Ubuntu

Ubuntu OS, can be use with Windows by dual-booting. If you don't know anything about dualbooting, you should make a google/youtube search or take a look the video of Mehmet AFACAN from Ankara University Computer Society.

https://youtu.be/BEEbB7D9tCE After installation of Ubuntu 18.06 LTS, you need to install applications which are needed in process. Use these commands:

1- sudo apt install vim git python3-pip gawk texinfo libgmp-dev libmpfr-dev libmpc-dev
2- sudo pip3 install pyelftools
3- sudo apt-get install autoconf automake autotools-dev curl python3 libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev
4- sudo apt install swig3.0 libjpeg-dev lsb-core doxygen python-sphinx sox graphicsmagick-libmagick-dev-compat libsdl2-dev libswitch-perl libftdi1-dev cmake scons libsndfile1-dev
5- sudo pip3 install twisted prettytable pyelftools openpyxl xlsxwriter pyyaml numpy configparser pyvcd
6- sudo apt install python-pip
7- sudo pip2 install configparser

Questa Sim

1- Go to https://www.intel.com/content/www/us/en/software-kit/750573/questa-intel-fpgas-pro-edition-software-version-21-1.html?

2- Download the both part1 and part2 of the questa 21.1

3- Go https://licensing.intel.com/psg/s/sales-signup-evaluationlicenses website to get Questa Licence

4- Register with your edu.tr email.

5- After verificated your e mail address and registered, go https://licensing.intel.com/psg/s/salessignup-evaluationlicenses website.

6- Select Questa*-Intel® FPGA Starter Edition SW-QUESTA and change the value of # of Seats to 1 and click Generate Licence.

7- In generate licence secton, click Create Computer. Then type a name to Computer name. Do selections as Computer Type=NIC ID, Licence Type = Fixed.

8- For Primary Computer ID section, type to console ifconfig . "Physical Address" is your Primary Computer ID. This value is defined as 12 HEX number. Example = ether xx:xx:xx:xx:xx:xx

9- After all, you'll get an e-mail which includes .dat Licence File.

10- Open a terminal in Ubuntu and type cd to go /home/user. Type "vim .bashrc" and edit the last line of this file as "export LM_LICENSE_FILE=LICENCELOCATION". After this command do the followings as "ESC, :x , Enter". This will save the file and return you back to root in terminal. Type "source .bashrc" and licence file will be included permanently to your PATH. —- Example command: export LM_LICENSE_FILE=/home/erkmen/Desktop/LR-102459 License.dat

11- Again type cd to terminal and go /home/user. Type "vim .bashrc" and add another line to add questa permanently to your PATH. You need to add bin folder of Questa. (With insert button, you can change editing mode. The key for insert mode is 'i') For example:

export PATH=/home/erkmen/Desktop/intelFPGA/21.1/questa_fe/bin:$PATH After that part, again do the followings as ESC, :x, Enter and type "source .bashrc". Now QuestaSIM is added to your PATH variable permanently.

Vivado 2018.3

1- Go to https://www.xilinx.com/support/download.html website
2- Click "Vivado Archive" from left.
3- Download Vivado from the part "Vivado HLx 2018.3: All OS installer Single-File Download", extract from .tar and install.

RISCV-GNU-Toolchain

1- Open a new terminal and go to the folder which you want to download RISCV-GNU-Toolchain's installation files. Go there with cd command. (For example: cd /Desktop/kurulum)
2- Type " git clone --recursive https://github.com/pulp-platform/riscv-gnu-toolchain.git" command to the terminal in this folder.
3- After cloned the files, go inside of this folder with cd command and type respectively "./configure –prefix=/opt/riscv –with-arch=rv32imc –with-cmodel=medlow –enable-multilib" and "make" commands. You can write any folder path to –prefix=/opt/riscv. You can define there another location path which you created in desktop or somewhere else na¨oed as RISCV-GNU-ROOT. (For example using –prefix=/home/erkmen/Desktop/RISCV-GNU-FOLDER, this will install riscv gnu toolchain to this folder.). (This will take some time.)
4- After installation type to terminal "cd" and go root. Again type "vim .bashrc" command and add a new line to and of the file. Type these commands respectively to their order:

   export PULP_RISCV_GCC_TOOLCHAIN=Installed Toolchain Folder Path

   export PATH=$PULP_RISCV_GCC_TOOLCHAIN/bin:$PATH

   For example:

   export PULP_RISCV_GCC_TOOLCHAIN=/home/erkmen/Desktop/RISCV-GNU-ROOT

   export PATH=$PULP_RISCV_GCC_TOOLCHAIN/bin:* $PATH

   Now riscv gnu toolchain is permanently added to your $PATH variable

PULPissimo

1- Open a terminal and go to the folder which you'll install pulpissimo.(For example: cd ~Desktop/proje)

