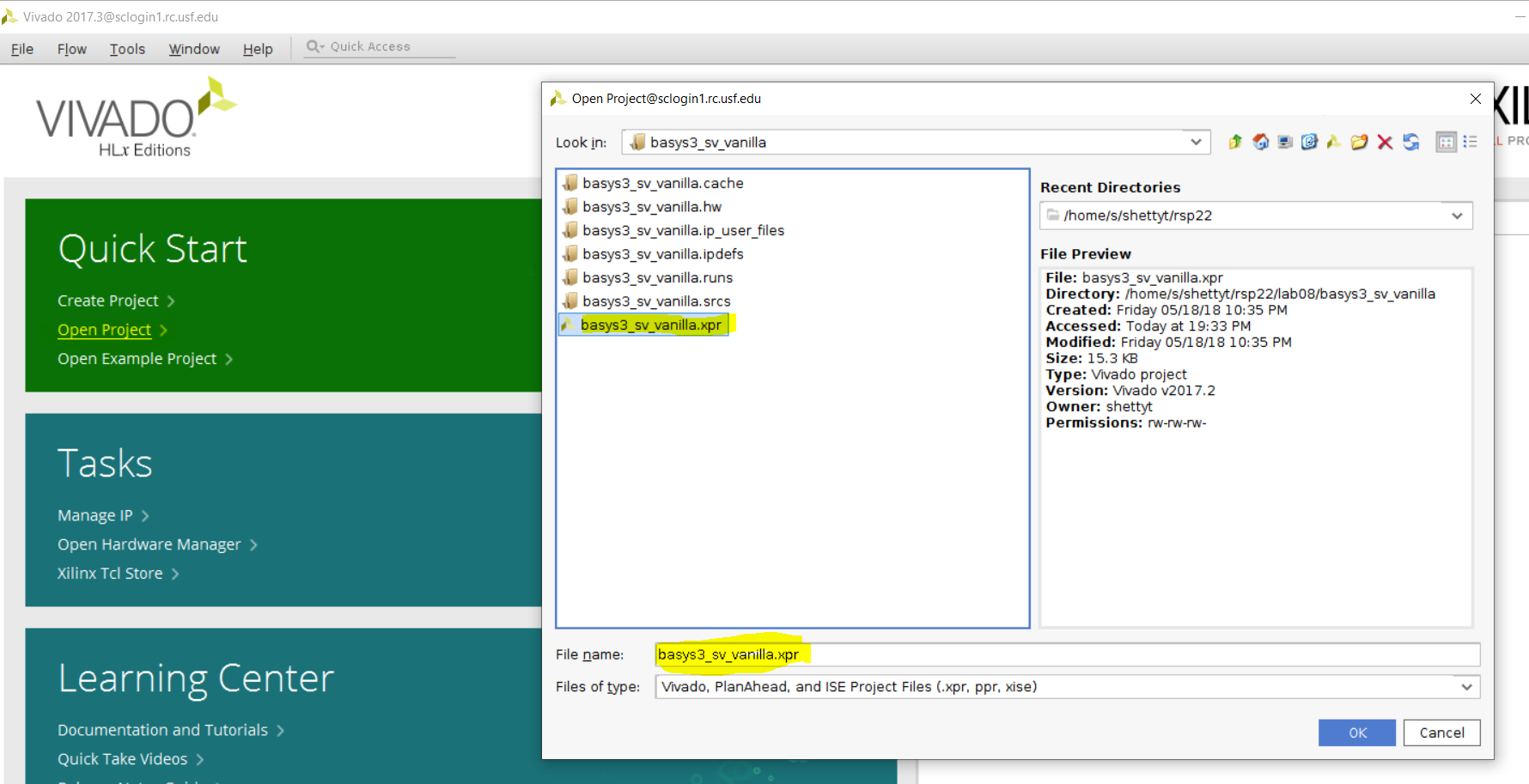
Steps for Lab08 – Vanilla

1. Copy ‘viva’, ‘sdk’, and ‘viva16’ from /shares/eel6729.001s22.jg/init22 to rsp22 folder in the server.(via command “cp /shares/eel6729.001s22.jg/init22/viva .”)
2. Change the file permissions for all the three software as ‘744’ via command “chmod 744 viva”.
3. Download the source code for Vanilla project from ‘’ and copy it to the server.
4. Open Xilinx Vivado in the server ( ./viva) and verify that the version of vivado is 2017 as shown in the below fig.

