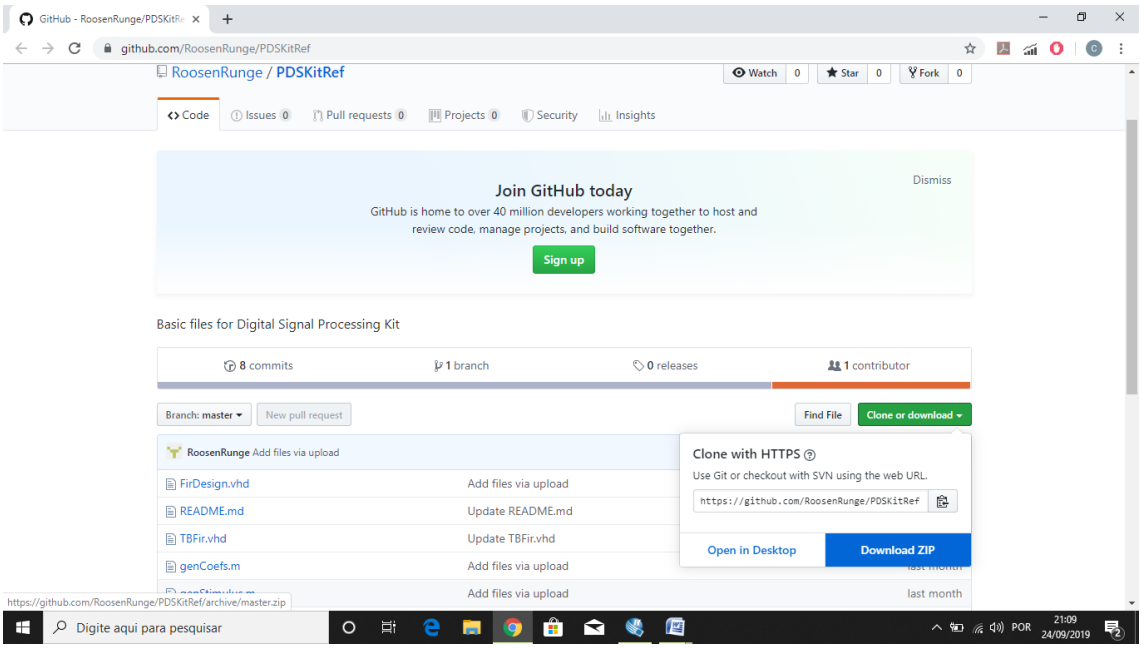
NOTES

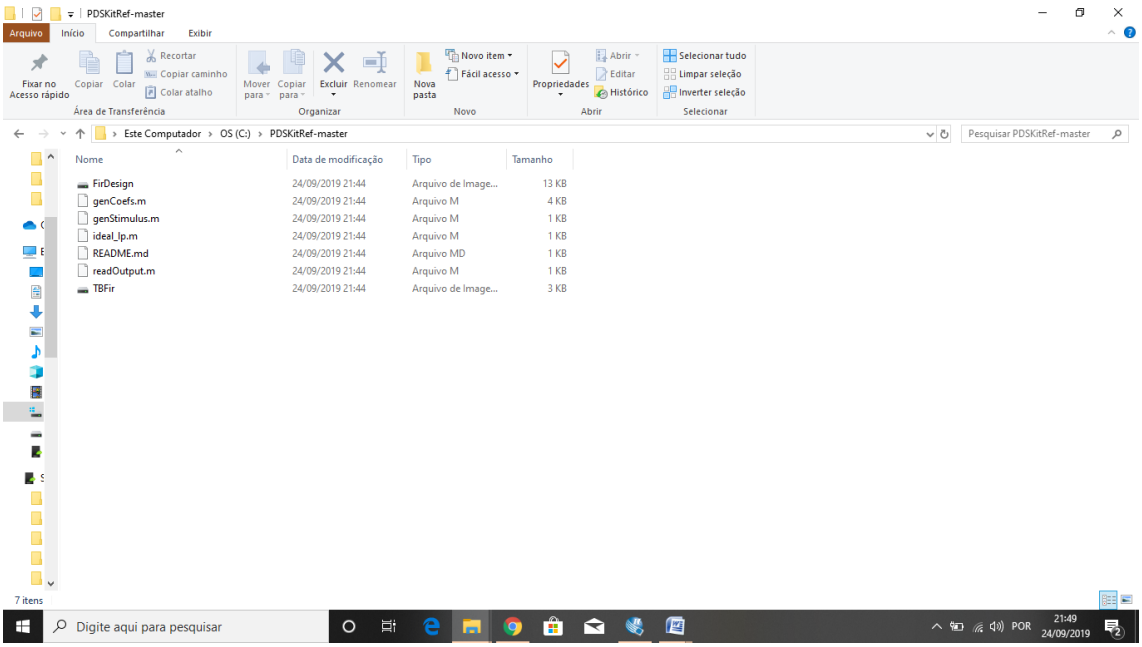
These notes show a sequence of screenshots that summarizes the basic steps to create, simulate, and synthesizes the project (with some additional hints). This sequence has been made with the files of GitHub repository and the hardware described in the paper "Introducing Programmable Logic Devices in Physics Laboratories: a practical guide for the implementation of experiments". If you are using another hardware or synthesizing a different FIR filter, you have to parameterize according to your specifications.

Before beginning to write these notes, download the project files in GitHub and extract them to your work directory as shown below:

Download de Zip file:



Extract them to your work directory:



These notes are organized in four sections. The websites where the softwares can be downloaded are indicated in Section 1. In Section 2, we start the simulation process running the files in Octave to obtain the FIR coefficients to the VHDL file that implements the FIR filter. In Section 3, the simulations are run and the response of the FIR filter can be visualized. Finally, in Section 4, the procedure for synthesizing the VHDL code and programming the FPGA device is shown.

1) Softwares Download

Firstly, you have to download the softwares to be used in all steps of the system development. These softwares are: QuartusII, Modelsim and Octave. All these softwares are free web edition.

a) Quartus II and Modelsim

We download a free Quartus II (web edition) from the Intel site: https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html.

There are many Quartus II versions. In the website you have to choose the one that supports the device you are using (in this project the device used was a Cyclone IV). We have downloaded Quartus II (web edition) 13.0 (service pack 1).

In the same page that you configure the items to download the software Quartus II, you may select the software Modelsim to download, as shown in Figure A1.

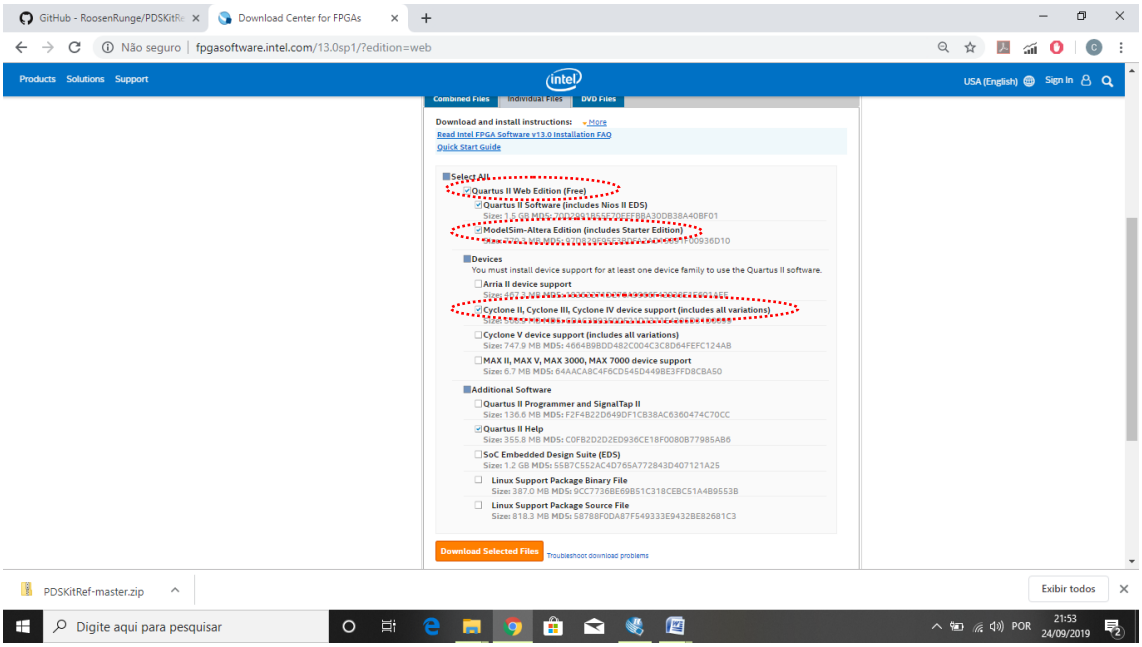


Figure A1. Quartus II and Modelsim download.

b) Octave

You can download the free Octave software in the site:

https://www.gnu.org/software/octave/download.html

2) The FIR Coefficients generation (Octave)

To start the simulation process, we will generate the FIR coefficients using the Octave. Running genCoefs.m, the file coefs.dat with the FIR coefficients to parameterize the VHDL design file FirDesing.vhd will be generated. Also the plot of the theoretical impulse response and frequency response of the filter will be generated.

For this project in genCoefs.m, we have used the windowing function 'Kaiser', therefore the first time you run it, the Octave software will exhibit in the prompt a message to run the package 'pkg load signal' as seen in Figure A2.

