A European Roadmap to Leverage RISC-V in Space Applications

Furano, Gianluca; Di Mascio, S.; Menicucci, A.; Monteleone, Claudio

DOI
10.1109/AERO53065.2022.9843361

Publication date
2022

Document Version
Final published version

Published in
2022 IEEE Aerospace Conference, AERO 2022

A European Roadmap to Leverage RISC-V in Space Applications

Gianluca Furano
European Space Agency
Keplerlaan 1, 2201AZ
Noordwijk, The Netherlands
gianluca.furano@esa.int

Stefano Di Mascio
Delft University of Technology
Department of Space Engineering
Delft, The Netherlands
s.dimascio@tudelft.nl

Alessandra Menicucci
Delft University of Technology
Department of Space Engineering
Delft, The Netherlands
a.menicucci@tudelft.nl

Claudio Monteleone
European Space Agency
Keplerlaan 1, 2201AZ
Noordwijk, The Netherlands
claudio.monteleone@esa.int

Abstract— **RISC-V is an open and modular Instruction Set Architecture(ISA) which is rapidly growing in popularity in terrestrial applications. This paper presents the place in future space embedded systems ESA's roadmap for RISC-V based processors. In order to satisfy different applications with contrasting requirements in satellite data systems, four different types of processors are identified: 1) General-Purpose (GP) processors for payloads 2) main platform On-Board Computers (OBCs) controllers 3) low-area/low-power microcontrollers (uCs), 4) enhanced payload processors with support for Artificial Intelligence (AI). We also describe the state of the art of the RISC-V software ecosystem, including the currently available hardware platforms, with a focus on developments for space applications and what has already been done in the European Space Industry. Finally, planned activities are presented, with a focus on the role of the European ecosystem.**

TABLE OF CONTENTS

1. INTRODUCTION

Traditionally processors employed in space were based on Instruction Set Architectures (ISAs) specifically designed for airborne computers (e.g. MIL-STD-1750 [1]). With the introduction of proprietary commercial ISAs like x86, MIPS and PowerPC the space sector could rely on software ecosystems and developments from the commercial field users base. This process followed the gradual demise of the specialized space electronic industry after the Apollo and Shuttle programs. In the nineties, the European Space Agency chose SPARC for its ERC32 and LEON series of rad-hard processors, as it was the only solution available at that time providing both openness and a large user base for software support. Currently the European Space Industry (and a large part of the worldwide space community) is using LEON-based System-on-Chips (SoCs) in all ongoing and planned missions [2]. Full rad-hard processors (where hardening is both in silicon process and in the microarchitecture design) typically lag more than a decade behind their commercial counterparts in terms of performance and the gap is widening every year [1]. This is mainly due to larger technology nodes, use of Rad Hard By Design (RHBD) cells, and the long qualification process for space-grade components (it took ESA almost a decade to qualify the new GR740 LEON4 processor [3]). Furthermore, the risk-averse procurement behavior of the space industry tends to advantage components and technologies which already have a history of proven spaceflight, with replacement of units that is more obsolescence-driven rather than progress-driven.

As maintaining a software ecosystem is by far the biggest industrial challenge in the development of a processor [4], the main advantage that comes with the selection of a mature commercial ISA is the reuse of the software ecosystem and of the heritage and experience of a much larger community working on terrestrial applications. The advantage is even stronger when this community is built around an open and free ISA, as everyone can contribute to advancing the state of the art of the ecosystem and there are few (commercial, legal) barriers in the developments of Intellectual Property (IP) cores and in their modification (for example, to introduce microarchitecture-level RHBD features, a use case very specific to space applications). For this reason, wide adoption of an ISA by industry and academia is crucial. These considerations were critical for the adoption of SPARC by ESA in the nineties. The rise of mobile embedded applications, with their novel requirements in terms of energy efficiency have led to the success of proprietary, vendor-tailored ISAs from ARM. Nevertheless, RISC-V (an open and free ISA like SPARC) has risen in popularity in recent years [5].

The reason behind this is that many developers are concerned about monopolistic positions in the embedded market, as ISA owners protect their IP by not allowing freely available implementations (thus ultimately preventing reuse) and by not allowing free-market competition from many IP core designers. Government-level actions on export of processor IPs have further exacerbated these concerns and increased the attention on open ISAs from defense and security markets [6]. The adoption of a popular, free and open ISA can thus lead to shorter time to market and lower costs from reuse. The space industry can spin-in developments from other industries, focusing limited resources mainly on improvements concerning specific needs in space applications and without wasting efforts on other activities. Furthermore, the adoption of an open and free ISA is not stopping companies from

978-1-6654-3760-8/22/$31.00 ©2022 IEEE

2022 IEEE Aerospace Conference (AERO) | 978-1-6654-3760-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/AERO53065.2022.9843361

Table 1. RISC-V User-level standard extensions.

