UNIVERSIDADE FEDERAL
DE SANTA CATARINA

RISC-V: An approach for learning the architecture

RISC-V: An approach for learning the architecture
Universidade Federal de Santa Catarina, Florianópolis - Brazil

May, 2023

RISC-V: An approach for learning the architecture
May, 2023

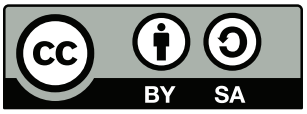
Project Chief:
Eduardo Augusto Bezerra <eduardo.bezerra@spacelab.ufsc.br>

Authors:
João Cláudio Elsen Barcellos <joaoclaudiobarcellos@gmail.com>
Rebecca Quintino Do Ó <rebeccaqquintino@gmail.com>
Yunior Alcantra Guevara <yunior.alcantra@posgrad.ufsc.br>

Contributing Authors:

Revision Control:

Version	Author	Changes	Date
0.0	J. C. E. Barcellos, Rebecca Q. Do Ó, Yunior A. Guevara	Document creation	2023/05/10



List of Figures

List of Tables

Contents

CHAPTER 1

Introduction

[1].

CHAPTER 2

Compiling and executing you first program

2.1 Downloading and installing the toolchain

First, you should access https://github.com/stnolting/riscv-gcc-prebuilt, to download the most recent available toolchains (today, 21/05/2023, **rv32i-4.0.0**). You should have now access to a .tar.gz file. Then, the next step, is to create the folder that the toolchain is going to be installed. You can open a terminal and type:

```
$ sudo mkdir /opt/riscv
```

Now, you need to navigate to the folder where the .tar.gz file was downloaded, like:

```
$ cd Downloads/
```

And then you have to extract the .tar.gz file to the folder previously created:

```
$ sudo tar -xzf <toolchain_version>.tar.gz -C /opt/riscv/
```

Finally, you should add the toolchain's bin folder to your system's PATH environment variable. You can open the .bashrc file:

```
$ sudo nano .bashrc
```

And then add the following line in the end of the .bashrc file:

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

To make sure everything works fine, navigate to the folder with the aplication examples and execute the following command:

```
$ make check
```

If everything is working fine you should se an "**OK**" appearing at the end, like in the Figure 2.1.

2.1.1 Some important commands

Now that the toolchain was installed, you should know some commands to generate the .hex, .bin, .vdh, as well as other types of files. As you can see in the Figure 2.2, you could use `hex` to generate the .hex file, which represents the machine language. You could use, for instance, the `image` to generate the .vhd file, which is the *neor32_application_image.vhd* used to store the main program that will run in the microprocessor.

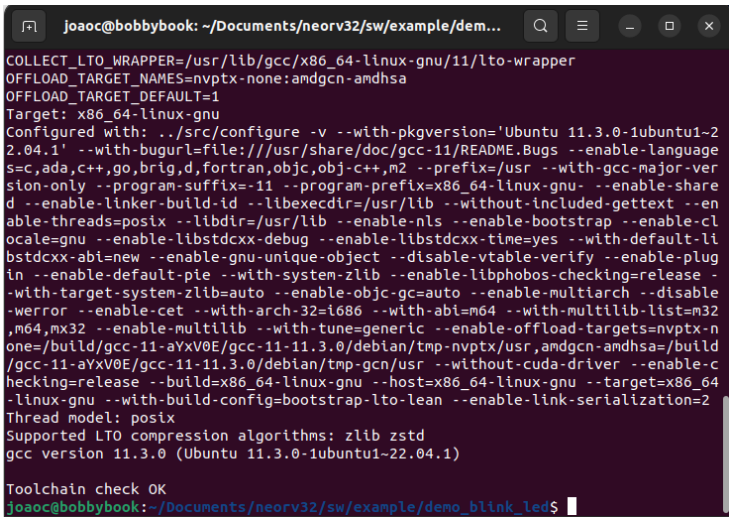


Figure 2.1: Indication that the installation was successful.

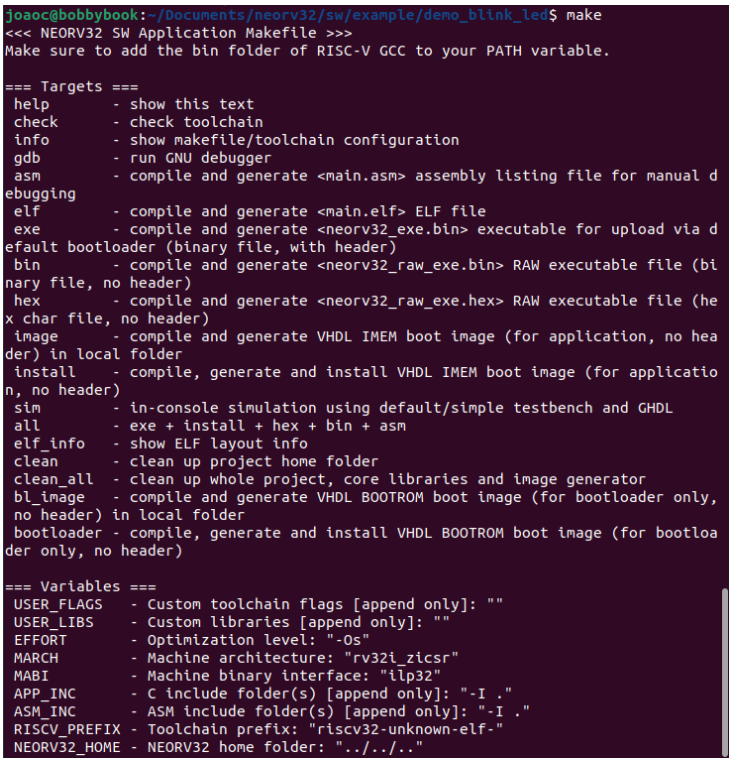


Figure 2.2: Some useful commands to use.

Bibliography

[1] J. Bouwmeester, A. Menicucci, and E.K.A. Gill. Improving CubeSat reliability: Subsystem redundancy or improved testing? *Reliability Engineering & System Safety*, 220:108288, 2022.