Submitted to SRvS *Workshop, November 7-8, 2024*

Hands-on experience for undergraduate Computer Architecture courses using Nios V-based soft SoCs and real board

Domingo Benitez
University of Las Palmas de Gran Canaria - Spain
domingo.benitez@ulpgc.es

October 13, 2024

Abstract

Problem statement. In the Computer Architecture course of Computer Science and Engineering undergraduate programs, there are often labs with simulation tools. This teaching method can give students misleading ideas about how hardware works because they are observing the behavior of a computer program, not the real hardware. In addition, the financial investment required to use real and non-reconfigurable equipment in a performance-oriented Computer Architecture course is relatively high.

Methodology. This paper proposes a set of hands-on laboratory experiments for undergraduate students enrolled in a Computer Architecture course focused on performance evaluation of RISC-V architecture computers. The most prominent feature of these labs is the multiple reconfiguration of a single real FPGA-based board. All FPGA configurations build soft system-on-chip computers based on Nios V processors.

Main findings. The methodology proposed here has been used in the training of more than 1,000 computer science students for more than 10 years. We have observed that this methodology allows: all students to interact with a hardware system that is more realistic than the one modeled by simulators, each student to interact with multiple system-on-chip computers of very different architectures without having to change physical equipment, and that low-cost hardware can be used for Computer Architecture labs where performance evaluation is an important pedagogical goal.

Principal conclusion. This method of teaching gives students a more realistic understanding of computer architecture than teaching with hardware simulators. In addition, by using reconfigurable hardware, the lab infrastructure is less costly than using various non-reconfigurable hardware.

Keywords: FPGA, soft processor, RISC-V, Nios V, reconfigurable hardware, system-on-chip (SoC), computer architecture, undergraduate, Computer Science, Computer Engineering, performance evaluation.

1. Introduction

As computer technology moves from sequential to parallel processing, a deeper understanding of the underlying computer architectures is required [1]. Computer Architecture encompasses a set of skills that are part of the ACM academic curricula called Computer Science [1] and Computer Engineering [2].

In the past decades, much academic effort in Computer Architecture courses has focused on developing pedagogical methods using software tools for hardware performance simulation [3, 4]. Students are often motivated by the hands-on nature of these types of engineering tools [5, 6].

Simulation tools are useful for modeling real systems. They are often desirable and necessary to allow students to study systems that are impractical to design and implement given the time and resources available [6]. Other advantages include low cost and immediate availability at any location, such as the student's residence.

Carefully planned practical assignments in a laboratory setting should help students develop confidence in their technical abilities [6]. However, the teaching methodology based on simulating hardware operation can establish ideas in the minds of students that are misleading. For example, the concept of program execution time is not fully assimilated when simply counting the number of cycles in a simulator window. This is because the student is observing the behavior of a computer program, not the actual hardware.

The purpose of this work was to propose a set of practical laboratory experiences for an undergraduate Computer Architecture course focused on performance evaluation. The main features of these labs are:

- They require a single real board based on a reconfigurable FPGA circuit for an entire academic semester.
- The FPGA is reconfigured multiple times to function as different System-on-Chip (SoC) computers that allow keyboard input and display output of results.
- One or more Nios V soft processors are used [7].

- A configurable memory hierarchy consisting of cache memory and various SRAM and DRAM modules is addressed.
- Multiple input-output controllers can be enabled.

In summary, the following topics are covered:

- Program generation using machine and assembly languages for the RISC-V instruction set architecture.
- Evaluation of the impact of the memory hierarchy on the performance of a SoC computer.
- Reverse engineering to determine the organization of the first-level cache memory.
- Performance evaluation of pipelined processors using benchmark programs.
- The roofline curves of different processors are obtained.
- Impact of the software instruction reordering on SoC performance.
- Multithreaded parallel programming and performance evaluation of a shared-memory Nios V-based soft multiprocessor.
- Analysis of the influence of cache coherency on the performance of a soft multiprocessor.
- Design of processors tailored for a software application.

The main contribution of this work is the definition of a set of hands-on experiences with real hardware for a university Computer Architecture course focused on performance evaluation. These practical experiences based on reconfigurable hardware are less expensive than the alternative situation where several different real hardware systems would have to be used.

We bring here our educational experience after this training has been used by more than 1.000 students for more than 10 years. We have observed that the methodology presented here allows all undergraduates to acquire a vision of the hardware that is more real than that acquired by simulators.