Subset	Name
Integer	I
Integer Multiplication and Division	M
Atomics	A
Single-Precision Floating-Point	F
Double-Precision Floating-Point	D
Quad-Precision Floating-Point	Q
Decimal Floating-Point	L
16-bit Compressed Instructions	C
Bit Manipulation	B
Dynamic Languages	J
Transactional Memory	T
Packed-SIMD Extensions	P
Vector Extensions	V
User-Level Interrupts	N

having proprietary or distribution controlled IPs, but this happens in a distributed industrial environment that protects everybody from monopoly and single sources.

The RISC-V ISA

RISC-V was originally developed by UC Berkeley to support computer architecture research and education oriented at hardware implementations [7]. The RISC-V manual is structured in two volumes, one for the user-level ISA and the other describing the privileged architecture.

RISC-V allows both standard extensions (see Table 1) and extensions defined outside the specifications (non-standard). The RISC-V ISA is defined as a base integer (I) ISA plus optional extensions. A subset of the integer base (E) can optionally be implemented for minimal implementations, with 16 general purpose registers instead of 32. The standard defines a ”general” subset (G) (comprising the IMAFD subsets) as the set of extensions required for general-purpose computing systems.

The privileged architecture contains three privilege levels: User (U), Supervisor (S) and Machine (M) mode. An implementation can employ just the user mode, the user and machine mode (when security is a concern) or all of the three modes for implementations targeting Linux-like Operating Systems (OSs).

Outline

In Sec. 2 the types of processors required in space embedded systems are introduced. Then, in Sec. 3, the state of the art of the RISC-V ecosystem is described and the relevance of each development to space embedded systems is highlighted. In Sec. 4 future activities concerning RISC-V in space systems are described. Finally, Sec. 5 draws conclusions.

2. Processors for Space Data Systems

In [1], the types of processors required in space data systems have been identified. They can be classified into General-Purpose (GP) processors, On-Board Computer (OBC) controllers, microcontrollers (uCs), and processors for on-board

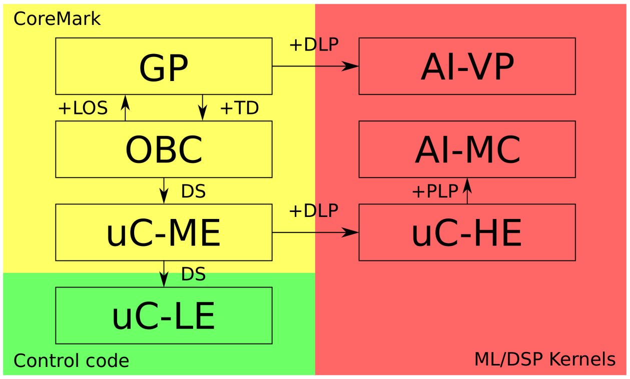

Figure 1. Identified profiles of processors for space applications, main differences among them and defining benchmarks. LOS stands for ”Support for Linux-like OSs”, DS stands for ”Downsized”, TD for ”Time Determinism”.

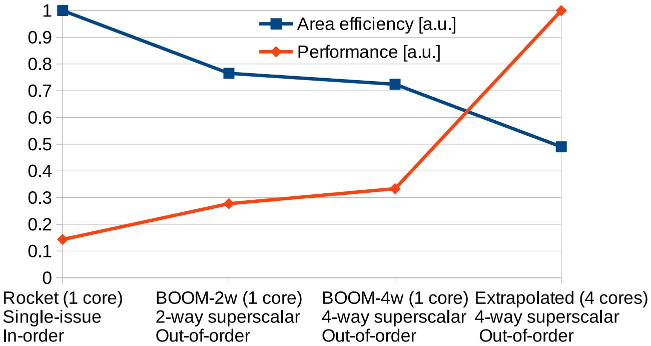

Figure 2. Comparison of different GP processors (data is from [1]). Data is normalized to the achieved peak.

Artificial Intelligence (AI) applications. Fig. 1 summarizes the relations between these profiles.

