A RISC-V Soft-Core-Based Embedded Systems Course Using Artix-7 FPGA Platforms

Mr. Zhihao Pan, Texas A&M University

Zhihao Pan is currently a PhD candidate in the Department of Electrical and Computer Engineering at Texas A&M University in College Station, Texas, USA. He earned his M.S. degree in Engineering Technology and his B.S. degree in Electronic Systems Engineering Technology from Texas A&M University. His research interests include RISC-V processor architecture, embedded systems, FPGA-based design, and hardware security.

Dr. Byul Hur, Texas A&M University

Dr. B. Hur received his B.S. degree in Electronics Engineering from Yonsei University, in Seoul, Korea, in 2000, and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Florida, Gainesville, FL, USA, in 2007 and 2011, respectively. In 2016, he joined the faculty of Texas A&M University, College Station, TX. USA, where he is currently an Associate Professor. He worked as a postdoctoral associate from 2011 to 2016 at the University Florida previously. His research interests include Mixed-signal/RF circuit design and testing, measurement automation, environmental & biomedical data measurement, and educational robotics development.

Dr. Wei Zhan, Texas A&M University

Dr. Wei Zhan is a Professor of Electronic Systems Engineering Technology and Associate Department Head for Undergraduate Studies at Texas A&M University. Dr. Zhan earned his D.Sc. in Systems Science and Mathematics from Washington University in St. Louis in 1991. From 1991 to 1995, he worked at University of California, San Diego and Wayne State University as a postdoctoral researcher and visiting assistant professor, respectively. From 1995 to 2006, he worked in the automotive industry as a system engineer. In 2006 he joined the Electronic Systems Engineering Technology faculty at Texas A&M. Dr. Zhan holds a joint appointment position in the department of Electrical and Computer Engineering at Texas A&M University. He is an American Society for Quality certified Six Sigma Black Belt and an ASQ Fellow. His research activities include control system theory and applications to industry, fault prognosis, system engineering, robust design, modeling, simulation, quality control, and optimization.

A RISC-V Soft-Core-Based Embedded Systems Course Using Artix-7 FPGA Platforms

Abstract

RISC-V is an open-standard Instruction Set Architecture (ISA) that enables engineers and educators across industries to learn and develop microcontroller and microprocessor architectures for a range of systems, including mobile, laptop, and server applications. A RISC-V soft-core processor can be either a complete or a prototype platform for an embedded computing system. Using FPGA boards, a soft-core RISC-V processor can be studied and learned through hands-on engineering methods. In this paper, the course content and hands-on laboratory examples using RISC-V soft-core systems on an Xilinx Artix-7 FPGA will be introduced. This work is the outcome of the RISC-V embedded system research and development effort, and the authors have found that a custom compact-sized RISC-V soft-core processor can be effectively studied in engineering education through experimentation and hands-on laboratories. This course can be an elective course or a training course for students who have already taken junior-level embedded systems courses or an engineering course for graduate students. It introduces the RISC-V soft-core processor for embedded systems using a Xilinx Artix-7 FPGA. Xilinx provides RISC-V softcore boards. A Xilinx Basys 3 board is a low-cost Artix-7 FPGA board suitable for FPGA education. This paper provides embedded system laboratory examples using an Artix-7 FPGA board, including GPIO and ADC labs. For the laboratories, a Xilinx Artix-7 FPGA board is used with a custom laboratory board. The designed and fabricated custom laboratory kit for the RISC-V board is introduced. This paper presents and discusses the educational values of the RISC-V embedded system.

1. Introduction

RISC-V is an open-standard Instruction Set Architecture (ISA) based on Reduced Instruction Set Computer (RISC) principles that has attracted significant attention in both industry and academia for its openness, simplicity, modularity, and extensibility from licensing restrictions [1-2]. Unlike proprietary ISAs, RISC-V allows educators and students to study, modify, and extend processor architectures without any constraints. These advantages make it particularly suitable for engineering education. With the growing adoption of RISC-V in embedded systems, edge computing, and high-performance platforms, there is an increasing need for education approaches that go beyond software-only exploration and provide students with hands-on experience in hardware implementation and system-level design.

Field-programmable gate arrays (FPGAs) provide an effective platform for this form of experiential learning. By implementing a RISC-V processor as a soft-core on an FPGA, students can directly observe how architectural concepts, including instruction execution, memory-mapped I/O, and peripheral integration, are realized at the hardware level. Compared to microcontroller-based education platforms, FPGA-based soft-core processors offer more significant architectural transparency. The advantages of openness enable students to engage with both the hardware and software layers of an embedded system. This hands-on experience is well-suited for upper-division undergraduate and graduate curricula in embedded systems, computer architecture, and digital design.

