

This is the post peer-review accepted manuscript of:

A. Bartolini et al, "A PULP-based Parallel Power Controller for Future Exascale Systems" 2019 26th IEEE International Conference on Electronics, Circuits and Systems (ICECS), Genoa, Italy, 2019, pp. 771-774. doi: 10.1109/ICECS46596.2019.8964699

The published version is available online at: <https://ieeexplore.ieee.org/abstract/document/8964699>

© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

A PULP-based Parallel Power Controller for Future Exascale Systems

Andrea Bartolini*, Davide Rossi*, Antonio Mastrandrea*, Christian Conficoni*,
Simone Benatti*, Andrea Tilli*, Luca Benini*[†]

*Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, Italy
{a.bartolini, davide.rossi, antonio.mastrandrea, christian.conficoni, simone.benatti, luca.benini, andrea.tilli}@unibo.it

[†]Integrated Systems Laboratory, ETH Zurich, Switzerland {lbenini}@iis.ee.ethz.ch

Abstract—Power management of digital circuits is raising of importance in a broad spectrum of computing domains. High-performance computing systems as the effect of the stop of Dennard’s scaling have become power and thermal limited. In this manuscript, we evaluate the feasibility of using an open-source RISC-V based power controller for the high-performance computing market.

Index Terms—PULP, RISC-V, HPC, power management

I. INTRODUCTION

With the end of Dennard’s scaling [1], [2], the last decade has seen a progressive increase of the power density required to operate each new processor generation at its maximum performance. Supercomputing installations have suffered from this power density increase, which over the years has pushed up the energy provisioning and cooling costs [3]–[5]. To mitigate these effects processors in this market segment, embed dedicated HW resources to control the power consumption dynamically, prevent thermal hazards, and increase the energy efficiency of the computation.

To achieve these goals the power controller has to: (i) interface with several on-chip and off-chip sensors, and power management interfaces and actuators; (ii) perform complex computational tasks, like automation control, signal processing, optimisation and machine learning algorithms.

PULP is an open-source parallel computing platform developed as a joint project between ETHZ and University of Bologna, originally developed with the aim of satisfying the computational demands of IoT applications requiring flexible processing of data streams typically generated by multiple sensors [6]. It consists of a set of register transfer level IPs, released under Solderpad license, assembling a complete system on chip infrastructure, hence including processors, communication system, memory system and peripheral system. OpenMP programming model is supported on PULP, as well as real-time operating systems such as Zephyr-OS and FreeRTOS, on top of a GCC 7.1 toolchain, enabling agile application porting, development, performance tuning and debugging.

In this manuscript, we propose the use of a PULP-based controller for power management in HPC compute nodes.

In Section II we present the power management problem in HPC systems. In Section III we introduce the PULP platform. In Section IV we introduce the firmware requirements for a

power controller in HPC systems and in Section V we evaluate the benefits of using a PULP based power controller w.r.t. state-of-the-art microcontrollers.

II. POWER MANAGEMENT IN HIGH PERFORMANCE COMPUTING

In this section, we describe the role of the power controller in High-Performance Computing (HPC) processors.

As shown in Figure 1 power controller is connected: (i) on-chip to the power control knobs (managing the power consumption and performance of the main processing elements) and to sensors (monitoring the process, temperature and voltage of the main processing elements); (ii) off-chip to the Voltage Regulator Modules (VRMs) which powers the chip, other onboard components, and the Board Management Controller (BMC).

The power management uses these hardware components and connections to support a set of out-of-band services and in-band services.

The in-band services are delivered to the applications and operating systems running in the processing elements of the chip and are composed of: (i) dedicated power governors and power-related telemetry at the operating system level; (ii) a dedicated interface to let applications and programming model runtimes to specify power management hints and prescriptions; (iii) a dedicated interface to the System and Resource Management to support CPU and node-level power capping as well as managing the trade-off between Throughput and Energy Efficiency. The out-of-band services are delivered to the system administrator and system management tools through the Board Management Controller (BMC). These services consist of the out-of-band power telemetry, system-level power capping and reliability and serviceability.

