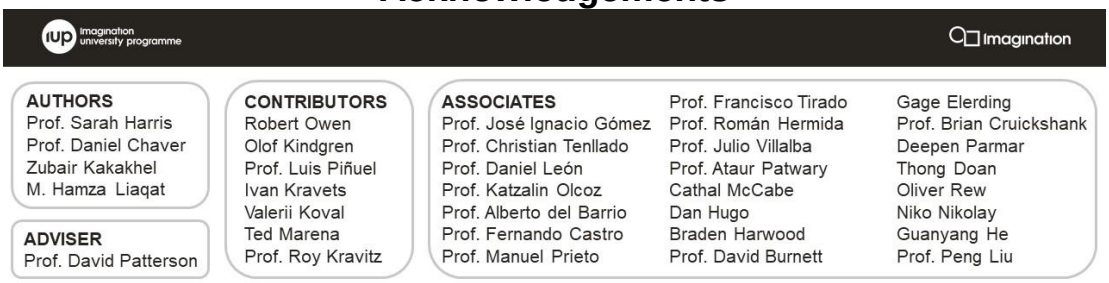
THE IMAGINATION UNIVERSITY PROGRAMME

Overview of the RVfpga Materials

Acknowledgements



Sponsors and Supporters

AUTHORS

- Prof. Sarah Harris (https://www.linkedin.com/in/sarah-harris-12720697/)
- Prof. Daniel Chaver (https://www.linkedin.com/in/daniel-chaver-a5056a156/)
- Zubair Kakakhel (https://www.linkedin.com/in/zubairlk/)
- M. Hamza Liaqat (https://www.linkedin.com/in/muhammad-hamza-liaqat-ab73a0195/)

ADVISER

- Prof. David Patterson (https://www.linkedin.com/in/dave-patterson-408225/)

CONTRIBUTORS

- Robert Owen (https://www.linkedin.com/in/robert-owen-4335931/)
- Olof Kindgren (https://www.linkedin.com/in/olofkindgren/)
- Prof. Luis Piñuel (https://www.linkedin.com/in/lpinuel/)
- Ivan Kravets (https://www.linkedin.com/in/ivankravets/)
- Valerii Koval (https://www.linkedin.com/in/valeros/)
- Ted Marena (https://www.linkedin.com/in/tedmarena/)
- Prof. Roy Kravitz (https://www.linkedin.com/in/roy-kravitz-4725963/)

ASSOCIATES

- Prof. José Ignacio Gómez (https://www.linkedin.com/in/jos%C3%A9-ignacio-gomez-182b981/)
- Prof. Christian Tenllado (https://www.linkedin.com/in/christian-tenllado-31578659/)
- Prof. Daniel León (www.linkedin.com/in/danileon-ufv)
- Prof. Katzalin Olcoz (https://www.linkedin.com/in/katzalin-olcoz-herrero-5724b0200/)
- Prof. Alberto del Barrio (https://www.linkedin.com/in/alberto-antonio-del-barrio-garc%C3%ADa-1a85586a/)
- Prof. Fernando Castro (https://www.linkedin.com/in/fernando-castro-5993103a/)
- Prof. Manuel Prieto (https://www.linkedin.com/in/manuel-prieto-matias-02470b8b/)
- Prof. Francisco Tirado (https://www.linkedin.com/in/francisco-tirado-fern%C3%A1ndez-40a45570/)
- Prof. Román Hermida (https://www.linkedin.com/in/roman-hermida-correa-a4175645/)
- Prof. Julio Villalba (https://www.linkedin.com/in/julio-villalba-moreno-97474824)
- Cathal McCabe (https://www.linkedin.com/in/cathalmccabe/)
- Dan Hugo (https://www.linkedin.com/in/danhugo/)
- Braden Harwood (https://www.linkedin.com/in/braden-harwood/)
- David Burnett (https://www.linkedin.com/in/david-burnett-3b03778/)
- Gage Elerding (https://www.linkedin.com/in/gage-elerding-052b16106/)
- Brian Cruickshank (https://www.linkedin.com/in/bcruiksh/)
- Deepen Parmar (https://www.linkedin.com/in/deepen-parmar/)
- Thong Doan (https://www.linkedin.com/in/thong-doan/)
- Oliver Rew (https://www.linkedin.com/in/oliver-rew/)
- Niko Nikolay (https://www.linkedin.com/in/roy-kravitz-4725963/)
- Guanyang He (https://www.linkedin.com/in/guanyang-he-5775ba109/)
- Prof. Ataur Patwary (https://www.linkedin.com/in/ataurpatwary/)
- Prof. Peng Liu (https://person.zju.edu.cn/liupeng)

Update History:

- Version 1.0 (Released November 2020):
  - Original release of the RVfpga course.