The remaining of this paper is as follows: Section 2 elaborates some related works, Section 3 states the educational context of our work, Section 4 gives a summary of the technical characteristics of the Computer Architecture Laboratory where a prominent part is an FPGA-based board, Section 5 describes the proposed hands-on experiences that use different soft SoC computers, Section 6 analyzes the results and states implications. The paper ends with the conclusion discussed in Section 7.

2. Literature Review

RISC-V International (`riscv.org`) opened a community-driven compilation of RISC-V resources and learning material [8]. This repository includes some university courses with different levels of experience. Most of the university courses included in this repository are oriented towards the use of simulators [9, 10]. There are many other university courses that use RISC-V based processors or software systems in their lab assignments but are not included in this repository. The following is a description of some of them that have motivated the publication of papers in conferences or journals.

RVfpga is a set of two freely available courses developed by the authors and taught at the Spanish UCM university [11]. Both courses use a commercial soft RISC-V processor to learn about architecture instruction set, digital design, processor and memory system design, SoC system design, programming and interaction with an operating system. The first course called Computer Organization is a second-year course in the official Spanish Computer Science degree program that is recognized by the European Higher Education Area [12]. All hands-on exercises are conducted in this course through simulations within a provided virtual machine [10]. The second course called Integrated Systems Architecture is a fourth-year course in another official Spanish degree program called Electronics and Communication Engineering. In this course, students are provided with a FPGA-based board that allows them to explore concepts related to the computer organization. In all practical exercises a single configuration for the soft processor is used. Simulations tools can be also used in this course [13].

Minev et al. presented several exercises in the Computer Architectures discipline [6]. They proposed 11 hands-on assignments to apply in the practical training of undergraduates the creation of a non-commercial pipelined soft RISC-V model using TL-Verilog and the Makerchip environment. These experiences are focused on implementing the microarchitecture of a RISC-V processor through simulation tools.

There are other university courses whose lab assignments use or have used soft processors with non-RISC-V ISA architecture. Mandalidis et al. proposed several exercises to undergraduate students of a Computer Architecture laboratory [4]. These exercises train them in issues concerning the FPGA implementation of a MIPS R2000 32-bit RISC processor, peripherals, programming, functionality and interfacing. This system was implemented on a real board. The authors recognize enthusiasm was especially relevant to motivation of students attending this type of course.

Harris et al. presented MIPSfpga and its accompanying set of learning materials that is a teaching infrastructure to learn Computer Architecture based on a soft-core processor [3]. Their labs focus on hands-on learning that emphasizes computer architecture, system-on-chip design

and hardware–software codesign. Students are proposed to set up a commercial MIPS soft-core processor on an FPGA, run programs, add new peripherals to the SoC system, extend the microarquitecture to support new features, experiment with different cache microarchitectures and run operating systems. MIPSfpga is the predecessor of RVfpga and the experience acquired from MIPSfpga is being transferred to the RISC-V based RVfpga infrastructure [11].

Intel Academy Program provides educational resources that supports the use of FPGAs in real DE-series boards such as DE0-Nano [14]. Our work was inspired by this program that includes labs for teaching a number of university courses that are part of a typical Engineering/Computer Science curriculum. Currently, the types of curses available are: digital design, computer organization, embedded systems and compute acceleration. However, a course for computer architecture focused on performance evaluation is not provided. Neither, this program offers a course based on Nios V-based soft SoC systems for DE boards. The teaching material for the Computer Architecture course presented here was designed from the study of SoC systems provided by the Intel Academy Program that are based on the soft Nios II processor.

3. Educational Context

The educational context of our proposal for an undergraduate course in Computer Architecture is situated in a Computer Science Undergraduate Program within the European Higher Education Area that attempts to connect different higher education systems [12]. The course described in this document is offered at a Spanish university as part of an official degree program approved by the Spanish government, which has a duration of four years and is called *Computer Science*.

The Computer Architecture course is a compulsory second-year course, offered in the spring semester and requires a dedication of approximately 150 hours, 60 of which correspond to classroom lectures and labs, 30 hours of lectures and 30 hours of hands-on practice. The number of students completing the lab assignments each semester is approximately 100. The students are organized in seven groups of 10-26 people. Each group is assigned a weekly lab session.

By the time a student takes Computer Architecture, he/she has previously spent approximately 450 hours in digital design, assembler programming and computer organization. Each spring semester, the set of Computer Architecture lab assignments to be offered in the spring semester of the following year is scheduled and approved by representatives of the local Computer Science School.