Several existing educational initiatives have demonstrated the value of RISC-V on FPGA platforms [3-9]. Notably, the RVfpga course offers a comprehensive instructional framework built around an industrial-scale RISC-V system-on-chip, which is a commercial RISC-V processor developed by Western Digital and hosted by CHIPS Alliance [9]. While such courses are practical for teaching software development, microarchitecture, and system integration, their scale and industrial feature richness can introduce a steep learning curve for students who are new to FPGA-based processor design. In an educational environment where time and prior hardware experience are limited, the complexity of highly optimized, production-grade cores may shift student effort toward understanding existing structures rather than emphasizing foundational architectural design and register-transfer (RTL) level design.

To address these educational considerations, this paper presents a course framework centered on a smaller, custom RISC-V soft-core processor implemented on a low-cost FPGA platform, the Basys 3 [10]. The Basys 3 board, based on a Xilinx Artix-7 FPGA, is widely utilized in undergraduate digital design courses and offers adequate resources, onboard peripherals, and accessibility for educational laboratories. By deploying a compact-sized RISC-V soft-core on this board, the course emphasizes architectural clarity, modular design, and direct hardware interaction rather than peak performance or industrial-feature completeness.

This paper introduces the structure and content of a RISC-V embedded systems course that employs a custom soft-core processor on an Artix-7 FPGA. The course is designed either as an elective for students who have completed junior-level embedded systems coursework or as a graduate-level engineering course. The paper outlines the motivation, course structure, and learning objectives for the course, details the hardware and software framework, and describes representative laboratory activities, including general-purpose input/output (GPIO) and analog-to-digital converter (ADC) experiments. Additionally, the paper discusses the anticipated educational value of this approach and presents an assessment plan suitable for future course deployment.

2. Background and Motivation

2.1 RISC-V in Computer Engineering Education

RISC-V is an open and modular ISA originally developed to support research and education in computer architecture. Its open specification and extensible design set it apart from proprietary ISAs, enabling instructors and students to study processor behavior without licensing barriers or unclear implementations. These features make RISC-V particularly appealing for engineering education, where transparency and modifiability are essential for gaining a deep understanding of processor design and embedded system integration.

From an educational standpoint, RISC-V allows instruction at multiple levels of abstraction. At the software level, students can learn assembly language programming, calling conventions, and compiler behavior using standard GNU toolchains [11]. At the hardware level, the same ISA can be implemented as a soft-core processor, facilitating exploration of datapath design, control logic, and memory systems. This vertical integration, from ISA specification to RTL implementation and physical deployment, aligns well with the learning objectives in embedded systems, digital design, and computer architecture courses.

As RISC-V adoption continues to grow in the industry, familiarity with it also provides practical value for students preparing for careers in hardware and embedded software development. Therefore, many institutions have begun including RISC-V in their undergraduate and graduate curricula, either as a replacement for or a complement to traditional architectures such as MIPS (Microprocessor without Interlocked Pipeline Stages) or ARM [12-13].

2.2 FPGA-Based RISC-V Teaching Platforms

FPGAs (Field-Programmable Gate Arrays) are commonly used in engineering education to connect theoretical concepts with practical hardware implementation. When paired with a RISC-V soft-core processor, FPGAs offer a flexible platform for hands-on learning, allowing students to observe and modify processor behavior in real time. Unlike commercial off-the-shelf microcontrollers, FPGA-based soft-core processors enable instructors to highlight architectural details that are typically covered in class, such as instruction decoding, register file organization, and memory-mapped I/O.

Several FPGA-based RISC-V teaching platforms have been proposed in the literature [3-9]. One notable example is RVfpga, which offers a comprehensive set of instructional materials centered around a commercial-grade RISC-V system-on-chip [9]. RVfpga emphasizes software development, system-level integration, and microarchitectural exploration using both simulation and FPGA deployment. Its extensive laboratory sequence illustrates the feasibility of teaching advanced topics such as pipelining, hazards, caching, and peripheral expansion using a realistic RISC-V core.

While these platforms are effective for advanced instruction, their complexity can pose challenges in introductory or intermediate courses. Large industrial cores are often highly optimized and may depend on extensive infrastructure, making it difficult for students to see the connection between ISA concepts and their RTL implementation. Consequently, students may primarily interact with software and pre-existing hardware blocks, limiting their ability to modify and fully understand the underlying processor design.