- Version 1.1 (Released June 2021):
  - Added description of Labs 11-20 in Lab 0.
  - Updated SweRVolf version to 0.7.3 and Verilator version to 4.106.
  - Added Boot ROM initialization program.
  - Added new Figure 1 and Table 1 in the GSG describing the RVfpga System
  - Added a UART exercise to Lab 10.
  - Fixed some typos.

- Version 2.0 (Released November 2021):
  - Added Labs 11-20: documents, figures, software sources, exercises and solutions.
  - Extended the slides with the new labs.
  - Added some minor things in the GSG and Labs 0-10, and fixed some typos.

- Version 2.1 (Released February 2022):
  - Renumbered Labs 1-5: moved Lab 1 to Lab 5 and renumbered Labs 2-5 as 1-4.

- Version 2.2 (Released May 2022):
  - Added *Workshop_Guide* document, which gives a guideline of a one-day workshop on RVfpga.
  - Added sources (such as some PlatformIO projects and Verilator binaries) required for the one-day workshop.
  - Added *ReadmeFirst* document (the one that you are reading right now), which enables ease of use for everyone. Removed Lab 0; most of this material is moved to the *ReadmeFirst* document.
  - Factored all the documents (GSG, slides, the IUP brochure, license agreement, and workshop guide) into a Documents folder.
  - Updated IUP brochure and license agreement.
  - Modifications in the slides.
  - Added some minor things in the GSG and Labs 1-20, and fixed some typos.

0. PREFACE

This RVfpga course in Computer Architecture provides hands-on understanding of a commercial RISC-V processor, RISC-V SoC, and the RISC-V ecosystem. The course provides an understanding of the system from the underlying digital design and signals to the instruction set architecture and processor to the programming environment, boot code, and compiler. RVfpga users walk away with a top-to-bottom understanding of the RISC-V system. After completing the RVfpga course, users will not only have a working RISC-V SoC and ecosystem, but they know how to use and expand the RISC-V processor and system for future projects and research.

Professor David Patterson, who shared the ACM Turing Award with John Hennessy for their contribution to RISC, says,

"RISC-V is transforming processor design and software/hardware co-design. RISC-V is an open architecture, which enables open-source hardware implementations. This new option means that software development can occur alongside hardware development, accelerating the design path. The RVfpga course enhances the understanding of not only RISC-V processors but also the RISC-V ecosystem and RISC-V SoCs. This course provides a deep understanding of an industrial-strength processor architecture and system of increasing popularity, which will prove useful throughout their academic and industry careers."

1. STRUCTURE OF THE RVfpga FOLDER

IMPORTANT: Before starting this course, copy the **RVfpga** folder that you downloaded from Imagination's University Programme to your Ubuntu/Windows/macOS machine. We will refer to the absolute path of the directory where you place this RVfpga folder as [RVfpgaPath].

The RVfpga folder includes the following elements:

1. **ReadmeFirst** (document): the document that you are currently reading describes the organization and main contents of the RVfpga materials.

   IMPORTANT: You should read the **ReadmeFirst** document completely before starting to use RVfpga.

2. **Documents** (folder): contains:
   a. **RVfpga_GettingStartedGuide** (GSG), which introduces the system and tools used in the RVfpga course. It is briefly described below (Section 2).
   b. **RVfpga_Slides**, which are the slides for the GSG and the Labs.
   c. **Workshop_Guide**, which includes the instructions that the attendees of a one-day RVfpga Workshop should follow.
   d. **IUP_Brochure** and **TeachingMaterial_LicenseAgreement**, which describe Imagination Technologies' teaching materials packages and license.
   e. **Figures_GSG** (folder): Figures used in the GSG document.

3. **examples** (folder): contains example programs that you will run while using the Getting Started Guide.

4. **src** (folder): contains the source code (Verilog and SystemVerilog) for the RVfpga System.

5. **verilatorSIM** (folder): contains the scripts for running the simulation of RVfpgaSim in Verilator.