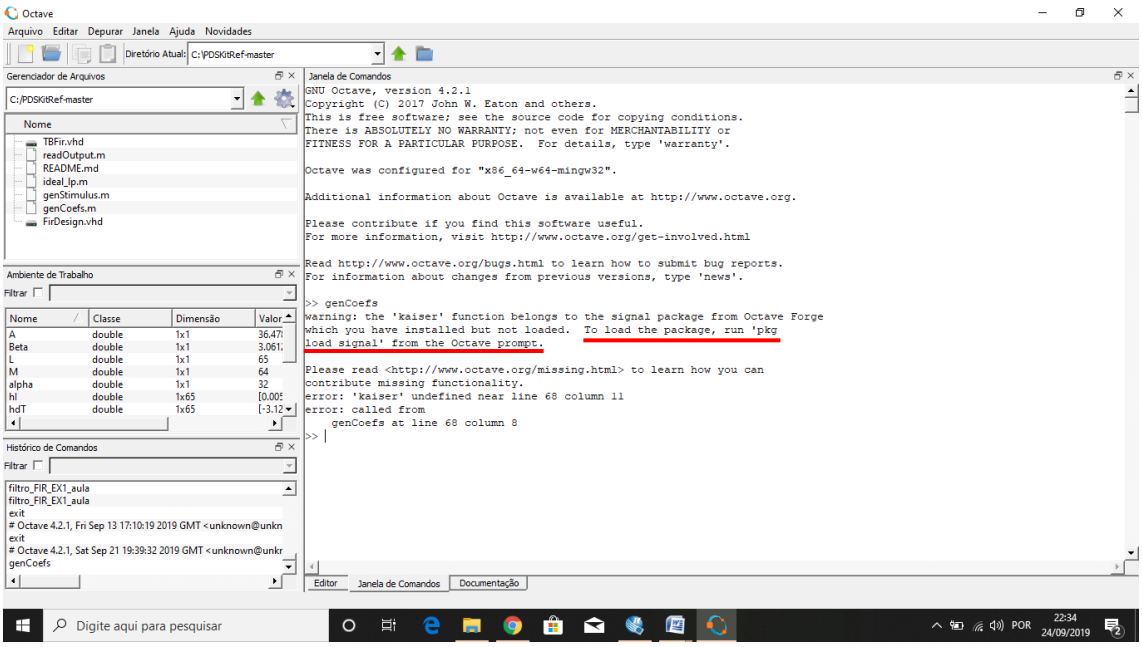


Figure A2.Octave genCoefs warning.

According to the instructions in the Figure 2, run the package "pkg load signal" and rerun genCoefs.m in Octave. At this time the file coefs.dat with the FIR coefficients are generated and the theoretical impulse response and frequency response of the FIR filter are plotted as shown in Figure A3.

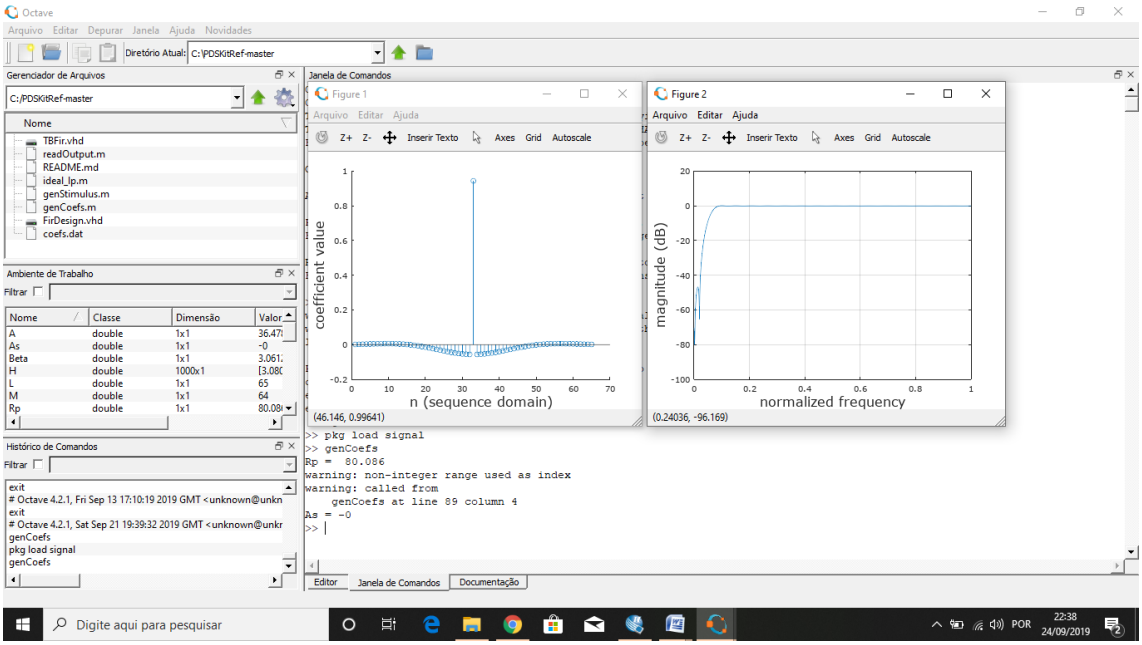


Figure A3. genCoefs rerun showing the theoretical FIR responses.

After manually modify the FirDesign.vhd with the coefficients in coefs.dat, we proceed to the simulation steps.

3) Simulation

In order to validate the circuit simulation, we use Octave to generate the inputSignal.dat file (that will be used by TBFir.vhd in MoldelSim) through of the file genStimulus.m. Running genStimulus.m, the inputSignal.dat and a plot with the corresponding input stimulus will be generated. Using the genStimulus.m code of GitHub, a sum of two sinusoids (high frequency and low frequency) waves are generated and graphically shown in Figure A4.

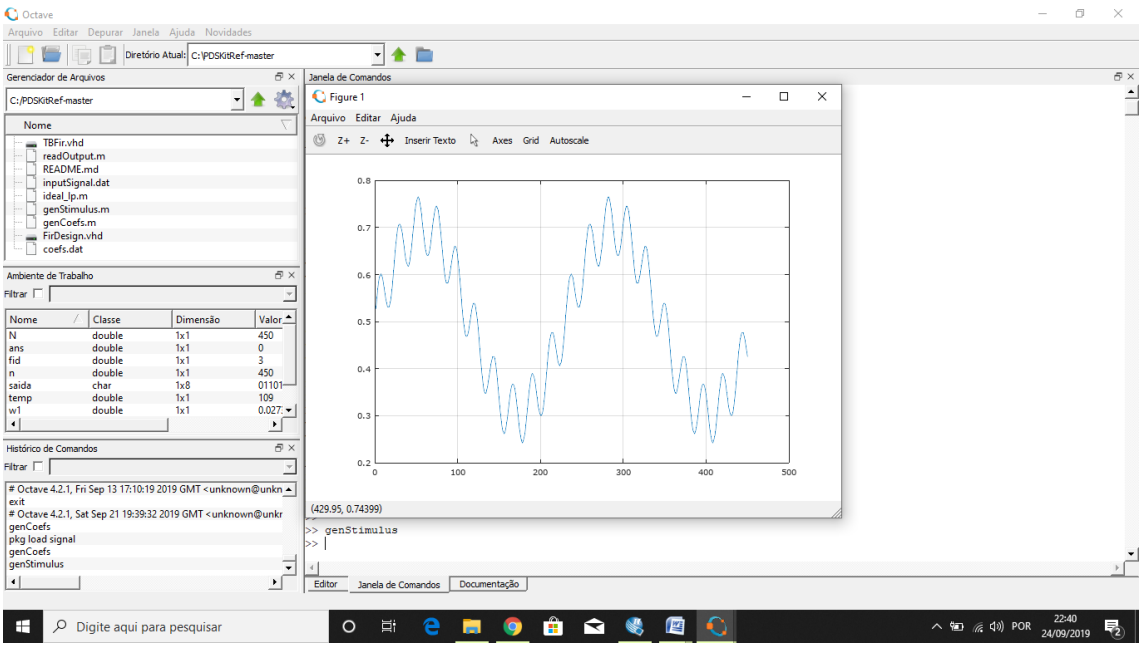


Figure A4. Plot of the two signals sum .

In the next steps, we create the project for the circuit simulation in ModelSim (Figure 5). We are starting with clicking the button JumpStart (Figure A6).

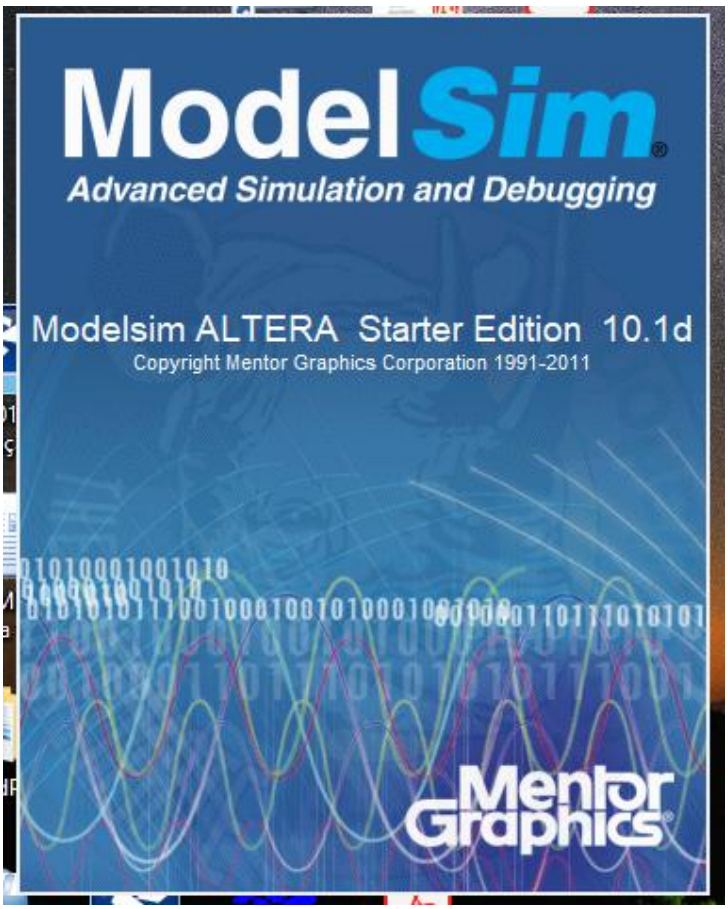


Figure A5. Initial screen of ModelSim.

