

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

Monte Cimone: Paving the Road for the First Generation of RISC-V High-Performance Computers

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:

Bartolini A., Ficarelli F., Parisi E., Beneventi F., Barchi F., Gregori D., et al. (2022). Monte Cimone: Paving the Road for the First Generation of RISC-V High-Performance Computers [10.1109/SOCC56010.2022.9908096].

Availability:

This version is available at: <https://hdl.handle.net/11585/905818> since: 2022-11-22

Published:

DOI: <http://doi.org/10.1109/SOCC56010.2022.9908096>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).
When citing, please refer to the published version.

(Article begins on next page)

This is the final peer-reviewed accepted manuscript of:

A. Bartolini *et al.*, "Monte Cimone: Paving the Road for the First Generation of RISC-V High-Performance Computers," 2022 IEEE 35th International System-on-Chip Conference (SOCC), Belfast, United Kingdom, 2022, pp. 1-6

The final published version is available online at:

<https://doi.org/10.1109/SOCC56010.2022.9908096>

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>)

When citing, please refer to the published version.

Monte Cimone: Paving the Road for the First Generation of RISC-V High-Performance Computers

Andrea Bartolini*, Federico Ficarelli[†], Emanuele Parisi*, Francesco Beneventi*, Francesco Barchi*,
Daniele Gregori[‡], Fabrizio Magugliani[‡], Marco Cicala[‡], Cosimo Gianfreda[‡],
Daniele Cesarini[†], Andrea Acquaviva*, and Luca Benini*[§]

*Università di Bologna, DEI, Bologna, Italy; [†]CINECA SCAI, Casalecchio di Reno, Italy;

[‡]E4 Computer Engineering, Scandiano, Italy; [§]ETH Zürich University, Zürich, Switzerland

*{a.bartolini, emanuele.parisi, francesco.beneventi, francesco.barchi, andrea.acquaviva, luca.benini}@unibo.it

[†]{d.cesarini, f.ficarelli}@cineca.it, [‡]{daniele.gregori, fabrizio.magugliani, marco.cicala, cosimo.gianfreda}@e4company.com

Abstract—The new open and royalty-free RISC-V ISA is attracting interest across the whole computing continuum, from microcontrollers to supercomputers. High-performance RISC-V processors and accelerators have been announced, but RISC-V-based HPC systems will need a holistic co-design effort, spanning memory, storage hierarchy interconnects and full software stack. In this paper, we describe Monte Cimone, a fully-operational multi-blade computer prototype and hardware-software test-bed based on U740, a double-precision capable multi-core, 64-bit RISC-V SoC. Monte Cimone does not aim to achieve strong floating-point performance, but it was built with the purpose of “priming the pipe” and exploring the challenges of integrating a multi-node RISC-V cluster capable of providing an HPC production stack including interconnect, storage and power monitoring infrastructure on RISC-V hardware. We present the results of our hardware/software integration effort, which demonstrate a remarkable level of software and hardware readiness and maturity - showing that the first generation of RISC-V HPC machines may not be so far in the future.

Index Terms—RISC-V, HPC, Power and Performance

I. INTRODUCTION

The strategic role of High Performance Computing (HPC) systems is widely acknowledged in many fields, from weather forecasting to drug design. With the pervasive digitalization of our society, high performance computers fuel the most disruptive mega-trends, from the deployment of artificial intelligence (AI) at scale (e.g. for training large machine learning models) to industrial internet-of-things (IoT) applications (e.g. for creating and maintaining digital twins). Thus, HPC systems are today strategic assets not only for academia and industry, but also as for public institutions and governments [1].

The key challenge in designing HPC systems today and in the foreseeable future is increasing compute efficiency, to meet the rapidly growing performance demand (10x every four years) within a constant or modestly increasing power budget, while facing the slow-down of Moore’s Law. To exacerbate the efficiency challenge, while integrated circuits technology is still delivering device density increases (albeit at a slower pace), power consumption does not scale down at the same rate. Hence power density grows and it is increasingly difficult to meet thermal design power specifications without

compromising performance. Disruptive technologies, such as quantum or optical computing may bring long-term relief in some specific application areas, but there is no silver bullet in sight.

