

Guide on Setting Up PULPissimo Gen2

This guide provides detailed steps to set up and use the PULPissimo Gen2 environment for FPGA boards. Follow these instructions carefully to ensure a successful setup.

Prerequisites

- Vivado 23.1 or higher installed on your system
- Access to a terminal on a Linux-based system
- FPGA board connected via JTAG

1. Cloning the PULPissimo Repository

1. Navigate to the PULPissimo GitHub page: [PULPissimo on GitHub](#).
2. Follow the README guide for initial setup instructions.

2. Setting Up Vivado Project

1. Open Vivado and select "Open Project".
2. Navigate to `project.xpr` to open the Vivado project.
3. Regenerate the Bitstream if necessary by following the Vivado documentation.

3. Programming the FPGA Board

1. Connect to the FPGA board's hardware server at `kintex7.doelab.site` using Vivado.
2. Program the FPGA boards through JTAG as detailed in the Vivado guide.

4. Installing RISC-V GCC

1. Open a terminal and navigate to `/tmp`.
2. Download the RISC-V GCC package using `wget`:

```
wget https://github.com/stnolting/riscv-gcc-  
prebuilt/releases/download/rv32i-4.0.0/riscv32-unknown-elf.gcc-  
12.1.0.tar.gz
```

3. Extract the downloaded package to `/opt/riscv`:

```
sudo mkdir /opt/riscv  
sudo tar -xzf riscv32-unknown-elf.gcc-12.1.0.tar.gz -C /opt/riscv/  
sudo rm -f riscv32-unknown-elf.gcc-12.1.0.tar.gz
```

4. Add RISC-V GCC to your system's PATH:

```
export PATH=$PATH:/opt/riscv/bin
```

5. Patching libpython3 (If Needed)

1. Install libpython3-dev using apt:

```
sudo apt install libpython3-dev
```

2. Create a symbolic link for the libpython version if required:

```
sudo ln -s /usr/lib/x86_64-linux-gnu/libpython3.11.so /usr/lib/x86_64-  
linux-gnu/libpython3.8.so.1.0
```

6. Preparing the Environment

1. Obtain the latest version of pulp-runtime from the provided source.
2. Navigate to the pulp-runtime directory and activate the environment:

```
source configs/fpgas/pulpissimo/gensys2.sh
```

Note: This step must be repeated every time you open a new terminal session.

7. Compiling and Running Your Application

1. In the terminal prepared in the previous step, switch to your application directory.
2. Start with the "periph/uart" example for a basic application.
3. Compile your application:

```
make clean all io=UART
```

This command compiles the application and links the ELF binary.

8. Setting Up JTAG and OpenOCD

1. On the PC connected to the FPGA via JTAG, download the configuration file for OpenOCD.
2. Modify the config file to allow remote connections by adding `bindto 0.0.0.0` before the `init` command.
3. Start OpenOCD with:

```
openocd -f openocd-genesys2.cfg
```

9. Running the Application

1. In a new terminal, start RISC-V GDB pointing to your ELF binary:

```
riscv32-unknown-elf-gdb build/test/test
```

2. Connect to your OpenOCD server:

```
target remote <ip>:3333
```

3. Open a UART console on the machine with UART connected.
4. Load the binary into PULPissimo's main memory and start execution:

```
load  
continue
```

5. Check the UART output for results.

Congratulations! You've successfully set up and run your application on PULPissimo Gen2.