ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 1, Number 1, 2022, 1–8

Dissertation Report: A review and comparison of the open-source and commercial soft-core processors

Mihai Paul FIRTALA[1]

[1]Politehnica University of Bucharest, Faculty of Electronics, Telecommunications and Information Technology

Email: mihai_paul.firtala@stud.aero.upb.ro*

Abstract. As embedded systems become more and more popular and accessible to the large public and the performance increases with every generation, new ideas and projects implying specific implementations are brought out requiring development and support. One such example is represented by the various soft processors already existing or in development in the public space. These processors are implemented using logic synthesis which translates a design described by a hardware description language (HDL) into a logic gate implementation. Both open-source and commercial soft processors have advantages and disadvantages in terms of cost, support, complexity, performance and the use-case for which each processor was designed. A detailed comparison will be made in this report for some of the most known solutions in order to create a general idea about the subject and to highlight the above-mentioned features.

Key-words: Soft processor, hardware description language (HDL), logic synthesis, FPGA, RISC, ARM, instruction set

1. Introduction

Embedded systems are complex systems containing hardware and software which are used to process input data from the outside world into signals that usually control a machine. Our daily life is filled with similar systems, from the simple ones controlling a washing machine or a refrigerator to the more complex ones used as on-board computers for satellites, planes and cars. An embedded system contains one or more microcontroller units (MCUs), memories and input/output peripheral devices (sensors) and are used to gather(input) data from the exterior world, be it sound, light, magnetic field or acceleration forces, and output them as control signals usually used to drive an actuator or to display information. The data processing is performed by

the microcontroller by executing the program stored in the memory. Since the first commercial field programmable gate array (FPGA) became available in 1985 and with the advance of modern technology, the performance has increased in such a manner that nowadays it is possible and more convenient to simulate a fully functioning microprocessor together with its peripherals on an FPGA board before the manufacturing process begins.

1.1. Related works

The "Soft processors as a prospective platform of the future" presents important aspects about the soft CPUs compared to the general-purpose CPUs and to the ASIC based processors. The paper concludes by mentioning that the modern FPGAs are the most promising platforms for supporting soft CPUs with a wide area of applications such as general-purpose products with digital control, tablet and mobile computers, computer terminals and personal computers with improved security and reliability. Vivek Jayakrishnan and Chirag Parikh compared in their paper [1] the power consumption, speed and resource utilization of two SoCs: a soft-core based design composed of a soft ARM Cortex M0 processor along with custom IP components and a hard-core based design containing a hard ARM Cortex A9 processor connected to the IP modules (VGA, I/O, Bus) loaded on the FPGA. Both designs run an oscilloscope application which inputs voltage signals via an ADC and outputs the scaled values on a display. The study concluded by observing that the hard- core design is much faster and utilizes less resources, but the power consumption is higher, in contrast to observations made in other studies, maybe due to some features that allow the Cortex M0 to set the sleep mode over the unused internal components. A detailed comparison of several open-source and commercial soft processors based on their features was made by Jason G. Tong, Ian D. L. Anderson and Mohammed A. S. Khalid [2]. They also provided some use cases in the industrial and research applications for the selected platforms. Their paper was the main source of inspiration and was the most relevant for the initial topic of this report. The next chapter was realized by combining all the information and knowledge gathered from the above-mentioned articles and from topic related websites that are included in the reference section.

2. Soft-core processors. State of the art

When it comes to the design of an embedded system, a series of approaches can be taken into consideration depending on the initial requirements. Cost, computation speed, memory size, the environment in which the system will function and other features play an important role in the initial design. The final approach needs to be balanced in order to exploit the advantages of the chosen solution and to make sure that the disadvantages present an insignificant impact. One can implement the system around a hard-core microcontroller which is a cheaper approach when it comes to developing complex algorithms in a short period of time and when the execution speed and the energy efficiency presents important aspects of the project. The control logic is much easier to implement, and a vast set of programming skills are available due to the large online community, thus allowing small companies and individuals to come up with innovative ideas. A second method that may be used for implementation is a system on chip containing a hard processor which combines the conventional programmable hard-core microprocessor and an FPGA to exploit the features of both elements. An SoC with a soft processor which includes the same hardware as stated in the previous point, but the FPGA is used to simulate the behavior of a microprocessor can be used as a third method. Let's further detail this approach in the next

rows. A soft-core processor is a reusable hardware module [3] whose architecture and behavior can be described using a synthesizable HDL code such as VHDL or Verilog, as opposed to the hard-core processors which are baked into the silicon. The use of soft processors includes considerable advantages:

- Flexibility: the soft processor can be configured and customized (i.e. adding custom instructions and/or components such as floating-point units, memory protection units, error detection and correction encoders/decoders etc) in order to adapt the system based on the user needs.
- Multiple cores: depending on the number of logic cells, the fpga can hold multiple cores thus having the possibility of parallelizing some tasks.
- The soft cores are technology independent meaning that they can be loaded on every FPGA board, thus they present a high tolerance to obsolescence.
- In the case of an SoC, distinct processes can be divided between the hard and the soft CPU.
- Portability: the soft core can be ported to any FPGA, as long as it is not vendor specific.
- The architecture and behavior are written at higher abstraction level using a hardware description language; therefore it is much easier to understand and to create a design.

Some of the disadvantages of a soft-core processor are represented by the overall high cost of an FPGA, lower computation speeds than an MCU due to the FPGA fabric, the higher complexity of a hardware description language compared to a programming language such as C and the less numerous online community. These days a large number of soft processors can be found on the internet, whether it be open-source platforms supported by individuals or commercial solutions provided by various vendors. In the following chapters we will present in detail the available platforms, mentioning important features regarding performance, number of stages, availability, required FPGA size, use cases, community etc.., and a comparison between them will be made at the end concluding with the most important aspects and the possible directions that could be approached in the future works.

3. Explore the platforms

The multitude of soft processors available online allow the users to choose the best solution suitable for their projects. In this chapter we will explore in depth these processors, provide a comparison in the form of a table and we will highlight the important features in terms of performance and availability.

3.1. Open-source processors

In this subsection we will explore the open-source soft processors available to the individuals or companies without a paid fee due to the accompanied license. The lack of an accepted open-source hardware license determined the use of software licenses such as GNU LGPL, GNU GPL or BSD. A large number of processors can be found on github or published on the OpenCores.org website. As stated in the wikipedia page [4], the Open Cores is a community formed

by individuals, universities or companies that develop open source hardware components by using electronic design automation (EDA) tools similar to how open source software is produced. The community comes up with a large variety and number of processors each having distinct and important features that are worth mentioning in this paper such as Amber, pAVR, Leon2/3/4 and BERI.

Amber is an open source 32-bit RISC processor based on the ARM architecture based on the ARMv2a instruction set, made by Conor Santifort. The processor is compatible with the ARMv2a instruction set, therefore it can be implemented without an ARM license being supported by the GNU toolset. The project is available for download on OpenCores.org and is part of the open-source hardware movement which aims to develop libraries that can be used in projects for free [5]. According to the website, it was created in 2010 and was updated several times, the final one being in 2019, which resulted in a stable version. The processor was developed in Verilog 2001 and implemented on the Xilinx Spartan 6 SP605 FPGA development board being optimized for FPGAs and provides a complete embedded system including the core and several peripherals such as UART, timers and Ethernet MAC. There is no reset logic, all registers being reset as part of FPGA initialization. Two versions of the core were developed as part of the Amber project. The Amber 23 was designed with a 3-stage pipeline, a unified instruction and data cache, a 32-bit wishbone interface and is capable of 0.75 DMIPS / MHz. Register based instructions execute in a single cycle, except for instructions involving multiplication. Load and store instructions require three cycles. The Amber 25 has a 5-stage pipeline, separate data and instruction caches, a 128-bit Wishbone interface, and is capable of 1.05 DMIPS / MHz. Both cores implement the same ISA and are software compatible. They do not contain a memory management unit so they can only run the non-virtual memory variant of Linux and they have been verified by booting a 2.4 Linux kernel. [5][6]

