

NEORV32 Development Environment

Basic information [↗](#)

The basic development environment for RISC-V NEORV-32 is the following:

- Vivado HL WebPACK: We use 2016.4 because it requires less memory on disk (8.41 GB) than newer Vivado versions.
- Toolchain for RISC-V: The GNU cross-compiler for RISC-V.
- Make
- UART terminal:
 - cutecom (linux), hterm (windows)
- Editor:
 - vim
 - MS code
 - other

Three Alternative Setups one can use [↗](#)

There are three alternative setups that one can have:

Setup_1: MS Windows PC [↗](#)

- Installation of Vivado 2016.4 (or any other version if you already have an install) on a native Windows machine
- A VirtualBox-based Virtual Machine (Lubuntu / disk size ~8GB) with all the necessary software and RISC-V toolchains.

Setup_2: Ubuntu PC [↗](#)

- Installation of Vivado 2016.4 (or any other version if you already have an install) on a native Ubuntu machine
- Installation of all necessary software and RISC-V toolchains.

Setup_3: Any PC [↗](#)

- A VirtualBox-based Virtual Machine (Lubuntu / disk size ~17.5GB) with all the necessary software, RISC-V toolchain, and Vivado 2016.4. (The PCs at Nikaia's lab use this solution)

i We propose to use Setup 3, which is the easiest way to start working fast on your PC

How to run Setup_3 [↗](#)

- Download [VirtualBox 7.1.6 platform packages](#)
- Download the VM by clicking this link [NEORV32-VirtualBox-VM.zip](#)
- Run the VM
- When you login in the VM with username=riscv and password=riscv
 - Setup git

```
1 git config --global user.email "you@example.com"
2 git config --global user.name "Your Name"
```

- Go to folder ~/wsp/neorv32 and commit the initial state of the neorv32 toolchain

```
1 cd ~/wsp/neorv32
```

```
2 git commit -m "init"
```

i Careful in MS Windows you should disable Hyper-V:

Open **Command Prompt (Admin)** and run:

```
1 bcdedit /set hypervisorlaunchtype off
```

Setting the RISC-V toolchain Ubuntu (Native or VM) [↗](#)

```
1 wget https://github.com/stnolting/riscv-gcc-prebuilt/releases/download/rv32i-4.0.0/riscv32-unknown-elf-gcc-12.1.0.tar.gz
2 sudo mkdir /opt/riscv
3 $ sudo tar -xzf riscv32-unknown-elf-gcc-12.1.0.tar.gz -C /opt/riscv/
```

- Add in `.bashrc`

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

Test the toolchain:

```
1 $ riscv32-unknown-elf-gcc -v
2 Using built-in specs.
3 COLLECT_GCC=riscv32-unknown-elf-gcc
4 COLLECT_LTO_WRAPPER=/opt/riscv/libexec/gcc/riscv32-unknown-elf/12.1.0/lto-wrapper
5 Target: riscv32-unknown-elf
6 Configured with: /tmp/rv_gcc/riscv-gnu-toolchain/gcc/configure --target=riscv32-unknown-elf --prefix=/opt/riscv --disable-shared --disable-threads --enable-languages=c,c++ --with-pkgversion=glea978e3066 --with-system-zlib --enable-tls --with-newlib --with-sysroot=/opt/riscv/riscv32-unknown-elf --with-native-system-header-dir=/include --disable-libmudflap --disable-libssp --disable-libquadmath --disable-libgomp --disable-nls --disable-tm-clone-registry --src=/tmp/rv_gcc/riscv-gnu-toolchain/gcc --disable-multilib --with-abi=ilp32 --with-arch=rv32i --with-tune=rocket --with-isa-spec=2.2 'CFLAGS_FOR_TARGET=-Os -mmodel=medlow' 'CXXFLAGS_FOR_TARGET=-Os -mmodel=medlow'
7 Thread model: single
8 Supported LTO compression algorithms: zlib
9 gcc version 12.1.0 (glea978e3066)
```

