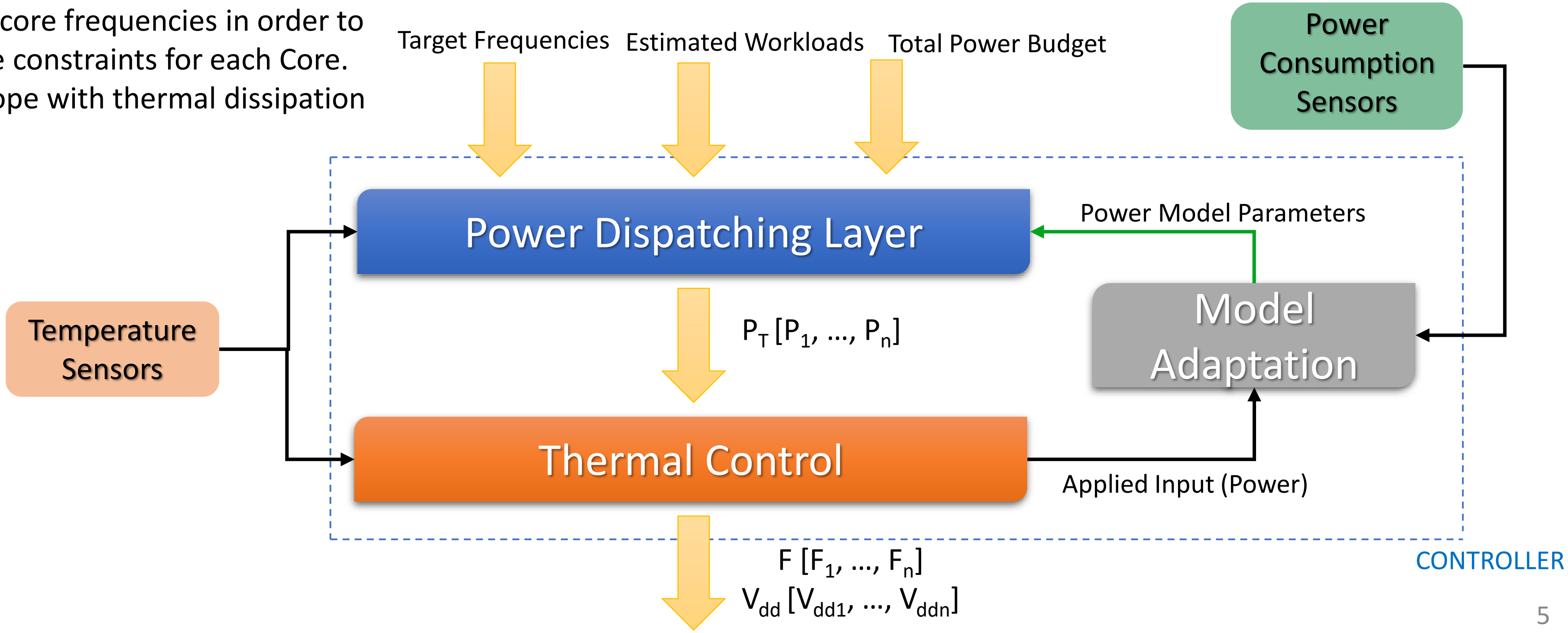
In charge of distributing the requested power among cores, according to power limitations and other constraints.

MODEL ADAPTATION

Online Model identification to improve the power model, improving the control performance. It can run with a slower period interval.

THERMAL CONTROL

PIDs that allocate core frequencies in order to meet temperature constraints for each Core. Period < 1ms to cope with thermal dissipation time constant.



Control Solution

The structure

POWER DISPATCHING LAYER

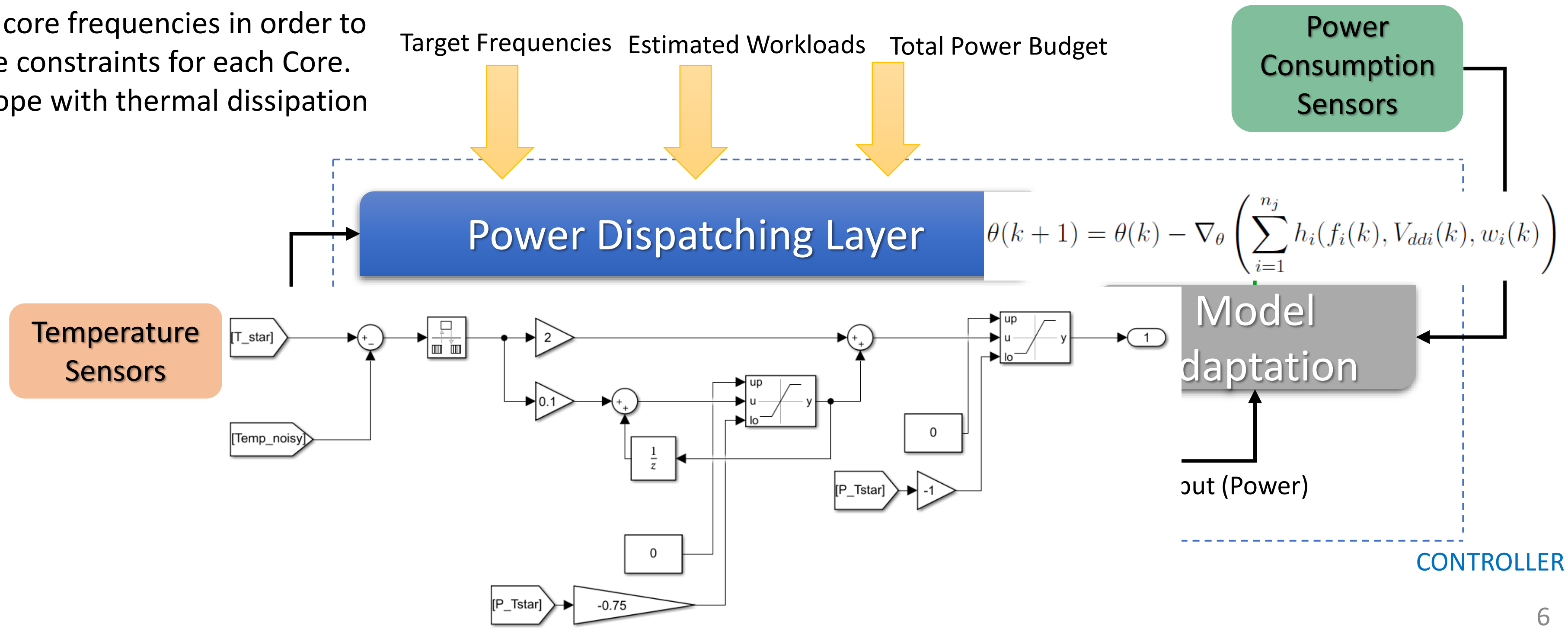
In charge of distributing the requested power among cores, according to power limitations and other constraints.

THERMAL CONTROL

PIDs that allocate core frequencies in order to meet temperature constraints for each Core. Period < 1ms to cope with thermal dissipation time constant.