A. The Power Controller

The power controller has the role in interfacing with all the physical sensors and actuators, and O.S. and user’s applications. Thanks to these interfaces the power controller periodically read the status of the main processing elements (Process, Temperature and Voltage) and sets accordingly to the power management policies the operating point of them (Voltage, Frequency). In addition to these metrics, the power controller reads the power consumption of the voltage rails from the

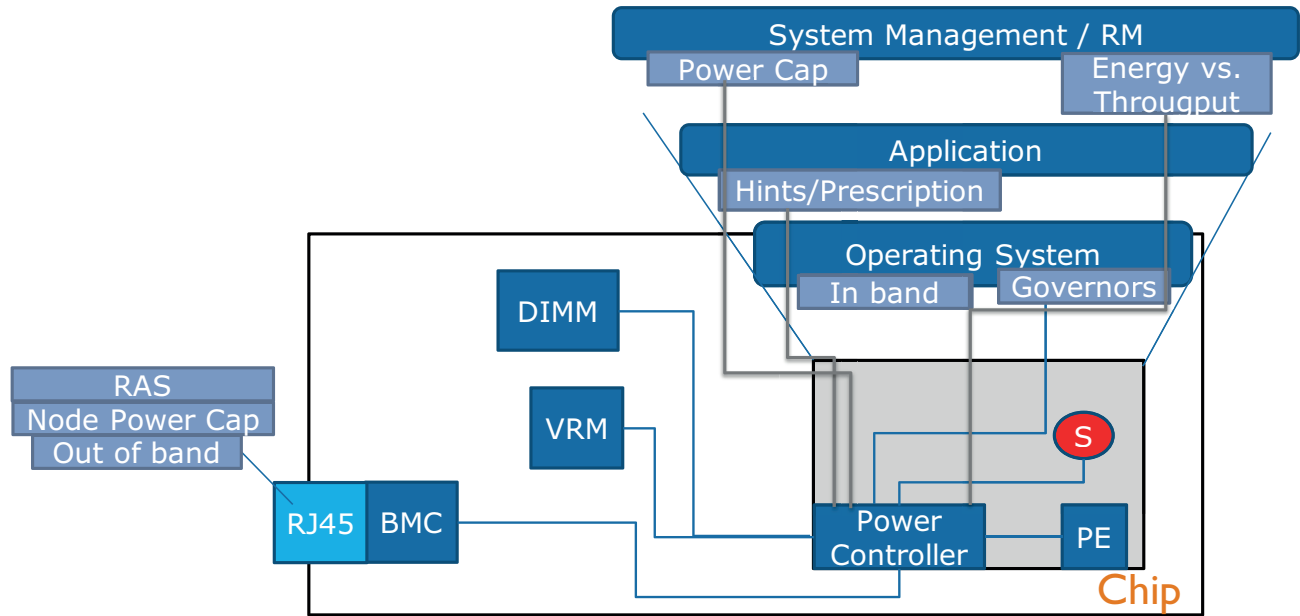


Fig. 1. Power management in HPC systems

VRM periodically and receives from the O.S. the requirements in terms of performance level (Target Frequency), power budget, and characteristics of the workload to be executed. The power management policy determines based on these parameters the best operating point at which executing the processing elements while ensuring the thermal stability, the power budget and the application constraints. The power management allows the application and the programming model run-time to require changes in the operating point asynchronously to track the application phases and enter in low-power operating points during –I/O, memory and communication- bounded phases to increase the energy-efficiency.

B. The In-Band Services

The power controller shares an internal memory region with the I/O address space of the processing elements. This interface allows the O.S. to periodically access a set of status data structures containing the power controller status, statistics, and power consumption of the different power rails and components. This information can be used and accessed by the applications, and the users, to monitor fine-grain the energy consumed by the applications, enabling energy-awareness.

C. The Out-of-Band Services

In addition to the power management policy and the In-band services, the power controller interfaces with the BMC to support out-of-band services. These comprise fine-grain telemetry on the chip power and performance status, chip level and system-level power capping and reporting of errors and faults in the chip and main processes.

III. THE PULP PLATFORM

PULP (pulp-platform.org) is an open-source energy-efficient RISC-V architecture. It is developed for more than microcontroller applications, and it is capable of delivering higher performance in the same power budget form factor than a standard microcontroller. PULP started as an academic project but now is becoming a reference implementation for a broad set of IoT appliances. PULP can be downloaded for free from its public git hub repository and used. PULP platform differentiates from its competitors by three folds:

- A powerful SoC based on a RISC-V core with DSP and SIMD operations
- The integration of SoC with a parallel cluster of cores which can deliver important speed up in a wide range of machine learning and signal processing applications.
- A design optimised for energy efficiency and event computing.