GP processors

GP processors for space applications are typically used in payloads, where the constraint of hard real-time can be loosened and a soft real-time approach can be adopted instead. In this case it is possible to use largely supported Linux-like OSs for GP computing, greatly increasing software modules reuse and enabling complex applications (running third-party code in protected mode being one of the most requested features, lately). GP processors employ a Memory Management Unit to translate page-based virtual addresses to physical addresses, as required by Linux-like OSs. Caches store recently-accessed data and other contiguous memory locations, to reduce the time spent accessing the main memory. Time-determinism is also penalized by speculation, typically required to efficiently use the resources in deep pipelines with large amounts of Instruction-Level Parallelism (ILP). Processor-Level Parallelism (PLP) is typically exploited employing a Symmetric MultiProcessing (SMP) approach, based on an OS scheduler assigning a thread for each core.

In Fig. 2 can be seen as, when performance is increased with

ILP, speculation and out-of-order execution, the efficiency of utilization of the resources of the processor decreases. The final step, applying SMP PLP with Level2 (L2) caches typically provides a large increase in terms of performance (around x3 for four cores [1]), although this comes at a decrease of area efficiency of 1.5x (assuming that this implementation with L2 cache requires 4.6 times the area of a single core).

OBCs

OBCs mainly deal with real-time tasks, leveraging simple and efficient Real Time Operating Systems (RTOS) like Real-Time Executive for Multiprocessor Systems (RTEMS) [8]. Even the newest ones are based on single-core single-issue in-order microarchitecutres like the LEON2 and the LEON3, because higher performance processors employ microarchitectural features (e.g. superscalar execution, out-of-order execution, speculation, multicore, several level of caches) that poses challenges to the development and verification of real-time software. For this reason, processors for OBCs can be seen as low-end GP processors (e.g. Rocket in Fig. 2).

Therefore, a RISC-V substitute of the LEON (when used as the main OBC processor) would require a similar microarchitecture, achieving similar performance. The drawback of giving up long fight heritage and large amount of OBC software legacy for the LEON processors in space applications shall be considered, and compensated by a ‘better’ software development toolset: a RISC-V substitute could leverage a larger software ecosystem in the future for new developments, building on a much larger user base for (e.g.) the maintenance of the software toolchain.

Given the criticality of the tasks carried out by OBCs, typically Fault-Tolerant (FT) IP cores are employed, even on radiation hardened silicon technology. FT IP cores usually employ Triple Modular Redundancy (TMR) at flip-flop level with separate clock trees, Error Detection And Correction (EDAC) codes on caches and on the register file [9]. These solutions cause penalties in terms of area, frequency and power compared to non-FT versions of the same core [10]. For instance, a LEON2FT occupies 40% more area compared to the non-FT version, while the penalty in frequency is relatively small (-7.4%) [9]. It should be noted that IP cores for terrestrial applications started to include similar features in recent years, as larger cache memories with smaller technology nodes tend to have considerable Single Event Rates (SERs) even in terrestrial applications.

Microcontrollers

The performance of space data systems can be increased by distributing software tasks for data acquisition/processing and simple control applications in spacecraft subsystems using remote terminal units (RTUs). For these applications, small, low-power, mostly bare metal (so, without OSs) implementations are preferred. In [1], three different kinds of uCs were identified:

- **Low End** (LE): These implementations are optimized for ”pure control code” [11] , as opposed to the CoreMark (based on arithmetic calculations). In [7], the use of RV32E instead of RV32I is expected to save 25% of area. Removing also the M extension (the hardware multiplier and divider) saves even more (38.62% [1]). Typically, also the C extension is employed to reduce code size by a 25%-30% [7] to fit into limited chip-level Non-Volatile Random Access Memories (NVRAMs). Implementation of the planned bit manipulation (B) extension reduces code size considerably and increases performance of control code, which typically has to clear or set a certain subset of bits in a peripheral register.
- **Mid End** (ME): These implementations typically achieve high area efficiency for CoreMark-like workloads, being essentially a down-scaled version of a GP processor. However, to increase time-determinism (and to cope with limited silicon resources) caching and speculation are typically not present.
- **High End** (HE): Obtained extending a ME microcontroller with Data-Level Parallelism (DLP) to achieve high-performance and efficiency when dealing with compute-intensive workload, as data-parallel processors execute an instruction on more elements of an array at the same time.