MODEL ADAPTATION

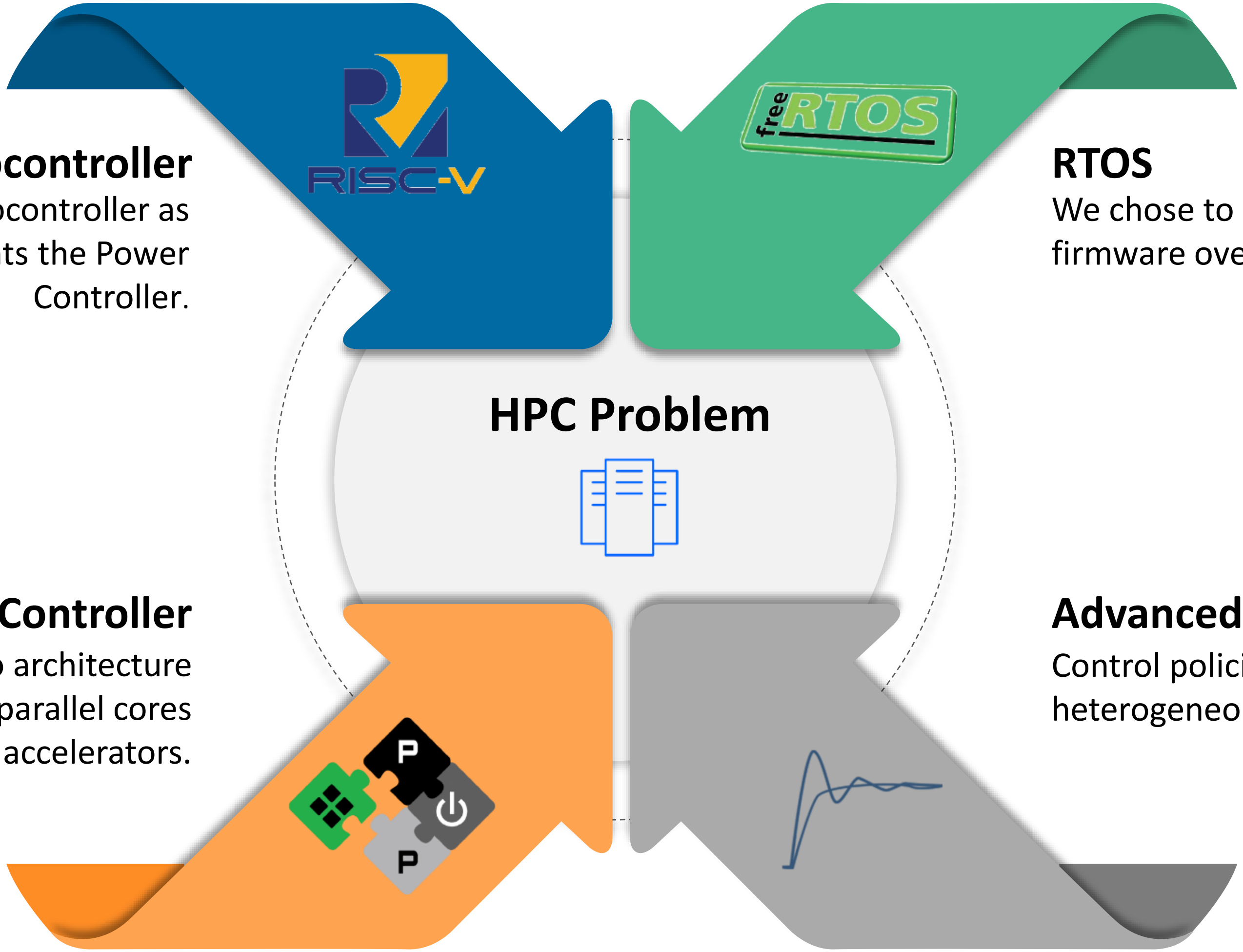
Online Model identification to improve the power model, improving the control performance. It can run with a slower period interval.



$$\theta(k+1) = \theta(k) - \nabla_\theta \left(\sum_{i=1}^{n_j} h_i(f_i(k), V_{ddi}(k), w_i(k) \right)$$

Addressing the Problem

The Choices



HPC Problem

RISC-V Microcontroller

We chose a RISC-V based microcontroller as the hardware that implements the Power Controller.

RTOS

We chose to Develop the Power Controller firmware over a Real-Time.

MultiCore Power Controller

We chose to adopt a PULPissimo architecture design. The uC can include an 8 parallel cores cluster and other accelerators.

Advanced Control Policies

Control policies able to meet multiple heterogeneous requirements.

Design Choices

Firmware Implementation

FREERTOS and TASKS

FreeRTOS is chosen as the RTOS due to its characteristics. We coded 3 tasks to implement the Control Structure and the I/O required.

500us PERIOD

The Periodic Interval for the execution of the control is 500us. The frequency is higher than other solution in the market to achieve better performance.

TIMER ISR

The Periodic Interval is set with TIMER Interrupts. Thus we are decoupled from the FreeRTOS structure increasing flexibility and Reliability.

NOTIFICATION SYSTEM

The FreeRTOS Notification System was used to set the periodicity of the Tasks.

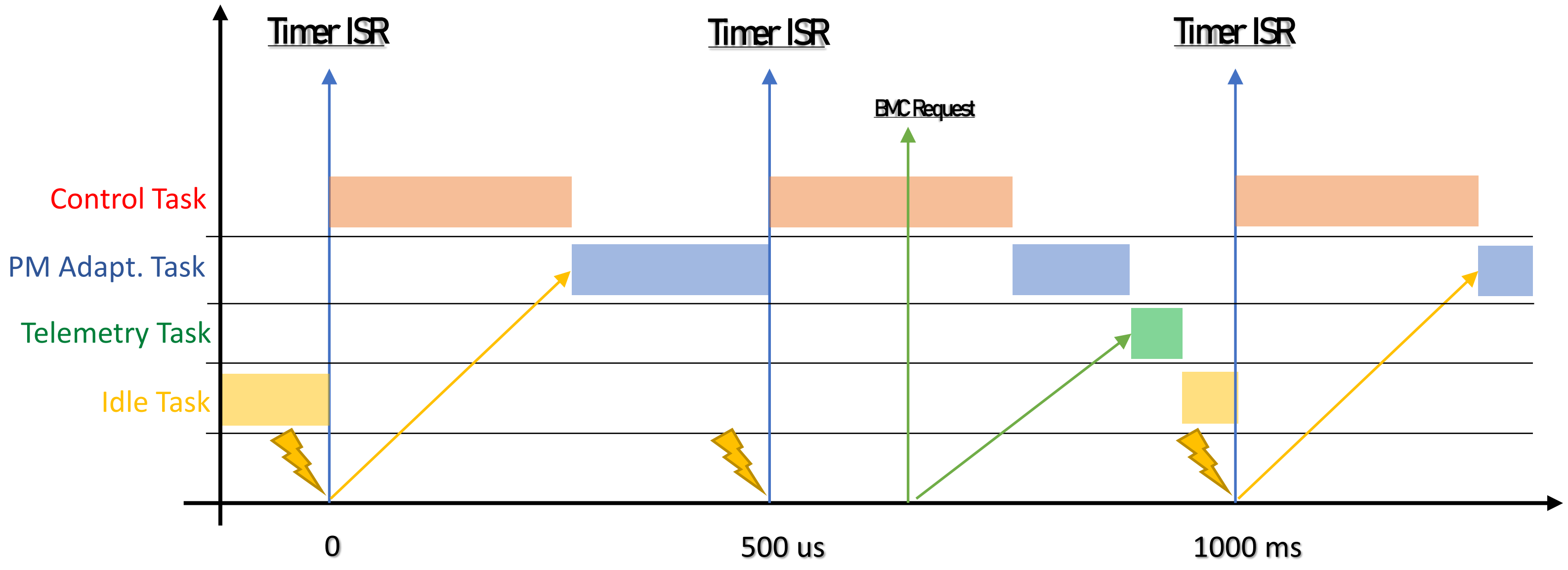
SHARED MEMORY ACCESS

The Tasks exchange data through shared memory variables that are accessed through Mutexes with Priority Inheritance mechanism.