pAVR is an AVR architecture compatible 8-bit microcontroller designed in VHDL by Doru Cuturela. It is a general representation of an AVR microcontroller which can be configured to simulate most of the AVR family members. The core was originally designed to obtain a powerful processor in terms of MIPS and concluded in a core 3x faster than the original AVR core. The pAVR is an easily maintainable and highly customizable design due to its modularity, code writing techniques and documentation. It has a 6-stage pipeline with 1 clock per instruction at peak (1.7 CPI typical) for most instructions, an estimated clock frequency of 50 MHz with 28 MIPS or 50 MIPS at peak (meanwhile the Atmel's cores run at 15-16 MHz with 10 MIPS). According to the Open Cores website the project was finished in the summer of 2002 and was updated a few times with the current status being "Alpha" due to the newly discovered bugs. [7]

Leon2 is a 32-bit processor model compatible with the SPARC V8 architecture created by the European Space Agency (ESA) and written in VHDL. It is a configurable core due to its modularity and has separate instruction and data caches, hardware multiplier and divider, interrupt controller, debug support unit with trace buffer, 2x24-bit timers, 2xUART, power down function, watchdog and 16-bit I/O ports. The model is synthesizable with most tools and can be implemented on ASICs and FPGAs. Leon2-FT was designed as a radiation tolerant processor to be included in the spacecraft's on-board computer and is the fault tolerant version of the Leon2 implementing features like triple modular redundancy for the flip-flops and protection for all the internal and external memories, using EDAC and parity bits. During a mission in outer space, the cosmic radiation may affect the data storage or data exchange between the various components by introducing single event upset (SEU) errors. These events appear due to the ionized particles striking nodes of logic elements and randomly changing bits, in a way that may

prevent the proper functionality of the system and thus concluding to the mission failure. The processor design incorporates advanced error detection and correction (EDAC) techniques that enable reliable delivery of data over unreliable communication channels in order to withstand arbitrary SEU errors that may cause the unwanted loss of data. The processor was used in two satellite missions: ESA's Intermediate eXperimental Vehicle (IXV) and China's Chang'e 4 lunar lander. [8][9]

Leon3, just like Leon2, is a synthesizable VHDL, 32-bit, SPARC V8 compatible soft processor and it is highly configurable. One major difference between the two processors is the number of pipeline stages: Leon3 has 7 stages while Leon2 has 5. It is based on a Harvard architecture by having separate instruction and data cache, containing features like hardware multiply/divide, MAC units, memory management unit, AMBA 2.0 AHB bus interface, provide debug capabilities, a maximum of 125 MHz on FPGA and 400 MHz on ASIC technologies and a performance of 1.4 DMIPS/MHz. Leon3-FT is the fault tolerant version of the Leon3 processor and includes similar methods for overcoming the single-event-upset (SEU) errors as Leon2-FT. Additional features of the Leon3-FT compared to the Leon3 are register file SEU error correction with 4 errors per 32-bit word, cache memory error correction, autonomous error handling and no timing impact due to error correction and detection. [9][10]

BERI (Bluespec Extensible RISC Implementation) is a soft-core processor implementing 64-bit RISC instruction set and designed by the University of Cambridge and SRI International using Bluespec System Verilog which is a high-level functional programming language for describing hardware designs. The processor implements a floating-point unit (FPU), programmable interrupt controller (PIC) and multicore operation (still in progress) and is able to boot the FreeBSD operating system which supports multitasking, TCP/IP networking, security features and third party open-source applications. The processor was tested on a Terasic DE4 board at 100 MHz, with CPI close to 1.0. The processor is used mainly in research purposes such as processor multithreading, verification and other instruction set level experimentations and the source code is available on the project website also provided in [11].

3.2. Commercial platforms

Commercial soft processors are available under a proprietary license that has to be purchased in order to be used in a proprietary product. The vendors contributed to the soft processor's world with revolutionary designs.