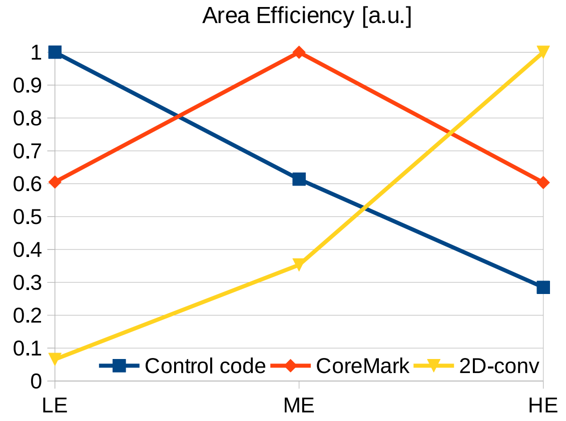

Figure 3. Comparison of the three profiles of microcontrollers targeting different applications (data is from [11], where LE is Zero-riscy, ME is Micro-riscy and HE is RI5CY). For each benchmark, data is normalized to the achieved peak. 2D-conv stands for ”2D-convolution kernel”.

The role of each microcontroller profile is clear from Fig. 3, where the area efficiency (taken from data in [11]) for a microcontroller of each category is shown for different benchmarks (representing different target applications).

Processors for On-Board Artificial Intelligence

There is a huge interest in using AI/ML application on board for several type of high and low computational load application, in space [12], [13]. An AI generated inference application (being tensor processing for actual learning currently out-of-scope for ‘on the edge’ space use) typically implies a large amount of matrix calculations on a large set of data [14]. GP processors perform poorly with such compute-intensive data-hungry simple-flow programs, as they are designed to speed up flow control (speculation) and data access with relatively small Level1 (L1) data caches (usually around 16KiB) compared to the amount of data employed by AI algorithms (a 64x64 matrix multiplication of 32-bit elements involves 48 KiB of data).

The tools available to increase performance for AI workloads are DLP and PLP [1]. There are two main approaches to obtain a processor for AI: replicating massively a simple core (i.e. a uC-ME or uC-LE) [15] or implementing DLP in one or a few complex cores (GP processors) [16]. In the second case there is a trend to replace packed-SIMD ISA extensions with AVL vector extensions [14]. The RISC-V 'V' extension aims to be flexible and reconfigurable for different data types and sizes on the run, with the goal to support both implicit auto-vectorization in (OpenMP) and explicit SPMD (OpenCL).

Processors for AI are typically benchmarked measuring the number of operations per second for several kernels relevant to Machine Learning (ML) and Digital Signal Processing (DSP) algorithms (e.g. axpy and gemm).

3. State of the Art of the RISC-V Ecosystem

When ESA started to investigate the use of the RISC-V ISA in 2018, there was widespread skepticism in the European space industry concerning the fact that RISC-V could catch-up with the most popular proprietary ISAs in terms of performance, adoption and ease of use, especially because of its limited software ecosystem at the time. The situation nowadays is quite different, and it gets better by the day. In the following subsections we mention, to the best of our knowledge, the most relevant facts about the state of the art of RISC-V. However, it is by no means possible for us to be comprehensive because of the wide range of developments based on the RISC-V ISA.

Software Ecosystem

The status of the RISC-V software ecosystem is captured in [17]. As for any other ISA, the bases of the software ecosystem are compilers and debuggers. The most popular C and C++ compilers are available, like GCC and Clang/LLVM [18]. Furthermore, the most popular command-line debugger, GDB, supports RISC-V [19]. Also Integrated Development Environments (IDEs) with Graphical User Interface (GUI) like Eclipse are available for some RISC-V boards [20].

Another important part of the software ecosystem is which OSs are available for processors based on a certain ISA. Several OSs have been ported to RISC-V:

- Real-Time Operating Systems (RTOSs): Free RTOS [21], Zephyr [22].
- OSs with focus on security: sel4 [23]
- Linux-based OSs: The Linux kernel has been ported to RISC-V and several distributions are available (Fedora, Debian, OpenMandriva, openSUSE, Gentoo, Ubuntu).

During the development of RISC-V processors and SoCs, also ISA emulators come at handy as reference models for hardware implementations and as software development tools. Beside a Register-Transfer Level (RTL) simulation of a reference implementation, that provides cycle-accurate results, there are instruction-accurate emulators like Spike (considered the "golden reference" of the RISC-V ISA during hardware development) and functional emulators like QEMU, which are very handy for software development as they require less computation efforts (hence development time), although they run a translation of the code in the native instructions of the host machine, losing instruction-by-instruction tracing [24].

Finally, automatic tools to translate machine learning algorithms with reasonabbly low memory footprint (TFLite) are available for some RISC-V boards [25].