TASKS IMPLEMENTATION

Control Task:

- Thermal Control
- Power Capping
- Telemetry Collecting
- 500us Periodicity
- Highest Priority to meet hard real-time deadline



PM Adaptation Task

- Power Model Adaptation
- Collects inputs from multiple agents
- Sends command to the Periodic Control Task
- Send Telemetry and Error Data
- Periodicity = k · 500us (k = 2)
- Medium Priority

Telemetry Task

- Board Management Control
- Manage BMC-related operations
- It is event-triggered
- Lower Priority

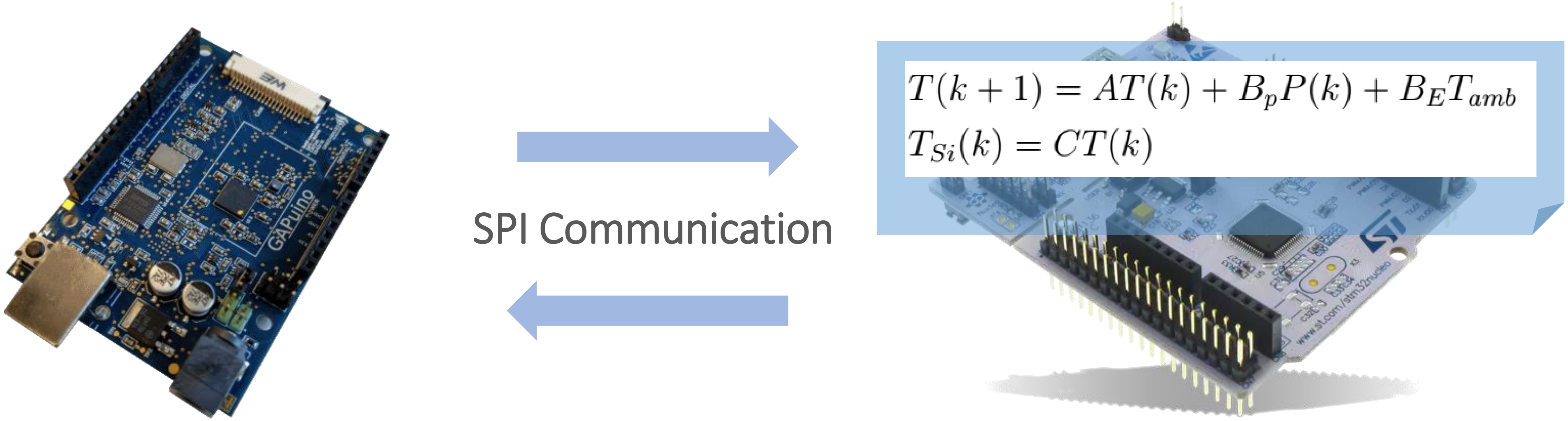
Setting Up the Hardware in the Loop (HiL) Test

Power Control Firmware Implementation:

- SoC: GreenWaves GAP8v1 @ 50MHz
- Simulated Time 100ms = 500us
- Floating Point Operations Emulation (FPU missing)
- SPI master @ 1MHz

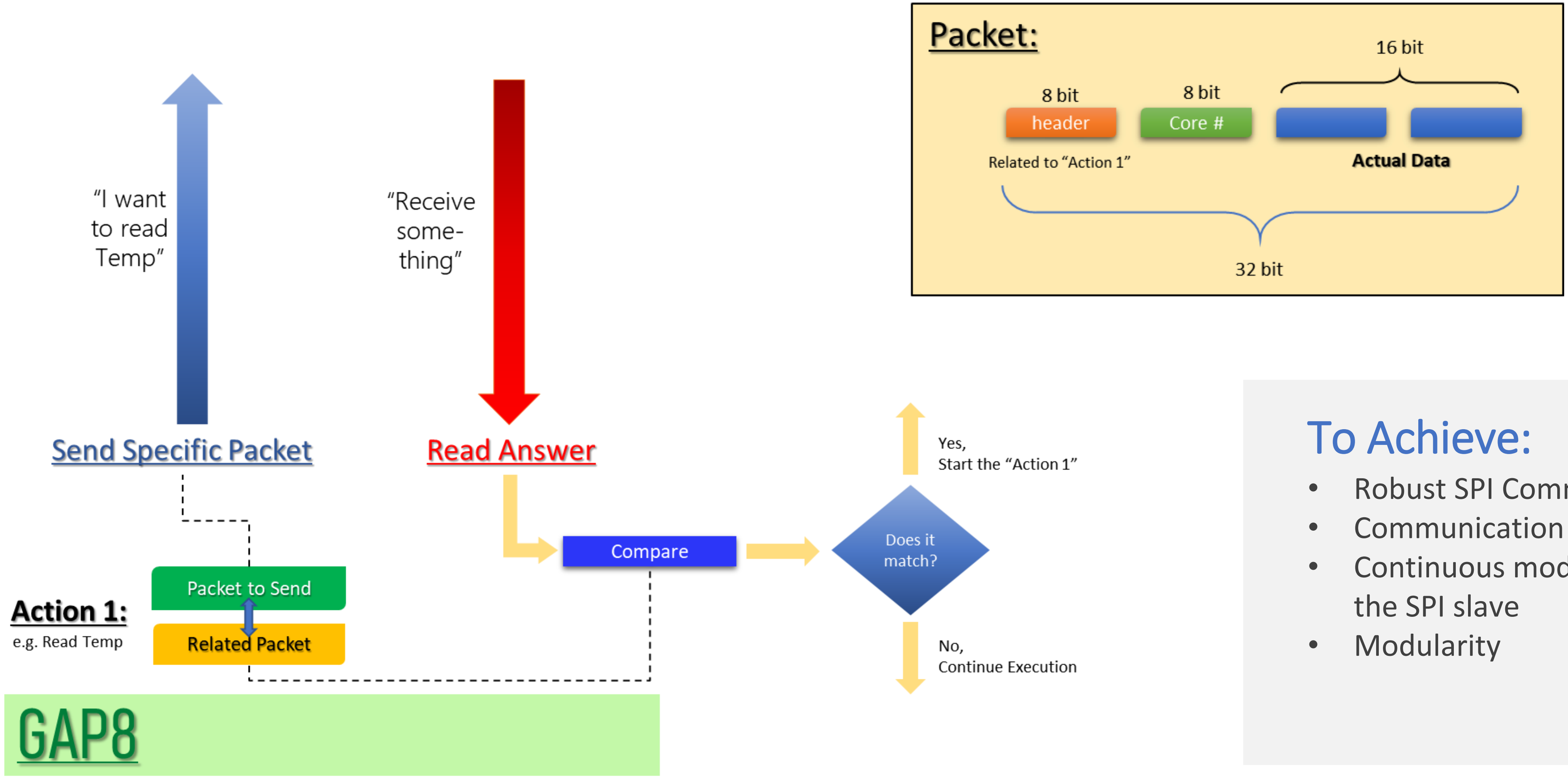
Plant Simultation – 1 Quadrant EPI

- SoC: STM32 Nucleo F401RE
- Discrete Thermal Model State Space Simulation
- Continuous Computation
- SPI slave managed through ISR and DMA
- UART communication to collect Simulation Data



$$T(k+1) = AT(k) + B_p P(k) + B_E T_{amb}$$
$$T_{Si}(k) = CT(k)$$

SPI HANDSHAKE SYSTEM



To Achieve:

- Robust SPI Communication
- Communication Flexibility
- Continuous model computation of the SPI slave
- Modularity

FREERTOS TEST: Feasibility

	Activation Time (in Cycles)	Context Switch (in Cycles)
Mean	1810	1453
Standard Deviation	8,78	13,16
Range (Max – Min)	34	55

Analisys:

- The jitter is negligible
- The overheads are not negligible
- Analyzing the Load stalls and Instructions miss, about 37% of the cycles overhead can be optimized by hardware modifications
- If the uC runs at 250MHz the overheads reduces to the 4% of the total cycles

Activation Time

It is the time needed to unblock and run a Periodic Task through the TIMER ISR and the Notification System.