2.3 Motivation for a Compact-Sized Custom RISC-V Soft-Core

To address the need for architectural clarity, several researchers have proposed compact, educational RISC-V processor implementations specifically designed for FPGA platforms [15-18]. These designs typically incorporate a subset of the RISC-V ISA, such as RV32I, and emphasize readability, modularity, and simplicity, which are ideal for instructional laboratories focused on fundamental concepts.

Custom RISC-V soft cores allow students to explore essential questions, including how instructions are fetched and executed, how control signals are generated, and how data flows through a processor datapath. Additionally, since these designs are often programmed using standard Verilog or SystemVerilog, students with prior digital design experience can easily analyze and modify them. This hands-on approach aligns with educational goals that prioritize learning through construction and experimentation, rather than the use of pre-built systems.

2.4 FPGA Platforms for Educational Environment

Selecting an FPGA platform is essential to ensuring both accessibility and scalability for a

RISC-V-based course. While high-end development boards may provide advanced features, they can be expensive and challenging to integrate into standard teaching laboratories. In contrast, low-cost boards commonly used in undergraduate programs offer a practical alternative.

A prime example of the needed FPGA platform is the Basys 3 board (Xilinx Artix-7 FPGA) [10]. It provides sufficient logic resources and on-board peripherals to implement a small RISC-V soft-core processor. Its widespread use in digital logic and FPGA courses lowers the barrier to entry for incorporating processor-based laboratories, as students are often already familiar with both the hardware and the development environment.

2.5 RISC-V Embedded Systems Course Framework

This paper presents a course framework centered on a small, custom RISC-V soft-core processor as the primary teaching tool. It targets the low-cost Artix-7 FPGA platform and focuses on creating modular, easily understandable RTL designs. The course aims to strike a balance between architectural transparency and practical application. Existing platforms, such as RVfpga, provide valuable context and inspiration. At the same time, research experiences with SoC-based RISC-V systems contribute to the overall educational framework. Together, these elements promote a course that emphasizes hands-on experimentation, hardware-software integration, and accessibility for upper-division undergraduate and graduate students in computer engineering.

3. Course Structure and Learning Objectives

3.1 Course Structure

The RISC-V embedded systems course is designed to fit flexibly within an engineering curriculum. It can be offered as an upper-division elective embedded systems course or as a graduate-level special topics or training course, depending on the institution's needs and the program's structure. In either case, the course aims to complement existing offerings in embedded systems, computer architecture, and digital design by emphasizing hands-on processor implementation and hardware–software integration.

At the undergraduate level, the course is most suited for junior or senior students who have completed foundational coursework in embedded systems and digital logic. In this context, it serves as a bridge between microcontroller programming courses and more advanced topics in computer architecture and system-on-chip design. By focusing on a RISC-V soft-core implemented on an FPGA board, students gain exposure to architectural concepts that are often abstracted in traditional embedded systems courses.

At the graduate level, the course may be offered as a special topic or training course for students in electrical engineering, computer engineering, or related fields. In this setting, the course emphasizes architectural exploration, customization, and experimentation, making it ideal for students preparing for research in processor design, hardware security, or hardware–software co-design. The modular structure of the laboratories also allows the course to be adapted for short-term training or focused workshops centered on RISC-V and FPGA-based systems.

To ensure that students are adequately prepared for the technical depth of the material, the course has the following prerequisites:

- An introductory embedded systems course, including experience with C programming and basic peripheral interaction.

- Prior exposure to digital logic or computer organization, covering concepts such as combinational and sequential circuits, finite state machines, and instruction execution.

These prerequisites enable the course to focus on processor-level design and integration, rather than introductory programming or logic fundamentals.

3.2 Learning Objectives

The course is organized around a set of learning objectives that focus on both conceptual understanding and practical implementation skills. Upon successful completion, students are expected to demonstrate their knowledge in the following areas.

First, students will develop a solid understanding of the RISC-V ISA fundamentals at both the register-transfer and system levels. The solid understanding includes comprehension of instruction formats, datapath operations, control logic, and the interaction between software instructions and hardware execution. By working directly with a RISC-V soft-core processor, students gain insight into how ISA specifications translate into actual hardware structures.

Second, students will be able to implement and modify a RISC-V soft-core processor on an FPGA board. Through guided laboratory sessions, students will interact with a modular processor design, examine its components, and make targeted modifications. This objective emphasizes architectural transparency and reinforces the connection between theoretical processor concepts and their RTL realization.