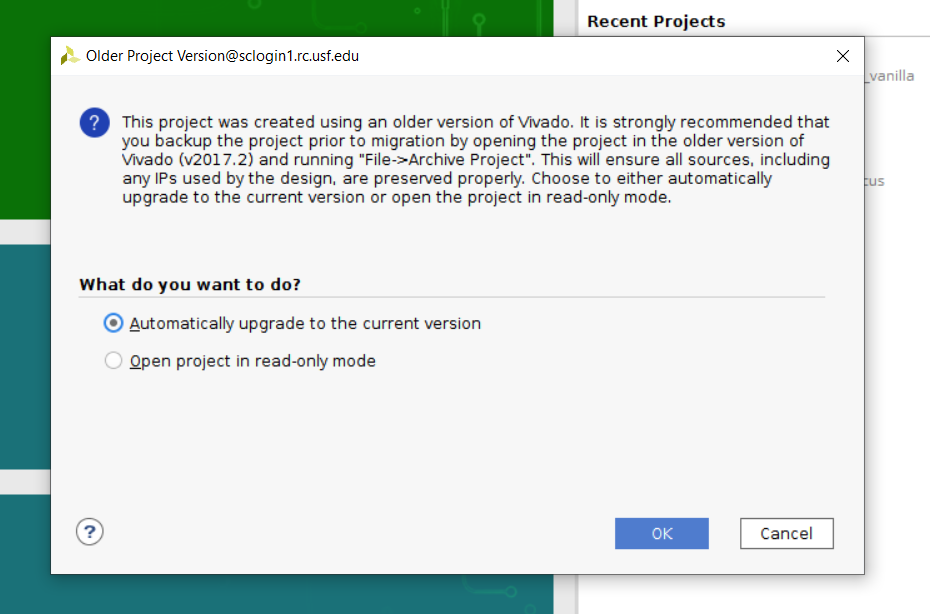
Text

Description automatically generated

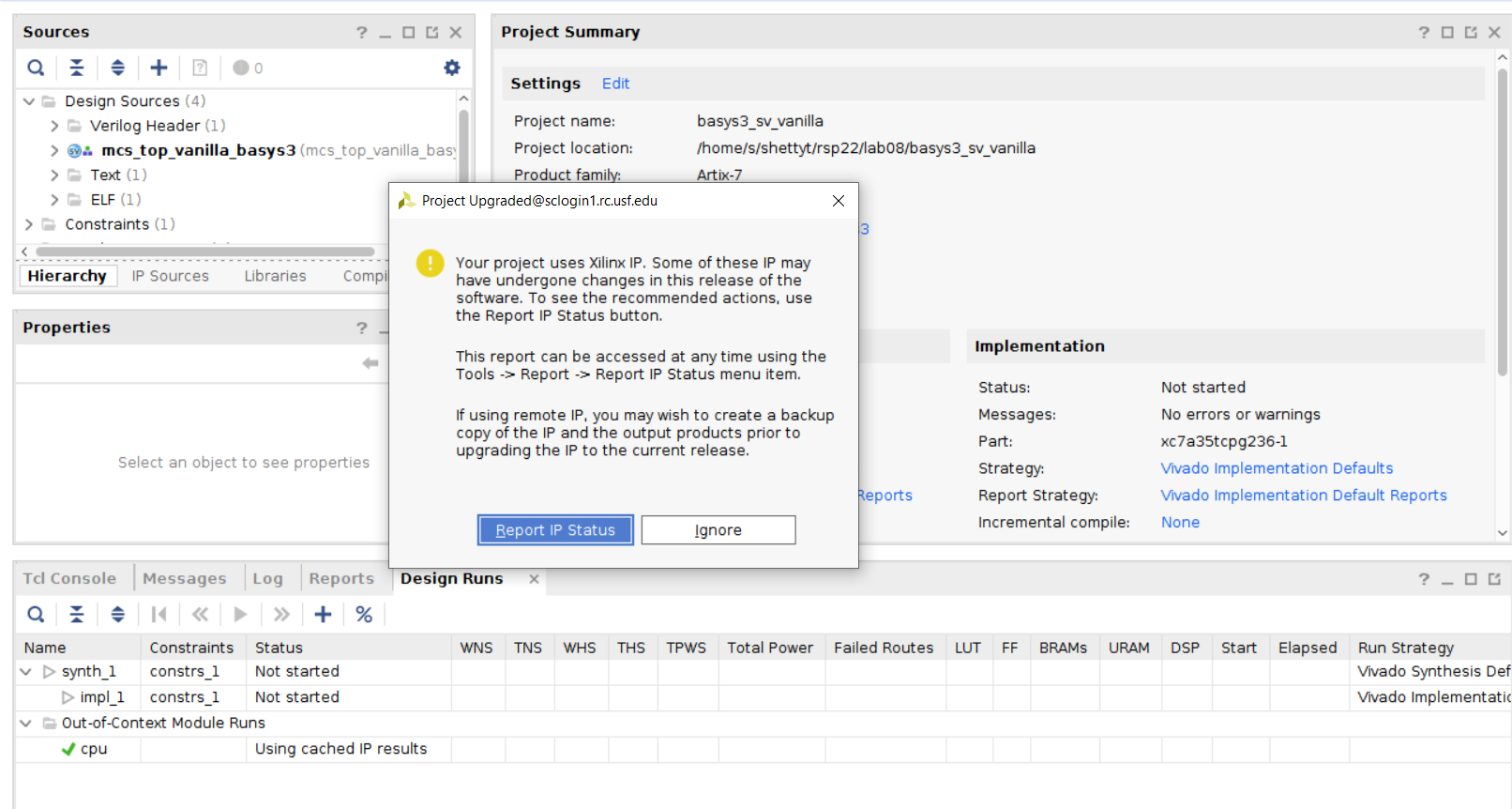
1. When the Vivado GUI is launched, go to ‘Open Project’ and open the .xpr file present in the project (/home/s/shettyt/rsp22/Vanilla/basys3\_sv\_vanilla/basys3\_sv\_vanilla.xpr).



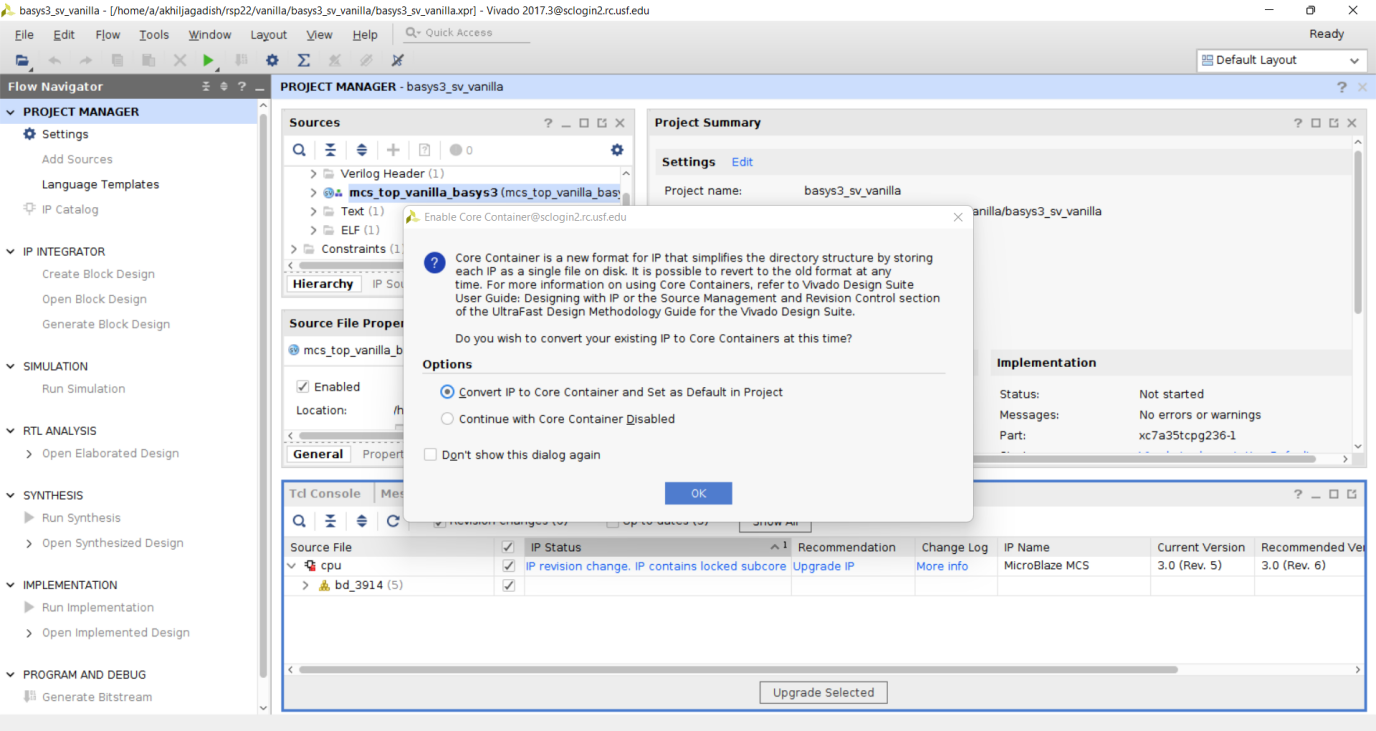
1. Once the project is opened, the below pop-up window occurs, for upgrading vivado. Select “Automatically upgrade” option.