Processors for terrestrial applications

The availability of a software ecosystem supported by a large open community ignited an unprecedented amount of open developments, with several announcements and/or releases of open-source implementations.

- UC Berkeley and SiFive have released an open-source SoC generator called Rocket Chip to automatically generate synthesizable Verilog. Building on the Rocket Chip generator, SiFive has released several IPs, components and development boards. An high-end board is available (HiFive Unmatched), based on a FU740-C000 Application-Specific Integrated Circuit (ASIC). This component contains a dual-issue in-order quad-core 64-bit SiFive U74 operating at up to 1.2 GHz and supports the RV64G [26]. Also HiFive1, a low-end board based on a FE310-G002 ASIC (containing a SiFive E31 clocked up to 320 MHz) [27] is available. A processor generated from the Rocket is also employed in the Polarfire SoC for terrestrial applications [28].
- ETH Zurich and University of Bologna open-sourced the PULP platform [15]. It contains several IP cores, ranging from a simple 2-stage 32-bit core (RV32EC) to a 6-stage Linux-capable core (RV64GC) with caches, and comprises several SoC architectures. One of the SoC architectures, PULP, is an example of AI-MC, an 8-core array of RI5CY cores plus a Zero-riscy for control operations. As the standard RISC-V ISA does not contain extensions to speedup compute-intensive calculations, RI5CY was extended with non-standard instructions for bit manipulation, auto-incremental load and store, Multiply-Accumulate (MAC), hardware looping and Single Instruction Multiple Data (SIMD) operations [1]. Based on PULP, the GAP8 [29] has been taped out and the GAPuino board is available. Also an improved version, the GAP9, has been announced [30].
- Western Digital has developed and open-sourced four different configurations of the SweRV processor, each of them targeting different performances: EH1 (5.7 CoreMark/MHz), single-issue EH2 (6 CoreMark/MHz), dual-issue EH2 (7.8 CoreMark/MHz), and EL2 (4.3 CoreMark/MHz). These cores will be used in future Western Digital products [31].
- Alibaba presented Xuantie-910 (7.1 CoreMark/MHz), a 16-core RV64GCV processor, featuring custom extensions for arithmetic operation, bit manipulation, load and store and cache operations [32]. Each core has a 12-stage out-of-order multi-issue pipeline achieving a maximum clock frequency 2.5 GHz on TSMC 12nm FinFET technology. Xuantie Field-Programmable Gate Array (FPGA) implementations have been deployed in the Alibaba data centers for applications-specific acceleration (e.g., blockchain transactions) [32].

Processors for space applications

The NOEL-V is a synthesizable Very High-Speed Integrated Circuit Hardware Description Language (VHDL) model of a RISC-V processor [33]. As shown in Table 2, the NOEL-V can be configured to satisfy most of the profiles identified in Sec. 2:

While NOEL-V configurations cover several profiles in the area of microcontrollers/OBC (TIN32-MIN32) and General Purpose processors(GPP32/64 and HPP32/64), Cobham Gaisler is still currently investigating ISA extensions like the P and the V to enable on-board decision making [14].

Furthermore, SiFive also provided IP cores for the RISC-V ecosystem by Microsemi (Mi-V) for their flash-based line of FPGAs, which comprises the radiation-tolerant RTG4 and Polarfile FPGAs [35].

Table 2. Currently available NOEL-V configurations [34]. SI stands for Single-Issue and DI for Dual-Issue.

Conf.	ISA	ILP	Cache	MMU	PMP
TIN32	32IM	SI	no	no	no
MIN32	32IMAC	SI	yes	no	yes
MIN64	64IMAC	SI	yes	no	yes
GPP32	32GCH	SI/DI	yes	yes	yes
GPP64	64GCH	SI/DI	yes	yes	yes
HPP32	32GCH	DI	yes	yes	yes
HPP64	64GCH	DI	yes	yes	yes

4. Future Developments

RISC-V is one of the most popular choices for new developments in space embedded systems. For instance, a hardware/software platform combining the XtratuM hypervisor and NOEL-V is being developed, to meet safety and validation requirements (e.g. time predictability) of space systems [36].