The PULP architecture is based on a tiny 32bits RISC-V CPU optimized for area, and a multicore cluster. The CPU is connected to memory, cluster, and peripheral subsystem via a low latency logarithmic interconnect. The main memory is divided into 4 banks with word-level interleaving to minimize banking conflicts during parallel accesses through multiple ports of the interconnect. All elements share access to an L2 memory area. The Cluster cores share access to an L1 Tightly-Coupled Data Memory (TCDM) area and instruction cache. Multiple DMA units allow autonomous and fast transfers between cluster L1 memory and L2 memory and external peripherals.

IV. THE POWER CONTROLLER FIRWMARE REQUIREMENTS

A. The SW Support

PULP supports OpenMP as parallel programming model, as well as real-time operating systems such as Zephyr-OS and FreeRTOS, on top of a GCC 7.1 toolchain, enabling the fast design of applications with multiple time constraints and computational demand.

B. The power controller firmware

Exploiting the software structure defined before, the functionalities of the power manager described in Section II have been deployed to three different tasks described in the following:

- **Thermal Control Task:** It is a hard-real time, high priority periodic task where decisions in terms of power budgeting and thermal regulation are made. Therefore it has to: (i) read the sensors measurements (temperatures, process, voltages), (ii) read settings (desired cores frequencies, power budget) written by the host O.S. in the shared memory¹, (iii) read calibration coefficients related to the core power model, and (iv) produce the corresponding inputs (frequency and voltage) for each core to meet budget and temperature constraints with minimal performance downgrade. It is further to note that distributed/decentralised strategies (possibly realising non-trivial optimisation and control algorithms) can be efficiently run in parallel, exploiting the HW/SW architecture of the considered system. Besides, such a task is in charge of saving the actuated commands each time it is executed, both for telemetry and possible learning purposes. To this aim, a copy of such information is stored in the shared memory. Finally, the task can handle pending BMC request which has not been served by the related task (see the next item) for thermal safety reasons.
- **Power Model Learning Task:** This is a periodic task with a cycle time greater than the control task but synchronised with it. Its activation period can be related to the refresh rate of the off-chip voltage regulator measurements. Indeed, the power consumption provided it is based on values obtained from the voltage regulators, the thermal controller previous period settings and the workloads (provided by the O.S through performance counter readings). Based on such information a learning algorithm (e.g. a non-linear regression or neural network identification) can be applied to estimate the map relating the cores features (frequency, voltage, workloads) to their power consumption. The coefficients of such map (in the form of function coefficients, or neural network weights) are then stored to be used by the thermal control task.
- **BMC handling task:** This is an asynchronous task which is triggered by the BMC whenever a request involving the power manager is generated (i.e. frequency, budget

¹To this aim some sort of synchronisation with the O.S tick can be foreseen so that as updated as possible data coming are used

changes, telemetry data acquisition). First, it reads the BMC data on a specific shared memory area, then it serves the requests, provided that they are feasible w.r.t. the thermal controller settings (that is frequency changes are allowed only if they are thermally safe), otherwise it sets such requests as pending, and they will be evaluated by the thermal control task at its next activation.

V. A PULP BASED POWER CONTROLLER

The PULP project can be at the base of the design of the power controller for HPC platforms for the following reasons:

- 1) A mature SoC design. PULP has been taped out in several technology nodes and configurations. Several variants of PULP SoCs have been implemented, fabricated and tested in several technology nodes by the UNIBO/ETHZ labs, including ALP 180 (3 chips), UMC 180 (3 chips), SMIC130 (4 chips), UMC65 (10 chips), TSMC 40 (1 chip), GF 28, STM 28 FD-SOI (3 chips), GF 22 FDX (3 chips).² Moreover, products and test chip based on PULP IPs have been fabricated by GreenWaves Technologies³, IBM⁴, Google⁵, NXP⁶, CEVA⁷
- 2) A more powerful SoC than SoA competitors (Table I,II):
 - (i) In Table I, the RI5CY processor used in PULP SoCs has been compared with M4 and H7 ARM cores, numbers are scaled to 65nm technology. Two frequency targets have been used for RI5CY, both a low-frequency (185 MHz) and high-frequency (560 MHz), to compute area (equivalent nand 2) and dynamic power (uW/MHz) of the IPs. The table also includes a comparison on a general-purpose benchmark (coremark).
 - (ii) A full embodiment of the PULP system implemented in 55nm technology, namely GAP8, is further compared with ARM Cortex M4-based SoC: STM32L4 implemented in 90nm technology, and with an ARM Cortex M7 based SoC: STM32H7 implemented in 40nm technology on a highly DSP intensive kernel (inference of an 8-bit Cifar 10 convolutional neural network - CNN). When compared with the STM32L4 SoC the PULP system achieve 30x lower latency for computing each CNN, achieving 36.8x higher performance at the maximum frequency (GMAC/s) with an overall increase in the energy efficiency of the 8.67x. Differently, when compared with the STM32H7 SoC the PULP system achieves 19.6x lower latency for computing each CNN, achieving 7.45x higher performance at the maximum frequency (GMAC/s) with an overall increase in the

²More information can be found on the PULP platform website (<https://pulp-platform.org/implementation.html>).

