THE IMAGINATION UNIVERSITY PROGRAMME

Overview of the RVfpgaEL2 Materials

0. STRUCTURE OF THE RVfpgaEL2 MATERIALS

IMPORTANT: You should read this **ReadmeSecond_EL2** document completely before starting to use RVfpgaEL2.

The RVfpgaEL2 folder contains four subfolders, one for each of the platforms supported in the course:

- **RVfpga_Boolean**: Documents and sources for Real Digital's Boolean Board (https://www.realdigital.org/hardware/boolean)
- **RVfpga_Basys3**: Documents and sources for Digilent's Basys 3 Board (https://digilent.com/reference/programmable-logic/basys-3/start)
- **RVfpga_NexysA7-DDR**: Documents and sources for Digilent's Nexys A7 (or Nexys 4 DDR) Board (https://digilent.com/reference/programmable-logic/nexys-a7/start). In this configuration, the SoC uses the DDR Memory available on the board.
- **RVfpga_NexysA7-NoDDR**: Documents and sources for Digilent's Nexys A7 (or Nexys 4 DDR) Board (https://digilent.com/reference/programmable-logic/nexys-a7/start). In this configuration the SoC does not use the DDR Memory available on the board.

IMPORTANT: We will identify the absolute path of each subfolder by the board name; specifically:
- *[**RVfpgaBasysPath**]* refers to the absolute path to the RVfpga_Basys3 subfolder
- *[**RVfpgaBooleanPath**]* refers to the absolute path to the RVfpga_Boolean subfolder
- *[**RVfpgaEL2NexysA7DDRPath**]* refers to the absolute path to the RVfpga_NexysA7-DDR subfolder
- *[**RVfpgaEL2NexysA7NoDDRPath**]* refers to the absolute path to the RVfpga_NexysA7-NoDDR subfolder

The structure of the four subfolders is the same and the contents are analogous but adapted to each platform. Each subfolder contains the following elements:

1. **Documents** (folder): contains:
   a. **RVfpga_GettingStartedGuide** (GSG), which introduces the system and tools used in the RVfpgaEL2 course. It is briefly described below (Section 2).
   b. **Figures_GSG** (folder): Figures used in the GSG document.
   c. **IUP_Brochure** and **TeachingMaterial_LicenseAgreement**, which describe Imagination Technologies' teaching materials packages and license.

2. **examples** (folder): contains example programs that you will run in Catapult SDK while reading the Getting Started Guide. It includes subfolder **examples_PlatformIO**, where the same examples are provided using the VSCode/PlatformIO SDK instead of the Catapult SDK.

3. **common** (folder): contains drivers and libraries that the different programs will need.

4. **src** (folder): contains the source code (Verilog and SystemVerilog) for the RVfpgaEL2 System.

5. **Simulators** (folder): contains scripts for running the simulation of the RVfpgaEL2 System using Verilator. Three simulators are provided:
   a. **RVfpgaEL2-Trace** generates a trace of the different internal signals of the SoC while executing a given program and then uses GTKWave for visualizing the

waveform of this trace.

b. **RVfpgaEL2-ViDBo** uses a virtual board (i.e., an image of an FPGA board using a web browser) to perform a simulation of the RVfpgaEL2 System and communicate with the peripherals of the simulated board. The following peripherals are supported in RVfpgaEL2-ViDBo: switches, LEDs, UART, pushbuttons, 7-Segment Displays, and tricolor LED (the latter peripheral is not available on the Basys 3 board).

c. **RVfpgaEL2-Pipeline** uses a VeeR EL2 Pipeline Simulator for analysing the evolution of the instructions through the pipeline.

6. **driversLinux** (folder): contains the drivers required for using the boards on Linux.

7. **PlatformIO_ChipsAlliance_Files** (folder): contains some files required for configuring PlatformIO.

8. **Labs** (folder): contains instructions, programs, and solutions that you will use during RVfpgaEL2 Labs 1-20. This folder contains several subfolders:
   a. **Lab1, Lab2, … , Lab19, Lab20** (folders): Instructions and resources to be used while completing the labs in Catapult SDK. Note that each of the 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 *Labs/Lab01/*. These lab documents give the instructions, examples, tasks, exercises, and figures for each of the RVfpgaEL2 Labs.
   b. **RVfpgaLabsFigures** (folder): Figures used in the lab documents.
   c. **RVfpgaLabsPlatformIO** (folder): labs that use the VSCode/PlatformIO SDK instead of the Catapult SDK.
   d. **RVfpgaLabsSolutions** (folder): a subset of exercise solutions for each of the labs. They include documents and software with the solutions for the proposed tasks and exercises, as well as the source code, the simulator binaries and the bitstreams for the different extended SoCs.

> **IMPORTANT:** Instructors should remove folder **RVfpgaLabsSolutions** before distributing RVfpgaEL2 to students.

> **Videos:** In addition to the RVfpga folder and the Virtual Machine, in the Imagination's YouTube channel we provide videos demonstrating different sections of the GSG and the Labs. These videos are a very good complement to the text explanations and the examples provided throughout the course.

2. RVfpgaEL2 GSG OVERVIEW

The RVfpgaEL2 Getting Started Guide (GSG) is the introductory document of the RVfpgaEL2 materials. It can be found inside the platform subfolder (*[RVfpgaBasysPath]*, *[RVfpgaBooleanPath]*, *[RVfpgaEL2NexysA7DDRPath]* or *[RVfpgaEL2NexysA7NoDDRPath]*), at: *Documents/RVfpga_GettingStartedGuide.docx*

> **IMPORTANT:** Before starting to work with the RVfpgaEL2 Labs, you should complete the RVfpgaEL2 Getting Started Guide (GSG).