Third, students will learn to design, integrate, and test memory-mapped peripherals within a RISC-V-based embedded system. Laboratory exercises will introduce general-purpose input/output and analog-to-digital conversion as representative peripherals, enabling students to understand address decoding, register interfaces, and hardware validation. These activities reflect everyday tasks in embedded system design while remaining accessible in an educational setting.

Lastly, students will gain experience in hardware-software co-design using C and assembly languages. By writing bare-metal software that interacts directly with custom hardware, students will learn to synchronize software control flow with hardware behavior. This objective highlights the interdisciplinary nature of embedded systems. It prepares students for real-world engineering scenarios where software and hardware must be developed together.

In combination, these learning objectives support a course structure that emphasizes experimentation, architectural insight, and practical skill development. The combination of a compact-sized RISC-V soft-core, FPGA-based laboratories, and integrated software development provides a cohesive framework for teaching modern embedded system design in an academic environment.

4. RISC-V Soft-Core Platform and FPGA Environment

4.1 Hardware Platform Selection

The hardware platform chosen for this course is the Basys 3, a cost-effective FPGA development board based on the Xilinx Artix-7 device [10]. This board includes a 4-digit 7-segment display, 16 switches and LEDs, 5 pushbuttons, and various sensors and connectors, including a VGA output, a USB port, and ADC ports [10]. The Basys 3 board was selected primarily due to its accessibility, suitability for educational purposes, and compatibility with existing undergraduate laboratory infrastructure. Its reasonable cost makes it an ideal option for classroom use, individual student projects, or shared laboratory environments, which is essential for scalable implementation in academic programs.

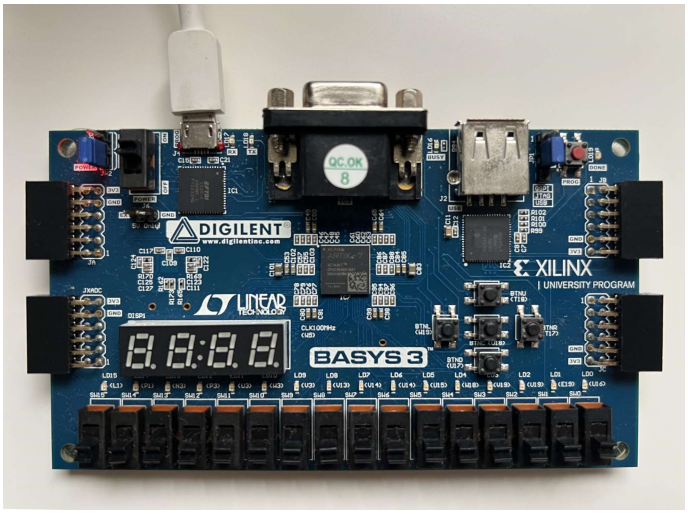

Figure 1. Basys 3 AMD Artix™ 7 FPGA Trainer Board [10]

In addition to its affordability, the Basys 3 board is widely available. It has already been integrated into many digital logic and FPGA-related courses. Consequently, students are often familiar with its physical layout, input/output peripherals, and development workflow before enrolling in the proposed course. This prior knowledge helps reduce the learning curve associated with new hardware. It allows the course to focus on processor architecture and embedded system concepts rather than on hardware setup.

The board offers a comprehensive array of on-board I/O resources, which are sufficient for implementing and validating memory-mapped I/O laboratories without additional external hardware. This capability enables students to directly observe processor behavior and peripheral interactions in real-time, reinforcing the connection between hardware design, software execution, and physical system responses. Overall, these features make the Basys 3 an effective and practical platform for deploying a RISC-V soft-core processor in the classroom.

4.2 Soft-Core Architecture

The course centers around a custom RISC-V soft-core processor specifically designed for educational

purposes. This processor implements a subset of the RISC-V ISA, focusing on the RV32I base integer instruction set. This design choice ensures compatibility with standard RISC-V toolchains while keeping architectural complexity at a level suitable for students who are implementing a processor on an FPGA for the first time.

The processor architecture uses either a single-cycle design or a simplified multi-stage design, depending on the laboratory configuration. In its simplest form, each instruction is executed in a single clock cycle, enabling students to easily trace the flow of instructions through the datapath and control logic. This approach emphasizes conceptual clarity and helps students relate instruction meaning to hardware behavior. The simplified design also lays the groundwork for future exploration into multi-cycle or pipelined execution models, should deeper instructional content be desired.

The overall design principles prioritize educational clarity over peak performance. The instruction set is intentionally compact-sized, avoiding advanced extensions or microarchitectural optimizations that might obscure fundamental concepts. The processor is described using modular, readable SystemVerilog code, with a clear separation of functional blocks, including the register file, arithmetic logic unit, control logic, and memory interface. This modular structure allows students to study individual components in isolation and understand their interactions at the system level.