To tackle the efficiency issue, academia and industry are aggressively pursuing architectural innovation and co-design strategies to develop HPC systems that mitigate the efficiency limitations of programmable architectures through various forms of specialization and domain-specific adaptation. Instruction Set Architectures (ISAs) have to evolve rapidly to sustain architectural evolution and domain adaptation, and the advent of the RISC-V open, royalty-free and extensible ISA has been a major step toward accelerating innovation in this area. An additional advantage of RISC-V with respect to the dominant proprietary ISAs (x86 and ARM) is that it is maintained by a global non-for-profit foundation with members across the world, ensuring a high degree of neutrality with respect to geopolitical tensions and their technology downfalls.

Currently, high-performance 64bit (RV64) RISC-V processors and accelerator chips are being designed, promising prototypes are demonstrated in numerous publications [2] and products are announced at a fast cadence [3], [4]. It is thus reasonable to expect that high-performance chips based on RISC-V will be available as production silicon within the next couple of years. However, building a HPC system requires significantly more than just high-performance chips. Many think that the RISC-V software stack and system platform are extremely immature, and will need several additional years of development effort before full applications could be run, benchmarked and optimized on a RISC-V-based HPC system. Our goal is dispel this overly conservative notion.

The main contribution of this work is to present Monte Cimone, the first physical prototype and test-bed of a complete RISC-V (RV64) compute cluster, integrating not only all the key hardware elements besides processors, namely main memory, non-volatile storage and interconnect, but also a complete software environment for HPC, as well as a full-featured system monitoring infrastructure. Further, we demonstrate that

it is possible to run real-life HPC applications on Monte Cimone today. Even though achieving strong double precision performance will be possible only with upcoming high-performance chips, we achieved the following milestones:

- We designed and set up the first RISC-V-based cluster containing eight computing nodes enclosed in four computing blades. Each computing node is based on the U740 SoC from SiFive and integrates four U74 RV64GCB application cores, running up to 1.2 GHz and 16GB of DDR4, 1 TB node-local NVME storage, and PCIe expansion cards. The cluster is connected to a login node and master node running the job scheduler, network file system and system management software.
- We ported and assessed the maturity of a HPC software stack composed of (i) SLURM job scheduler, NAS filesystem, LDAP server, Spack package manager (ii) compilers toolchains, scientific and communication libraries, (iii) a set of HPC benchmarks and applications, (iv) the ExaMon datacenter automation and monitoring framework.
- We measured the efficiency of the HPL benchmark and STREAM benchmark with the toolchain and libraries installed by the SPACK. We compared the attained results against the one obtained for other RISC ISA architectures used in the 1st and 2nd ranked Top500 supercomputers (namely, Summit and Fugaku). Results show that upstream HPL achieved 46.5% utilization on Monte Cimone, the Marconi100 [5] and Armida [6] compute nodes achieved 59.7% and 65.79% of their peak respectively. The Monte Cimone node achieves slightly lower FPU utilization but in the range with the state of the art. When running an unoptimized Stream benchmark, Monte Cimone obtained just the 15.5% of the peak bandwidth, while Marconi100 and Armida obtained an efficiency of 48.2% and 63.21% respectively, pointing to significant margins for improvement in application and software stack tuning to the hardware target.
- We characterised the power consumption of various applications executed on Monte Cimone.

II. RELATED WORKS

The most recent successful effort to introduce a new ISA to HPC has involved the ARM ISA. Bringing the Arm ISA and software ecosystem to HPC maturity has required almost a decade and several funding rounds: The Mont-Blanc EU project series started in 2011, leading to the first ARM-based HPC cluster deployed in 2015 [7], based on SoCs developed for the embedded computing market. Notably, since June 2020, Fugaku [8], the fastest supercomputer in TOP500, is based on ARM scalable vector extension (SVE) ISA, and achieves more than 400 PFLOPs. Further, high-performance ARM-based SIMD processors are being adopted in servers and datacenters worldwide. We observe that it took approximately a decade for ARM to become a strong player in these highly competitive markets, even though X86 is still by far the dominant architecture in HPC and cloud.

The RISC-V ISA has been conceived just a decade ago, thus clearly its market penetration is much smaller than the

incumbent ARM and X86 ISAs. Today, only a few 64-bit RISC-V (RV64G ISA) SoCs are available commercially and none is in volume production for HPC or performance servers. Nevertheless, several high-performance RISC-V processors have been announced for high-performance general-purpose and accelerated computing markets [9]–[11]. In addition, a few research prototypes have been presented in the recent literature that demonstrate on silicon the technical feasibility and competitiveness of high-performance RISC-V computing engines [12]–[14]. Furthermore, the European Processor Initiative (EPI) launched in 2019 is funding a major research thrust to develop RISC-V based accelerators for HPC [15].