Several ESA-led activities are in preparation to perform in orbit Demonstration of several classes of RISC-V processors. In the short term, as logical, they will focus on soft core processors on FPGA, on longer term there is the idea of producing ASICs for the categories described in Sec. 2 that will gradually replace LEON-based ones. In detail:

- Launch of first FPGA demonstrator for RISC-V NOEL core is foreseen in 1Q2023 on the Trisat-R satellite (First CubeSat in Medium Earth Orbit, 6000 km altitude). RISC-V 32-bit processor (RV32IMAF) will be hosted in Microsemi PolarFire FPGA and will run self check experiments.
- Test Chip in UDSM ($\leq$ 28 nm) is in preparation, with ESA tender in open competition in preparation (Q1/2022). The objective is to select and tailor a RISC-V IP core and associated peripherals, implement and verify chip design in Hardware Description Language (HDL), select ASIC technology (ST 28FDSOI, GF-22FDX, or smaller), define and implement the radiation hardening(-by-design) concept.

We foresee space qualified RISC-V based SoCs to be available already before 2025, probably starting from the low end (32-bit microcontrollers) performance class.

IP Cores for FPGAs

In the space embedded systems domain we are now witnessing a very novel trend. While “classical” processing ASICs keep suffering of the usual, widening, performance gap between ‘terrestrial’ processors and space grade ones (as discussed in [2]), on the counter recent developments especially in SRAM-based FPGAs have brought ‘edge’ technologies for space use, providing a possible quantum leap for processing in space. One of the most notable novelties is the availability of Xilinx Versal Adaptive Compute Acceleration Platforms (ACAPs) [37] also in space-grade packages, which deliver previously impossible application and system-level performance for space edge computing applications. Being based on 7nm node, they are a big step from previous 65 and 45 nm space grade “pure” FPGAs since they also include heterogeneous compute engines with a breadth of hardened memory and interfacing technologies for far superior performance/watt targets. Especially for payload data processing tasks, these devices will likely replace any future dedicated ASSP and call for availability of sophisticated, RHBD, processor IP cores for specific applications.

5. Conclusion

Most space-grade microprocessors today implement the SPARC instruction set architecture, and although this was initially devised in Europe, its success is now global. Adhering to a standard architecture facilitates software reuse and accumulation of design and engineering knowledge. SPARC was selected as a preferred architecture by the European Space Agency in the 1990’s and today several implementations of SPARC microprocessors exist in devices such as SCOC3 (Airbus Defense and Space), EPICA-NEXT (Thales Alenia Space), AT697 (Atmel), UT699/UT700 (CAES), GR712RC and GR740 (CAES). The dominance of, in particular ARM-developed ISA, in the commercial sector has fuelled the need for a new modern, open, and unrestricted instruction set architecture. In recent years this has surfaced as part of the RISC-V initiative that now shows a major momentum in adoption by companies such as Google, AMD and HPE (Hewlett Packard Enterprise).

The RISC-V architecture offers several potential benefits for end users:

- Attractive license model (BSD open source license), enabling students, end users and auditors to become familiar with the details of the chip design. Eventually this will lead to enhancements and better products allowing dedicated RHBD techniques to be implemented in space IP cores.
- Strong academic and industrial support which confirms that the RISC-V architecture is a good potential candidate for future processing alternatives.
- Low silicon floor space requirements resulting in more logic resources being available to peripheral functions and interfaces. This leads to higher integration density which is a significant advantage in spacecraft applications. In the end an improved system will be obtained, exhibiting higher reliability at a lower cost.
- The slightly larger memory size requirement typically imposed by RISC architectures can be easily compensated by larger memory capacities becoming nowadays available for use in spacecraft avionics equipment. In addition the expected gain on Silicon floor space (as stated in the previous bullet) will also compensate for this minor loss.

References

[1] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, “Leveraging the Openness and Modularity of RISC-V in Space,” *Journal of Aerospace Information Systems*, vol. 16, no. 11, pp. 454–472, 2019.

[2] G. Furano and A. Menicucci, *Roadmap for on-board processing and data handling systems in space*. Springer International Publishing, 8 2017, pp. 253–281.

[3] J. Andersson, M. Hjorth, F. Johansson, and S. Habinc, “LEON Processor Devices for Space Missions: First 20 Years of LEON in Space,” in *2017 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, 2017, pp. 136–141.

[4] K. Asanović and D. A. Patterson, “Instruction sets should be free: The case for RISC-V,” *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2014-146*, 2014.

[5] C. H.-M. Su, "The Rise of RISC-V from Edge to Cloud," in *2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2020, pp. 1–1.

[6] William Li, "RISC-V to Shake up $8.6-Billion Semiconductor IP Market," https://www.counterpointresearch.com/riscv-semiconductor-ip-market-2025/, Accessed: 2021-09-29.

[7] A. Waterman and K. Asanovic, "The RISC-V instruction set manual-volume I: User-level ISA-document version 2.2," *RISC-V Foundation*, May 2017.