The materials for each lab assignment include a tutorial and several files of two types. One type of files contains the soft configurations of the various SoC computers used in the lab assignment. And the other type of files contains the source code of assembly or C programs.

Lab sessions are offered in a laboratory located in the Computer Science School building. Each lab session is taught by a professor for two hours. The first part of each lab session lasts between 30 and 60 minutes. In the first part of each lab session, the background necessary for the lab session is explained or reviewed. In the second part, the student does the practical work for the remainder of the time. Before the end of the lab session, the instructor will suggest a series of exercises for the students to complete before the next lab session. In some lab sessions, there is an exam on the last lab assignment. Each lab assignment takes three or four lab sessions. The final grade for the lab portion of the Computer Architecture course is calculated as the geometric mean of the grades received on the exams.

4. Laboratory Infrastructure

To develop the hands-on experiences described in this document, it is necessary for each student to have an technological infrastructure divided into hardware and software parts. In our Computer Science School, we have a physical space called the *Computer Architecture Laboratory*, also known as *CAlab*. Inside the CAlab, each student sits at an individual workstation. In our case, 26 students can participate in each lab session. At least one technician is always available to quickly resolve any unusual technical problems that may arise with both hardware and software.

4.1. Real hardware equipment

Each workstation includes a Terasic DE0-Nano board, a desktop PC with keyboard, mouse and display, a USB-A - miniUSB cable, a table and a chair. The DE0-Nano board consists mainly of a Cyclone IV FPGA, 32 MB SDRAM, and a miniUSB connector for power and PC interface. The price of each board is $87 for academics in 2024.

The hands-on exercises proposed here use a RISC-V instruction set architecture for the soft configurations of the SoC computers that reconfigure the FPGA of the DE0-Nano board. The microarchitectures of the SoC computer configurations are based on the soft pipelined processors called Nios V [7]. Two of the Nios V family soft processors are used. The Nios V/m soft processor supports the RV32IZicsr instruction set and has no cache. The Nios V/g processor supports the RV32IMZicsr instruction set and has first-level instruction and data cache. In addition, the SoCs integrate input/output controllers such as timers, UART-JTAG, SRAM and SDRAM controllers, and an Avalon bus interface.

4.2. Software tools

The PC desktop computer of each workstation has the software tools installed that are required to complete the hands-on exercises. A required tool is called *Nios V Command Shell*, which is included in the *Intel/Altera Quartus Prime* package. In our case, we used the Standard Edition

23.1 Design Suite. This software contains all the tools to compile, assemble and link C programs and to generate RISC-V executable code.

5. Hands-on lab assignments

A total of five lab assignments have been developed for the undergraduate Computer Architecture course. The main feature of these assignments is the use of a single board called Terasic DE0-Nano that integrates an FPGA. This electronic device is reconfigured up to five times for different soft SoC computers to execute programs built with RISC-V instructions. The educational objectives of each lab assignment are summarized below.

5.1. Lab 1. RISC-V instruction set architecture and programming of Nios V/m processor

It is appropriate for students to write several test programs directly in assembler for RISC-V instruction set architecture [6]. Thus, this Lab assignment is an introductory exercise that involves Nios V/m processor and its RISC-V assembly language. The assignment uses a simple soft SoC computer called *DE0-Nano Nios V/m Basic Computer*. The soft SoC system is implemented as a circuit that is downloaded into the FPGA device on the DE0-Nano board. This exercise illustrates how programs written in the RISC-V assembly language can be executed on the a real board. The Nios V Command Shell of the Quartus Prime Design Suite is used to compile, load and run the application programs.

The educational objectives to be achieved by the students are the following:

- Description of the RISC-V instruction set architecture of the Nios V/m processor.
- Description of the software tools for managing the DE0-Nano board in the laboratory.
- Development of assembler programs and their execution on the DE0-Nano Nios V/m Basic Computer soft system: subroutines, modification of the machine code of a program, implementation of proposed algorithms.

The time estimate for this lab assignment is four lab sessions in CAlab.

5.2. Lab 2. Performance evaluation of the memory hierarchy of a computer and reverse engineering of the data cache memory

In this exercise, two different soft SoC configurations called *DE0-Nano Nios V/m Basic Computer* and *DE0-Nano Nios V/g Basic Computer* are implemented in the FPGA of the DE0-Nano board. In both soft SoC computers, two levels of memory hierarchy called *Main memory* and *Cache memory* are implemented. The main memory level on the DE0-Nano board is implemented in two different ways. Firstly, with SRAM circuits inside the FPGA but outside the Nios V soft processor. Secondly, with SDRAM circuits that are on the board but outside the FPGA. The cache level is implemented with on-chip SRAM.