By emphasizing simplicity, transparency, and modifiability, the soft-core processor serves as an adequate teaching tool. Students are encouraged not only to use the processor but also to study and modify its internal structure, reinforcing their understanding of computer architecture and embedded system design.

5. Laboratory Design and Implementation

The laboratory component of the course aims to reinforce architectural concepts through hands-on experimentation. Each laboratory exercise is designed to build directly on the custom RISC-V soft-core platform described in the section on RISC-V Soft-Core Platform and FPGA Environment and to highlight the visible relationships between software instructions and physical hardware. The laboratory component of the course is organized as a progressive sequence that moves from fundamental processor interaction to system-level integration. Each laboratory builds upon the previous one, reinforcing architectural concepts through hands-on FPGA implementation while maintaining accessibility for upper-division undergraduate and graduate students.

Table 1. List of Laboratories for the Course

Laboratories	Description
1	Introduction to the RISC-V Soft-Core and FPGA Environment
2	Memory-Mapped GPIO Laboratory
3	Timer and Simple Control Peripheral
4	ADC and Peripheral Extension Laboratory
5	Software-Hardware Co-Design
6	System Integration and Peripheral Design

7	Performance Observation and Architectural Exploration
Term Project	RISC-V Embedded System Design Project

Next, the general-purpose input/output (GPIO) and analog-to-digital converter (ADC) laboratory examples are described in detail.

5.1 Memory-Mapped GPIO Laboratory

The Memory-Mapped GPIO laboratory is an example of introducing students to general-purpose input/output (GPIO) using a memory-mapped I/O model. In this exercise, GPIO registers are mapped into the RISC-V processor's address space, allowing software to interact with external hardware through standard load and store instructions. This approach reinforces the concept that peripheral devices in embedded systems are accessed using the exact mechanisms as memory, a foundational principle in modern processor-based design.

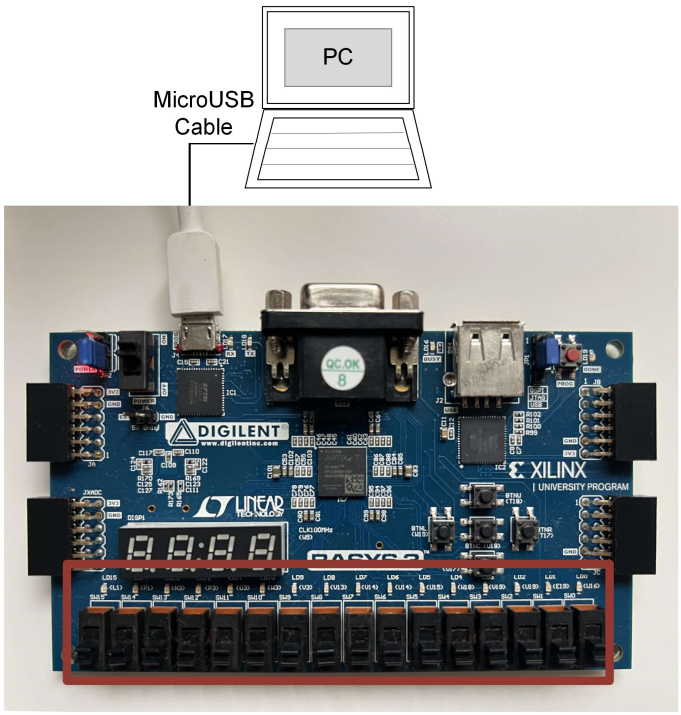

Figure 2. Basys 3 Board - Memory-Mapped GPIO Laboratory

As shown in Figure 2, the onboard LEDs and switches enclosed in red boxes are accessible to students to implement basic input and output functionality. Switch states are read through memory-mapped input registers, while LED patterns are controlled by writing values to output registers. These interactions enable students to directly observe the effects of their software execution on physical hardware, providing immediate feedback and strengthening the connection between instructions, datapath behavior, and external signals.

At the instruction level, students analyze how RISC-V load and store instructions translate into hardware activity within the soft-core processor. By stepping through simple programs and observing

the LED and switch behavior, students gain insights into address decoding, register access, and the timing of memory-mapped operations. This laboratory lays the foundation for subsequent laboratories by linking conceptual architectural concepts to actual system behavior.

