The Case for RISC-V in Space

Stefano Di Mascio, Alessandra Menicucci, Gianluca Furano, Claudio Monteleone and Marco Ottavi

Abstract This paper presents preliminary position on the use of the novel, free and open RISC-V Instruction Set Architecture (ISA) for on-board electronics in space. The modular nature of this ISA, the availability of a rich software ecosystem, a rapidly growing community and a pool of open-source IP cores will allow Space Industry to spin-in developments from terrestrial fields (in terms of security, artificial intelligence, support for operating systems, hardware acceleration etc.) while focusing its efforts mainly on aspects related to the specific needs of on-board electronics for space applications (e.g. fault tolerance, observability, error signaling, etc.). This will improve reuse and avoid the necessity of developments from scratch when not strategically needed, eventually increasing productivity and reducing costs. The use of an open, non proprietary ISA will allow ad-hoc design of microarchitecture-level soft error countermeasures that can greatly increase the robustness of Application Specific Standard Products (ASSP) and FPGA implementations.

1 Introduction

While open-source software has been around for decades, being the driving force behind most of the Internet and all of the top-500 supercomputers [1], hardware has not yet fully experienced the disruptive effects of openness. Nevertheless, over the last years RISC-V has risen in popularity, drawing the attention of several universities and companies previously focusing on other open and free ISAs, proprierary ISAs

S. Di Mascio · A. Menicucci
Delft University of Technology, Faculty of Aerospace Engineering, Space Systems Engineering, Delft, The Netherlands

G. Furano · C. Monteleone
European Space Agency, European Space Technology Centre, Noordwijk, The Netherlands

M. Ottavi (✉)
Department of Electronic Engineering, University of Rome Tor Vergata, Rome, Italy
e-mail: ottavi@ing.uniroma2.it

S. Saponara and A. De Gloria (eds.), *Applications in Electronics Pervading Industry, Environment and Society*, Lecture Notes in Electrical Engineering 550, https://doi.org/10.1007/978-3-030-11973-7_37

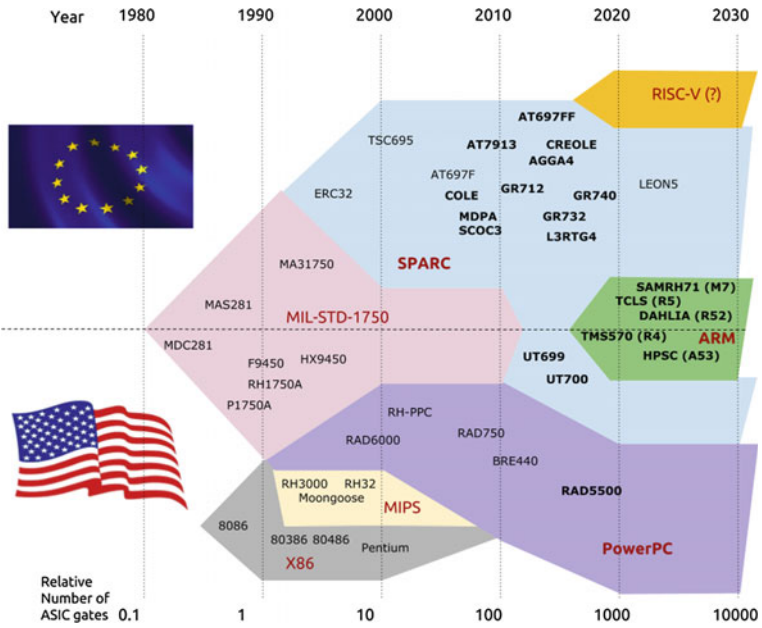

Fig. 1 Main CPU ISAs and their market share evolution in Europe and US. In bold SoCs. TCLS and DAHLIA are H2020 EU-funded research projects, while SPARC IPs and SoCs development in Europe have been largely funded by ESA

or even on ISAs designed in-house (with the big drawback of having to design and maintain a software ecosystem). LEON is a precursor applied to the space industry to what is currently happening now with RISC-V in the COTS market. Use of an open and free ISA like SPARC had compelling motivations at the time the introduction of LEON for space [2] and the code of the non-FT version is available to the public. 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 [3]. Most of the cost associated for those custom SoCs is in design, and introduction of reusable standard IP libraries allowed to simplify design complexity thus allowing all major European players to develop their own LEON-based processor and product lines (Fig. 1).