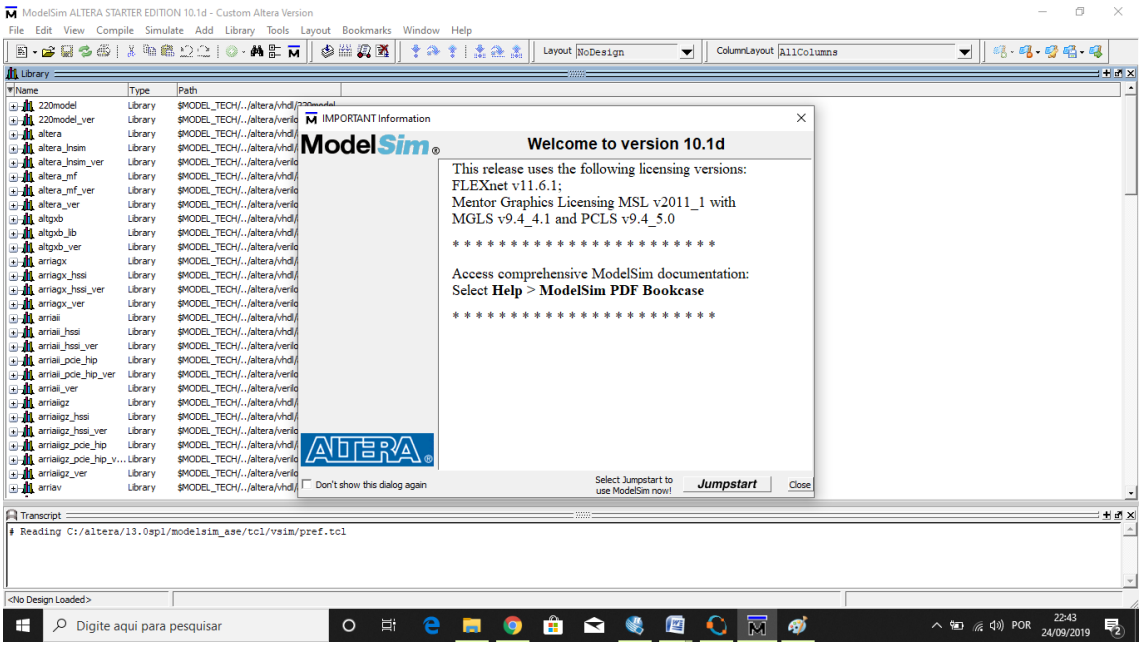


Figure A6. Starting ModelSim.

Subsequently, we create the project for the simulation (Figure A7), specify the file location and project name (Figure A8), add the simulation files (Figures A9 and A10), compile all files (in this case the files are FirDesing.vhd and TBFir.vhd) (Figure A11, and finally, start the simulation (Figure A12). When we start the simulation, the Testbench File (in this project TBfir) is also specified (Figure A13).

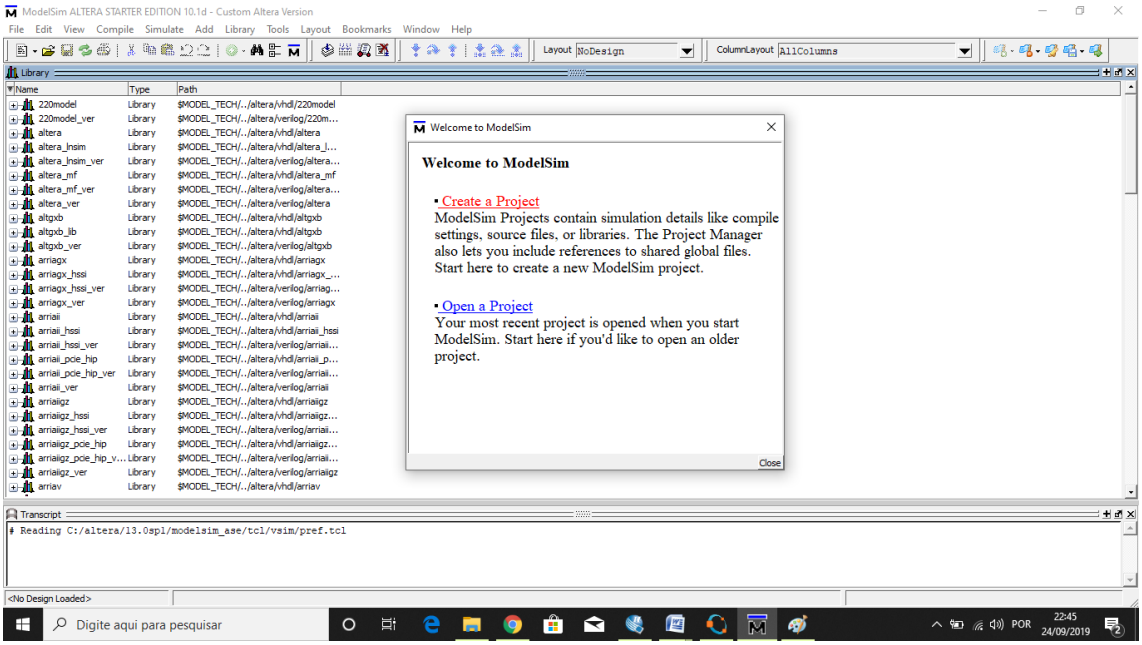


Figure A7. Creating a Project with ModelSim.

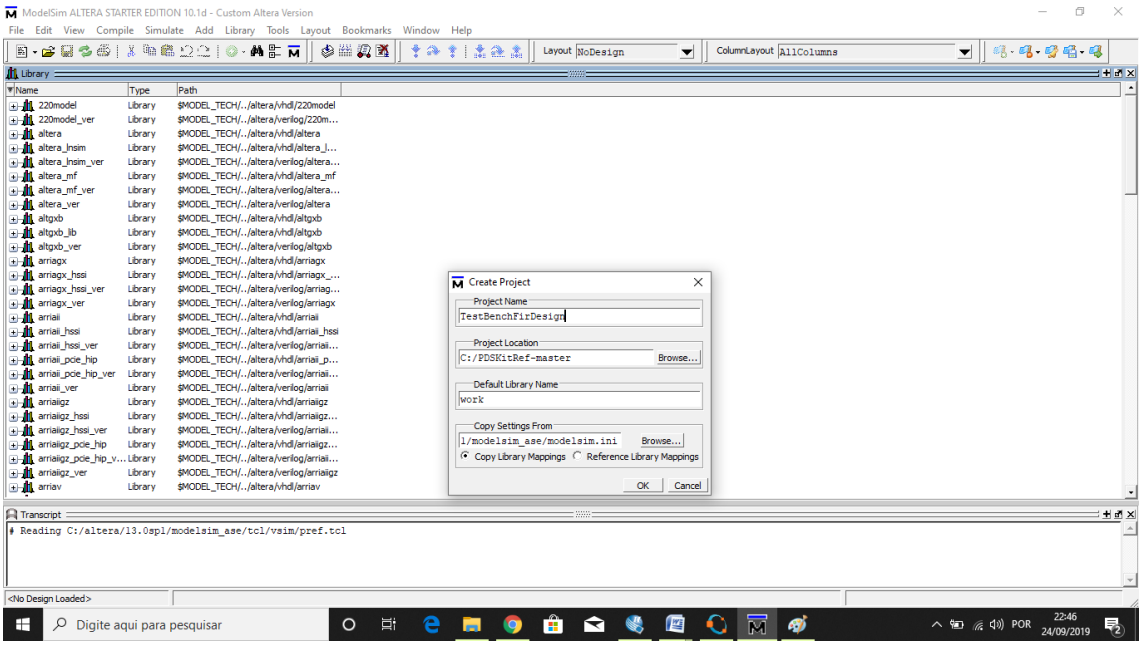


Figure A8. Creating a Project with ModelSim. Naming Project and directory.