Among the RV64G chips available in low volumes on the market, for our cluster we chose the SiFive Freedom U740 SoC, featuring a 64-bit dual-issue, superscalar RISC-V U7 core complex configured with four U74 cores and one S7 core, an integrated high speed DDR4 memory controller, a root complex PCI Express Gen 3 x8 and standard peripherals. The availability of a main memory interface with reasonable performance and a PCIe root complex for connecting fast storage, IOs and accelerators, makes this SoC a good basis for exploring the deployment of RISC-V processors in a scalable cluster and working on the software stack. Still, it is apparent that the performance and number of cores in the SoC is not sufficient to achieve performance comparable to mature ARM and X86 cores.

The maturity of the software ecosystem around RISC-V has been growing at a very fast rate. A reasonably complete snapshot of major software packages available for RISC-V is maintained by the RISC-V foundation [4]. While the list is not complete, due to the very fast growth of the RISC-V community of developers, it is clear that porting efforts so far have mainly focused on embedded and AI applications. A HPC special interest group (SIG) for RISC-V has been founded in 2019 [16]. However, to the best of our knowledge, the demonstration of a complete software stack and HPC applications running on real hardware on RISC-V nodes in a multi-blade cluster is still missing. The present work aims at filling this gap.

III. MONTE CIMONE HARDWARE

Monte Cimone is based on the SiFive Freedom U740 RISC-V SoC HiFive Unmatched board integrated in an HPC node form factor. The E4 RV007 blade prototype system, adopted as Monte Cimone building Block, is a dual-board platform server, with a form factor of 4.44 cm (1 RackUnit) high, 42.5 cm width, 40 cm deep. Two 250 W power supplies, one for each board (compute node), are installed inside the case, making the system ready with abundant power headroom for future expansions with hardware accelerators and PCIe Network Card connector.

The board follows the Industry Standard Mini-ITX with a size of 170 mm per 170 mm. It features one SiFive Freedom U740 SoC, 16 GB of 64-bit DDR4 memory operating up to 1866s MT/s and high-speed interconnects with PCIe Gen 3 x16 (but it's limited to x8 lanes), one Gigabit Ethernet, and four USB 3.2 Gen 1, see Figure 1.

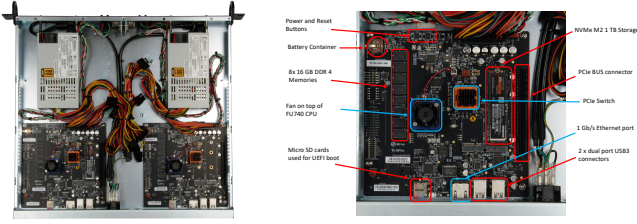


Fig. 1. The E4 RV007 Server Blade is based on a dual SiFive Freedom U740 SoC, the form factor is 4.44 cm (1 RackUnit) high, 42.5 cm width, 40 cm deep. The size of each RISC-V development board is 170 mm per 170 mm.

In RV007 node the M.2 M-key expansion slot is occupied by a 1 TB NVME 2280 SSD Module storage device used by the Operating System. The Micro SD card is present and used for the UEFI Boot. Two buttons for reset and power up operations are available on top of the board and in front of the case.

The FU740-C000 is a Linux-capable SoC powered by SiFive’s U74-MC, the first (to the best of our knowledge) commercially available superscalar heterogeneous multi-core RISC-V Core Complex. The FU740-C000 is compatible with all applicable RISC-V standards.

The U74-MC core complex is composed of four 64-bit U74 RISC-V (Application) cores. Each U74 core has a dual issue in-order execution pipeline, with a peak sustainable execution rate of two instructions per clock cycle. The U74 core supports Single-Precision Floating Point, Double-Precision Floating Point, Atomic, and Compressed RISC-V extensions.