LEON was designed for a single function (processor), targeting an ASSP that came in 2009 with Atmel's AT697F but it has been increasingly being used as SoC platform since the very beginning. This generated a positive feedback with availability of IP libraries and use of commercial ones in several space grade SoCs. The availability of large rad-hard FPGAs (especially with introduction of ACTEL/Microsemi RTAX series) made the rest, since now space components and units manufacturers have a very efficient SoC platform that helps in minimizing design and SW development work, resulting in many first-silicon-good ASICs. The LEON 'ecosystem' is now mature (mostly thanks to ESA and Cobham Gaisler) and thanks to its portability and

CAD tool independence allows to support of both ASIC and FPGA technologies, taking the best of both worlds. Any adoption of a new ISA for the design of safety and mission critical processors require robustness and dependability of the whole infrastructure around the processor IP:

- Complete, easy to reproduce verification with dedicated tool set
- Clear documentation on how to use it, with reference designs
- Mature, maintained SW tools
- Robustness validated through Silicon implementation and test
- Widespread knowledge in academia and industry.

2 The RISC-V ISA

RISC-V was originally developed by UC Berkeley to support computer architecture research and education oriented at hardware implementations, as they could not find an ISA fit for such purpose. An example of the limits of previously existing free ISAs is OpenRISC, which features micro-architectural choices like branch delay slots and is not designed to be modular. DARPA funded RISC-V in its very beginning [4] and is continuing to fund other activities related to the spin-in of open-source IPs in trustable electronic systems. The reason behind this is that open-source IPs and open ISAs can reduce the resources, time, and complexity required for 'secure and trusted' custom SoC design, as detailed information about open-source IP cores can be found by inspection and ad-hoc improvements or modifications for security are much easier, avoiding the need to design everything from scratch and thus ultimately increasing reusability [5]. The Space Industry can apply much of those considerations to enhancements in fault-tolerance of existing open-source COTS IPs. RISC-V is backed also by big players of the commercial field, as companies like Google, Qualcomm, IBM, NVIDIA, Samsung and Western Digital are members of the RISC-V foundation. The reasons for this is that concerns are growing about monopolistic positions in the embedded market, as ISA owners protect their IP not allowing freely available implementations and free-market competition from other core designers, thus ultimately preventing reuse and ad-hoc designs. The adoption of a free and open ISA can thus lead to shorter time to market and lower costs from reuse.

The main feature of RISC-V is modularity. The RISC-V manual is structured in two volumes, one for the user-level ISA and the other describing the privileged architecture with three privilege levels (User, Supervisor and Machine 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 Linux-like Operating Systems (OSs). The user-level ISA is defined as a base integer (I) ISA, which must be present in any implementation, plus optional extensions to the base ISA. The integer base is restricted to a minimal set of instructions sufficient to provide a reasonable target for compilers, assemblers, linkers, and OSs (with additional supervisor-level operations), and so provides a convenient ISA and software toolchain skeleton around

which more customized processor ISAs can be built. A subset of the integer base can optionally be implemented (E) when an implementation targets small 32-bit microcontrollers, with 16 general purpose registers instead of 32. The standard defines a "general" subset (G) as the set of extension required for general purpose computing systems (e.g. this subset of the ISA is enough to run Linux from an instruction perspective). RISC-V allows both standard and non-standard extensions (defined outside the specifications). Whereas other ISAs are treated as a single entity which changes to a new version as instructions are added over time, RISC-V aims at keeping the base and the standard extension constant over time, and instead layer new instructions as further optional extensions. This will ultimately increase reuse of software, especially in the long-term.