[8] J. Seronie-Vivien and C. Cantenot, "RTEMS operating system qualification," *Proceedings of the DASIA 2005 Conference, Edinburgh, Scotland, June 2005.*, 2005.

[9] J. Gaisler, "A portable and fault-tolerant microprocessor based on the SPARC v8 architecture," in *International Conference on Dependable Systems and Networks. Proceedings.* IEEE, 2002, pp. 409–415.

[10] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, "Open-source IP cores for space: A processor-level perspective on soft errors in the RISC-V era," *Computer Science Review*, vol. 39, p. 100349, 2021.

[11] P. D. Schiavone, F. Conti, D. Rossi, M. Gautschi, A. Pullini, E. Flamand, and L. Benini, "Slow and steady wins the race? A comparison of ultra-low-power RISC-V cores for Internet-of-Things applications," in *2017 27th International Symposium on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, Sep. 2017, pp. 1–8.

[12] G. Furano, G. Meoni, A. Dunne, D. Moloney, V. Ferlet-Cavrois, A. Tavoularis, J. Byrne, L. Buckley, M. Psarakis, K.-O. Voss *et al.*, "Towards the use of artificial intelligence on the edge in space systems: Challenges and opportunities," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, no. 12, pp. 44–56, 2020.

[13] G. Giuffrida, L. Fanucci, G. Meoni, M. Batič, L. Buckley, A. Dunne, C. Van Dijk, M. Esposito, J. Hefele, N. Vercruyssen *et al.*, "The ϕ-sat-1 mission: the first on-board deep neural network demonstrator for satellite earth observation," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[14] S. Di Mascio, A. Menicucci, E. Gill, G. Furano, and C. Monteleone, "On-Board Decision Making in Space with Deep Neural Networks and RISC-V Vector Processors," *Journal of Aerospace Information Systems*, vol. 18, no. 8, pp. 553–570, 2021.

[15] 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.

[16] Y. Lee, A. Waterman, R. Avizienis, H. Cook, C. Sun, V. Stojanović, and K. Asanović, "A 45nm 1.3GHz 16.7 double-precision GFLOPS/W RISC-V processor with vector accelerators," in *ESSCIRC 2014 - 40th European Solid State Circuits Conference (ESSCIRC)*, Sept 2014, pp. 199–202.

[17] "RISC-V Exchange: Available Software," "https://riscv.org/exchange/software/", Accessed: 2021-09-29.

[18] M. Poorhosseini, W. Nebel, and K. Grüttner, "A Compiler Comparison in the RISC-V Ecosystem," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, 2020, pp. 1–6.

[19] "GDB: The GNU Project Debugger," "https://sourceware.org/gdb/news/", Accessed: 2021-09-29.

[20] "Eclipse Embedded CDT - The RISC-V corner," "https://gnu-mcu-eclipse.github.io/arch/riscv/", Accessed: 2021-09-29.

[21] "Using FreeRTOS on RISC-V Microcontrollers," "https://www.freertos.org/Using-FreeRTOS-on-RISC-V.html", Accessed: 2021-09-29.

[22] "Zephyr - Supported Boards," "https://docs.zephyrproject.org/latest/boards/index.html", Accessed: 2021-09-29.

[23] RISC-V Foundation, "seL4 is verified on RISC-V!" "https://riscv.org/blog/2020/06/sel4-is-verified-on-risc-v/", Accessed: 2021-09-29.

[24] "RISC-V QEMU Part 1: Privileged ISA v1.10, HiFive1 and VirtIO," "https://www.sifive.com/blog/risc-v-qemu-part-1-privileged-isa-hifive1-virtio", Accessed: 2021-09-29.

[25] "Tensorflow Lite in Zephyr on LiteX/VexRiscv," "https://antmicro.com/blog/2019/12/tflite-in-zephyr-on-litex-vexriscv/", Accessed: 2021-09-29.

[26] SiFive, "HiFive Unmatched Datasheet," https://sifive.cdn.prismic.io/sifive/d0556df9-55c6-47a8-b0f2-4b1521546543_hifive-unmatched-datasheet.pdf, Accessed: 2021-09-29.

[27] ——, "HiFive1 Rev B Getting Started Guide," https://sifive.cdn.prismic.io/sifive/cf239fd0-ae4f-4fd8-a944-fdafb5018153_hifive1b-getting-started-guide_v1.2.pdf, Accessed: 2021-09-29.