The SiFive Freedom board features a single port gigabit Ethernet copper interface. Moreover, we equipped two of the compute nodes with an Infiniband FDR HCA (56Gbit/s) to leverage RDMA communications among different nodes to improve the network throughput and the communication latency. We used a Mellanox ConnectX-4 FDR HCA interconnect through the PCI-e interface available on the compute node. This HCA support x8 free PCIe Gen 3 lanes, which are currently supported by the vendor. The first experimental results show that the kernel is able to recognise the device driver and mount the kernel module to manage the Mellanox OFED stack. We are not able to use all the RDMA capabilities of the HCA due yet-to-be-pinpointed incompatibilities of the software stack and the kernel driver. Nevertheless we successfully run an IB ping test between two boards and between a board and an HPC server showing that full Infiniband support could be feasible. This is currently a feature under development.

In addition, the SiFive Freedom U740 SoC features 7 separated power rails including the core complex, IOs, PLLs, DDR subsystem and PCIe one. The HiFive Unmatched board implements separated shunt resistors in series with each of the SiFive U740 power rails as well as for the on-boards memory banks [17].

IV. MONTE CIMONE SOFTWARE STACK

Since our goal was to build a software environment as close as possible to a production HPC cluster, we

leveraged the Spack [18] package manager to deploy the full software stack and make it available to all system users via environment modules [19]. Actual Spack architecture and micro-architecture support, in the form of platform-specific toolchain flags, is provided by the archspec [20] module. Explicit support for the linux-sifive-u74mc target triple was already present (archspec version 0.1.3) and tested to be working without modifications. The user-facing software stack installed successfully via Spack (version 0.17.0) and presented to users is gcc:10.3.0, openmpi:4.1.1, openblas:0.3.18, fftw:3.3.10, netlib-lapack:3.9.1, hpl:2.3, netlib-scalapack:2.1.0, stream:5.10, quantumESPRESSO:6.8, (transitive dependencies omitted for brevity). All of the nodes are running upstream Ubuntu 21.04 deployed from riscv64 server images without modifications and mount a remote NFS.

We ported on Monte Cimone all the essential services needed for running HPC workloads in a production environment, namely NFS, LDAP and the SLURM job scheduler. Porting all the necessary software packages to RISC-V was relatively straightforward, and we can hence claim that there is no obstacle in exposing Monte Cimone as a computing resource in a HPC facility. However, full integration requires integrating Monte Cimone within a holistic monitoring framework. For that purpose we use the ExaMon framework [21]. In the next sub-section we describe its configuration and the measured metrics.

V. EXPERIMENTAL RESULTS

In this section, we report the characterisation of the Monte Cimone Cluster and of its software stack with the objective of assessing its maturity. In subsection V-A we focus on the software stack by compiling and running three different applications without manual optimisations. This lets us assess the available toolchains and libraries’ capability to extract the application’s performance given the new in HPC RISC-V ISA. Finally, we will then focus in subsection V-B one the power characterisation of one compute node.

A. Application performance

Considering the peak theoretical value of 1.0 GFLOP/s/core, inferred from the micro-architecture specification [17], leading to a 4.0 GFLOP/s peak value for a single chip, the upstream HPL [22] benchmark (built on top of the software stack shown in Section IV) reached a sustained value of 1.86 ± 0.04 GFLOP/s on a single node (on a N=40704 and NB=192 HPL configuration and a total runtime of 24105 ± 587 s); this amounts to 46.5% of the theoretical peak, a result we deem to be promising considering the upstream, unmodified software stack used in this phase. The same experiment, run on both the Marconi100 [5] system at Cineca and the Armida [6] system at E4 using the same upstream software stack (and no vendor libraries) with the same MPI topology of 1 MPI task per physical core attained 59.7% and 65.79% of a single node’s CPU-only theoretical peak respectively, a result that is comparable to what we observed on Monte Cimone. The same HPL configuration has been used to carry out a

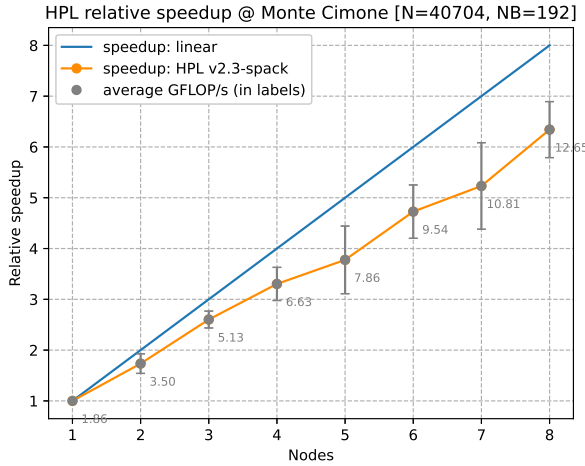


Fig. 2. HPL strong scaling tests on Monte Cimone. Average attained throughput values are shown in labels. Standard deviations are calculated on 10 repetitions.