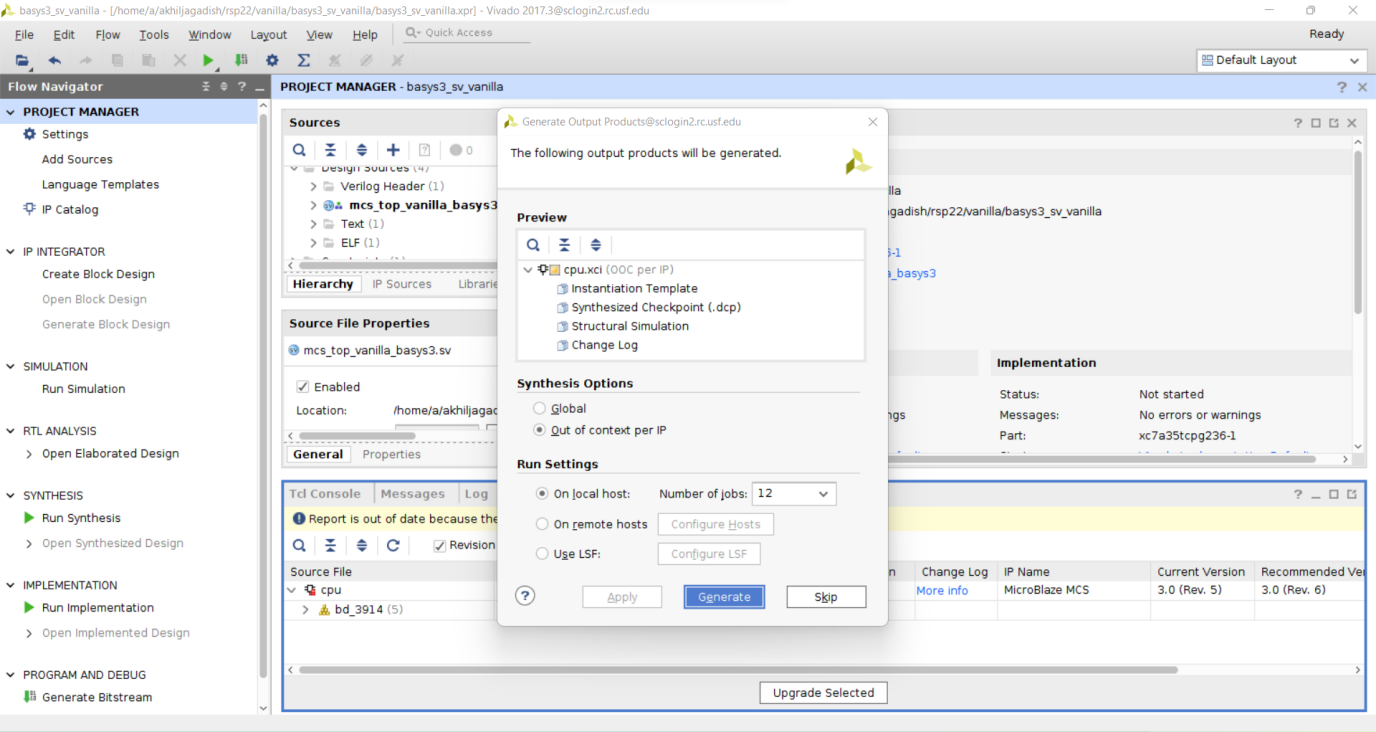
Next, the IP upgrade request pop-up occurs due to updating of vivado version, as shown below. Select “Report IP Status” option.



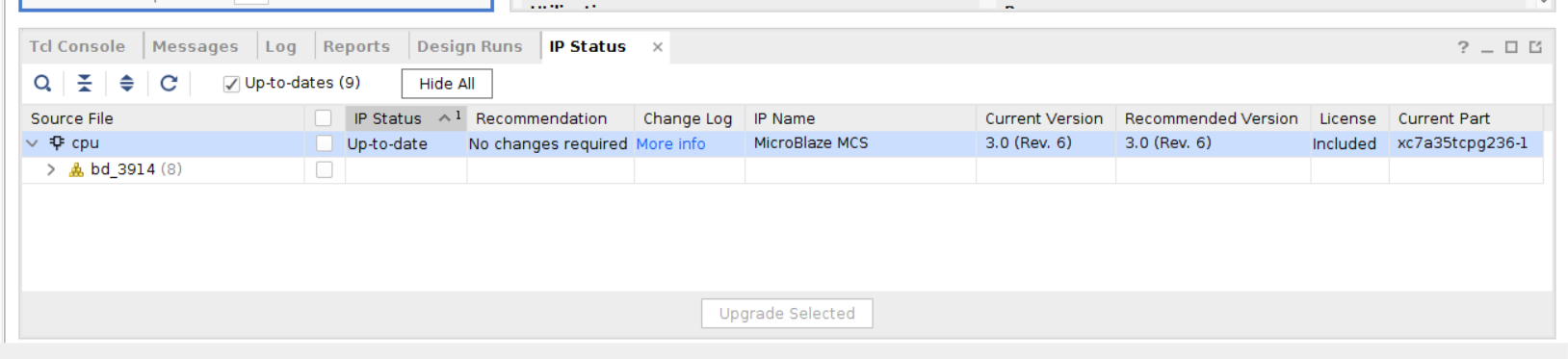
1. In the IP status report bar, select the ‘cpu’ option and give “Upgrade Selected” option.The below pop-up window will be generated.



Choose “Convert IP to core Container” option for the IP upgrade. After that, choose “generate” option for the below pop-up to generate the new IP.