2- Type git clone https://github.com/pulp-platform/pulpissimo command to clone the pulpissimo repo.
3- After 2, type git clone https://github.com/pulp-platform/pulp-runtime/ command to clone pulpruntime repo
4- After 3, type git clone https://github.com/pulp-platform/pulp-runtime-examples command to clone pulp-runtime-examples repo. 5- Type "cd pulpisimo" command and go inside of cloned pulpissimo folder.
5- Type
curl --proto '=https' --tlsv1.2 https://pulp-platform.github.io/bender/init -sSf | sh ccommand
6- After 5 you must run the following command
Make checkout

Simulation

1- Go inside of the pulp-runtime folder with terminal (cd pulp-runtime).
2- Type "source configs/pulpissimo cv32.sh" command.
3- Type "cd ." to go back in terminal and go inside pulpissimo folder..
4- Type "source setup/vsim.sh" command to source vsim.sh.
5- Type "make build" command.
6- After 5, use "cd ." command to go back in terminal and use "cd pulp-runtime-examples" command to go inside of the pulp-runtime-examples folder.
7- Go to hello folder in terminal(cd hello).
8- Type "make clean all run" command. That command compiles that C file to RISCV instructios and it'll make simulation on PULPissimo by using QuestaSim.

Bit File Generation

1- Type to terminal "sudo apt-get install cutecom" command.
2- Go to pulpissimo folder in terminal.
3- Type command "make scripts".
4- After 3, type command "cd ." to go back in terminal and type command "cd pulp-runtime" to go pulp-runtime folder.
5- Type command "source configs/pulpissimo cv32.sh"
6- Type command "source configs/fpgas/pulpissimo/nexys.sh". ATTENTION: nexys.sh is the file of Nexys FPGA Boards. If you are using another supported FPGA board, you should source it from the same configs/fpgas/pulpissimo folder
7- Type command "cd ." to go back and type command "cd pulpissimo/fpga". After that, type command "make nexys rev=nexysA7-100T". ATTENTION: In this Nexys A7-100T parameters are used to generate bitfile. To learn the parameters for other boards Go https://github.com/pulp-platform/pulpissimo/tree/master/fpga and open your board's repo. Read the README part.
8- It will take time to generate bit file. After it ends, generated bit file will be in fpga folder.
9- Connect your FPGA board with USB to your computer and activate its programming mode.
10- Type "vivado" command in terminal to launch Vivado 2018.3.

11- Connect your device and program it with program device section. Choose bit file which you generated under the fpga folder.

12- If PULPissimo booted correctly, LED0 will be activated. ATTENTION: When you disconnect the FPGA board from USB, booted pulpissimo will gone. Don't disconnect your FPGA board after this. 12- Disconnect from Hardware Manager first then close hardware manager and Vivado.

Run C code on FPGA

To do this, you should end the section: bit file generation above.

1- First, OpenOCD should be installed.

2- Open a terminal in the workspace folder and type command "git clone https://github.com/openocd-org/openocd".

3- Type command "cd openocd", then run followings
"./bootstrap"
"./configure"
"make"
"sudo make install"

4- Go pulp-runtime-examples/hello folder in terminal and paste your C code to hello.c or if you justwant to see "Hello" as an output, don't touch anything. Type command "make clean all io=UART".

5- You can learn PULPissimo's JTAG pinmap from the websitehttps://github.com/pulp-platform/pulpissimo/tree/Go to your board's repo and take a look at the README. Make connections first, then connect external JTAG programer to your computer with USB.

6- Open a new terminal and type command "openocd -f .../pulpissimo/fpga/pulpissimo-nexys/openocd-nexys-hs2.cfg". ATTENTION: This command is valid for Nexys boards and you should change ... part. If you are using another FPGA board, you should select your board's cfg file. For Example:openocd -f /home/erkmen/Desktop/kurulum/pulpissimo/fpga/pulpissimo-nexys/openocd-nexys-hs2.cfg

7- Open another terminal and type command "riscv32-unknown-elf-gdb /home/erkmen/Desktop/kurulum/pulp-runtime-examples/hello/build/test/test". After that type command "target extended-remote: 3333". You should change /home/erkmen/Desktop/kurulum part

8- So now we have 2 active terminal. One is gdb and one is OpenOCD. We should open third terminal and launch the cutecom with command "sudo cutecom".

9- Listen the FPGA from connected port with 115200 baudrate. (Baudrate of your board is defined in FPGA repo)

10- Go back to gdb termnial and do the commands load then continue. You should see the output from cutecom.

In process, there are many possible errors. Most of them has solutions in pulpis-simo github's issues section. You can solve other basic errors by using google.

Contact: erkmen17@itu.edu.tr , serto17@itu.edu.tr