³https://greenwaves-technologies.com/ai_processor_gap8/

⁴<https://content.riscv.org/wp-content/uploads/2018/05/16.10-16.25-Seiji-Munetoh-IBM-Japan.pdf>

⁵<https://content.riscv.org/wp-content/uploads/2018/05/13.15-13.30-matt-Cockrell.pdf>

⁶<https://content.riscv.org/wp-content/uploads/2018/05/11.20-11.45-Rob-Oshana-NXP.pdf>

⁷<https://www.ceva-dsp.com/wp-content/uploads/2018/05/Ceva-First-to-Launch-802.11ax-IP-CEVA.pdf>

energy efficiency of the 25x. (iii) We can conclude that a single RISC-V core of PULP has similar general-purpose performance, area and power consumption as an ARM Cortex M4 processor. With the big advantage of having significant more capabilities than single-core ARM platforms in terms of digital signal processing throughput (and efficiency) than the parallel nature and energy-efficient DSP extensions of the cluster. For this reason, PULP enables to power the power controller with more horsepower than SoA and competitor versions. It should be noted that similar DSP extensions are being evaluated by ARM, but will not be available in any of the ARM ISA currently [7].

- 3) The PULP architecture has been applied to a wide set of smart applications featuring edge artificial intelligence and signal processing [8], [9]. If used as a power controller, it will enable to combine AI and predictive control in the power management subsystem, paving the way to smarter and greener servers. The parallel nature of PULP will enable to scale the power control policies with the number of cores integrated with the tile. Some practical examples of how this will be done are: (i) Take advantage of the openness of the PULP design to obtain an SoC capable of interfacing with IO at high frequency to handle the on-chip and off-chip sensors and communication channels without loss of information. (ii) The use of the additional computational power to update internal models of the temperature evolution and power consumption of the chips. (iii) The use of the additional computational power solve optimisation algorithms needed by model predictive control algorithms. (iv) The use of the MISO extensions to support the integration of signal processing and deep learning for automated fault identification and isolation.

Processor	RISCV	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

VI. CONCLUSION

In this paper we have described the role of power controller in HPC systems and evaluated how the pulp project can be used to create a power controller for HPC systems.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 826647.

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.)

REFERENCES

- [1] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct. 1974.
- [2] H. Esmailzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May 2012.
- [3] F. Fraternali, A. Bartolini, C. Cavazzoni, G. Tecchiolli, and L. Benini, "Quantifying the impact of variability on the energy efficiency for a next-generation ultra-green supercomputer," in *Proceedings of the 2014 international symposium on Low power electronics and design*. ACM, 2014, pp. 295–298.
- [4] C. Conficoni, A. Bartolini, A. Tilli, C. Cavazzoni, and L. Benini, "Hpc cooling: A flexible modeling tool for effective design and management," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2018.
- [5] M. Maiterth, G. Koenig, K. Pedretti, S. Jana, N. Bates, A. Borghesi, D. Montoya, A. Bartolini, and M. Puzovic, "Energy and power aware job scheduling and resource management: Global survey—initial analysis," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 685–693.
- [6] D. Rossi, F. Conti, A. Marongiu, A. Pullini, I. Loi, M. Gautschi, G. Tagliavini, A. Capotondi, P. Flatresse, and L. Benini, "Pulp: A parallel ultra low power platform for next generation iot applications," in *2015 IEEE Hot Chips 27 Symposium (HCS)*, Aug 2015, pp. 1–39.
- [7] ARM, "ARM community new vector extension for arm m," <https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/arm-helium-the-new-vector-extension-for-arm-m-profile-architecture>, 2019, accessed: 2019-07-05.
- [8] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64mw DNN-based Visual Navigation Engine for Autonomous Nano-Drones," *IEEE Internet of Things Journal*, pp. 1–1, 2019, arXiv: 1805.01831. [Online]. Available: <http://arxiv.org/abs/1805.01831>
- [9] PULP, "PULP platform website," <https://www.pulp-platform.org/publications.html>, 2019, accessed: 2019-07-05.