The various exercises measure the time in seconds required to execute two benchmark programs. A hardware clock cycle counter with a frequency of 50 MHz is used for this purpose. The first benchmark is used to compare the performance of the two soft SoCs. The main difference between the two SoCs is that the Nios V/m processor has no cache memory and the Nios V/g processor has cache memory. In this part, the student will be able to perceive the benefit that is achieved in the performance of a computer when SRAM or SDRAM technology is used in the main memory and when the cache memory of a processor is enabled or not.

The second benchmark is used to address different subsets of bytes of a very large data vector with different patterns. In this part of the lab, only the DE0-Nano Nios V/g Basic Computer SoC is used, whose processor has a first-level cache memory separated into data and instructions. The student must determine the memory capacity and block size of the data cache of the Nios V/g soft processor by observing the time measurements in seconds from the second benchmark runs.

The educational objectives to be achieved by the students are the following:

- Description and implementation of the method for evaluating the performance of the memory hierarchy composed of cache and main memory levels in a soft SoC computer implemented on the DE0-Nano board.
- Description and implementation of the method for determining the parameters of the microarchitecture of the data cache memory of the Nios V/g processor.

The time estimate for this lab assignment is four lab sessions in CAlab.

5.3. Lab 3. Performance evaluation of pipelined processors

The main objective is to compare the performance of the Nios V/m and Nios V/g pipelined soft processors using a roofline curve. For this purpose, each student will obtain an X-Y graph where the horizontal X-axis represents the ratio between the number of ALU operations and the number of bytes stored in memory that are addressed by the program during a given time. The vertical Y-axis represents the number of ALU operations committed by the processor per second. In addition, this lab assignment will analyze the effect of the software instruction reordering technique on the performance of both processors. For all these activities, students will work with three different benchmark programs and the two SoC soft configurations *DE0-Nano Nios V/m Basic Computer* and *DE0-Nano Nios V/g Basic Computer*. All these materials will be provided to the students by the instructors.

The educational objectives to be achieved by the students are the following:

- Analysis of the mix of instruction types in a benchmark program.
- Analysis of the limitations of the *ALU Operations per Second* ratio observed during the execution of a benchmark program on Nios V/m and Nios V/g pipelined processors.
- Analysis of the effects of instruction reordering on Nios V pipelined processors.

The time estimate for this lab assignment is four lab sessions in CAlab.

5.4. Lab 4. Nios V multiprocessor implementation, parallel programming, and performance evaluation

This hands-on exercise introduces students to parallel programming through three C programming tutorials. At the end of these tutorials, each student has implemented a multithreaded parallel program that runs matrix-vector multiplication. The student is then asked to implement a new parallel algorithm consisting of matrix multiplication. The Nios V Command Shell tools installed on the PC are then used to execute various RISC-V multithreaded programs on the DE0-Nano board. Two new soft SoC configurations are used in this exercise, named *DualCoreNiosVm* and *DualCoreNiosVg*. The difference between the two configurations is the enabled processor, Nios V/m and Nios V/g, respectively.

The educational objectives to be achieved by the students are the following:

- Parallel programs on two dual-core Nios V multiprocessors are implemented with the goal of significantly reducing program execution time.
- Measure and compare the execution times of parallel programs with the corresponding sequential versions using the DE0-Nano board.
- Performance evaluation of a pair of dual-core Nios V multiprocessors for different amounts of processed data.
- Data cache coherency analysis of the Nios V/g dual-core multiprocessor.

The time estimate for this lab assignment is three lab sessions in CAlab.

5.5. Lab 5. Nios V processor with customized architecture for a software application

The main goal of this lab assignment is to discover and explain the key mechanisms by which processors with a specialized instruction set reduce the execution time of determined programs. The goal is to optimize the Cyclic Redundancy Check (CRC32) algorithm to detect errors that leave the data intact and add a checksum at the end.

The student is provided with a set of files described in Verilog that allow the creation of a specialized functional unit in the Nios V/g soft processor. These files are necessary to generate a SoC configuration of the DE0-Nano board FPGA called *AC_CI_CRC*, whose file is also provided. In addition, a file with C source code is provided. This file allows to obtain a program that implements the CRC32 algorithm in two different ways, with and without the use of a custom instruction of the Nios V/g soft processor.