Monte Cimone full-machine benchmark experiment leveraging the 1 Gb/s network currently available, reaching a sustained value of 12.65 ± 0.52 GFLOP/s using all of the eight nodes (with a total runtime of 3548 ± 136 s); this amounts to 39.5% of the entire machine’s theoretical peak and to 85% of the extrapolated attainable peak in case of perfect linear scaling from the single-node case. Relative speedup obtained during the HPL strong scaling experiment are shown in Figure 2. Again, we consider these results to be promising and deserving both further optimization on the software side and tuning (or technology upgrade) on the interconnect side.

The STREAM [23] benchmark has been used to measure the attainable memory bandwidth on a single node. Out of the peak 7760 MB/s [17], a 4-thread experiment measured the values shown in Table I. Being the node a UMA system, no topology configuration had to be taken into account. We consider the results attained via upstream, unmodified STREAM unsatisfactory: the results on Monte Cimone show an attained bandwidth of no more than 15.5% of the available peak bandwidth. The same experiment involving an upstream, unoptimized STREAM benchmark ran on both Marconi100 [5] and Armida [6] (using the same topology with 1 OpenMP thread per physical core) attained 48.2% and 63.21% of the peak bandwidth respectively, suggesting that a result higher than the lower quartile should be easily attained with little to no effort. This observation is worth of further experimentation, in particular:

(i) the L2 prefetcher provided by the micro-architecture [17], being able of tracking up to eight streams per core, should be perfectly capable of reducing the gap between the two experiments shown in Table I (DDR-bound and L2-bound) given the large degree of spatial and temporal locality shown by the STREAM memory access patterns. Further analysis is needed to understand how the prefetcher is currently operating and the modifications needed to leverage it properly; (ii) the overall data size used by STREAM is currently limited by the RISC-V code model. The median code model used by RV64

TABLE I
STREAM, 4 THREADS

Test	STREAM.DDR	STREAM.L2
	1945.5 MiB [MB/s]	1.1 MiB [MB/s]
copy	1206 ± 3.26	7079 ± 2.11
scale	1025 ± 4.94	3558 ± 3.72
add	1124 ± 4.93	4380 ± 3.72
triad	1122 ± 5.63	4365 ± 3.56

requires that every linked symbol resides within a $\pm 2GiB$ range from the pc register [17], [24]. Since the upstream, unmodified STREAM benchmark uses statically-sized data arrays in a single translation unit preventing the linker to perform *relaxed* relocations, their overall size cannot exceed 2 GiB. Further experiments on available workarounds for the absence of a *large code model* [25] and modifications to the STREAM source itself to overcome this limitation are needed; (iii) while the architecture provides both the Zba and Zbb RISC-V bit manipulation standard extensions [17], the upstream GCC 10.3.0 toolchain isn’t capable of emitting them nor the underlying GNU as assembler (shipped with GNU Binutils 2.36.1) is able to properly assemble them. Experiments with the latest upstream GCC version (*minimal* support for bit manipulations code generation landed in GCC 12 [26]) and the upstream development version of GNU Binutils (patches already merged [27], expected to be shipped with GNU Binutils 2.37.x) are needed to assess its impact on current STREAM measurements.

Regarding user applications, we carried out benchmarks for the quantumESPRESSO [28] suite, in particular using its LAX test driver, compiled with OpenMPI, that performs a blocked (and optionally distributed) matrix diagonalization as a benchmark representative of the full-scale application workload. For a 512^2 input matrix we obtained a value of 1.44 ± 0.05 GFLOP/s (36% of the theoretical FPU efficiency) on a single node over a total test duration of 37.40 ± 0.14 s.

B. Power characterization

We characterised the system’s power consumption under test, exploiting the set of nine power lines available on-boards with embedded shunt resistors for current monitoring.

Power consumption of a cluster node is characterised using a set of standard HPC benchmarks run on a single node with the maximum allowed parallelism. Additionally, we measured the system’s power consumption in idle, when only normal OS services and daemons are running in the background to evaluate the impact of benchmark running on power consumption. Power measurement results are collected in Table II. Figure 3 reports 8 seconds of power traces for each of the benchmark executed.