[28] M. J. Cannizzaro, E. W. Gretok, and A. D. George, "Risc-v benchmarking for onboard sensor processing," in *2021 IEEE Space Computing Conference (SCC)*, 2021, pp. 46–59.

[29] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini, "A 64mW DNN-based Visual Navigation Engine for Autonomous Nano-Drones," 2018.

[30] GreenWaves Technologies, "GAP9 next generation processor for hearables and smart sensors," https://greenwaves-technologies.com/wp-content/uploads/2021/06/Product-Brief-GAP9-Sensors-General-V1_14_new.pdf, Accessed: 2021-09-29.

[31] Western Digital, "RISC-V: Accelerating Next-Generation Compute Requirements," "https://www.westerndigital.com/company/innovations/risc-v", Accessed: 2021-09-29.

[32] C. Chen, X. Xiang, C. Liu, Y. Shang, R. Guo, D. Liu, Y. Lu, Z. Hao, J. Luo, Z. Chen, C. Li, Y. Pu, J. Meng, X. Yan, Y. Xie, and X. Qi, "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.

[33] J. Andersson, "Development of a NOEL-V RISC-V SoC Targeting Space Applications," in *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2020, pp. 66–67.

[34] Cobham Gaisler, "NOEL-V Processor," https://www.gaisler.com/index.php/products/processors/noel-v, Accessed: 2021-09-29.

[35] "Mi-V RISC-V Ecosystem," "https://www.microsemi.com/product-directory/fpga-soc/5210-mi-v-embedded-ecosystem", Accessed: 2021-09-29.

[36] F. Gómez, M. Masmano, V. Nicolau, J. Andersson, J. Le Rhun, D. Trilla, F. Gallego, G. Cabo, and J. Abella Ferrer, "De-RISC–Dependable Real-Time Infrastructure for Safety-Critical Computer Systems," *Ada User Journal*, vol. 41, no. 2, pp. 107–112, 2020.

[37] K. Vissers, "Keynote 2: Versal: The new xilinx adaptive compute acceleration platforms (acap)," in *2018 IEEE/ACM 8th Workshop on Irregular Applications: Architectures and Algorithms (IA3)*, 2018, pp. 10–10.

BIOGRAPHY

***Gianluca Furano** PhD in Microelectronics Engineering, works in European Space Agency's Data System Division since 2003. He is in charge of operational missions as well as research and development activities and he coordinates ESA research on on-board artificial intelligence. Among Gianluca's interest in ESA are on-board data handling systems and their major components, such as space grade microprocessors and support electronics, meeting very stringent requirements in terms of radiation tolerance, reliability, availability, and safety; key avionics building blocks such as platform mass memories, remote terminal units, on-board buses and data networks; on-board and space to ground data communication protocols including protocol security aspects. Gianluca also provides support to European standardisation (CCSDS, ECSS) in areas such as telemetry, telecommand and on-board data, wireless and monitoring control interfaces. He has authored or co-authored more than 100 publications.*

***Stefano Di Mascio** obtained his MSc in Electronic Engineering from the University of Rome "Tor Vergata" in 2016, after an internship at ESA. In 2016 he joined Cobham Gaisler as a Hardware Engineer and worked on the development of Application-Specific Integrated Circuits (ASICs) and Intellectual Property (IP) cores. In 2018 he started a PhD on the use of the RISC-V ISA in space applications at Delft University of Technology in the frame of a Networking Partnering Initiative cooperation with ESA and Cobham Gaisler.*

***Alessandra Menicucci** graduated in 2000 in experimental particle physics from the University of Rome La Sapienza. In 2004 she received a PhD in Physics from the University of Rome Tor Vergata working on the on-board computer of the cosmic rays experiment PAMELA. In 2005 she was granted a Marie-Curie postdoctoral fellowship at AGFA-Gevaert. In 2006 she joined ESA in the Space Environment and Effects section. Since 2015 she is Assistant Professor in the Space System Engineering of TU Delft. Her research interests are miniaturized radiation sensors onboard of micro-satellites and the radiation hardness assurance of COTS.*

***Claudio Monteleone** received a degree in Physics in at the University of Milan, Italy. He worked as digital designer of on board computers for space applications until 2001 when he started working at the technical centre of the European Space Agency, ESTEC, located in The Netherlands. He was involved in the development and procurement of the avionics (On Board Computers, Remote Terminal Units, Mass Memories) for a number of space missions. As technical officer he contributed to the development of several rad-hard on-board computers based on the ERC32 and LEON processors families and to several digital and mixed-signal ASICs and micro-controller based system-on-a-chip.*