The educational objectives to be achieved by the students are the following:

- Understand Verilog and, in particular, the contents of the files provided by the instructors.
- Understand how the custom functional unit described in Verilog is integrated into the microarchitecture of a Nios V/g processor.
- Understand how the use of the new custom instruction is integrated into the C compiler tools based on the RISC-V instruction set architecture.
- Analyze in the CRC32 algorithm the types of operations performed, the data structures used, and the data dependencies between operations.
- Discover which operations of the CRC32 software application consume most execution time.
- Compile, load into memory, execute, and measure the execution times of the C program for the CRC32 algorithm on the DE0-Nano board.
- Evaluate the performance of the specialized Nios V/g processor and compare it to the non-specialized processor.

This lab is optional and takes four lab sessions to complete in CAlab.

6. Discussion

The work presented in this paper proposes an educational method for the Computer Architecture course based on the use of a single real board integrating an FPGA that can be reconfigured to function as different computers. We use several soft SoC computers based on Nios V/m and Nios V/g processors with RISC-V instruction set architecture to teach computer science students basic computer architecture knowledge related to performance evaluation.

In the five lab assignments described in this paper, there is no need for each student to run different real computers, which would be prohibitively expensive. By simply using a single reconfigurable FPGA-based board on which multiple soft SoCs can be configured, it is possible for a

student to run the same program on different computers without having to change the physical hardware.

A major drawback that had to be overcome in order to have multiple soft SoCs was the need to build the FPGA circuit configurations. The author built these SoCs by modifying some of the Golden System Reference Design (GSRD) provided by Intel-Altera. In addition, the author had to build the benchmark programs that are run by the students in CAlab. These benchmarks include access to hardware counters to measure cycles, as well as input/output drivers to display program output on the PC monitor.

The main claim proposed here is that our educational methodology produces in the student an understanding of the content of a Computer Architecture course focused on performance evaluation that is closer to reality than simulator-based teaching. The additional cost of the boards is negligible compared to all the costs required to provide the physical space of a laboratory and the teaching and technical staff.

Many computer science students who have done hands-on exercises for a Computer Architecture course in different parts of the world have only used hardware simulators. This fact may have influenced a significant percentage of them to think that the computer is a black box whose configuration they do not need to interfere with. In some universities, students are not taught to distinguish the elements that make up the main board of a computer. By forcing the student to interact with the motherboard of a soft SoC computer, which can adopt different configurations and analyze its performance, the student becomes aware of the possibility of intervening in the definition of the computer's configuration.

6.1. Example of teaching schedule

Currently, in the Computer Architecture course of the Computer Science major at the University of Las Palmas de Gran Canaria, a teaching program of 15 laboratory sessions is being implemented. In this course, labs 1, 2, 3 and 4 are compulsory and lab 5 is optional. The first three labs take four lab sessions and the fourth lab takes three lab sessions.

We have developed all of the labs using the Nios II processors, which have a RISC-type instruction set, but not RISC-V. Taking advantage of the ease with which a Nios II processor can be swapped out for a Nios V processor [15], starting next spring semester 2025 we will be using NiosV processors with RISC-V architecture.

In each of the lab assignments, there is an exam that is scored from 0 to 10. A student passes the lab assignments portion of the Computer Architecture course if the geometric mean of the four scores is equal to or greater than 5.

This scheme has been used for more than 10 years for the education of more than 1.000 students of our university. Approximately 85% of the students obtain a grade equal to or greater than 5 on the first attempt. This percentage is higher than that of students who pass the theoretical part of the Computer Architecture course. The academic success in the exercises proposed here is probably caused by the higher motivation of the students. This motivation is observed to result from the ease of passing the course by examining the practical skills they have acquired in the lab sessions.

The hands-on exercises presented here do not involve the use of an operating system to manage the processor. Therefore, the influence of an operating system on the measured performance of the soft SoC computer based on the RISC-V architecture is not observed by the student. Furthermore, the microarchitectures of the soft SoC systems used in the proposed exercises have been implemented by the author. The students do not create the FPGA configurations that are evaluated in the lab. For a computer science graduate, this limitation may not be relevant. However, computer engineers may find it useful to replace some of the lab assignments in the sample syllabus described above with ones that involve modifying the microarchitecture of the soft SoC.

7. Conclusions

This paper has presented a set of practical hands-on exercises using soft SoC systems based on two Nios V processors to teach the content of a Computer Architecture course focused on performance evaluation. This proposal allows all students to work with real hardware at a low cost. Regarding the use of a free simulator, in our case the additional cost of purchasing 26 DE0-Nano boards is approximately $2,500. Regarding the use of computers with different architectures whose hardware is not reconfigurable, our Computer Architecture teaching proposal is less expensive because we use only one reconfigurable FPGA circuit. In the future, we plan to extend these practices by studying the influence of an operating system on the performance of soft SoC computers when running a given program.