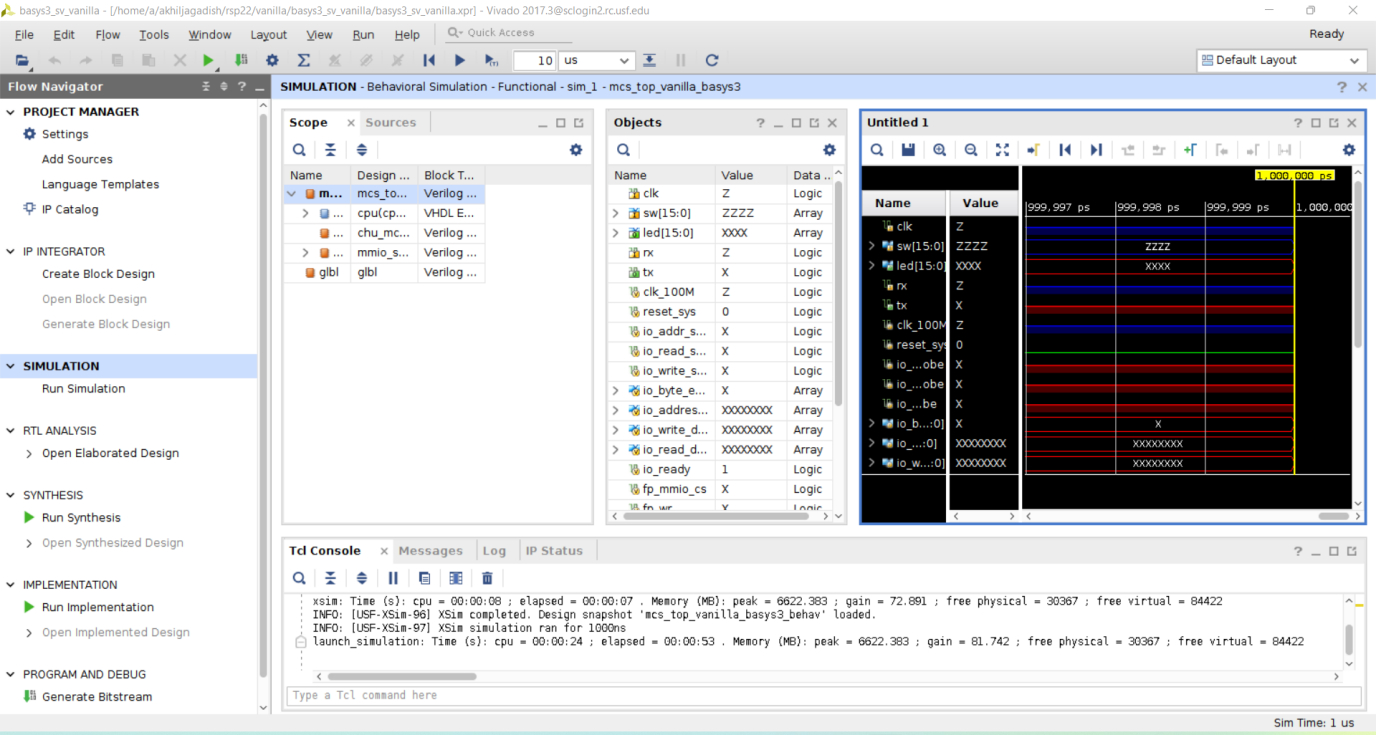


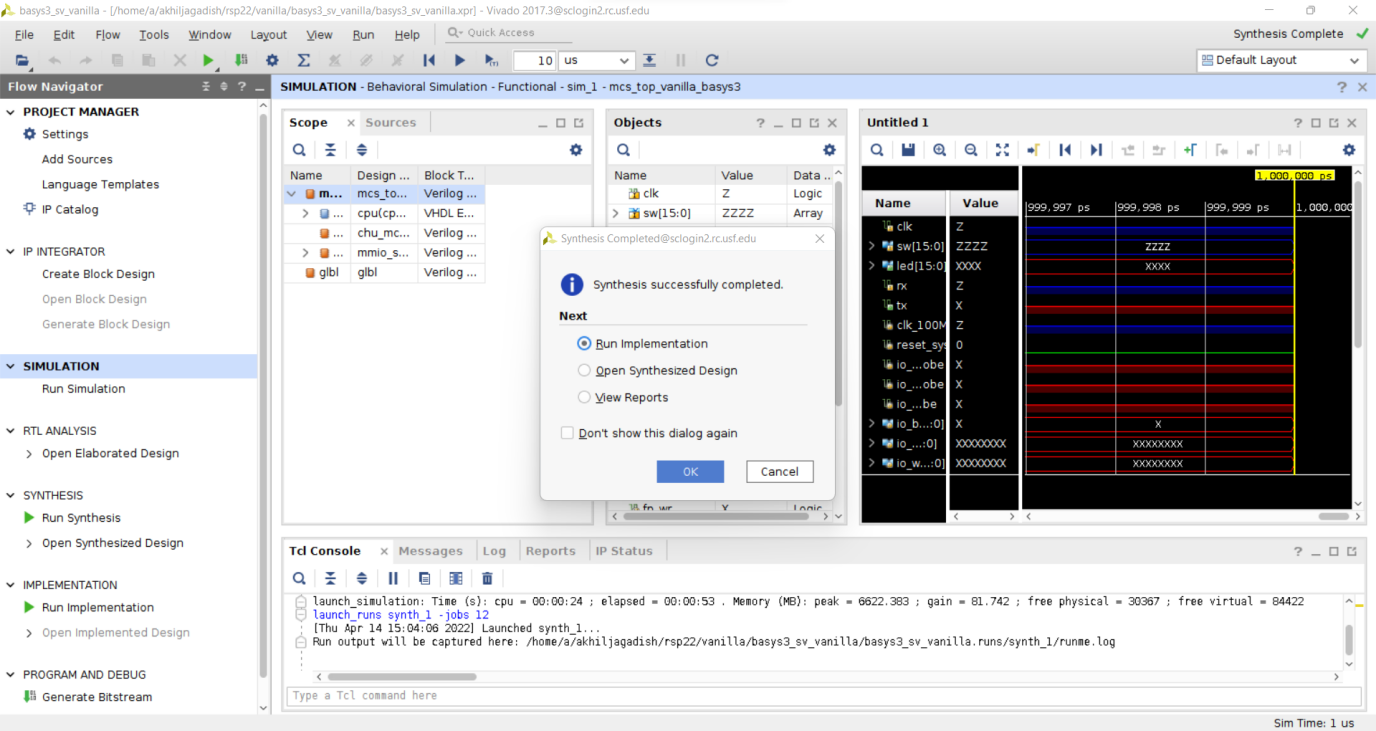
Wait until the update is completed and the current IP status after the update is as below.