The GSG is an extensive document that includes software installation instructions, an overview of the RISC-V architecture and RVfpgaEL2 (including in-depth descriptions of the VeeRwolf SoC and VeeR EL2 core), and instructions about how to write, simulate, and run programs on RVfpgaEL2 both in simulation and, optionally, in hardware on the FPGA board. Here is an overview of the GSG contents:

- **Section 1:** An overview of the GSG contents, RVfpgaEL2 System, required software and optional hardware, and expected prior knowledge.
- **Sections 2 and 3:** A brief introduction to the RISC-V computer architecture, the RVfpgaEL2 SoC, and the organization of the Verilog and SystemVerilog files that make up the RVfpgaEL2 system.
- **Section 4:** Shows how to install all of the software tools needed to use RVfpgaEL2 both in simulation and hardware.
- **Section 5**: Shows how to use Catapult SDK to both download the RVfpgaEL2 SoC onto the FPGA board and download and run several example programs on RVfpgaEL2.
- **Sections 6 - 9:** Show how to simulate RVfpgaEL2 source code (Verilog and SystemVerilog) using Verilator with three different simulators (RVfpgaEL2-Trace, RVfpgaEL2-ViDBo and RVfpgaEL2-Pipeline) and how to simulate RISC-V code on the Whisper instruction set simulator (ISS).
- **Appendices:** The Getting Started Guide also includes appendices that show additional features such as how to install the required tools on Windows and macOS machines.

3. RVfpgaEL2 LABS OVERVIEW

The RVfpgaEL2 Labs, listed in Table 1, provide hands-on understanding of RISC-V hardware and software (remember that before starting to work with the RVfpgaEL2 Labs, you must complete the RVfpgaEL2 Getting Started Guide). The RVfpgaEL2 Labs materials can be found inside the platform subfolder (*[RVfpgaBasysPath]*, *[RVfpgaBooleanPath]*, *[RVfpgaEL2NexysA7DDRPath]* or *[RVfpgaEL2NexysA7NoDDRPath]*), in the *Labs* folder.

Table 1. RVfpgaEL2 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 (only included for the Nexys A7 board)
Part 2	11	VeeR EL2 Configuration and Organization. Performance Monitoring
	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~~ (This lab is included in RVfpga 2.2, RVfpgaEH1, because VeeR EH1 is superscalar. It is not included in the RVfpgaEL2 course, as the VeeR EL2 core is not superscalar).
	18	Adding New Instructions to the VeeR EL2 Core
	19	Memory Hierarchy: The Instruction Cache (I$)
	20	The ICCM, DCCM, and Benchmarking

The labs are divided into two parts. Part 1 shows how to program RVfpgaEL2, build the source code in Vivado, and extend the RVfpgaEL2 System to include additional peripherals. Part 2 focuses on the RISC-V VeeR EL2 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, Labs 1-4 show how to run C and RISC-V Assembly programs on the board and on simulation, Lab 5 shows how to analyse the VeeRwolfX SoC source code, create an RTL project and generate a bitstream for the RVfpgaEL2 on the FPGA board, and Labs 6-10 show how to modify the RVfpgaEL2 System to add new peripherals.

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.

These labs are well-suited for a two-semester course for undergraduates. Labs 11-20 could also be taught to master's level students. Prior to completing this RVfpgaEL2 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. RVfpgaEL2 SOFTWARE AND HARDWARE

Table 2 lists the required software and optional hardware needed to use these labs. All of the software is free. An 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-based simulation tools that we will show how to use in this course. The RVfpgaEL2 Getting Started Guide shows how to install and use all software and optional hardware.

Table 2. Required Software and Optional Hardware

Software		
Name	**Website**	**Cost**
Vivado 2022.2 WebPACK*	https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/2022-2.html	free
Catapult SDK	https://developer.imaginationtech.com	free
Verilator (an HDL simulator)	https://github.com/verilator/verilator	free
GTKWave	http://gtkwave.sourceforge.net/	free
LibWebSockets	https://libwebsockets.org/	free
Whisper (RISC-V Instruction Set Simulator) **	https://github.com/chipsalliance/VeeR-ISS	free
Optional Hardware		
Name	**Website**	**Cost**
Nexys A7 FPGA Board***	https://store.digilentinc.com/nexys-a7-fpga-trainer-board-recommended-for-ece-curriculum/	$349 (academic price at https://digilent.com/shop/academic/academic-price-list/: $261.75)
Boolean Board	https://www.realdigital.org/hardware/boolean	$105 (academic: $87)
Basys 3 FPGA Board	https://digilent.com/shop/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/	$165 (academic: $124)
RISC-V Core and System-on-Chip (SoC)**		
Name	**Website**	**Cost**
VeeR EL2 Core **	https://github.com/chipsalliance/Cores-VeeR-EL2	free
VeeRwolf **	https://github.com/chipsalliance/VeeRwolf	free

* Vivado WebPACK is optional and is only needed when using the optional hardware.
** Already provided with the RVfpgaEL2 download from Imagination Technologies
*** All of the steps described in this guide also work on Digilent's Nexys4 DDR FPGA board.