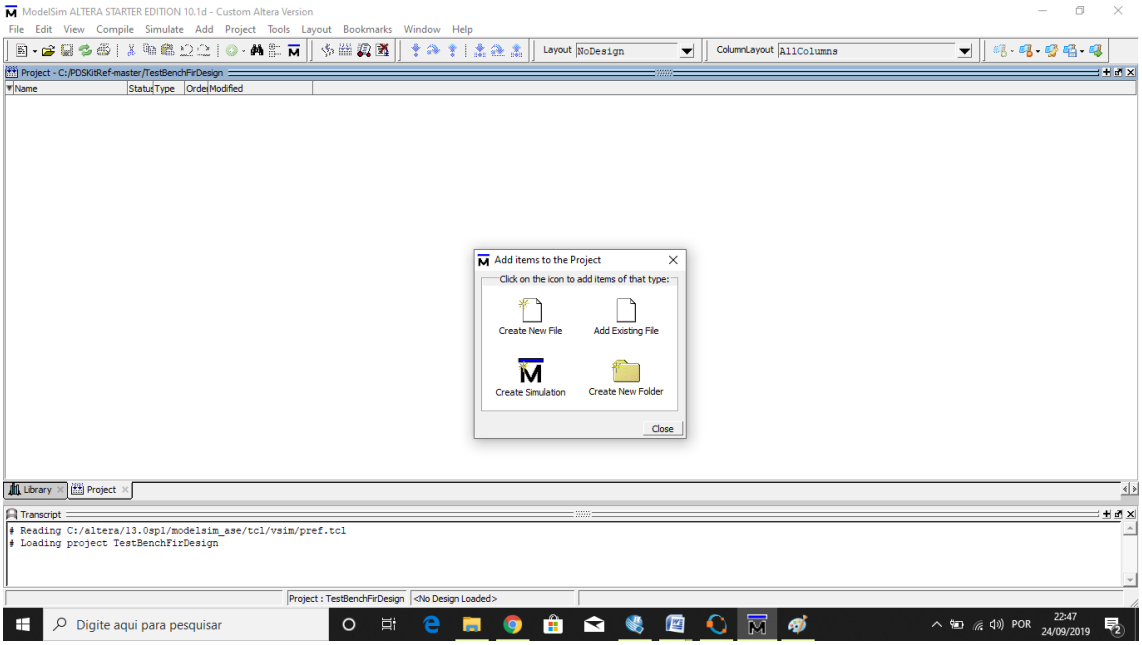


Figure A9. Creating a Project with ModelSim. Open a folder.

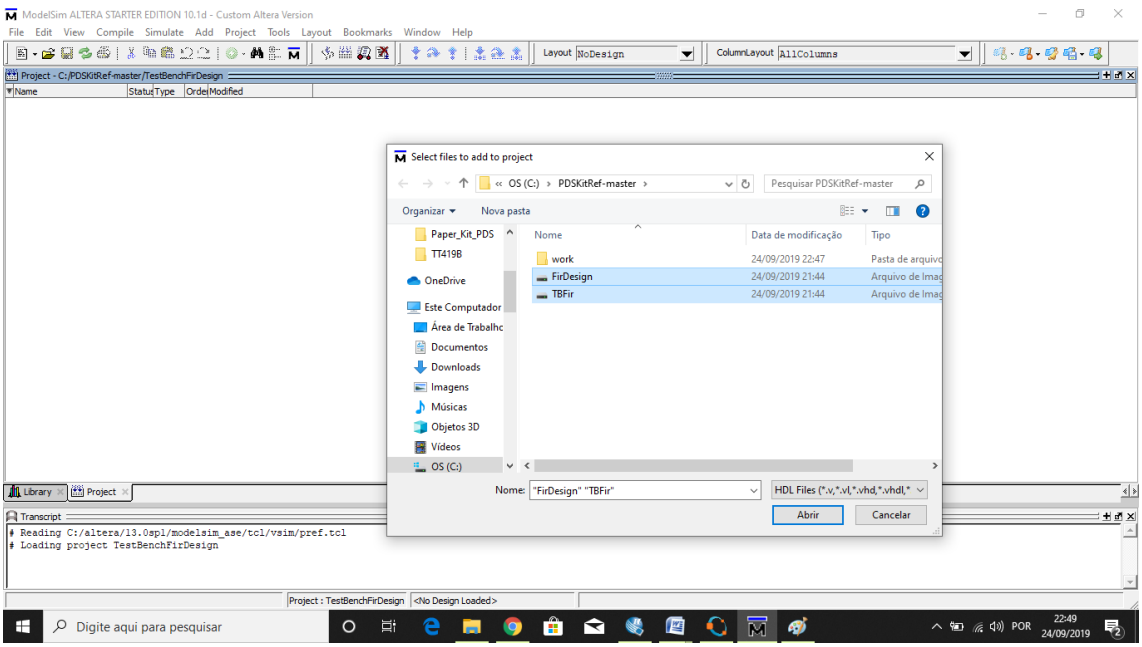


Figure A10. Adding files.

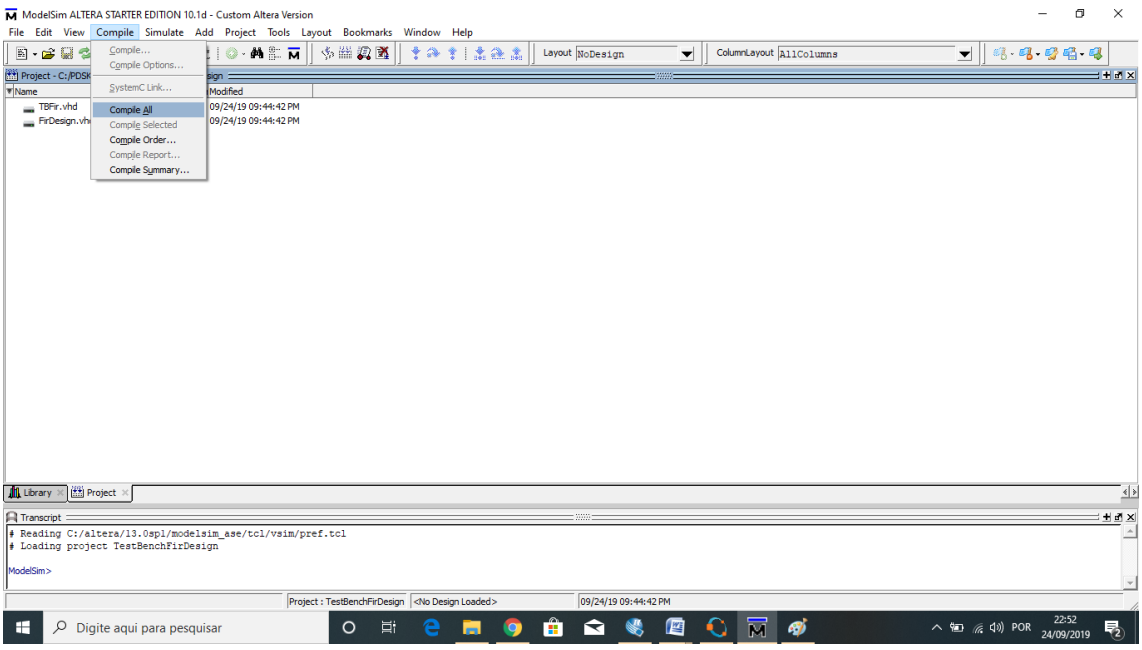


Figure A11. Compiling the file.

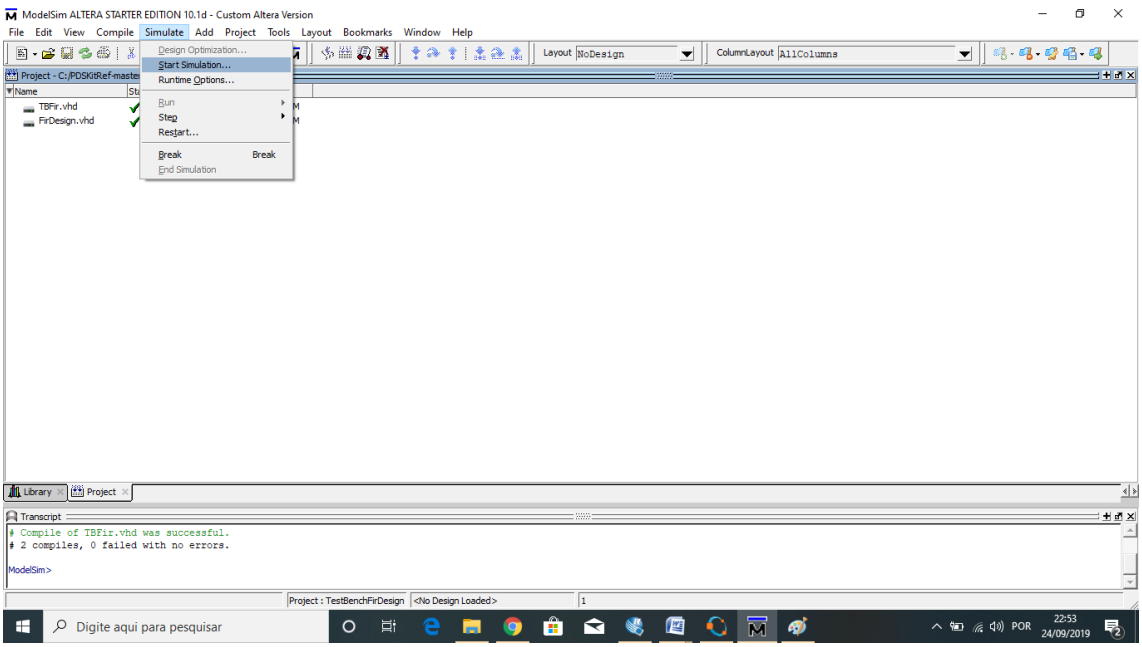


Figure A12. Starting simulation.