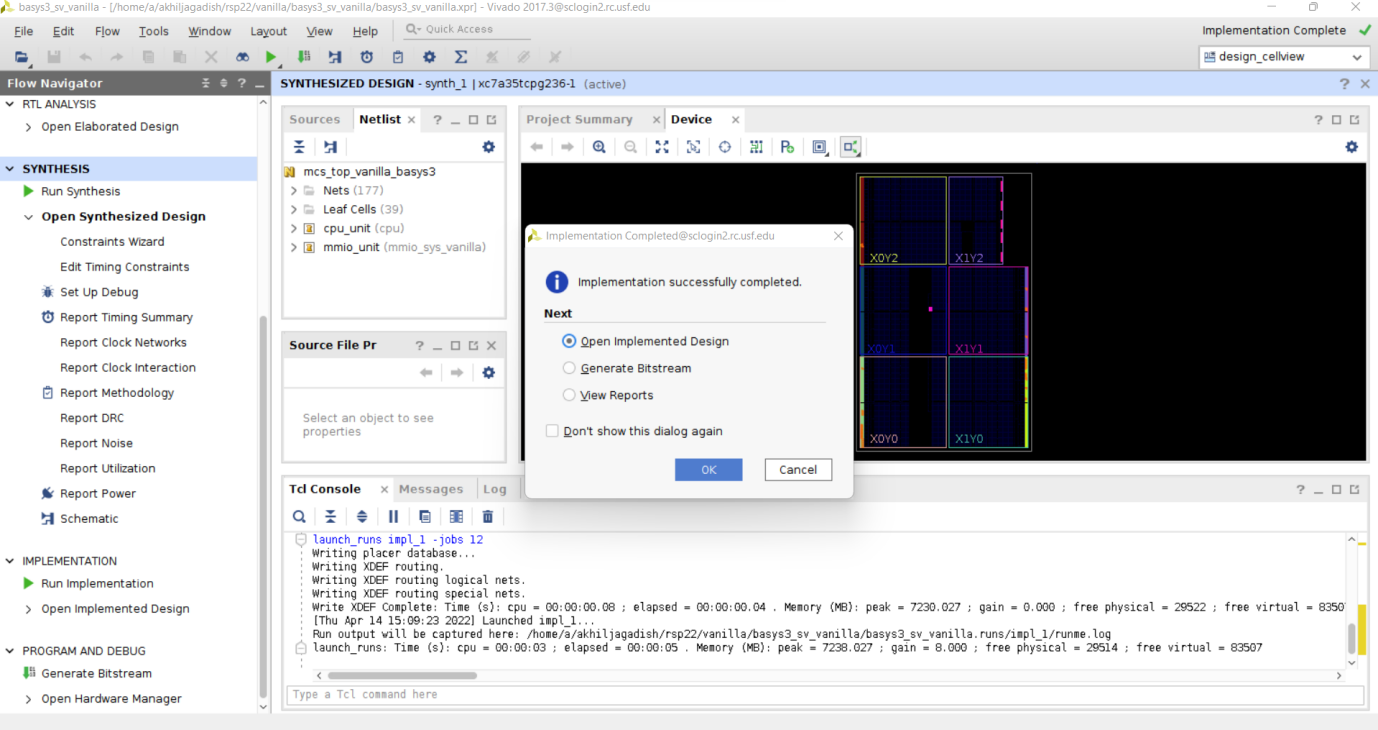


**Generate .bit file:**

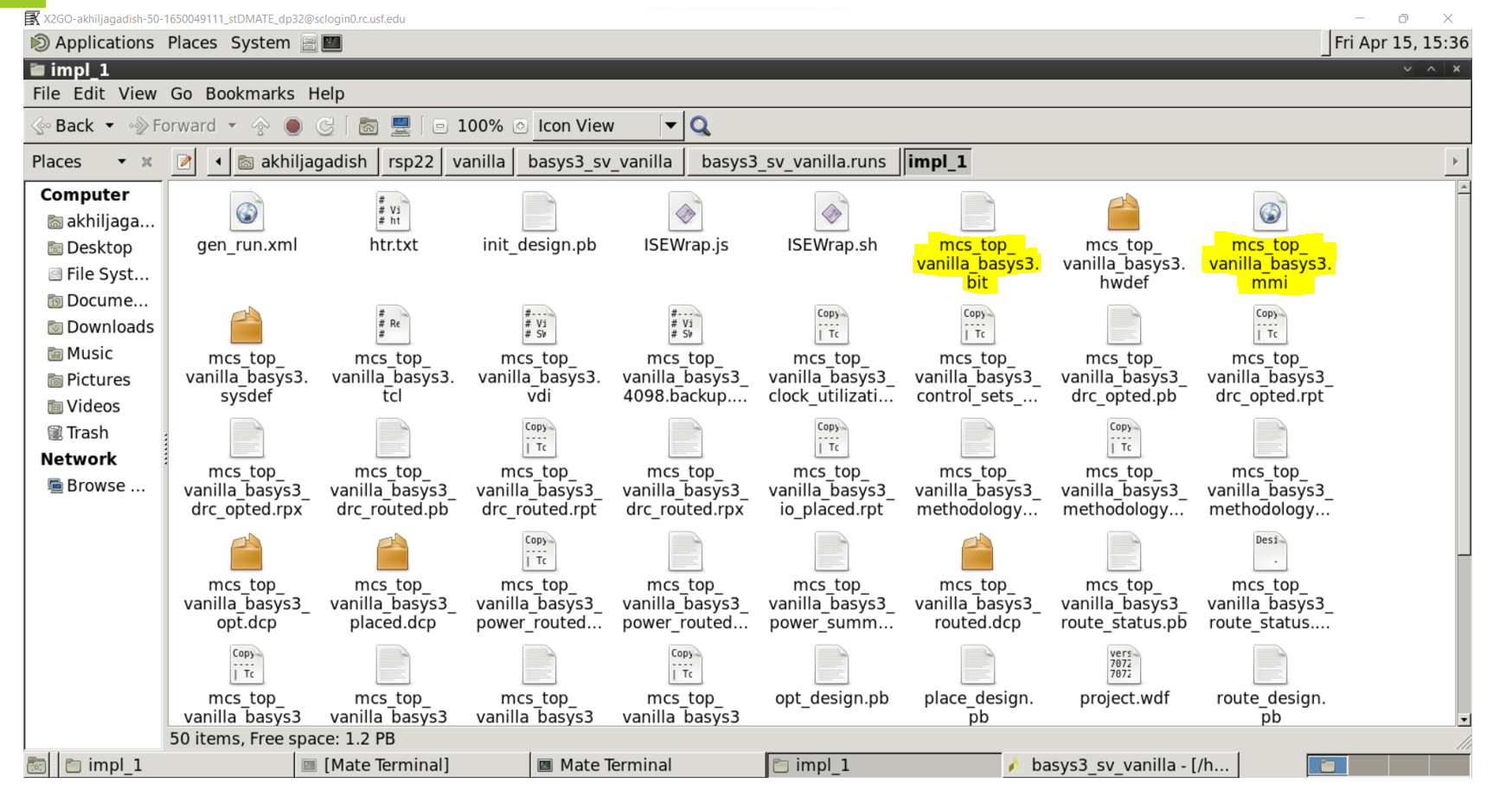
1. First select “run simulation” option available at the left hand side of the vivado window. Next, Run the synthesis, and then run the implementation of the design. After implementation is successful, Generate bitstream. Each steps are shown in the below figure.