7,2%

Context Switch Time

It is the time needed to block a task through the Notification system API call, perform a context switch and start running a task.

5,8%

Total Time 25000 Cycles

At 50 MHz in a 500us periodic interval we have 25.000 cycles. **FreeRTOS overheads are about 20% of the total amount of cycles.**

-20%

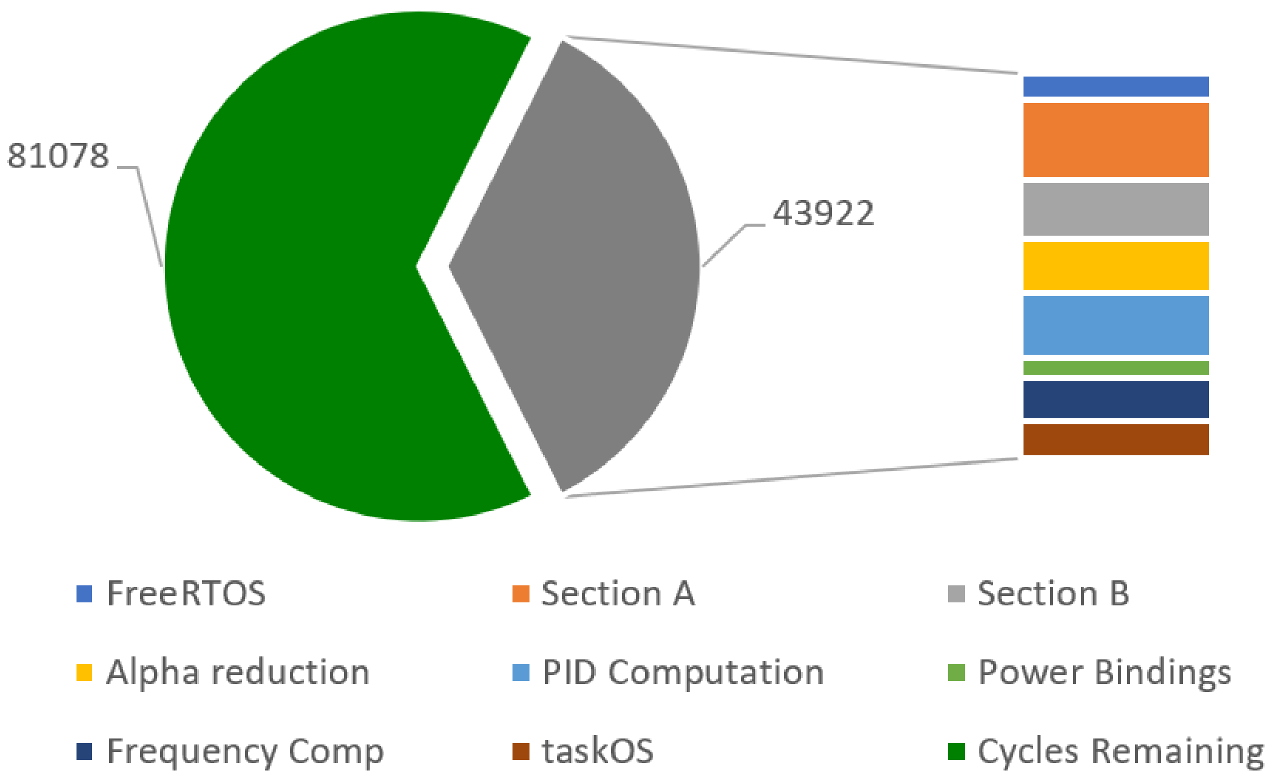
TEST: Control Execution Timings Verification

Cycles used at 250MHz

35%

Section A	9153
Section B	6696
Alpha Reduction	6156
PIDs Computation	7498
Power Bindings	2324
Frequency Computation	4806
TaskOS Execution	4310