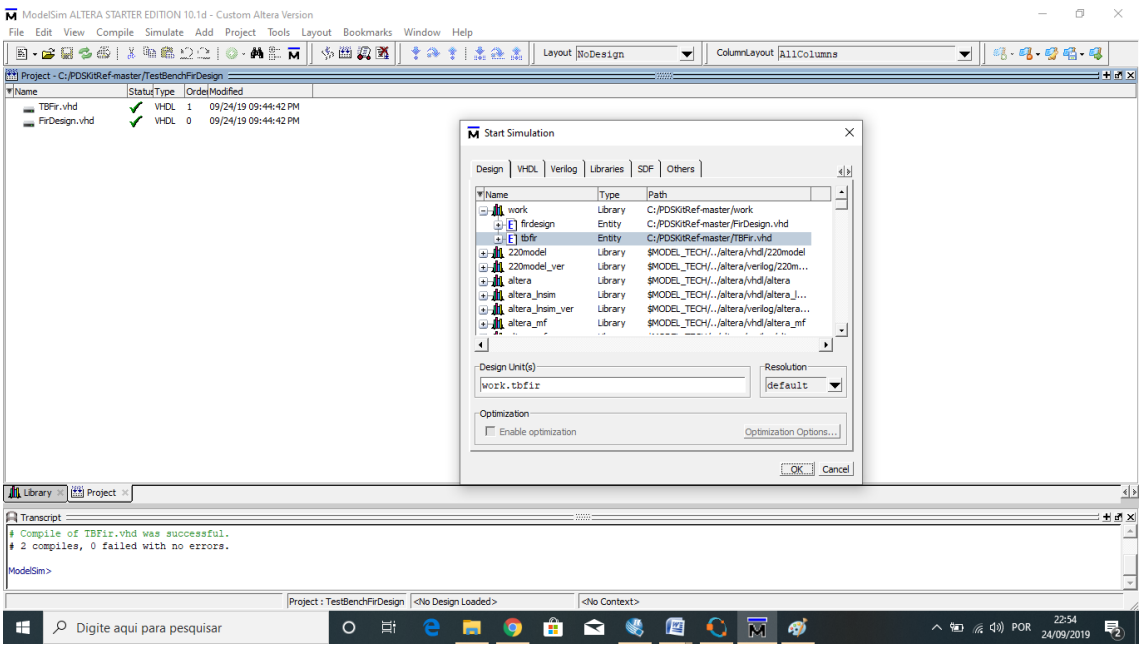


Figure A13. Specifying the Testbench file.

The simulation is run in the environment of Figure A14. In the project the input vector is 9 µs long. After the simulation, the **TBFir.vhd** will generate the **outputSignal.dat** file using the simulation outputs provided by ModelSim (Figure A15).

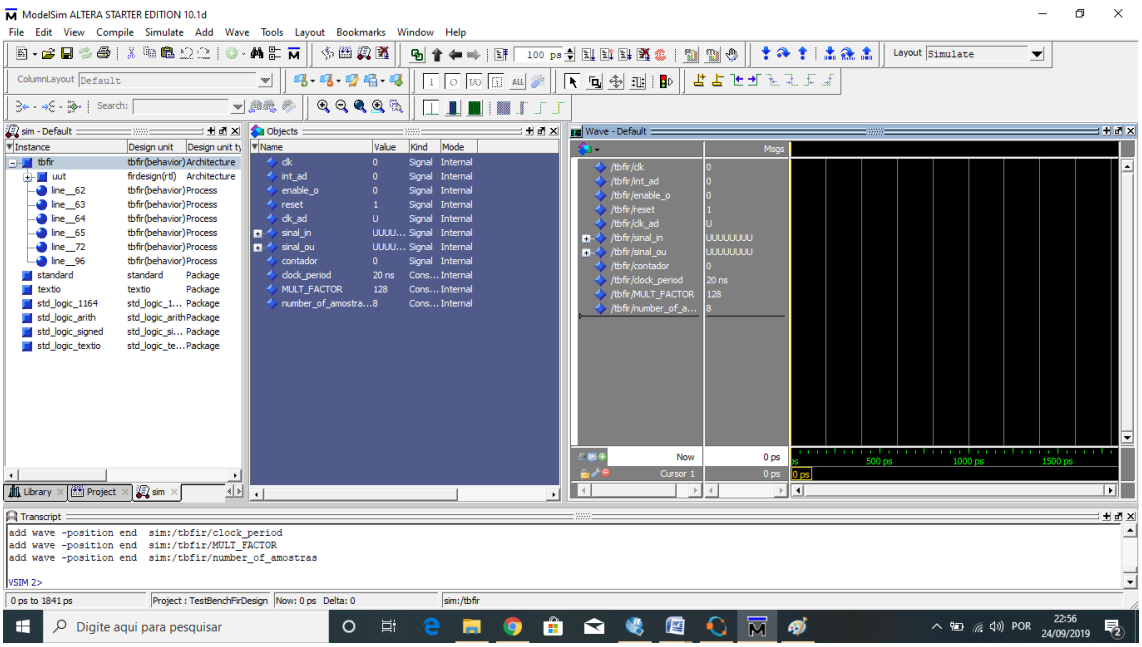


Figure A14. Environment of the simulation.

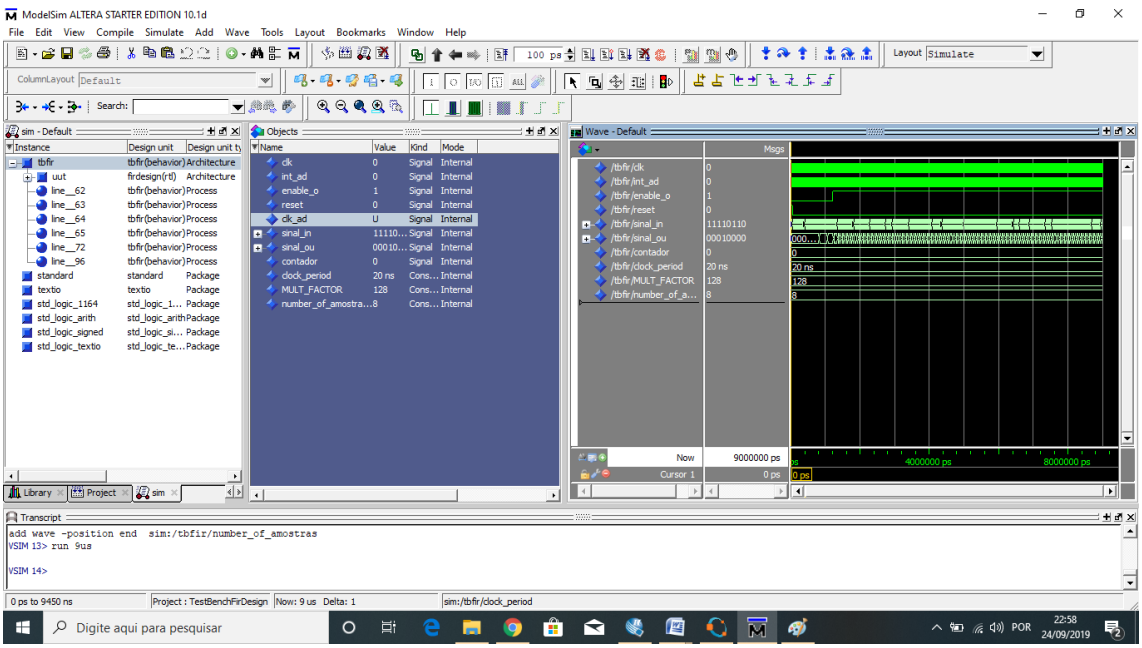


Figure A15. The simulation result provided by ModelSim.

The results are graphically presented in Octave. Running **readOuput.m** (in Octave), we obtain plots with the input (**inputSignal.dat**) and output (**outputSignal.dat**) simulation wave forms (Figure A16). In this case a high-pass filter was implemented cutting off the

low frequency input and allowing high frequency pass. This is the filter example you will find in the GitHub repository.

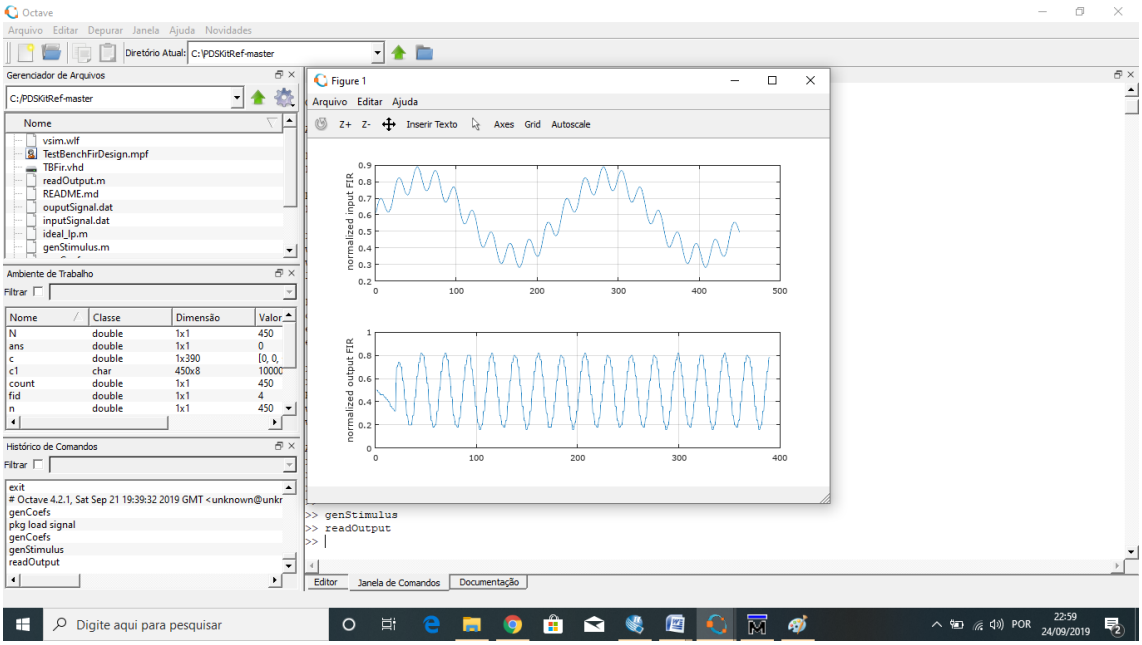


Figure A16. Simulation result.