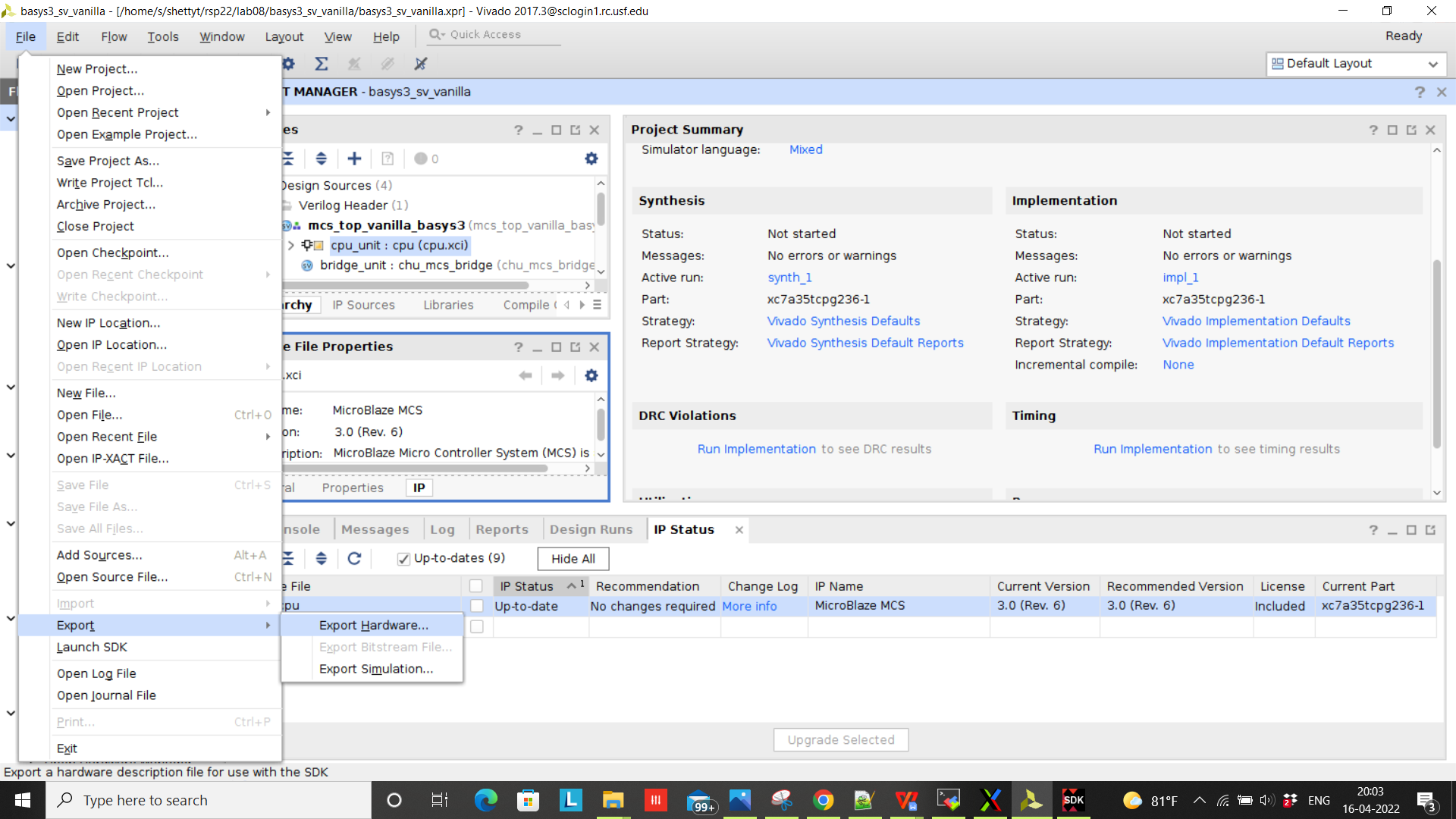


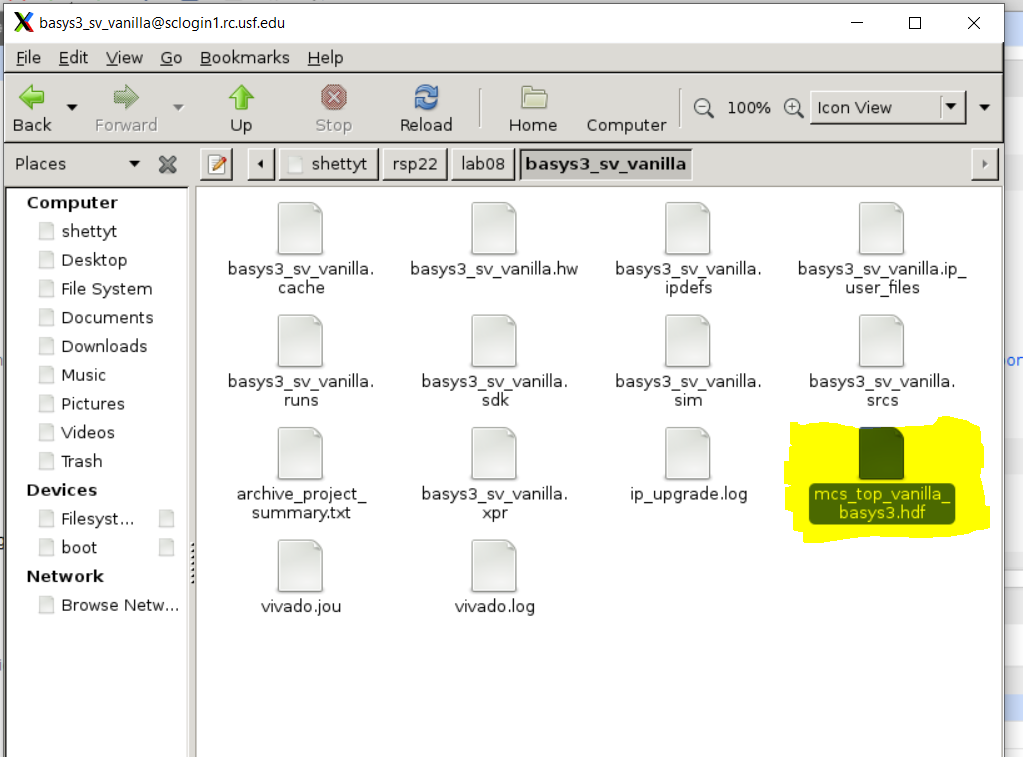
1. Once he bitstream generation is successful, go to ‘/\*project\*/basys3\_sv\_vanilla/basys3\_sv\_vanilla.runs/impl\_1’ folder and check the .bit and .mmi file generated.



**To generate .hdl file:**

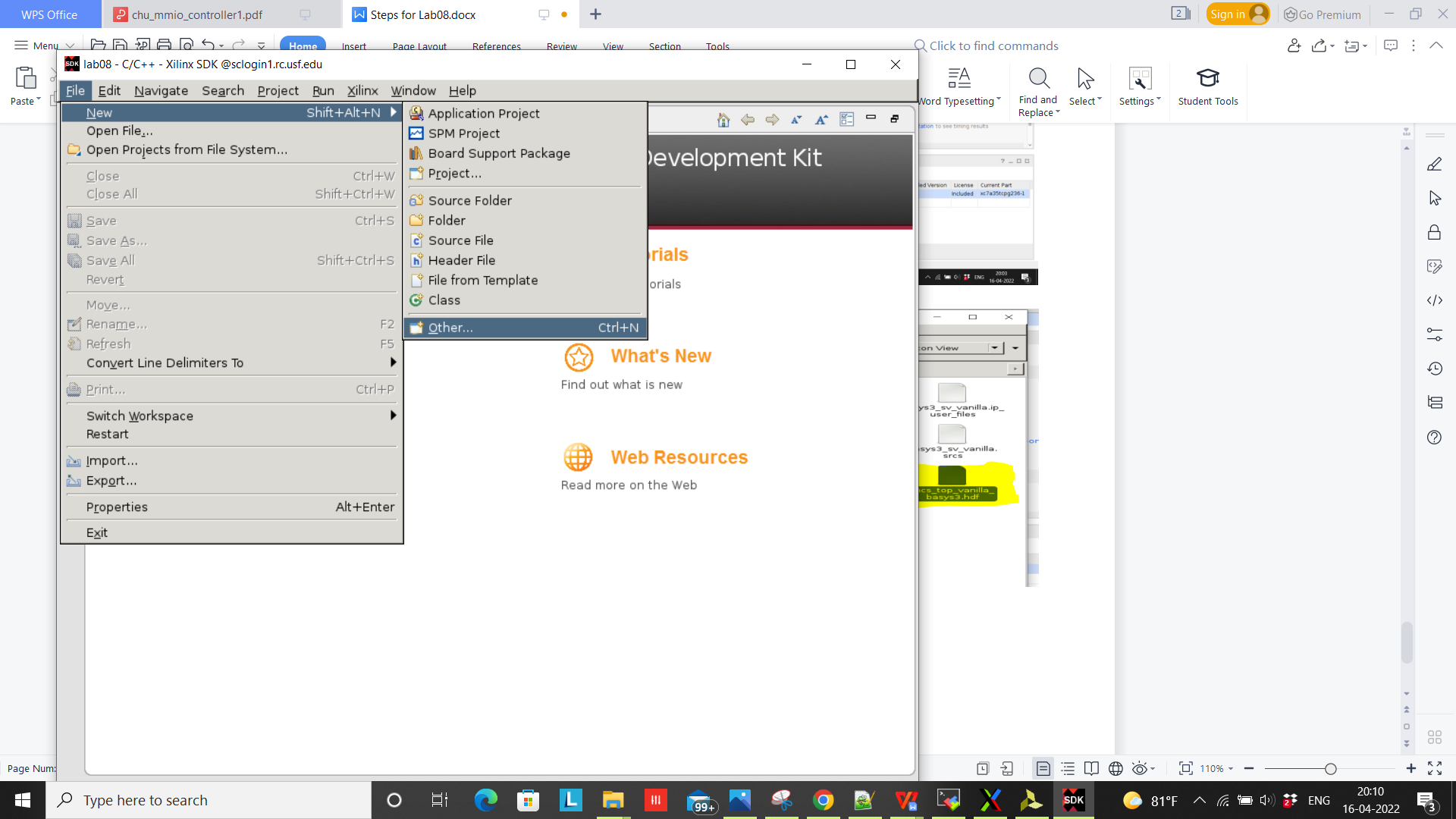
1. Go to File --> Export --> Export Hardware. And choose the specific location where the file needs to be saved. (ex : /home/s/shettyt/rsp22/lab08/basys3\_sv\_vanilla)