Periodic Interval breakdown in Cycles



Scalability Linearity and Max Cores Projection

Scalability Equation

of the Periodic Control Task

$$Cycles = 5721 + 899 \cdot \#Cores$$

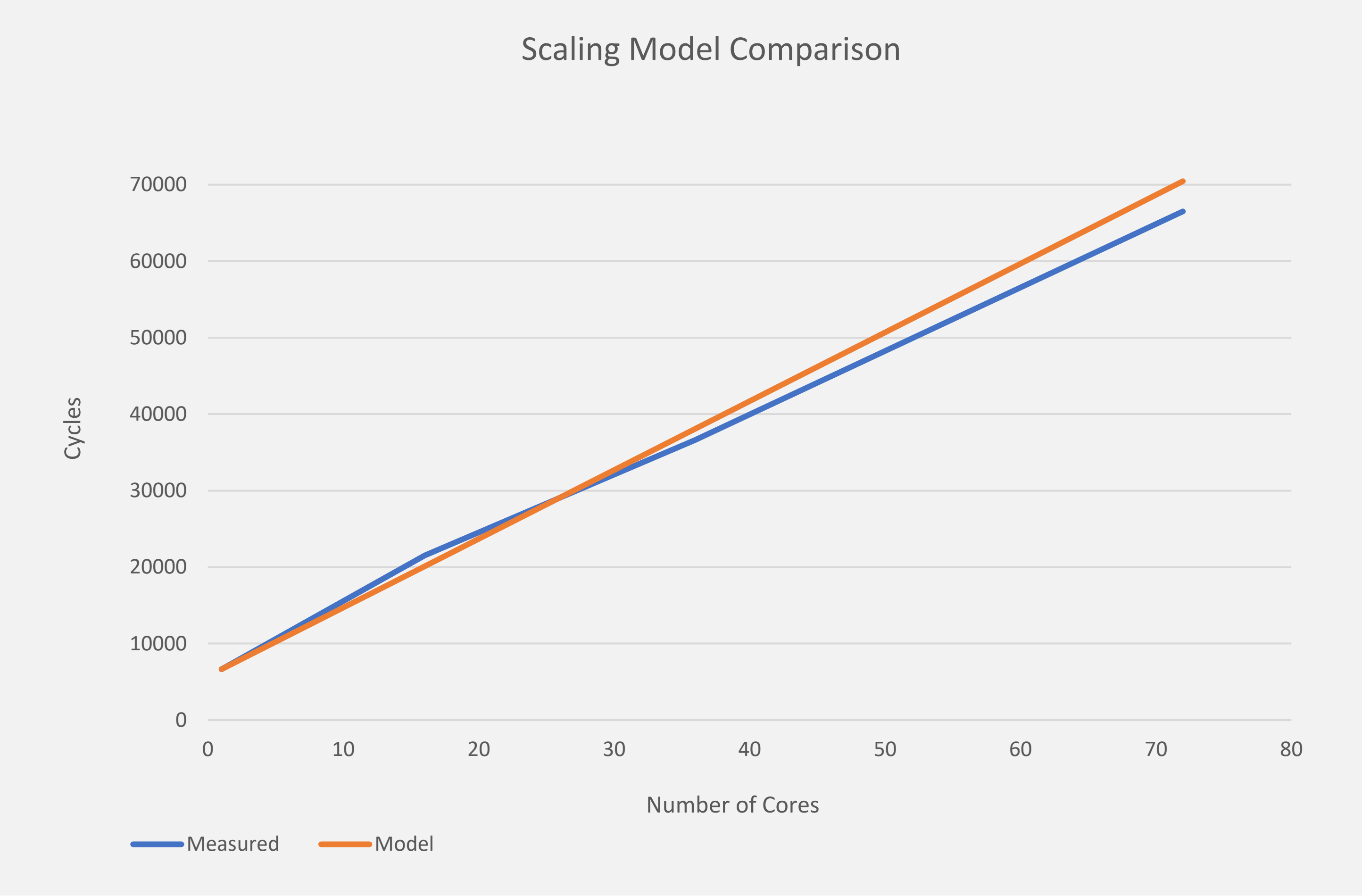
Analysis

The amount of Fixed Cycles is high (4,5% of the amount of Cycles in the Periodic Interval). The amount of Cycles per Core is good (0,7% of the total).

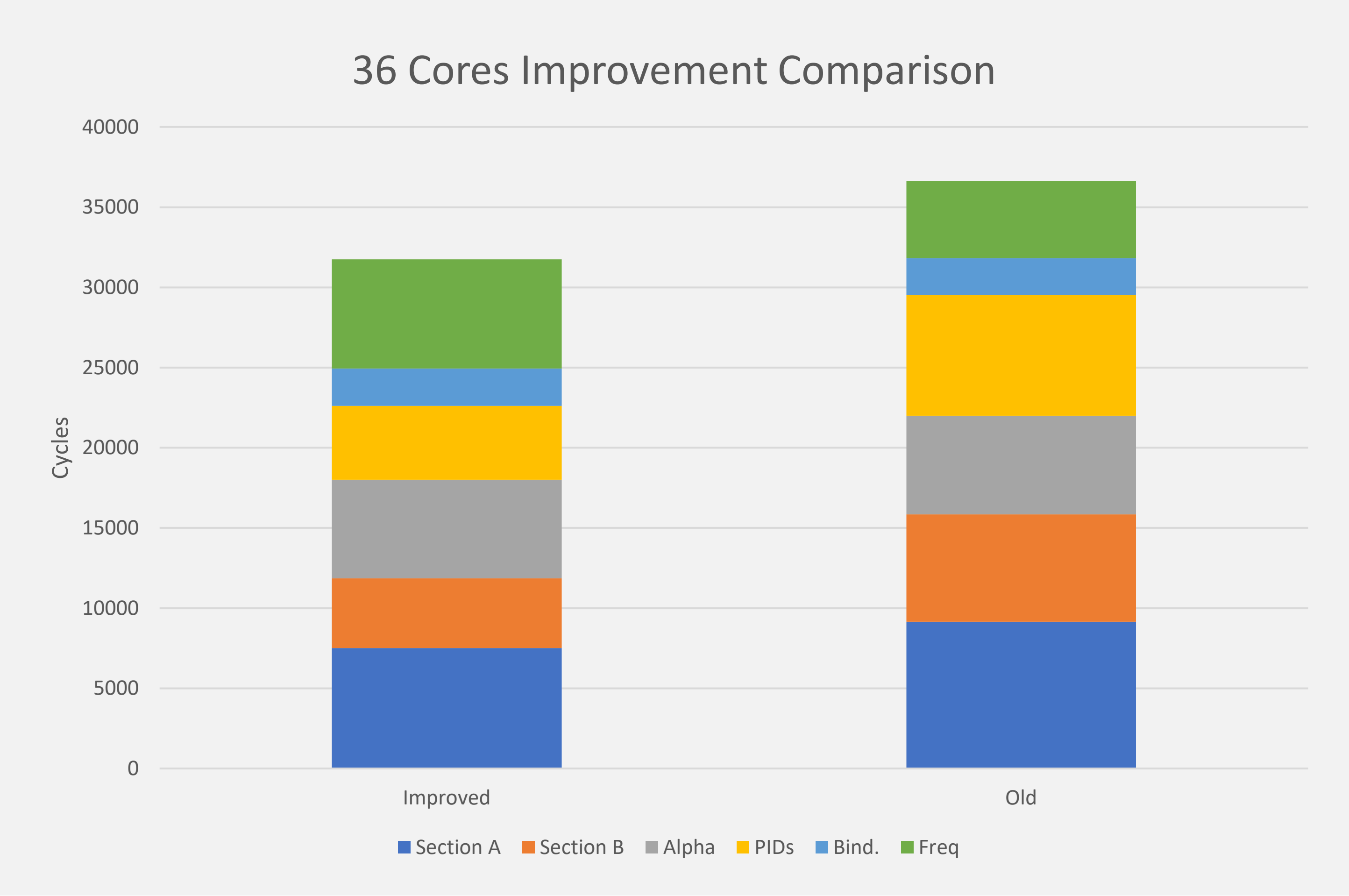
Projected Maximum number of Cores

Considering to occupy only just 51% of the available cycles per periodic iteration