Supplementary Materials

This section contains the links to the materials for teaching Computer Architecture using the methodology described above. The Nios V- and Nios II-based labs are accessible through the online platforms [16] and [17], respectively.

Acknowledgments

The author acknowledges the support provided by the DIS computer science department, EII computer science school, and SIANI research center, all of them belonging to the University of Las Palmas de Gran Canaria. The author acknowledges his colleague Ricardo Perez for many helpful conversations.

References

[1] A. N. Kumar, R. K. Raj, Computer Science Curricula 2023 (CS2023): Community Engagement by the ACM/IEEE-CS/AAAI Joint Task Force, in: Proceedings of the 54^{th} ACM Technical Symposium on Computer Science Education V. 2, SIGCSE 2023, Association for Computing Machinery, New York, NY, USA, 2023, pp. 1212–1213. doi:10.1145/3545947.3569591.

[2] Association for Computing Machinery and IEEE Computer Society, Computer Engineering Curricula 2016, Tech. rep., ACM and IEEE (2016).

[3] S. L. Harris, D. M. Harris, D. Chaver, R. Owen, Z. L. Kakakhel, E. Sedano, Y. Panchul, B. Ableidinger, MIPSfpga: Using a commercial MIPS soft-core in computer architecture education, IET Circuits, Devices and Systems 11 (4) (2017) 283–291. doi:10.1049/IET-CDS.2016.0383.

[4] D. Mandalidis, P. Kenterlis, J. N. Ellinas, A Computer Architecture Educational System based on a 32-bit RISC Processor, in: 1^{st} International Scientific Conference eRA, 2006.
URL https://api.semanticscholar.org/CorpusID:114513798

[5] D. Benitez, Learning the relationship between computer architecture and technology by reconfiguring, in: Proceedings of the 1998 Workshop on Computer Architecture Education, WCAE '98, Association for Computing Machinery, New York, NY, USA, 1998, pp. 30–es. doi:10.1145/1275182.1275212.

[6] P. Minev, V. Kukenska, I. Varbov, M. Dinev, A Practical Computer Architecture Education with RISC-V and TL-Verilog, in: 2023 XXXII International Scientific Conference Electronics (ET), 2023, pp. 1–6. doi:10.1109/ET59121.2023.10279744.

[7] Intel Corporation, Nios V Processor Reference Manual, Tech. rep., Intel Corporation (2023).

[8] RISC-V International, Learn RISC-V, https://github.com/riscv/learn?tab=readme-ov-file (2024).

[9] P. Pisa, J. Dupak, K. Koci, QtRVSim – RISC-V Simulator for Computer Architectures Classes, in: Embedded World 2022 Exhibition & Conference, 2022.

[10] D. Chaver, Computer Organization at UCM, https://github.com/artecs-group/RVfpga-sim-addons/tree/main/Computer_Organization (2024).

[11] S. L. Harris, D. Chaver, L. Piñuel, J. Gomez-Perez, M. H. Liaqat, Z. L. Kakakhel, O. Kindgren, R. Owen, RVfpga: Using a RISC-V Core Targeted to an FPGA in Computer Architecture Education, in: 2021 31^{st} International Conference on Field-Programmable Logic and Applications (FPL), 2021, pp. 145–150. doi:10.1109/FPL53798.2021.00032.

[12] European Education and Culture Executive Agency and Eurydice, The European higher education area in 2024 - Bologna process implementation report, Publications Office of the European Union, 2024. doi:doi/10.2797/483185.

[13] D. Chaver, Integrated systems architecture, https://github.com/artecs-group/RVfpga-sim-addons/tree/main/Integrated_Systems_Architecture (2024).

[14] FPGAacademy, A hub for learning about computer hardware with FPGAs (2024).
URL https://fpgacademy.org/index.html

[15] Altera, AN 978: Nios V Processor Migration Guidelines, Tech. rep., Altera Corporation (2024).

[16] D. Benitez, Computer Architecture hands-on exercises based on the Nios V soft processors (2024).
URL https://github.com/vipl-dbd/ComputerArchitecture_NiosV

[17] D. Benitez, Computer Architecture hands-on exercises based on the Nios II soft processors (2024).
URL https://github.com/vipl-dbd/ComputerArchitecture_NiosII