The base of RISC-V is similar to the original RISC developed in the Berkeley RISC project [6], but is updated to account new trends and needs of the embedded market, since it is relatively new and, being and open standard, allows open discussion on what must be included in the standard. For instance the standard defines 32-bit (RV32), 64-bit (RV64) and even 128-bit (RV128) address space variants, it is thought to support manycore implementations (including heterogeneous multiprocessors), to be fully virtualizable to ease hypervisor development and provides features like a 16-bit compressed Instruction extension (C) to increase performance, code density and power efficiency. The standard atomic instruction extension (A) adds instructions that atomically read, modify, and write memory for inter-processor synchronization using load-link/store-conditional instructions instead of compare-and-swap instructions, thus avoiding the ABA problem affecting the latter and allowing a straightforward use of modern crossbars that don't support locked accesses, like AMBA AXI4. Furthermore, RISC-V is little-endian, allowing straightforward integration with the most popular state-of-the-art embedded infrastructures and proprietary architectures.

3 State of the Art

UC Berkeley and SiFive have released an open-source SoC generator called Rocket Chip to easily configure a SoC and automatically generate the synthesizable RTL (Verilog). It is written in Chisel, a hardware construction language based on Scala, and can be configured to generate a wide range of SoCs. Based on the Rocket Chip generator, SiFive has released several IPs, components and development boards (e.g. the Arduino-compatible HiFive1 and the Linux-capable HiFive Unleashed). SiFive also provided IP cores for the Mi-V ecosystem by Microsemi for their flash-based line of FPGAs, which comprises the radiation-tolerant RTG4 FPGA. The RISC-V software ecosystems is maturing quickly. Several ISA simulators (e.g. Spike and QEMU), C compilers (e.g. GCC), C libraries (e.g. glibc and newlib) and debugging tools (e.g. gdb) are already available. The availability of such ecosystem ignited the development of several open-source hardware platforms from several universities and companies. For instance ETH Zürich and University of Bologna are working on the Parallel Ultra Low Power (PULP) Platform, an ultra-low-power processing

platform mainly targeted to the Internet of Things (IoT) [7]. It is based on several processors, ranging from a simple 2-stage 32-bit core (RV32EC) supporting compressed instructions to a 6-stage Linux-capable core (RV64GC) with caches and TLBs, and comprises several SoCs architectures (e.g. the single core PULPino and the many-core PULP). The platform provides also other IPs for communication (I2C, UART, SPI, etc.) and can be easily integrated in FPGAs and extended, as it is built around the popular AMBA crossbar AXI4 and the peripheral bus APB. The source code is written in SystemVerilog, an HDL fit both for design and verification which is becoming increasingly more popular in the commercial field. Based on the PULPino SoC, Sapienza University of Rome released Klessydra, an open-source SoC with multi-threaded CPU in VHDL [8]. Other open-source implementations are already available, as VectorBlox Computing's Orca, a stand-alone VHDL implementation of RV32I and RV32IM intended to target FPGAs and developed to be also included in their proprietary products. Also established players announced developments based on RISC-V. For instance Western Digital, a founder of the RISC-V Foundation, announced that over the next few years all the processors shipped within their products will be transitioned over to RISC-V [9] and several other big players are allegedly working on such transition.

4 Future Developments and Benefits for Space Industry

Processors in space face unique challenges due to the effects of the space environment. While components must be designed keeping into account the physical effects of a wide temperature range, mechanical vibrations, vacuum and radiations, when considering an IP core everything is lumped into functional faults. It is up to the IP core designer to consider how the space environment will induce functional faults and how to counteract to them with an effective area/power/frequency trade-off. The designer can typically chose between:

- a Fault-Tolerant (FT) IP Core designed from scratch
- a FT processor obtained modifying a COTS IP core
- a COTS IP core without any modifications.