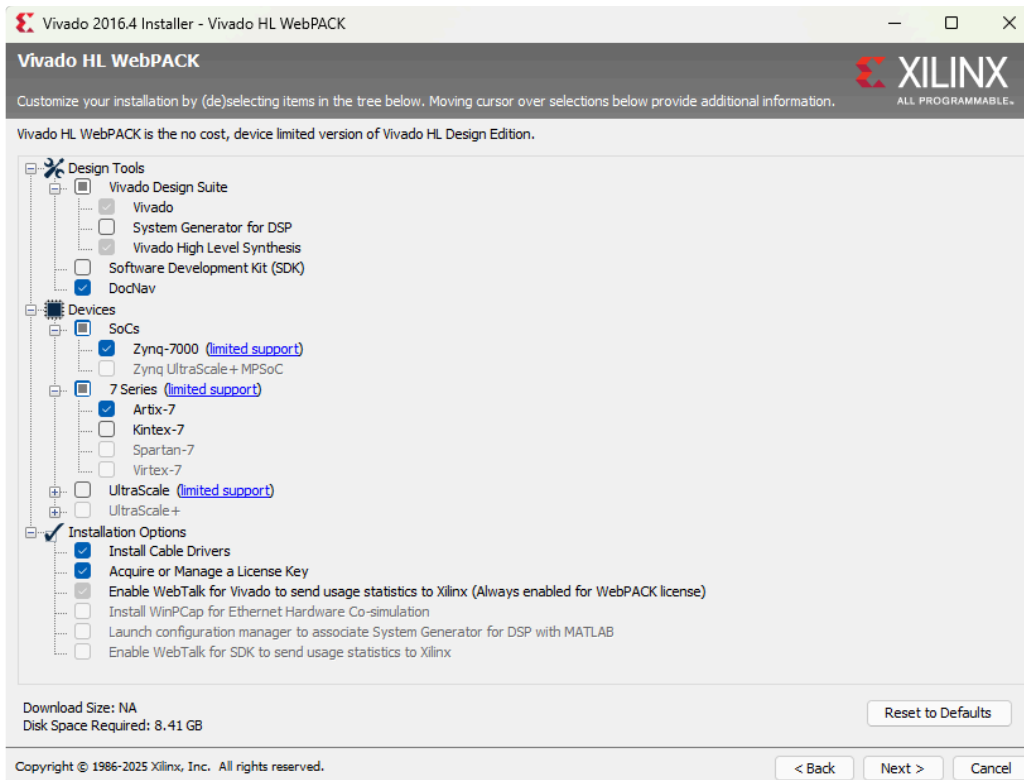
- Also install cutecom uart terminal

```
1 sudo apt install cutecom
2 sudo usermod -aG dialout $USER
```

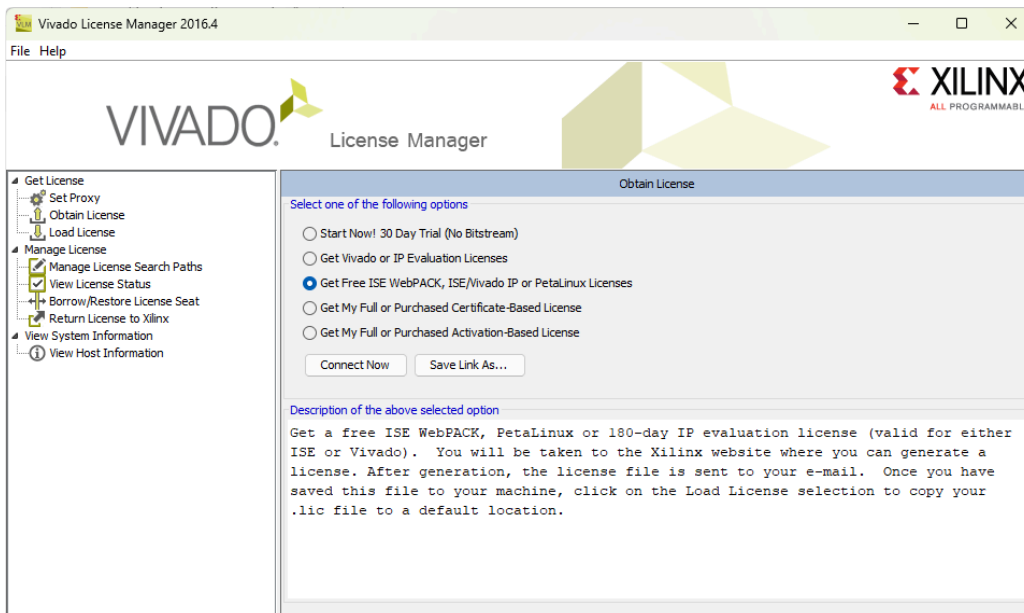
- Also install MS code or any other editor you prefer.

Setting the Vivado in Windows or Ubuntu (Native or VM) [↗](#)

- Download Vivado HL WebPACK 2016.4 from [Xilinx Download Archives](#)
- During Installation select the following options (to save space you may deselect SoCs devices since we do not use them in the NEORV32 lab)



- When the Vivado License Manager Opens, simply close the window. You do not need to download any license.



Expanding the VirtualBox VDI Disk [↗](#)

In case you are out of space, follow the instructions below

This guide explains how to resize a **VirtualBox** virtual disk image (**VDI**) using the `VBoxManage` command-line tool.

- Ensure the virtual machine (VM) is **shut down** (not suspended).
- Run the following command to resize the disk:

```
1 VBoxManage modifyhd "<path_to_vdi_file>" --resize <size_in_MB>
```

<path_to_vdi_file>: Full path to the VDI file.

<size_in_MB> : New total size of the disk in megabytes (e.g., 18500 = 18.5 GB)

- Start the VM and run

```
1 sudo growpart /dev/sda 1
2 sudo resize2fs /dev/sda1
3 df -h
```