The power required by the system to run is comprised between 4.810 Watts, in idle, and 5.935 Watts when the most power-hungry computation is run. Most of the system consumption is due to the core subsystem, which absorbs 3.543 Watts on average, reaching a peak consumption of 4.097 Watts for CPU intensive benchmarks such as HPL. The

TABLE II
POWER CONSUMPTION

Line	Idle		HPL		STREAM.L2		STREAM.DDR		QE		Boot	
	[mW]	[%]	[mW]	[%]	[mW]	[%]	[mW]	[%]	[mW]	[%]	R1 [mW]	R2 [mW]
core	3075	64	4097	69	3714	68	3287	62	3825	67	984	2561
ddr_soc	139	3	177	3	170	3	232	4	176	3	59	197
io	20	0	20	0	20	0	20	0	20	0	5	20
pll	1	0	1	0	1	0	1	0	1	0	0	2
pcievp	521	11	527	9	524	10	522	10	530	9	12	231
pcievph	555	12	554	9	554	10	555	10	561	10	1	395
ddr_mem	404	8	440	7	401	7	592	11	434	8	275	467
ddr_pll	28	1	28	1	28	1	28	1	28	1	0	29
ddr_vpp	67	1	90	2	73	1	98	2	95	2	49	122
Total	4810	100	5935	100	5486	100	5336	100	5670	100	1385	4024

results show two more main sources of power consumption. i) The PCIe subsystem consistently requires 1 Watt, roughly 20% of system consumption, even if nothing is attached to the HiFive Unmatched PCIe connector. ii) DDR4 memory requires between 0.638 Watts when the system is idle and 0.950 Watts when the STREAM benchmark is run with a data size sufficient to disrupt L2 data locality. In general, DDR memory subsystem power consumption sits between 12% and 18% of the overall. The PLL subsystem and the IO interfaces together stand below 1% of the overall consumption for the tested workloads.

Figure 4 reports 80 seconds of power traces measured during the boot process. It is interesting to note a region of power consumption ($4s < t < 10s$) at which the core complex is powered on, but PLL (reported in yellow) is not active yet, we call these regions, *R1*. The average power consumption of the core complex in that region is 0.984 Watt.

As soon as the PLL activates, the power consumption jumps to a value of 2.561 Watts (*R2*) which increases to the value of 3.082 Watts, comparable with idle power for $t > 40s$ *R3*. These three regions are of interest as they allow us to estimate the three components of the power consumption, which are hard to extract from a commercial off-the-shelf device without complex laboratory equipment. As in region *R1*, only the power supply but no clock is applied to the core complex, which is consuming only leakage power, which accounts for 32% of the idle power. In region *R2*, the clock is propagated to the core complex, but the operating system is not yet loaded, memory is initialising, and boot-loader tasks are ongoing. This power consumption accounts mainly for the clock tree and core's dynamic power. In region *R3*, the operating system is executing, but no active workload is in execution.

We can thus conclude that the operating system power accounts for the gap between *R3* and Idle power (3.072 Watts) and *R2* power consumption (2.561 Watts), which is (0.514 Watts) the 17% of the idle power. Conversely the difference between *R2* and *R1* accounts for the dynamic and clock tree power, which is 1.577 Watts equal to the 51% of the core idle power. By focusing to the DDR subsystem (ddr_mem) we can make the similar consideration having in *R1* 0.275 Watts of leakage power, which is the 68% of their idle power

