1 Software Requirements

PULP Software development needs tools to program, simulate and optimize software code. PULP Project provides as open-source, all these tools in the PULP-PLATFORM organization on GitHub (`https://github.com/pulp-platform`).

1.1 PULP Compiler

PULP toolchain is available at `https://github.com/pulp-platform/pulp-riscv-gnu-toolchain`. It is based on standard RISC-V toolchain of GNU GCC project and includes PULP ISA support.

On Ubuntu 18.04 this packages should be required and you can install them with:

- $ sudo apt-get install autoconf automake autotools-dev curl libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev

To install the PULP toolchain follow these steps:

- $ git clone https://github.com/pulp-platform/pulp-riscv-gnu-toolchain
- $ cd pulp-riscv-gnu-toolchain
- $ git submodule update –init –recursive
- $ export PATH=INSTALL_DIR/bin:$PATH
- $ ./configure –prefix=INSTALL_DIR –with-arch=rv32imc –with-cmodel=medlow –enable-multilib
- $ make

More details at: `https://github.com/pulp-platform/pulp-riscv-gnu-toolchain`

1.2 PULP-SDK

PULP-SDK is available at `https://github.com/pulp-platform/pulp-sdk`. It is a software development kit for PULP developers. It includes run-time functions, a test-suite, scripts and GVSoC simulator. Run-time functions (PMSIS) allows to abstract to the programmers the registers layer and support RTOS (Real-Time Operating System) functionalities such as tasking and memory management.

On Ubuntu 18.04 this packages should be required and you can install them with:

- $ sudo apt-get install -y build-essential git libftdi-dev libftdi1 doxygen python3-pip libsdl2-dev curl cmake libusb-1.0-0-dev scons gtkwave libsndfile1-dev rsync autoconf automake texinfo libtool pkg-config libsdl2-ttf-dev

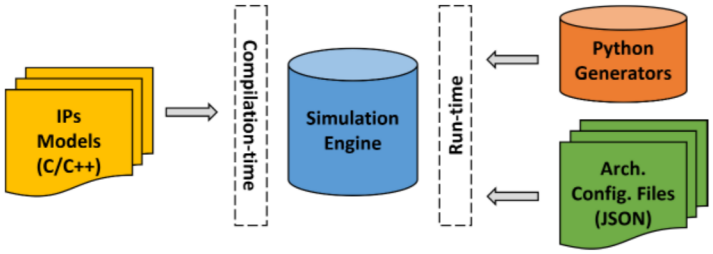

Figure 1: GVSoC structure overview

You may have needed of other python3 packages and you can install them with:

- $ pip install argcomplete pyelftools six

Assuming that PULP Toolchain and PULP-SDK are correctly built and installed:

- $ cd pulp-sdk/

Setting up PULP Toolchain path:

- $ export PULP_RISCV_GCC_TOOLCHAIN=INSTALL_DIR

Setting up the environment sourcing a configuration file:

- $ source configs/pulp-open.sh

Build and compile GVSoC (if target or GVSoC is changed):

- $ make build

1.3 GVSoC

GVSoC is the PULP simulator. It is written in C++ and models every important module of the target PULP architecture. It is composed by C++ models such as cores, I-cache, memory, peripherals, and interconnect, a set of configuration files and python scripts to wrap the target architecture, as depicted in Fig.1. It is extremely fast and enough accurate to be an important tool for software development, and the to pursuit this exercise you will use GVSoC as simulation platform.

2 Write a 2D matrix multiplication for PULP

The focus of this task is approaching to PULP programming writing a simple matrix multiplication between 2D matrices on a PULP system depicts in Fig.2. In this architecture, a single RISC-V core is directly connected to an interleaved

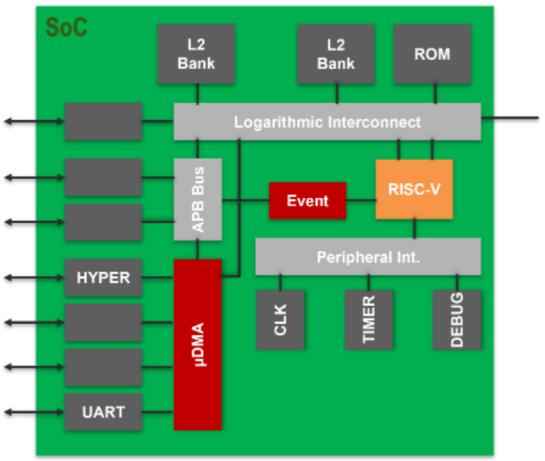

Figure 2: PULP architecture overview

multi-banked L2 memory (512 kiB) via a logarithmic interconnect. First, you have to choose the right matrix dimensions to properly fit the available memory space, then allocate L2 space for that and finally write the code to implement:

$$\mathbf{C} = \mathbf{A} \times \mathbf{B}$$

You can find several examples in the PULP-SDK test-suite.

2.1 Output

You should functionally verify the correctenss of **C** considering pre-generated input data for **A** and **B**.