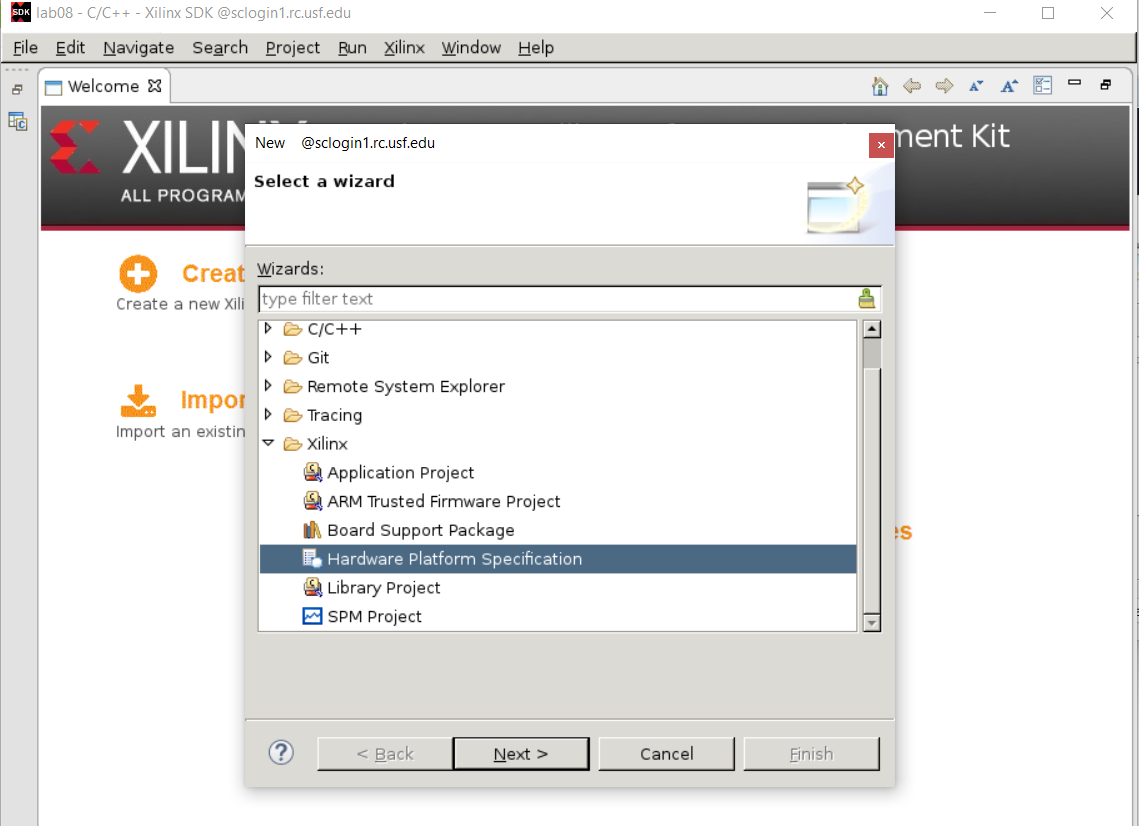




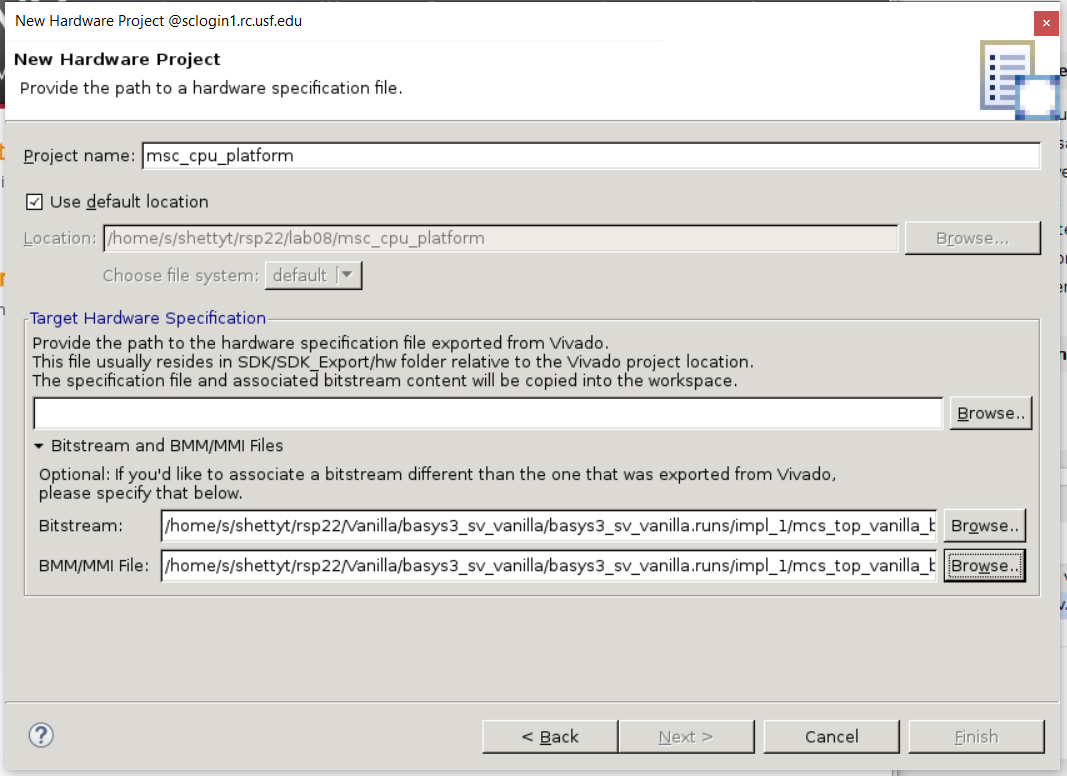
**To generate .elf file:**

1. Run the sdk from the command window ( ./sdk). Once he SDK is launched, follow the below steps.
2. Firstly, choose File--> New --> Others --> Xilinx --> Hardware Specification Program