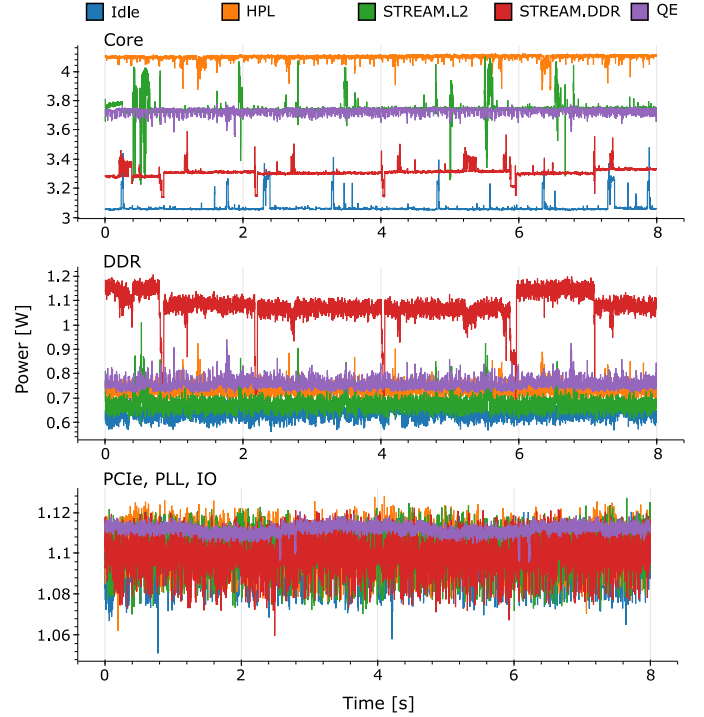


Fig. 3. Snapshot of the power consumption of the core (top), the DDR (middle) and the PCIe, PLL and IO subsystems (bottom). The traces are obtained observing power consumption for 8 seconds during benchmark run and averaging raw data using 1 ms windows.

the remaining part 32% is expected to be self-refresh and O.S. accesses for house keeping during O.S. idle period.

VI. CONCLUSIONS

In this manuscript we presented Monte Cimone : To the best of our knowledge, this is the first RISC-V cluster which is fully operational and supports a baseline HPC software stack, proving the maturity of the RISC-V ISA and the first generation of commercially available RISC-V components. We also evaluated the support for Infiniband network adapters which are recognised by the system, but are not yet capable to support RDMA communication.

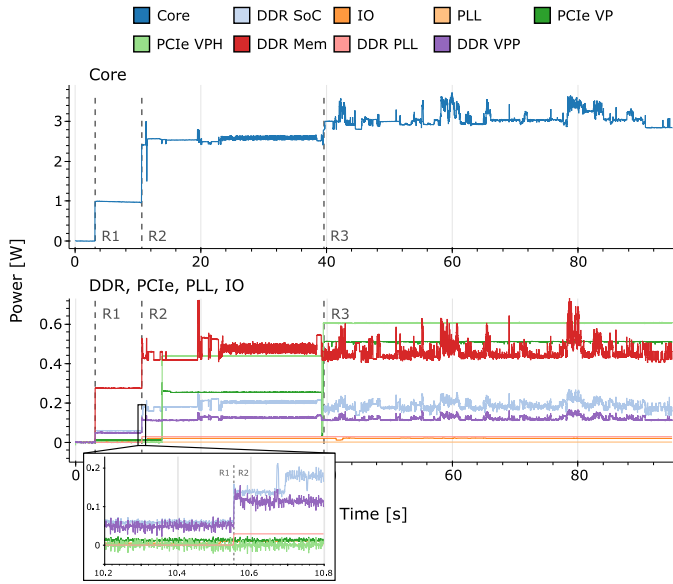


Fig. 4. Power consumption for the Core (top), DDR, PCIe, PLL and IO (bottom) subsystems during system boot. Boot phases: power-on (R1), bootloader (R2), O.S. boot (R3). The Figure also shows the detail of PLL activation.

We characterised in detail the power consumption of the SiFive Freedom U740 SoC for different workloads, measuring 4.81W in idle, with 64% due to core power (32% of leakage power, 51% dynamic and clock tree power and 17% by the O.S. workload), 13% related to DDR and 23% to the PCI subsystem. The power consumption increases to 5.935W under CPU intensive workloads.

Future work will focus on improving the software stack to achieve higher memory utilisation (i), to implement dynamic power and thermal management (ii), overcome the limitation in the Infiniband support (iv), extend Monte Cimone with PCIe RISC-V based accelerators (v).

VII. ACKNOWLEDGMENTS

The study has been partially supported by the PNRR National Centre for HPC, Big Data and Quantum Computing and the following EuroHPC JU projects: the european-project-initiative EPI-SGA2 (FPA 800928, g.a. 101036168), the REGALE project (g.a. 956560), the EUPEX project (g.a. 101033975).