5.2 ADC and Peripheral Extension Laboratory

Building on the GPIO laboratory, the second laboratory introduces peripheral expansion with the analog-to-digital converter (ADC). This laboratory utilizes an external custom-designed board connected to the FPGA, enabling students to explore analog input acquisition in a RISC-V-based embedded system. The ADC interface is made accessible to the processor through a memory-mapped register set, consistent with the design principles introduced in the GPIO laboratory.

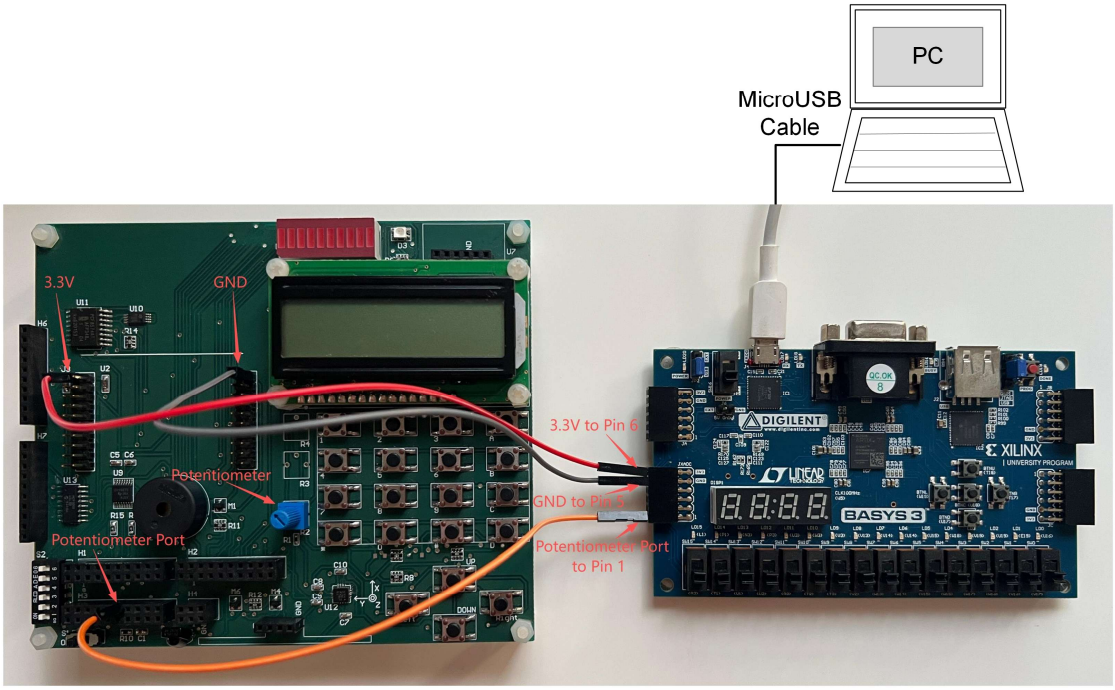

Figure 3. Basys 3 Board - ADC and Peripheral Extension Laboratory

As shown in Figure 3, several pins of the external custom-designed board are connected to the Basys 3 FPGA board. The 3.3V (voltage) pin is connected to Pin 6 of the Analog signal Pmod connector (XADC). The GND (ground) pin is connected to Pin 5 of the XADC. The potentiometer port is connected to Pin 1 of the XADC. The potentiometer is the blue knob on the external custom-designed board. Students design and integrate additional hardware logic to interface the ADC with the RISC-V soft-core. They examine how analog signals are sampled, converted to digital values, and made available to software through the processor's address space. As students adjust the potentiometer (the blue one, connected by an orange jumper) on the BH Board, the digital value will continuously update to reflect these changes. The red jumper is used to supply power (3.3V) to the BH Board, while the gray jumper serves as the ground connection. The 7-segment display can display the converted digital value. This process emphasizes system-level design considerations, including peripheral timing, data-width adaptation, and register-interface definition.

The ADC laboratory also provides an opportunity to compare FPGA-based soft-core designs with system-on-chip (SoC) platforms that integrate hard processor cores and peripherals. In contrast, the custom soft-core approach requires students to explicitly define and implement the peripheral interface, offering greater transparency and reinforcing their architectural understanding. This comparison highlights the trade-offs between educational clarity and system complexity, helping students to notice the differences between instructional platforms and commercial SoC devices.

6. Educational Outcomes and Observations

The course has been developed as part of an ongoing research and development initiative. Also, this course has not been taught in class yet. Therefore, the assessment framework outlined in this section is both planned and prospective. This section aims to present a structured evaluation strategy applicable to future course offerings, clearly conveying the anticipated educational benefits of this approach in line with ASEE education research practices.