Cortex M1 is the first ARM processor designed to be implemented on a FPGA and includes support for synthesis tools. It is a 32-bit processor compatible with ARMv6-M architecture based on the Thumb and Thumb2 instruction set with a 3-stage pipeline and debug capabilities. As important features we can count the 0.88 DMIPS/MHz. The processor is supported by many FPGA boards such as Altera's Cyclone and Stratix series, Xilinx Spartan-3, Virtex-2,3,4 and Artix-7 and Microchip Fusion, ProASIC3L and other boards.

Nios II is a 32-bit RISC processor designed specifically for the Altera FPGA family. It is suitable for a wide range of embedded applications such as digital signal processing (DSP) or system control. The processor is highly configurable by adding a predefined memory management unit or by defining custom instructions or custom peripherals. The processor family comes in different variants Nios II classic (fast, standard and economy) and Nios II gen2 (fast and economy) each having specific features. Nios II/s (standard) version is a 5- stage pipeline processor, with up to 256 custom instructions that can be defined by the user, having a JTAG debug module which may include hardware breakpoints or real-time trace with a performance of 250 MIPS.

PowerPC 440 is a 32-bit RISC processor core belonging to the Power PC family and based on the Power ISA instruction set. It was designed to be used in various applications such as SoC microcontrollers, television decoders, network appliances, ASICs and FPGAs applications, storage devices and supercomputers. It is a high-performance processor with a 7-stage pipeline, memory caches, with speeds up to 800 MHz. Xilinx Virtex-5 FXT FPGA board can support 2 PowerPC 440 cores obtaining a frequency of 550 MHz. BRE440 Rad Hard SOC includes an IBM licensed PowerPC 440 core and is designed to be used in radiation-rich environments for deep space, nuclear, defense and medical applications. The processor includes an FPU, memory controllers, cache, ethernet ports, serial ports, DMA channels, EDAC controller and PCI interface supporting a speed of 133 MHz and 2 MIPS/MHz.

4. A comparison of the processors

This chapter contains a summarize of some of the important features presented in the previous chapter. Each processor was designed for specific tasks, and it is hard to select a one-fits-all solution. Depending on the project initial requirements, the designers need to study in depth and consider additional characteristics in order to decide which processor is more suitable.

Table 1. Soft-core processor features

Processor	Architecture	No. of stages	Cache memory [Kbytes]	Speed [MHz]	DMIPS	Platform
Amber	32-bit RISC	3/5	unspecified	80	0.75/1.05	FPGA
pAVR	8-bit AVR	6	unspecified	50	unspecified	FPGA
Leon2	32-bit SPARC V8	5	unspecified	125	unspecified	FPGA
Leon3	32-bit SPARC V8	7	Up to 256	125(FPGA)	1.4	FPGA/ASIC
BERI	64-bit RISC	6 (BERI2)	64	100	unspecified	FPGA
Cortex M1	32-bit ARMv6-M	3	no	0.88	unspecified	FPGA
Nios II	2-bit RISC	5	Up to 64	200	150	FPGA
PowerPC440	32-bit RISC	7	Up to 256	800	2	FPGA/ASIC

5. Conclusion

In this paper we presented the work that has been done regarding the FPGA and soft processors world and a detailed comparison between the various solutions available, whether it be open-source or commercial. We observed the different methods used for testing and comparing the soft-core processors with the hard-core ones, the promising future that modern technology brings and the distinct industrial domains in which these processors offer reliable and performant solutions. Starting from the display applications to the deep space exploration, the soft cores posses unlimited possibilities in terms of configuration, complexity and performance and the technology advance will improve continuously these features. A multitude of future options appeared during the research and the writing of this paper. From the perspective of a fast processor to the radiation hardened one, a large number of aspects can be approached.