REFERENCES

- [1] M. Malms, M. Ostasz, M. Gilliot, P. Bernier-Bruna, L. Cargemel, E. Suarez, H. Cornelius, M. Duranton, B. Koren, P. Rosse-Laurent *et al.*, "Etp4hpc's strategic research agenda for high-performance computing in europe 4," Ph.D. dissertation, European Technology Platform for High-Performance Computing (ETP4HPC), 2020.
- [2] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, 2019.
- [3] A. Dörflinger, M. Albers, and *et al.*, "A comparative survey of open-source application-class risc-v processor implementations," in *Proceedings of the 18th ACM International Conference on Computing Frontiers*, ser. CF '21, 2021, p. 12–20.
- [4] "Risc-v exchange: Cores & socs," <https://riscv.org/exchange/cores-socs>.
- [5] The marconi100 hpc system at cineca. <https://www.hpc.cineca.it/hardware/marconi100>.
- [6] "The armida hpc system at e4," <https://www.hpc.cineca.it/hardware/marconi100>.
- [7] N. Rajovic, A. Rico, F. Mantovani, and *et al.*, "The Mont-Blanc Prototype: An Alternative Approach for HPC Systems," Nov. 2016, pp. 444–455.
- [8] M. Sato, Y. Ishikawa, and *et al.*, "Co-design for A64FX manycore processor and "Fugaku"," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '20, Nov. 2020, pp. 1–15.
- [9] "Risc-v targets data centers," <https://semiengineering.com/risc-v-targets-data-center/>.
- [10] "Semidynamics high bandwidth risc-v ip cores," <https://semidynamics.com>.
- [11] "Esperanto.ai," <https://esperanto.ai>.
- [12] F. Zaruba, F. Schuiki, and L. Benini, "Manticore: A 4096-core risc-v chiplet architecture for ultraefficient floating-point computing," *IEEE Micro*, vol. 41, no. 2, pp. 36–42, 2021.
- [13] C. C. Chen, X. Xiang, and *et al.*, "Xuantie-910: A commercial multi-core 12-stage pipeline out-of-order 64-bit high performance risc-v processor with vector extension : Industrial product," in *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, 2020, pp. 52–64.
- [14] C. Schmidt, J. Wright, and *et al.*, "4.3 an eight-core 1.44ghz risc-v vector machine in 16nm finfet," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64, 2021, pp. 58–60.
- [15] "European processor initiative," <https://www.european-processor-initiative.eu>.
- [16] "Special interest group: High-performance computing (hpc)," <https://lists.riscv.org/g/sig-hpc>.
- [17] SiFive u74-MC core complex manual. https://sifive.cdn.prismic.io/sifive/cde638d4-1346-4bc2-9724-17e6acf0edd0_u74mc_core_complex_manual_21G2.pdf.
- [18] T. Gamblin, M. LeGendre, and *et al.*, "The spack package manager: bringing order to HPC software chaos," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, pp. 1–12.
- [19] J. L. Furlani, "Modules : Providing a flexible user environment," 1991.
- [20] M. Culpo, G. Becker, and *et al.*, "archspec: A library for detecting, labeling, and reasoning about microarchitectures," in *2020 2nd International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)*. IEEE, pp. 45–52.
- [21] A. Bartolini, F. Beneventi, and *et al.*, "Paving the Way Toward Energy-Aware and Automated Datacentre," in *Proceedings of the 48th International Conference on Parallel Processing: Workshops*, ser. ICPP 2019. New York, NY, USA: ACM, 2019, pp. 8:1–8:8.
- [22] A. Petitet, R. Whaley, and *et al.*, "Hpl – a portable implementation of the high-performance linpack benchmark for distributed-memory computers," 01 2008.
- [23] J. D. McCalpin, "Memory bandwidth and machine balance in current high performance computers," *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter*, pp. 19–25, Dec. 1995.
- [24] R.-V. International. RISC-v ABIs specification. <https://wiki.riscv.org/display/TECH/GitHub+Repo+Map>.
- [25] SiFive. RISC-v large code model software workaround. <https://www.sifive.com/documentation>.
- [26] RISC-v: Minimal support of bitmanip instructions. <https://gcc.gnu.org/git/?p=gcc.git;a=commit;h=149e217033f01410a9783c5cb2d020cf8334ae4c>.
- [27] RISC-v: Add support for Zbs instructions. <https://sourceware.org/git/?p=binutils-gdb.git;a=commit;h=9455c91957590ca6d4520cfe0955f9f9f1349f82>.
- [28] P. Giannozzi, O. Baseggio, and *et al.*, "Quantum espresso toward the exascale," *The Journal of Chemical Physics*, vol. 152, no. 15, p. 154105, 2020.