$$n_{Core} = \frac{51\% - 4,5\%}{0,72\%} = 64$$



CODE PERFORMANCE IMPROVEMENT



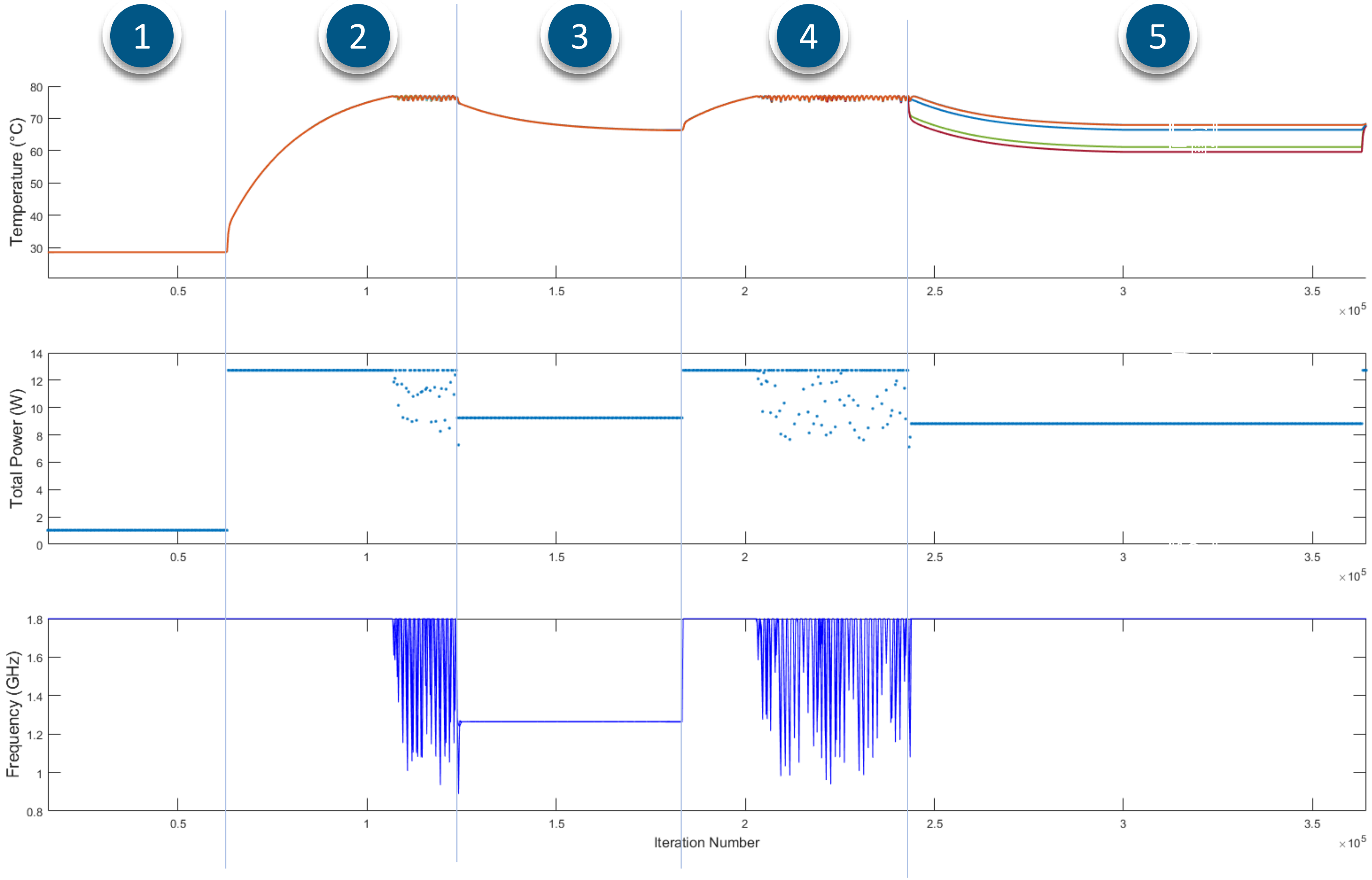
Fixed Cycles value is too high

Caused by multiple accesses to the Shared Memory variables: Mutex add overheads. New Design access Shared variables only once (at start or in the end) storing the data into a Local Variable, increasing the memory footprint of the Firmware.

Improvements

With the new design the cycles decreased in multiple sections. In the last they increased because is where memory is accessed. Now the Periodic Task executes in 31.750 cycles with 13,30% cycles improvement.

HIL RESULTS

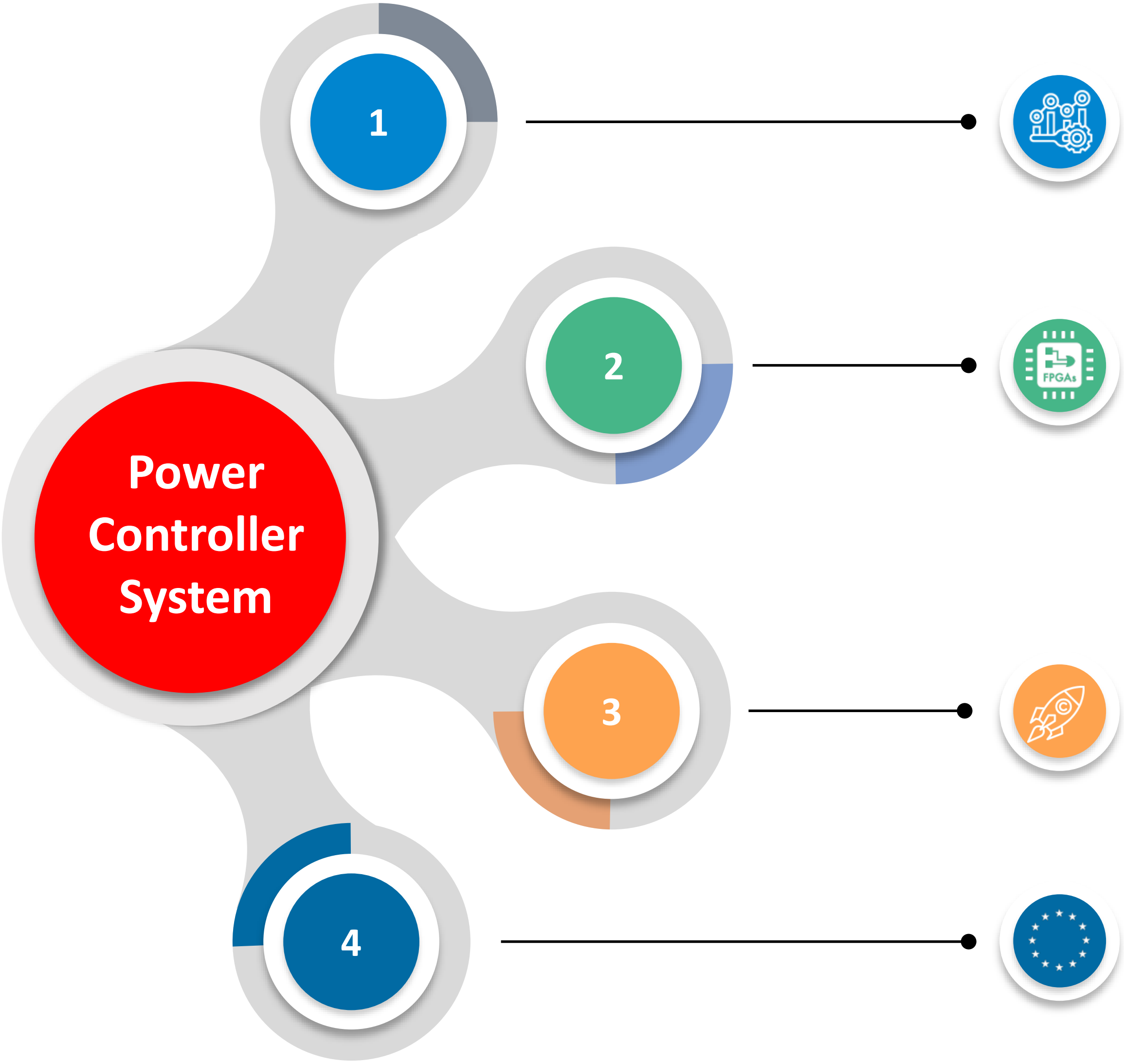


HPC Workflow Simulation:

- Action 1: Low Workload
- Action 2: Thermal Constraint
- Action 3: Power Capping
- Action 4: Thermal Constraint
- Action 5: Workload Relaxation

Future Works

The EPI project



Further Improve Firmware and Control
Improve Code optimization. Improve Security regarding the input, output and control values. Adding new Features.

Test on FPGA with Hardware Optimization
An FPGA version of the PCS is being developed. Further tests could be done to optimize performance and carry on a Hardware/Software co-design.

Optimal Control Implementation – Exploit Cluster Accelerator
Deploy and optimize for parallel computation an Optimal Control

EPI Project
Release the Project open-source. Include the developed PCS inside EPI project.

SAMSUNG
Tektronix
TBS 2000 SERIES DIGITAL OSCILLOSCOPE
lenovo
GAP8
Thermal Sensors (Raw values)
Predictive Controller
Router + Clock

Grazie dell’attenzione

Outline

Introduction

01 **INTRODUCTION – HPC & PCS**
Introduction to the HPC problem and PCS Requirements.