4) Synthesizing

The next step is to synthesize the VHDL code (**FirDesign.vhd**) in order to create the FPGA programming file to the evaluation board. For this purpose, we use the free version of Quartus II (Figure A17).

Figure A17. Opening the Quartus II.

The first screen of the Quartus II environment is shown in Figure A18.

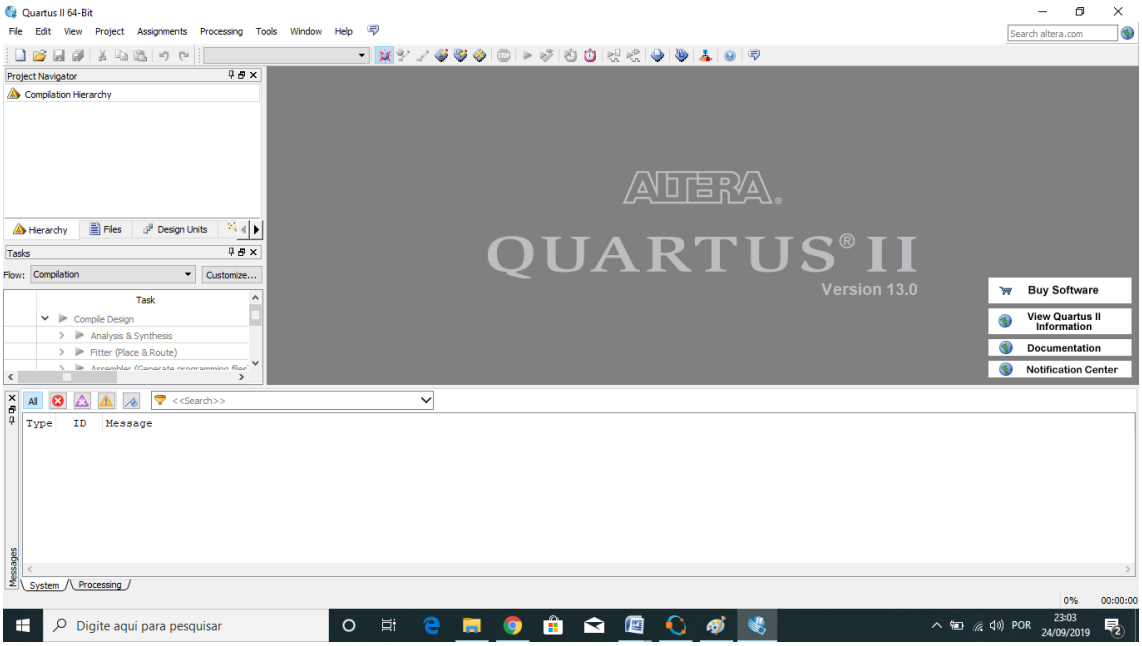


Figure A18. Initial screen of Quartus II.

We start the process creating a new project using the project Wizard (Figures A19 and A20).

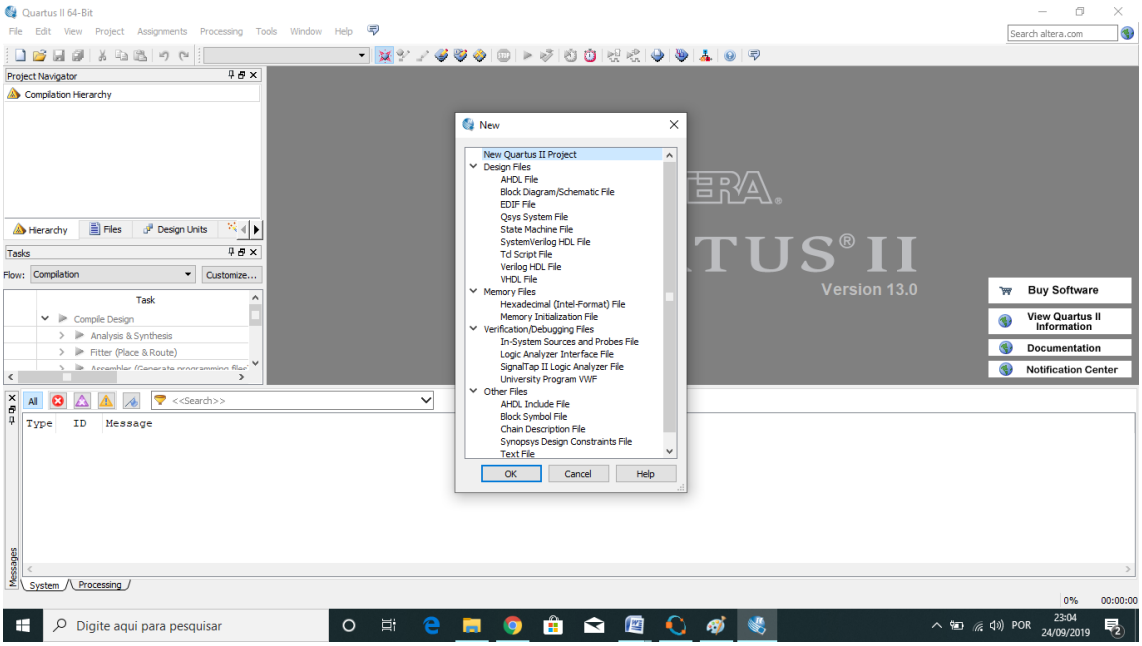


Figure A19. Creating a new project in Quartus II.

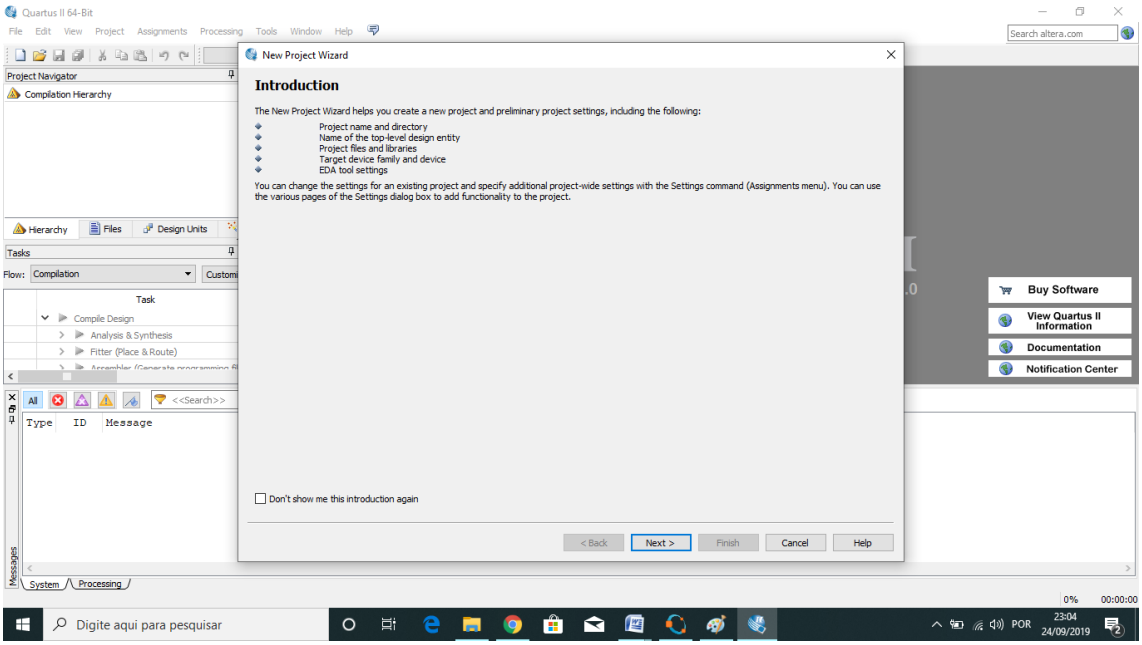


Figure A20. Creating a new project in Quartus II.

In the next steps, we name the project and directory (Figure A21), add the FIR VHDL file (**FirDesing.vhd**) to the project (Figure A22), and specify the device according to the choosen hardware (Figure A23). By using Pin Planner in Quartus II we assign the FPGA pins according to the chosen hardware with a graphically drop down pin assignment (Figure A24).