6.1 Assessment Plan

The assessment plan combines formative and summative details to evaluate students' conceptual understanding and practical skills. The design emphasizes alignment between course learning objectives and observable student outcomes.

On a conceptual level, the authors will implement pre-course and post-course concept checklists to measure students' gains in understanding RISC-V architecture and embedded system fundamentals. These checklists will assess topics such as instruction execution, datapath behavior, memory-mapped I/O, and the relationship between software instructions and hardware implementation. Distributing the same or closely aligned mechanisms before and after the course will allow for a quantitative comparison of learning gains.

At the laboratory level, the authors will use lab check-offs and demonstrations as a primary assessment mechanism. In these check-offs, students will demonstrate the correct function of their RISC-V soft-core system and associated peripherals on the FPGA board. Evaluation will focus on functional correctness, understanding of design decisions, and the ability to explain observed system behavior. This approach emphasizes knowledge of hands-on skills and ensures that students engage directly with the hardware platform.

Additionally, the authors will require reports and reflective assignments to capture deeper learning outcomes. Written reports will document laboratory design choices, implementation details, and debugging processes. The reflective assignments will encourage students to articulate what they learned and how their understanding evolved throughout the course. These artifacts will provide qualitative insights into student thinking and complete the quantitative assessment measures.

6.2 Student Learning Scenarios

Based on prior experience as a laboratory instructor in embedded systems courses, the author anticipates several hypothetical student learning scenarios for this course. These scenarios describe expected patterns of student progression and inform both instructional design and assessment

planning.

Students are expected to begin the course with a foundational understanding of the RISC-V ISA at the software level, typically acquired through prior exposure to assembly programming or computer organization coursework. Early laboratory exercises reinforce the foundation by translating the concepts into observable hardware behavior, including data flow between registers, logic operations, and memory-mapped I/O interactions.

As the course progresses, students are expected to develop RTL-level comprehension of the RISC-V soft-core. Through guided exploration of SystemVerilog modules, students will gain an understanding of its internal structure, including the datapath, control logic, and memory interfaces. The deeper understanding is a critical learning milestone and often represents a shift from software-centric thinking to hardware–software co-design.

In later stages of the course, students are expected to achieve system-level integration skills. This includes modifying the soft-core or peripheral logic, integrating new hardware components, and writing software that reliably interacts with custom hardware. At this stage, students demonstrate the ability to reason across abstraction boundaries and debug issues arising from the interaction between hardware and software.

Several challenges can be anticipated in this process. Students may initially struggle with RTL code comprehension or with relating high-level software behavior to low-level hardware signals. To mitigate these challenges, the course design emphasizes modular hardware structure, incremental laboratory complexity, and extensive use of physical I/O for feedback. By allowing students to observe system behavior directly through LEDs, switches, and other FPGA outputs, abstract architectural concepts become tangible, observable outcomes.

6.3 Educational Value Metrics

The educational value of the course will be assessed using a combination of qualitative and quantitative metrics that reflect both student engagement and learning outcomes. One important metric is student interaction with the hardware. The use of an FPGA-based RISC-V soft core is expected to enhance student motivation and sustained involvement compared to instruction that relies only on simulators [14]. This increase in engagement can be measured through attendance, the time students spend on laboratory tasks, and feedback collected from surveys or reflective assignments.

Another key metric is students' ability to modify processor components. Successful completion of laboratory tasks that require changes to the soft core or peripheral logic demonstrates that students have learned the material. The extent and accuracy of these modifications provide concrete evidence of their understanding of architectural concepts and design skills.

Finally, the course will support a comparative evaluation against simulator-only instruction. Simulators can be accessible and convenient and often abstract hardware constraints and limit physical interaction. The assessment framework from the course will encourage future comparisons of learning outcomes between FPGA-based instruction and simulation-only instruction, particularly regarding student understanding of timing, input/output behavior, and system integration.

Together, these metrics create a comprehensive evaluation framework that can align with educational goals. Although the assessment plan is prospective, it offers a clear structure for future data collection. It establishes a foundation for evaluating the point of FPGA-based RISC-V instruction in embedded systems education.

Conclusions

This paper presented the design and planned implementation of a RISC-V–based embedded systems course centered on a small, custom RISC-V soft-core processor deployed on a low-cost Artix-7 FPGA platform. Motivated by the openness, simplicity, modularity, and extensibility of the RISC-V instruction set architecture, the course framework emphasizes hands-on learning through direct interaction with processor hardware, rather than simulation or commercial hardware. By selecting the Basys 3 FPGA board as the primary hardware platform, the course balances accessibility, affordability, and technical capability. The laboratory sequence described in this work progresses from basic memory-mapped GPIO interaction to more advanced peripheral integration, including analog-to-digital conversion and software–hardware co-design. At the same time, the course is designed to support embedded systems education at both the upper-division undergraduate and graduate levels, providing students with a coherent way from architectural fundamentals to system-level understanding.