6. **driversLinux_NexysA7** (folder): contains the Linux drivers for the Nexys A7 FPGA board.

7. **Labs** (folder): contains instructions, programs, and solutions that you will use during RVfpga Labs 1-20. This folder contains several subfolders:
    a. **Lab1, Lab2, … , Lab19, Lab20** (folders): Instructions and resources to be used while completing the labs. Note that each of the 20 labs has an instruction document that is located within the Labs directory under the specific lab's folder. For example, the instructions for Lab 1 are in [RVfpgaPath]/Labs/Lab01/RVfpga_Lab01. These lab documents give the instructions, examples, tasks, exercises, and figures for each of the 20 RVfpga Labs.
    b. **RVfpgaLabsFigures** (folder): Figures used in the lab documents.
    c. **RVfpgaLabsSolutions** (folder): A subset of exercise solutions for each of the labs.
        i. **ProgramsAndDocuments** (folder): documents and software with the solutions for the proposed tasks and exercises.
        ii. **Modified_RVfpgaSystem** (folder): Modified RVfpga System source code (Verilog and SystemVerilog) extended as guided by the exercises in Labs 6-10 and in Lab 18. Both different platformIO projects and the source code and the bitstreams for the SoC (that you can directly use on the FPGA) are provided.

> **IMPORTANT:** Instructors should remove folder **RVfpgaLabsSolutions** before distributing RVfpga to students.

2. RVfpga GSG OVERVIEW

The RVfpga Getting Started Guide (GSG) is the introductory document of the RVfpga materials. It is available at:

[RVfpgaPath]/RVfpga/Documents/RVfpga_GettingStartedGuide.docx,

> **IMPORTANT:** Before starting to work with the RVfpga Labs, you should complete the RVfpga Getting Started Guide.

The GSG is an extensive document that includes a Quick Start Guide, software installation instructions, an overview of the RISC-V architecture and RVfpga (including in-depth descriptions of the SweRVolf SoC and SweRV EH1 core), and instructions about how to write, simulate, and run programs on RVfpga both in simulation and, optionally, in hardware on the Nexys A7 FPGA board.

- **GSG Section 1** gives an overview of the GSG contents, RVfpga System, required software and optional hardware, and expected prior knowledge.
- **GSG Section 2** is the Quick Start Guide, which describes the minimal software installation needed for RVfpga and then shows how to download and execute a simple example program on RVfpga.
- **GSG Sections 3 and 4** give a brief introduction to the RISC-V computer architecture, the RVfpga SoC (similar to Section II.C of this document), and the organization of the Verilog and SystemVerilog files that make up the RVfpga system.

- **GSG Section 5** shows how to install all of the software tools needed to use RVfpga both in simulation and hardware.
- **GSG Section 6** shows how to use PlatformIO to both download the RVfpga SoC onto the Nexys A7 FPGA board and download and run several example programs on RVfpga.
- **GSG Sections 7 and 8** show how to simulate RVfpga source code (Verilog and SystemVerilog) using Verilator and how to simulate RISC-V code on the Whisper instruction set simulator (ISS), respectively.
- **Appendices:** The Getting Started Guide also includes appendices that show additional features such as how to use RVfpga at the command prompt in Linux and how to install the required tools on Windows and macOS machines.

3. RVfpga LABS OVERVIEW

The RVfpga Labs, listed in Table 1, provide hands-on understanding of RISC-V hardware and software. Before starting RVfpga Labs, you must complete the RVfpga Getting Started Guide. The RVfpga Labs materials are provided in the following folders:

- *[RVfpgaPath]/RVfpga/**Labs*** contains resources for each of the 20 labs, including a instructions document for each lab. These lab resources are contained within each of the subfolders: *Lab01*, *Lab02*, etc.
- *[RVfpgaPath]/RVfpga/Labs/**RVfpgaLabsSolutions*** contains a subset of exercise solutions for each of the 20 RVfpga Labs. This folder should be deleted before distributing the RVfpga package to students.