6. List of papers selected by title

1. “Embedded Processors on FPGA: Soft vs Hard”, by Vivek Jayakrishnan and Chirag Parikh, Grand Valley State University, Grand Rapids, MI

2. “The Microarchitecture of FPGA-Based Soft Processors”, by Peter Yiannacouras, Jonathan Rose, and J. Gregory Steffan, University of Toronto, Canada
3. RVCoreP: An Optimized RISC-V Soft Processor of Five-Stage Pipelining, by Hiromu Miyazaki, Takuto Kanamori, Md Ashraful Islam and Kenji Kise, Tokyo Institute of Technologym Japan
4. On Data Forwarding in Deeply Pipelined Soft Processors, by Hui Yan Cheah, Suhaib A. Fahmy and Nachiket Kapre, Nanyang Technological University, Singapore
5. Optimal Reduction in the Number of Test Vectors for Soft Processor Cores Implemented in FPGA, by Mariusz Wegrzyn, Ernest Jamro, Agnieszka Dabrowska Boruch and Kazimierz Wiatr, Technical University of Cracow, AGH University of Science and Technology Krakow, Poland
6. Design and Implementation of a soft processor with an embedded FPU, by Angelos- Efstathios Ntasios and Minas Dasygenis, University of Western Macedonia, Greece
7. Soft-Core Processors for Embedded Systems, by Jason G. Tong, Ian D. L. Anderson and Mohammed A. S. Khalid, University of Windsor, Canada
8. Soft processors as a prospective platform of the future, by Dmitry Efanov, Konstantin Grigoryev, Pavel Roschin, and Vassili Leonov, National Research Nuclear University, Scientific Research Institute of Rheumatology, Russian Federation
9. Supporting Multithreading in Configurable Soft Processor Cores, Roger Moussali, Nabil Ghanem, and Mazen A. R. Saghir, American University of Beirut, Lebanon
10. TTA-SIMD Soft Core Processors, by Kati Tervo, Samawat Malik, Topi Lepanen and Pekka Jaaskelanien, Tampere University, Finland

7. List of papers rejected by abstract

1. RVCoreP: An Optimized RISC-V Soft Processor of Five-Stage Pipelining, by Hiromu Miyazaki, Takuto Kanamori, Md Ashraful Islam and Kenji Kise, Tokyo Institute of Technologym Japan
2. On Data Forwarding in Deeply Pipelined Soft Processors, by Hui Yan Cheah, Suhaib A. Fahmy and Nachiket Kapre, Nanyang Technological University, Singapore
3. Optimal Reduction in the Number of Test Vectors for Soft Processor Cores Implemented in FPGA, by Mariusz Wegrzyn, Ernest Jamro, Agnieszka Dabrowska- Boruch and Kazimierz Wiatr, Technical University of Cracow, AGH University of Science and Technology Krakow, Poland
4. Design and Implementation of a soft processor with an embedded FPU, by Angelos- Efstathios Ntasios and Minas Dasygenis, University of Western Macedonia, Greece
5. Supporting Multithreading in Configurable Soft Processor Cores, Roger Moussali, Nabil Ghanem, and Mazen A. R. Saghir, American University of Beirut, Lebanon

6. TTA-SIMD Soft Core Processors, by Kati Tervo, Samawat Malik, Topi Lepanen and Pekka Jaaskelanien, Tampere University, Finland

8. References

References

[1] *Why use a soft cpu core*. https://lxp32.github.io/docs/why-use-a-soft-cpu-core/. (n.d.).

[2] Jason G. TONG. *Soft-Core Processors for Embedded Systems*. Windsor, Ontario, Canada.

[3] *Amber processor*. https://en.wikipedia.org/wiki/Amber_(processor)

[4] https://www.cl.cam.ac.uk/research/security/ctsrd/beri/

[5] Vivek JAYAKRISHNAN, CHIRAG PARIKH. *Embedded Processors on FPGA: Soft vs Hard*. Grand Valley State University.

[6] *Amber ARM-compatible core*. Conor SANTIFORT. https://opencores.org/projects/amber

[7] *pAVR*. Doru CUTURELA. https://opencores.org/projects/pavr

[8] *LEON2FT, European Space Agency*. https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Microelectronics/LEON FT

[9] *LEON processor*. https://en.wikipedia.org/wiki/LEON

[10] Gaisler, LEON3FT Fault-tolerant processor, https://www.gaisler.com/index.php/products/processors/leon3ft

[11] Wikipedia, OpenCores, https://en.wikipedia.org/wiki/OpenCores