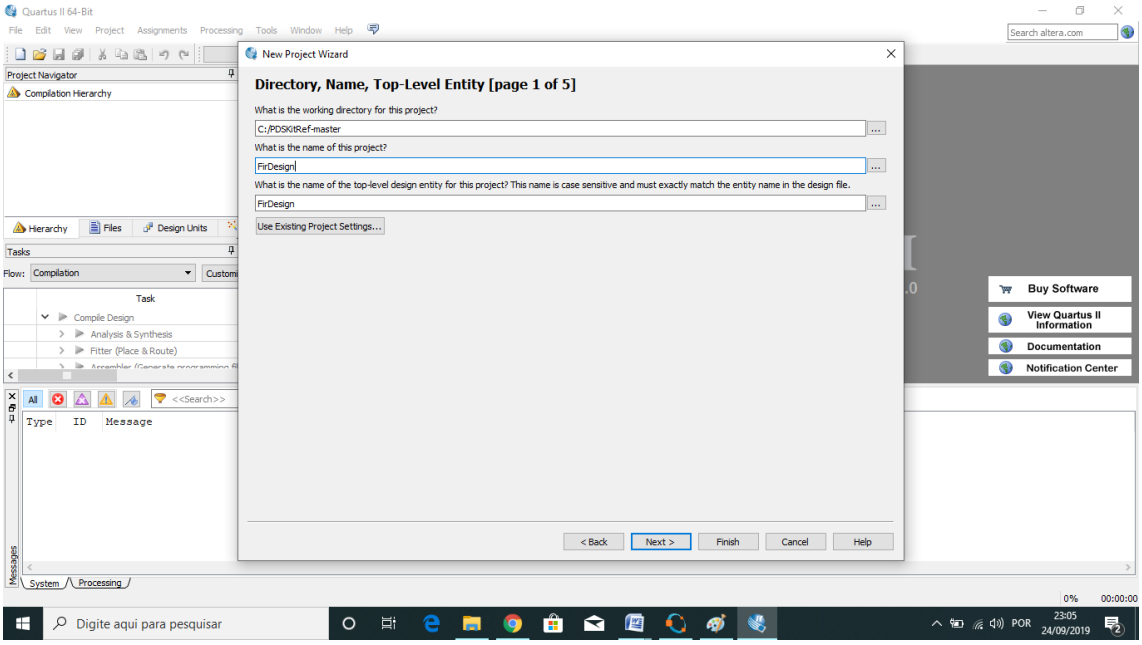


Figure A21. Naming Project and directory in Quartus II.

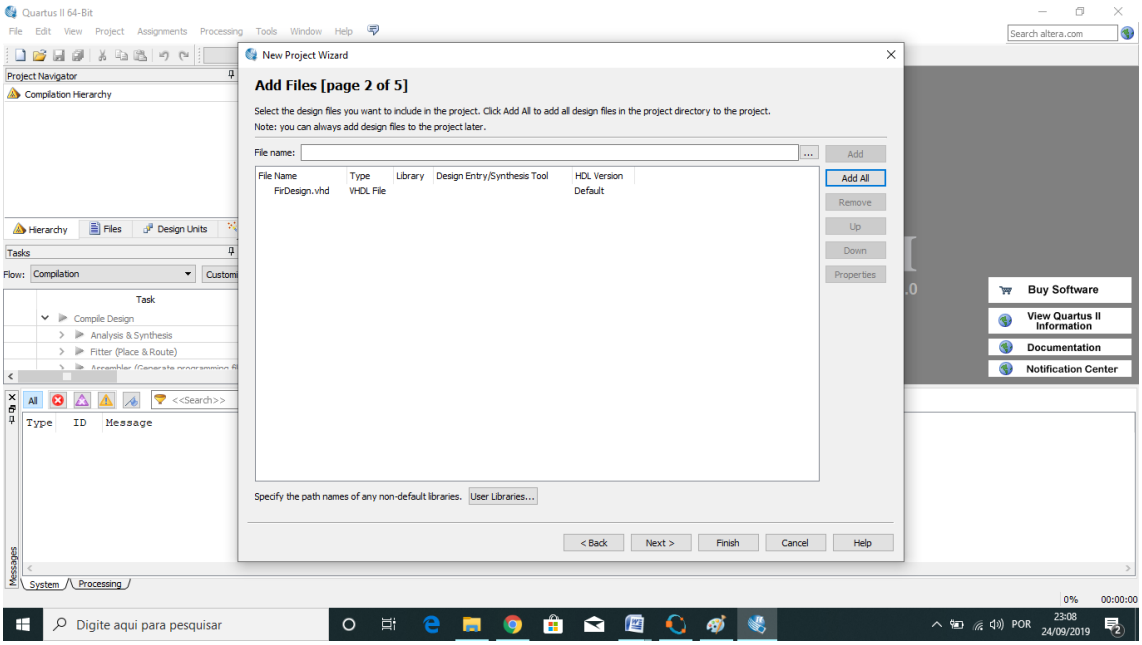


Figure A22. Selecting and adding design files.

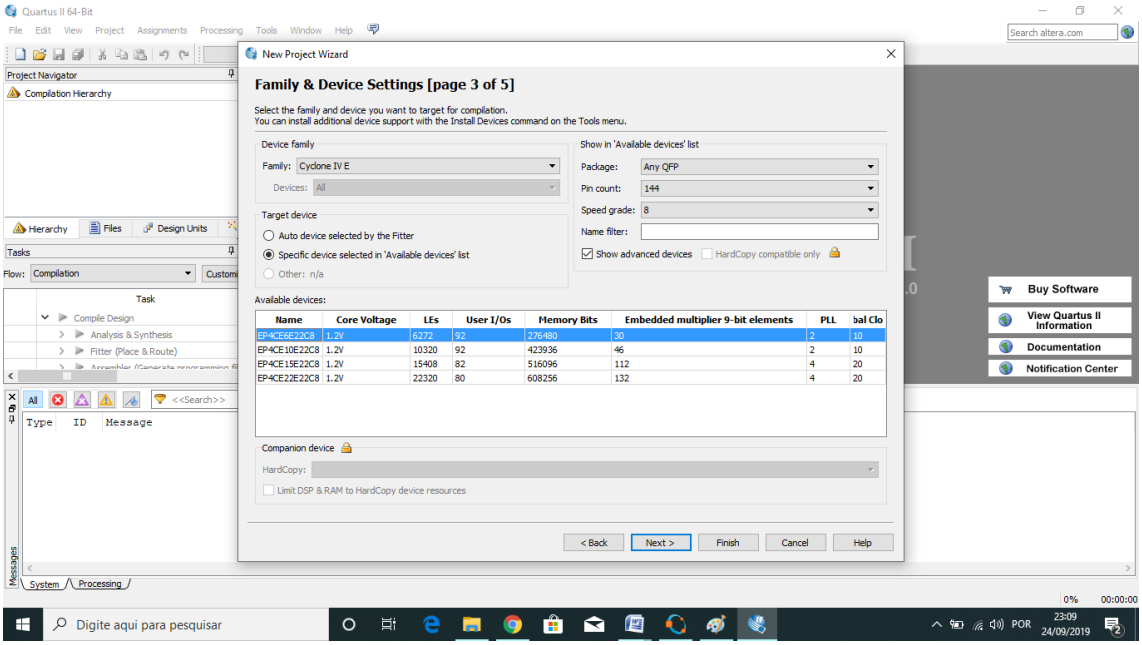


Figure A23. Choosing the device.

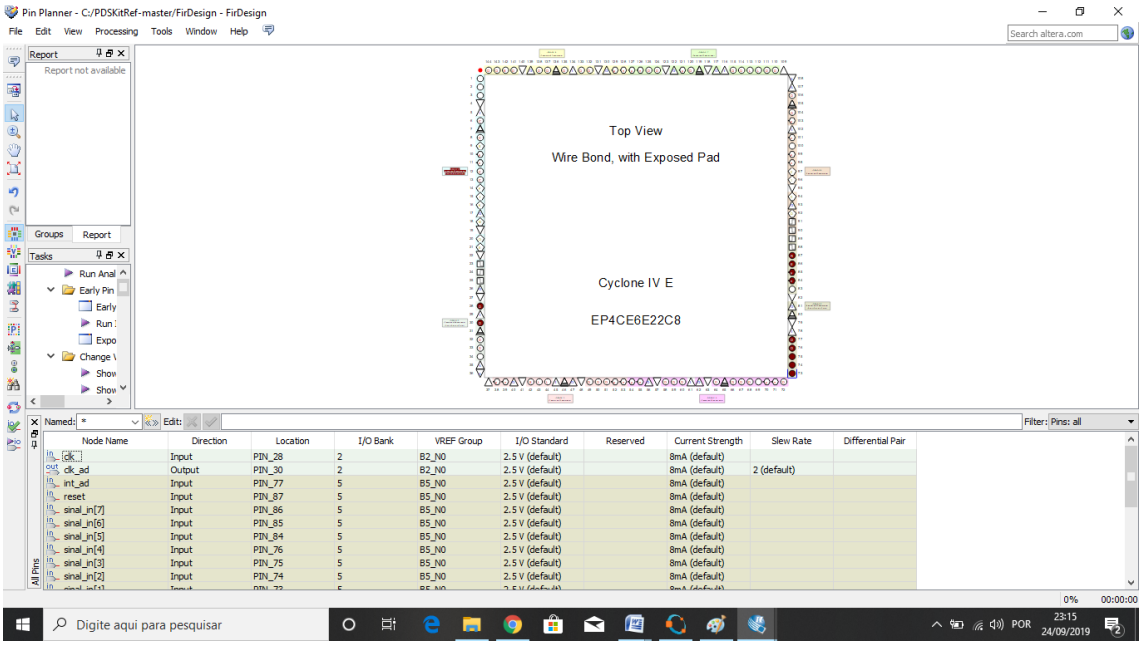


Figure A24. Device pins assignment.

The next step is to synthesize the code (Figures A25 and A26), generate the program file (Figure A27), and load the output file in the FPGA device using "Programmer" in QuartusII (Figure A28). A ByteBlaster Cable must be connected to the computer to the evaluation FPGA hardware.