02 **CONTRIBUTION**
Contribution and obtained results of this Thesis.

03 **PCS ARCHITECTURE**
Design and implementation choices.

04 **GAP8 PROTOTIPATION**
Implementation work of the PCS on GAP8 Hardware

05 **CO-SIMULATION SETUP**
Design and Set-up of the Hardware-in-the-Loop test

06 **EXPERIMENTAL RESULTS**
Test methodology. Results and improvements.

OPTIMIZATION PROBLEM

$$\min_{f} \sum_{j=0}^{N-1} |f_t(k+j|k) - f(k+j|k)|_R^2$$

subject to:

$$T_{Si,i}(k+j+1|k) \leq T_{CRITi} \quad \forall i=1,\ldots,n_s \quad \forall j=0,\ldots,N$$

$$f_{min}\mathbf{1}_{n_s} \preceq f(k+j|k) \preceq f_{max}\mathbf{1}_{n_s}, \quad \forall j=0,\ldots,N-1$$

$$V_{dd\_min}\mathbf{1}_{n_s} \preceq V_{dd}(k+j|k) \preceq V_{dd\_max}\mathbf{1}_{n_s}, \quad \forall j=0,\ldots,N-1$$

$$f_{ti} = f_{tj} \text{ if } B(i,j) = 1$$

$$\sum_{i=1}^{n_s} P_i \leq P_{budget}$$

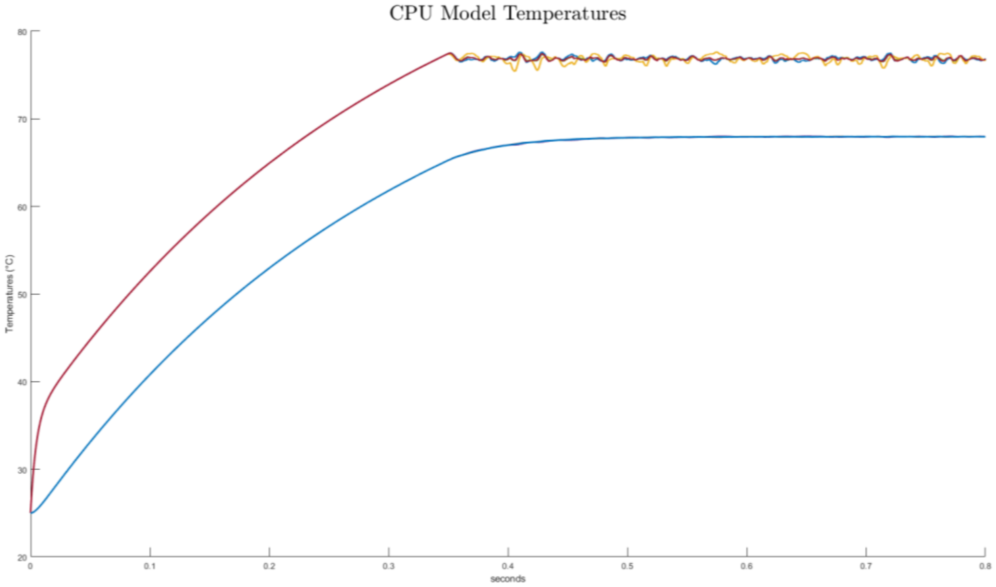
Where:

- *f* are the decision variables (i.e. Frequencies to be assigned)
- *k* denotes the discrete time variable
- *N* is the number of steps assumed are the horizon
- T_{crit} is the critical temperature
- $T_{si,i}$ are the thermal states of the chip (i.e. the temperature in the silicon of each component)
- N_s is the number of elements/cores
- *B(i, k)* is a n_s x n_s symmetric matrix to describe binding constraints

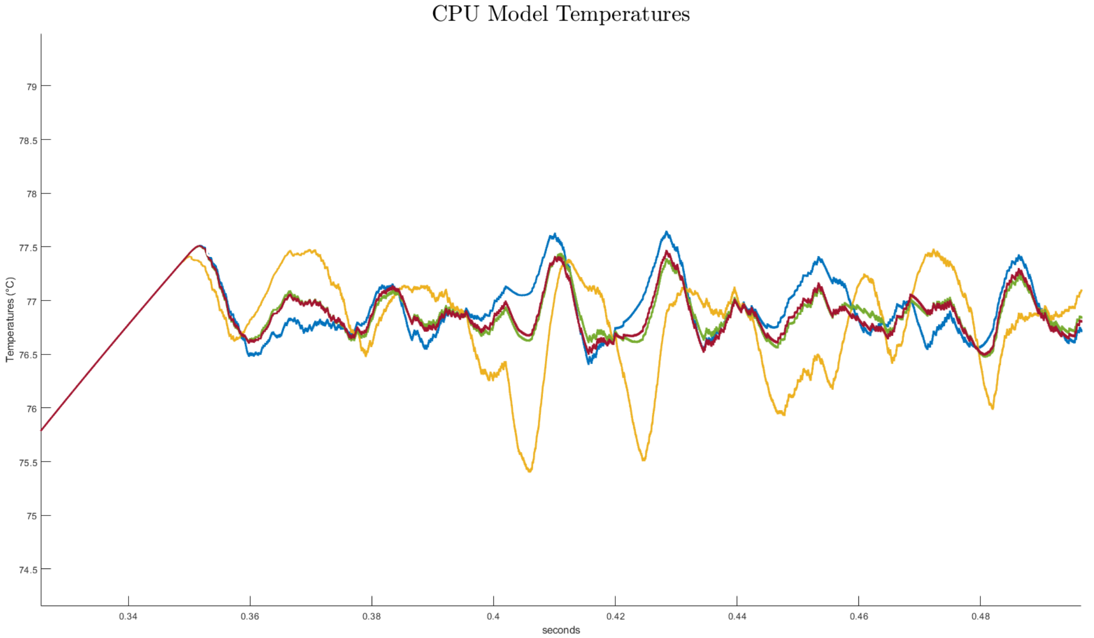
The Optimization Problem

Is a Convex Quadratic Problem that can be solved by numerical iterative optimization algorithm