The last choice has become very popular over the last years, as the use of proprietary COTS processors has been indicated as a solution to reduce costs and increase performance, relying only on system level enhancement to fault tolerance. However, such processors were not intended for the specific needs of electronics for space and sometimes their lack of functionalities can't be compensated at system level [10]. A typical example for this is observability: if the IP core or the ISA was not designed to let the user know what is happening inside the box, the response at system level may be inadequate or inefficient, resulting in reduced safety and availability. Implementing fault-tolerance at system level often implies heavy redundancies (e.g. Triple Modular Redundancy of the processors) with big penalties in terms of power and size.

The use of an FT-enhanced COTS IP core, still compatible with the original software ecosystem, seems then the best approach to effectively keep into account the effects of the space environment while keeping costs and development time for both hardware and software under control. The main obstacle to this is that typically commercial processors and ISAs are proprietary: accessing the source of the IP core or obtaining a license to use the ISA may be too expensive, limit what can be done with the final product or simply be not possible. If the base of the FT processor is a open-source IP core based on an open and free ISA, then modifying the RTL becomes a viable way. The adoption of an open and free ISA that is designed to be modular and easily extendable like RISC-V enabled a vast field of research activities both for terrestrial applications (e.g. security, artificial intelligence, multi-threading, digital signal processing, etc.). The Space Industry can then spin-in developments from other fields, focusing its efforts mainly on improvements concerning the specific needs in space applications and without wasting efforts on non-strategic activities.

Nevertheless open and free ISAs offers advantages also in the case of developments of FT processors from scratch, as there is no need to pay for a proprietary ISA and the final product can be owned by the designer.

5 Conclusion

Thanks to the open and modular nature of RISC-V, designers are free to implement whatever architecture is deemed best for their applications, from low performance/low power microcontrollers to high performance CPU for payload applications to reliable processors handling a large number of tasks. The introduction of RISC-V in space will contribute to providing a range of alternatives to proprietary solutions in a frame of new architectures for on-board embedded systems, as concerns are growing about monopolistic positions in the embedded market. RISC-V looks like the best solution in this case, as it is backed by big commercial players and academia exactly for this reason. Further works and studies will give to the European Space Industry the tools and the knowledge required to choose proprietary solutions when actually needed and choose several degrees of openness when possible. This will lead the European Space Industry to build the next generation of embedded systems for space with a harmonious and effective mix of the two approaches, taking the best from both worlds.

References

1. Margan, D., et al.: The Success of Open Source Software: A Review. MIPRO (2015)
2. Gaisler, J.: A Portable and fault-tolerant microprocessor based on the SPARC V8 Architecture. In: International Conference on Dependable Systems and Networks (2002)
3. Furano, G., et al.: Roadmap for on-board Processing and Data Handling Systems in Space. Springer International (2017)

4. Waterman, A., et al.: The RISC-V instruction set. 2013 IEEE HCS
5. Salmon, L.G.: A perspective on the role of open-source ip in government electronic systems. In: 7th RISC-V Workshop Proceedings, November 2017
6. Patterson, D.A., et al.: A VLSI RISC, IEEE Computer (1982)
7. Rossi, D., et al.: PULP: a parallel ultra low power platform for next generation IoT applications. In: 2015 IEEE Hot Chips 27 Symposium (2015)
8. Cheikh, A., et al.: The microarchitecture of a multi-threaded RISC-V compliant processing core family for IoT end-nodes. In: APPLEPIES (2017)
9. Fink, M.: RISC-V: enabling a new era of open data-centric computing architectures. In: 7th RISC-V Workshop Proceedings, November 2017
10. Furano, G., et al.: A novel method for SEE validation of complex SoCs using Low-Energy Proton beams. 2016 IEEE DFT