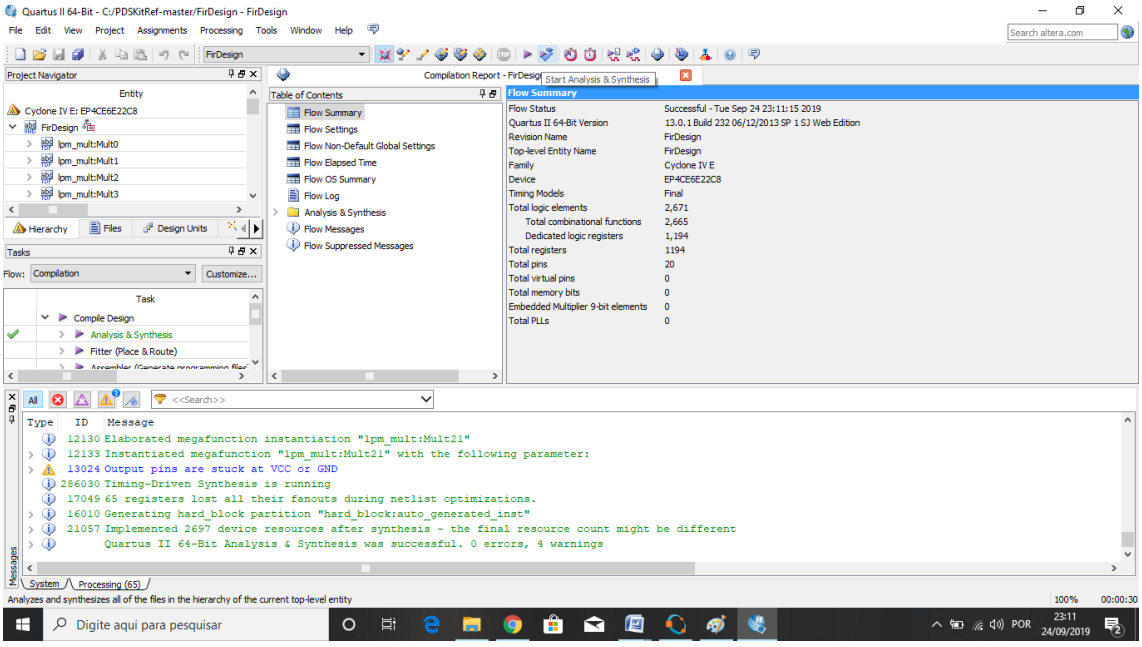


Figure A25. Screen of Synthesis of Quartus II.

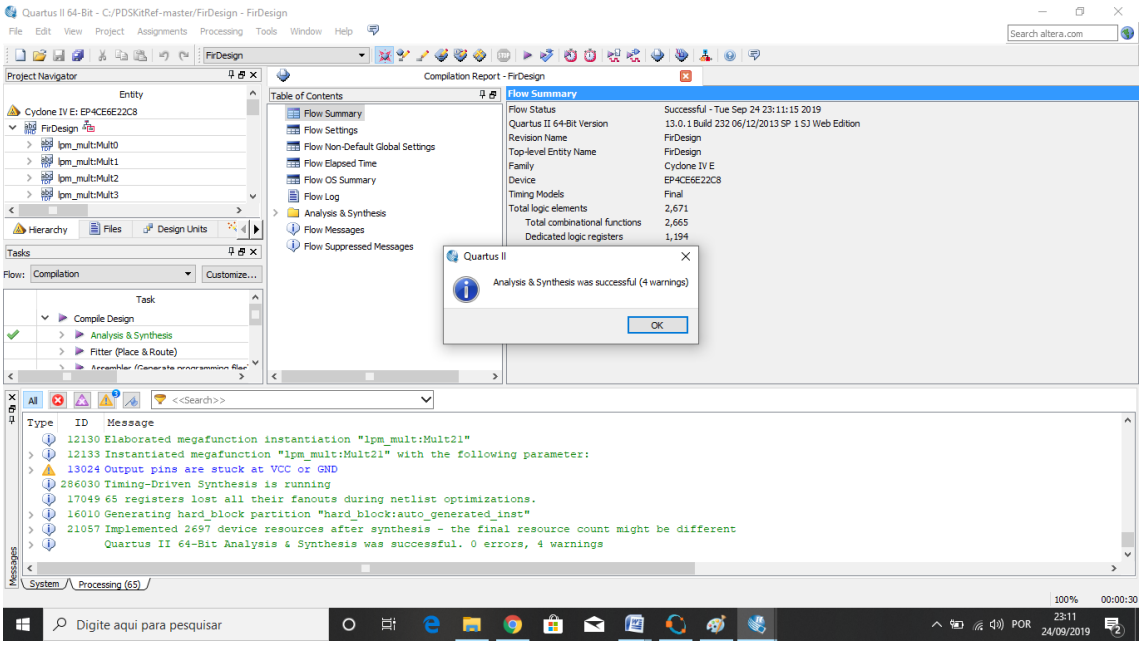


Figure A26. Last screen of synthesis of Quartus II.

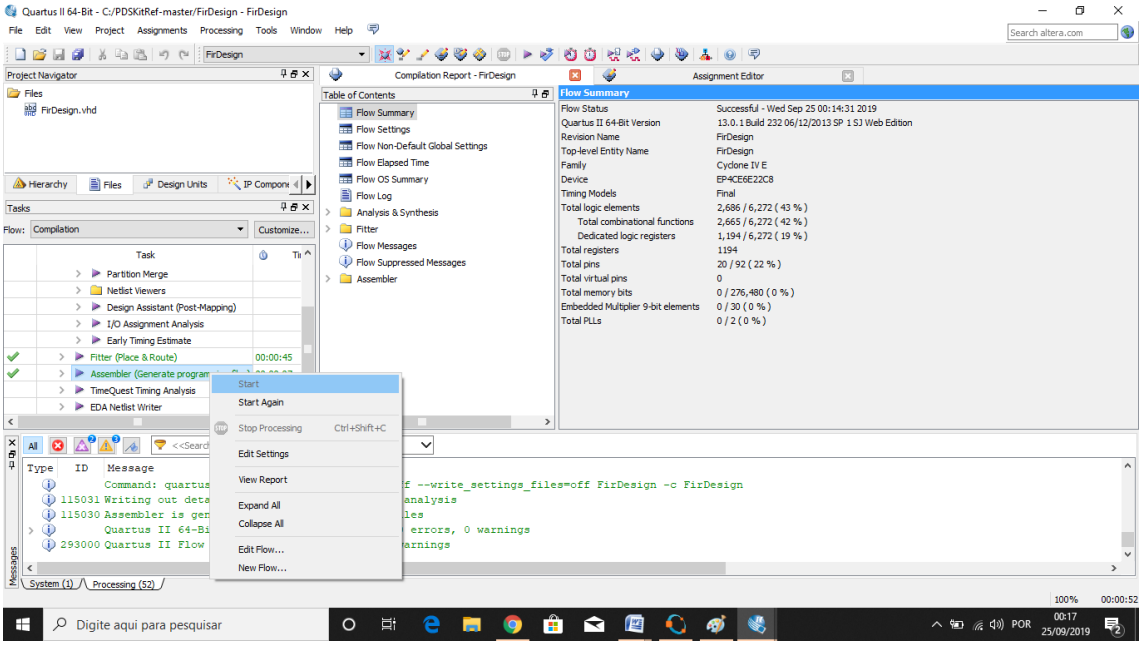


Figure A27. Generation of program file.

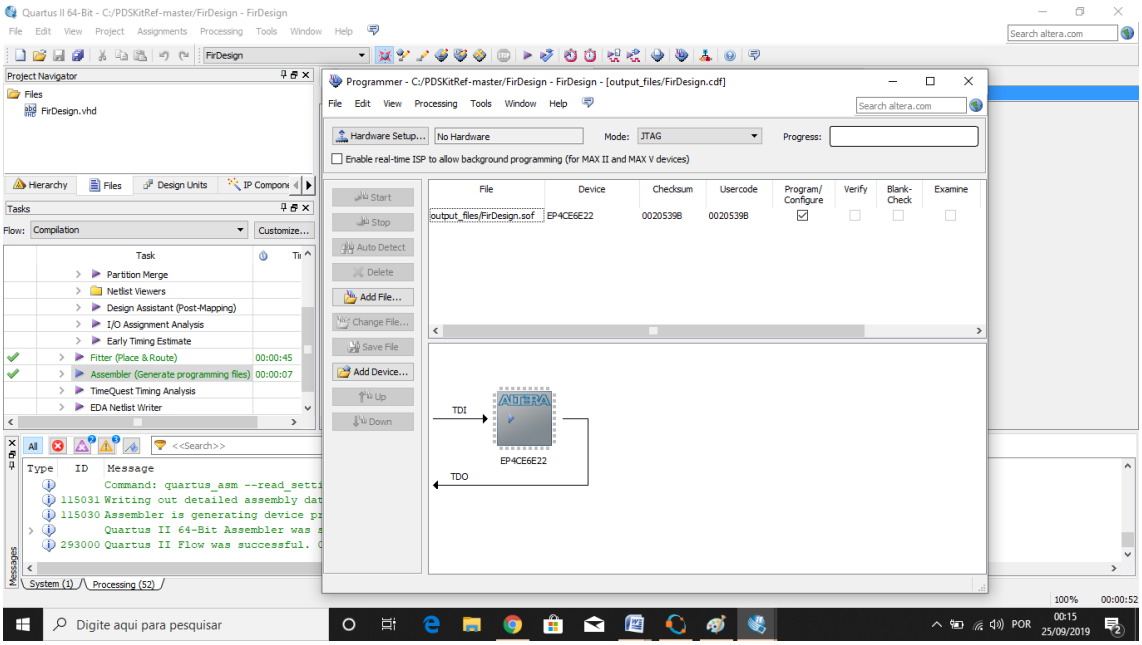


Figure A28. Loading the file in the device using Programmer in QuartusII.