Table 1. RVfpga Labs

	#	Title
Part 1	1	C Programming
	2	RISC-V Assembly Language
	3	Function Calls
	4	Image Processing: Projects with C & Assembly
	5	Creating a Vivado Project
	6	Introduction to I/O
	7	7-Segment Displays
	8	Timers
	9	Interrupt-Driven I/O
	10	Serial Buses
Part 2	11	SweRV EH1 Configuration and Organization. Performance Monitoring
		SweRV EH1 Reference
	12	Arithmetic/Logical Instructions: `add`
	13	Memory Instructions: the `lw` and `sw` Instructions
	14	Structural Hazards
	15	Data Hazards
	16	Control Hazards. Branch Instructions: `beq` and the Branch Predictor
	17	Superscalar Execution
	18	Adding New Features (Instructions, Hardware Counters) to the Core
	19	Memory Hierarchy: The Instruction Cache (I$)
	20	I$, ICCM, DCCM, and Benchmarking

The labs are divided into two parts. Part 1 shows how to program RVfpga, build the source code in Vivado, and extend the RVfpga System to include additional peripherals. Part 2 focuses on the RISC-V core and memory system.

Specifically, Labs 1-10 (Part 1) show how to use the RISC-V SoC and toolchain (compilers and simulators), and they show how to add peripherals to the SoC. Specifically, Lab 1 shows how to analyse the SweRVolfX SoC source code, create an RTL project and generate a bitstream for the RVfpga on Nexys A7 board (Lab 1), how to run programs on RVfpgaNexys, RVfpgaSim and Whisper (Labs 2-5), and how to modify the RVfpga System to add peripherals (Labs 6-10).

Labs 11-20 (Part 2) focus on microarchitecture and memory hierarchy; they show how to understand the RISC-V pipeline and use or add features to the RISC-V core, including additional instructions, other branch predictors, and memory features. The RVfpga_SweRVref document is provided inside *[RVfpgaPath]/RVfpga/Labs/Lab11* and gives extra instructions on several topics: Sigasi Studio, Configuration of the SweRV EH1 processor, RVfpga System hierarchy of modules and their most relevant signals, Main structures/types for grouping control bits, RISC-V compressed instructions, Real Benchmarks.

These labs are well-suited for two-semester course for undergraduates. Labs 11-20 could also be taught to master's level students. Prior to completing this RVfpga course, students should understand the fundamentals of logic design, computer architecture, processor design, input/output systems and C/assembly programming. This material is covered in the textbook *Digital Design and Computer Architecture: RISC-V Edition*, Harris & Harris, © Morgan Kaufmann 2021.

4. RVfpga SOFTWARE AND HARDWARE

Table 2 lists the required software and optional hardware needed to use these labs. All of the software is free. The Nexys A7 FPGA board (or, equivalently, Nexys DDR FPGA board) is not required to complete the labs. Instead, all labs can be completed using Whisper (Western Digital's Instruction Set Simulator) and Verilator (an open-source HDL simulator).

Table 2. Required Software and Optional Hardware

Software	
Vivado 2019.2 WebPACK*	https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2019-2.html
VS Code	https://code.visualstudio.com/Download
PlatformIO	https://platformio.org/ (Installed within VSCode)
Verilator and GTKWave	https://github.com/verilator/verilator http://gtkwave.sourceforge.net/
Whisper (Western Digital's RISC-V Instruction Set Simulator)	https://github.com/chipsalliance/SweRV-ISS (Installed within PlatformIO)
RISC-V Toolchain and OpenOCD	https://github.com/riscv/riscv-gnu-toolchain, https://github.com/riscv/riscv-openocd (Installed within PlatformIO)
Hardware	
Nexys A7 FPGA Board*	https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/
RISC-V Core and System-on-Chip (SoC)	
Western Digital's SweRV EH1	https://github.com/chipsalliance/Cores-SweRV (included in RVfpga package)
SweRVolf	https://github.com/chipsalliance/Cores-SweRVolf (included in RVfpga package)

* optional

5. RVfpga 1-DAY WORKSHOP OVERVIEW

Two additional documents are included in the RVfpga package that can be used by instructors to help an instructor run an RVfpga course or run a 1-day RVfpga Workshop:

- *[RVfpgaPath]/RVfpga/Documents/**RVfpga_Slides.pptx***: These slides describe the entire RVfpga package: overview, installation, Getting Started Guide, labs, etc. The trainer should follow these in an RVfpga workshop.

- *[RVfpgaPath]/RVfpga/Documents/**Workshop_Guide.docx***: This document is the guide that the instructor will use and the attendees will follow in an RVfpga Workshop. It describes the workshop topics, demonstrations, and hands-on activities.