References

1. K. Asanović and D. A. Patterson, “Instruction sets should be free: The case for RISC-V,” EECS Department, University of California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2014-146, 2014.
2. A. Waterman and K. Asanović, *The RISC-V Instruction Set Manual, Volume I: User-Level ISA*, Version 2.2, RISC-V Foundation, 2019.
3. RISC-V International, “University resources,” [Online]. Available: https://riscv.org/learn/. Accessed: Apr. 2026.
4. R. Agrawal, S. Bandara, A. Ehret, M. Isakov, M. Mark, and M. Kinsy, “The BRISC-V Platform: A Practical Teaching Approach for Computer Architecture”, in *Proceedings of the Workshop on Computer Architecture Education*, pp. 1-8, Jun. 2019. DOI: 10.1145/3338698.3338891.
5. S. L. Harris *et al.*, “RVfpga: Using a RISC-V core targeted to an FPGA in computer architecture education,” in *Proc. 31st Int. Conf. Field-Programmable Logic Appl. (FPL)*, Dresden, Germany, 2021, pp. 145–150, doi: 10.1109/FPL53798.2021.00032.
6. S. L. Harris *et al.*, “RVfpga: Computer architecture course and MOOC using a RISC-V SoC targeted to an FPGA and simulation,” in *Proc. ASEE Annu. Conf. Expo.*, Baltimore, MD, USA, Jun. 2023, doi: 10.18260/1-2--44172.
7. M. Li, K. Garcia-Estala, and I. Gutierrez, “An FPGA-based toolchain for computer architecture courses,” in *Proc. ASEE Annu. Conf. Expo.*, Montreal, QC, Canada, Jun. 2025, Paper 57616, doi: 10.18260/1-2--57616.
8. R. Thompson, D. L. Harris, J. Stine Jr., and S. L. Harris, “RISC-V system-on-chip design textbook and course,” in *Proc. ASEE Annu. Conf. Expo.*, Montreal, QC, Canada, Jun. 2025, Paper 57147, doi: 10.18260/1-2--57147.
9. D. Chaver *et al.*, “The RISC-V FPGA (RVfpga) teaching package,” *IEEE Access*, vol. 14, pp. 18455–18475, 2026, doi: 10.1109/ACCESS.2026.3658743.
10. Digilent Inc., “Basys 3 Artix-7 FPGA trainer board,” [Online]. Available: https://digilent.com/shop/basys-3-amd-artix-7-fpga-trainer-board/. Accessed: Apr. 2026.
11. RISC-V Collab, “RISC-V GNU toolchain,” GitHub repository. [Online]. Available: https://github.com/riscv-collab/riscv-gnu-toolchain. Accessed: Apr. 2026.
12. MIPS Technologies, “MIPS architecture,” [Online]. Available: https://mips.com/. Accessed: Apr. 2026.
13. Arm Ltd., “ARM CPU architecture,” [Online]. Available: https://www.arm.com/architecture/cpu. Accessed: Apr. 2026.
14. N. Binkert *et al.*, “The gem5 simulator,” *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, May 2011, doi: 10.1145/2024716.2024718.
15. C. Wolf, “PicoRV32—A size-optimized RISC-V CPU,” GitHub repository, 2015. [Online]. Available: https://github.com/cliffordwolf/picorv32. Accessed: Apr. 2026.

16. K. Asanović *et al.*, “The Rocket Chip generator,” EECS Dept., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2016-17, 2016.
17. L. Poli, S. Saha, X. Zhai, and K. D. Mcdonald-Maier, “Design and implementation of a RISC-V processor on FPGA,” in *Proc. 17th Int. Conf. Mobility, Sensing Netw. (MSN)*, Exeter, U.K., 2021, pp. 161–166, doi: 10.1109/MSN53354.2021.00037.
18. A. N. Ben Yousuf, D. Khaled Alamen, and M. M. Eljhani, “Design and implementation of five-stage pipelined RISC processor on FPGA,” in *Proc. IEEE 3rd Int. Maghreb Meeting Conf. Sci. Techn. Autom. Control Comput. Eng. (MI-STA)*, Benghazi, Libya, 2023, pp. 300–304, doi: 10.1109/MI-STA57575.2023.10169818.