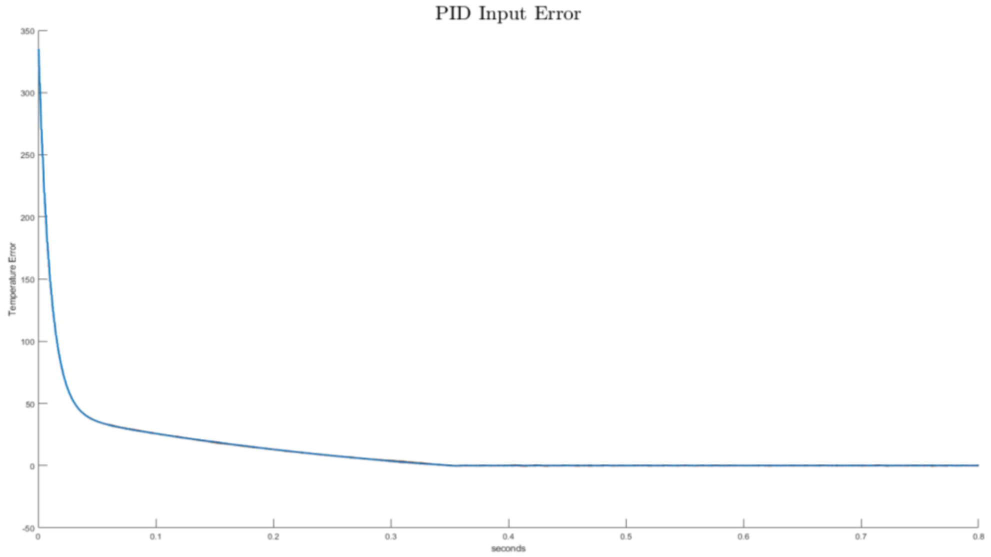
MATLAB/SIMULINK TEST



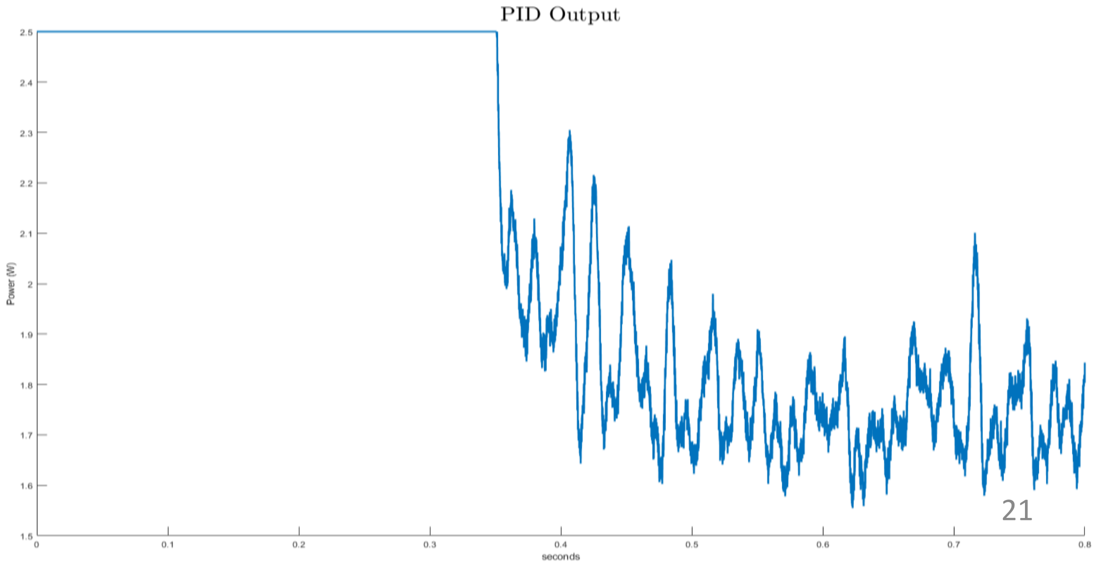
CPU Model Temperatures



CPU Model Temperatures



PID Input Error



PID Output

Setting Up the HiL Test

Bugs and Journey

Timer ISR Priority Bug

A bug in SDK v2.7 wouldn't allow to call FreeRTOS APIs inside ISR

SPI API Utilization

Lack of Documentation and examples along with confusing names made it difficult to use SPI APIS

uDMA Bugs

uDMA was afflicted by multiple bugs that made SPI communication not consistent, application crashes, GPIOs bugs, …

3+ SPI Config. Bugs

SDK v3.2 had multiple SPI configuration bugs and Configuration APIs missing

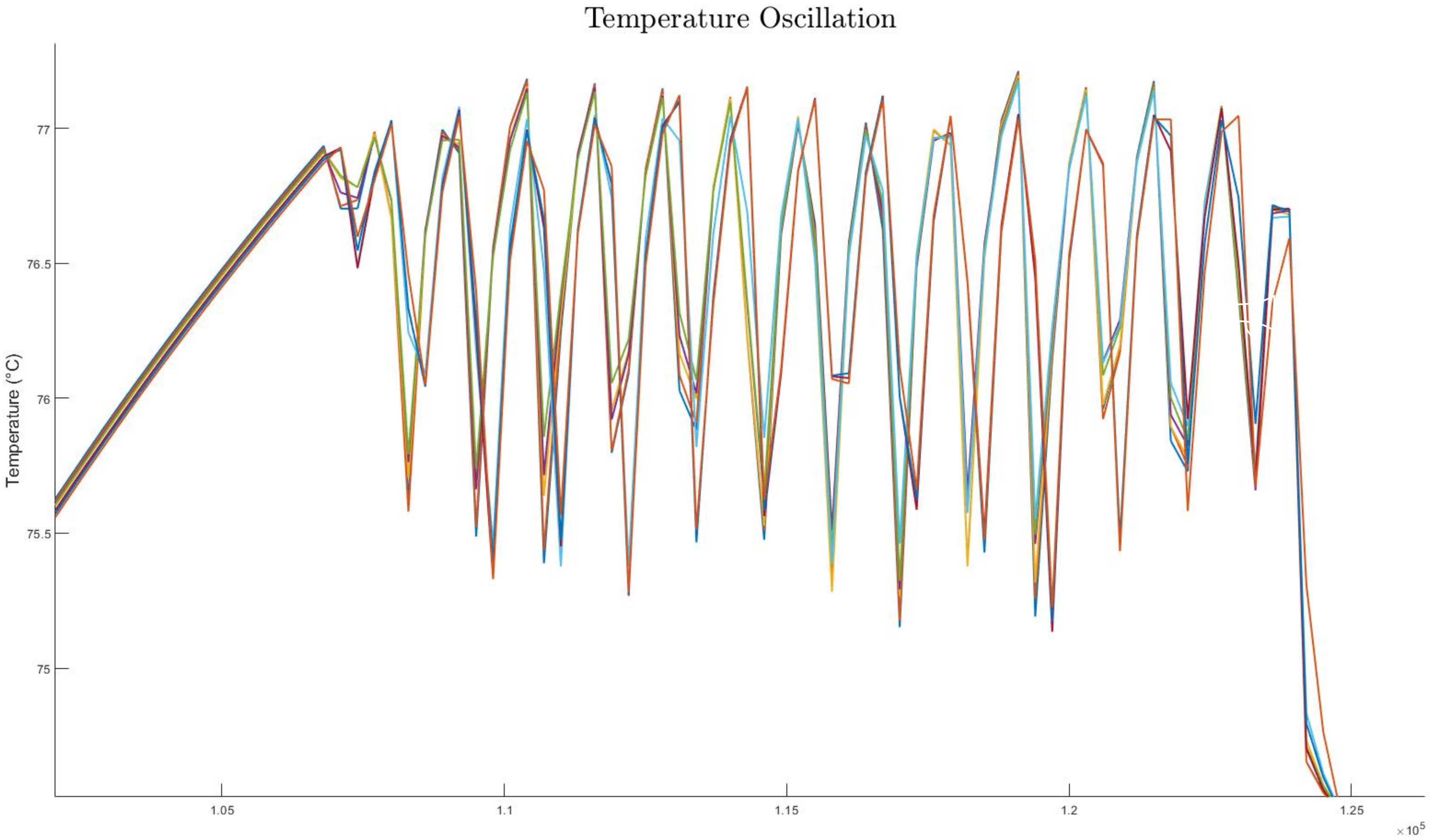
FreeRTOS Priority Config.

The maximum number of FreeRTOS priority was 2

SPI Clock – PLL Bug

GAP8v1 has a bug in the configuration of the SPI clock related to the PLL

HIL RESULTS



HPC Workload Simulation:

- Action 1: Normal
- Action 2: Thermal Constraint
- Action 3: Power Capping
- Action 4: Thermal Constraint
- Action 5-6: Workload Relaxation
- Action 7-8: Thermal Constraint
- Action 9 Power Capping