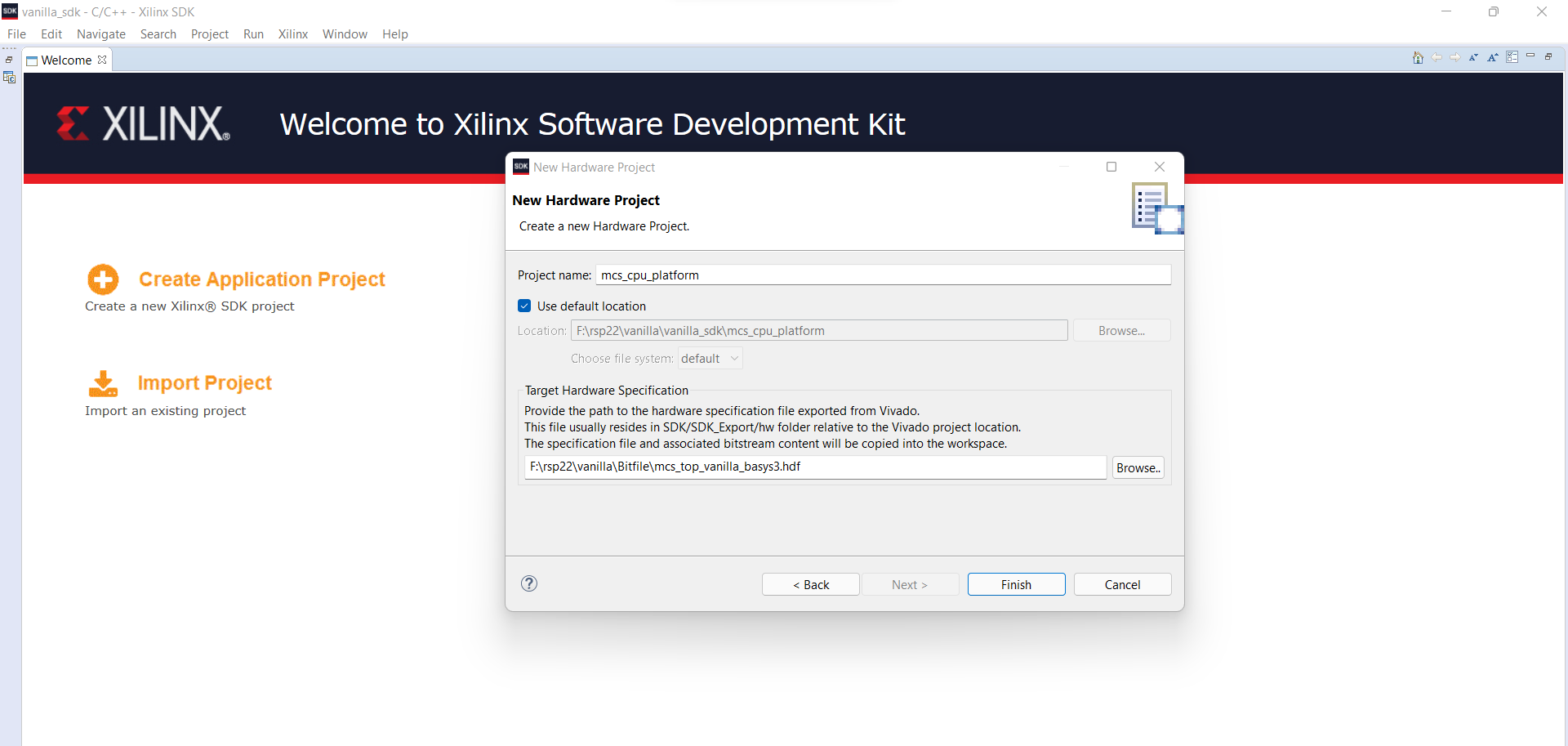




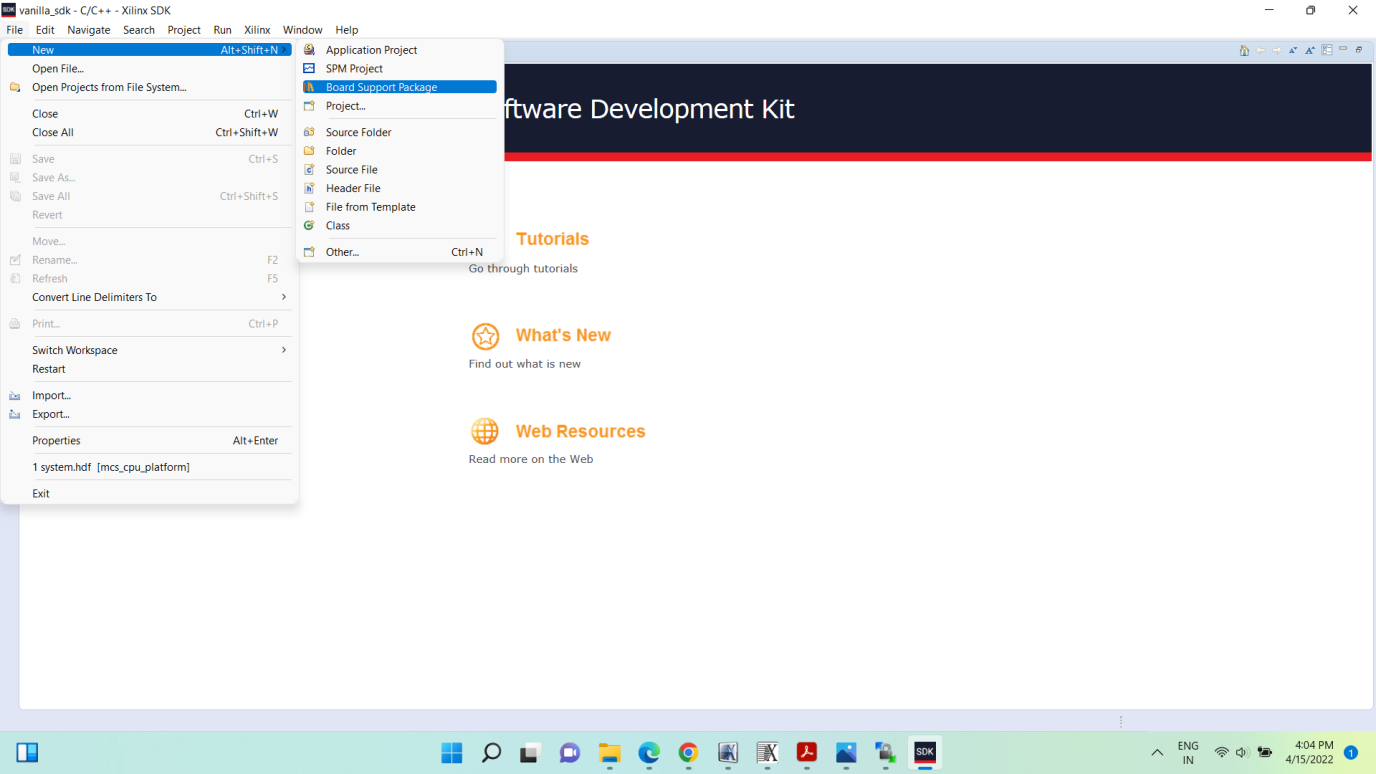
Next, give the project name you desire and choose the .bit and .mmi file geneated in step 9 as below.

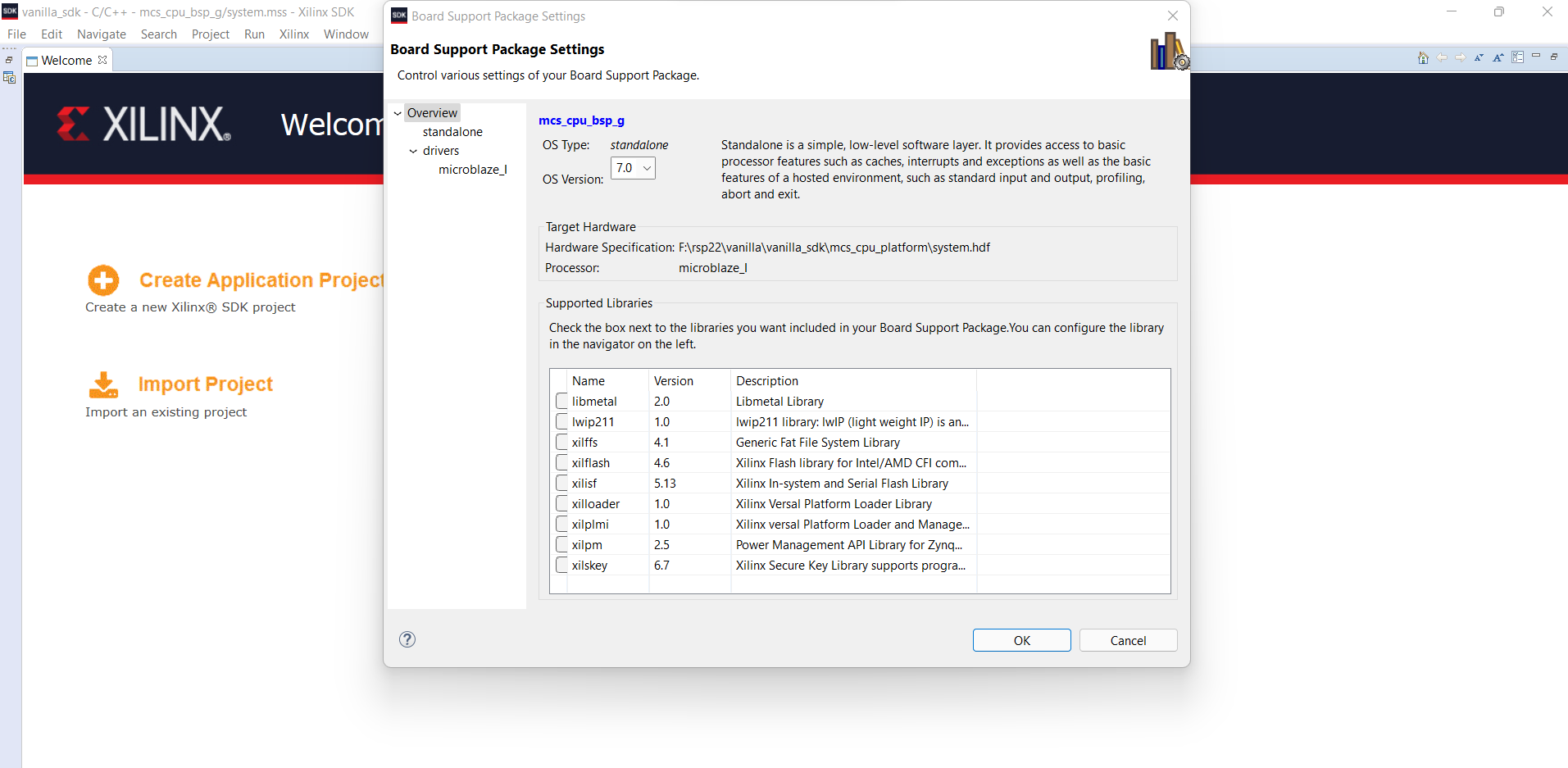


Once these files are chosen, next choose the .hdf file generated in step 10 as below and give “Finish”.

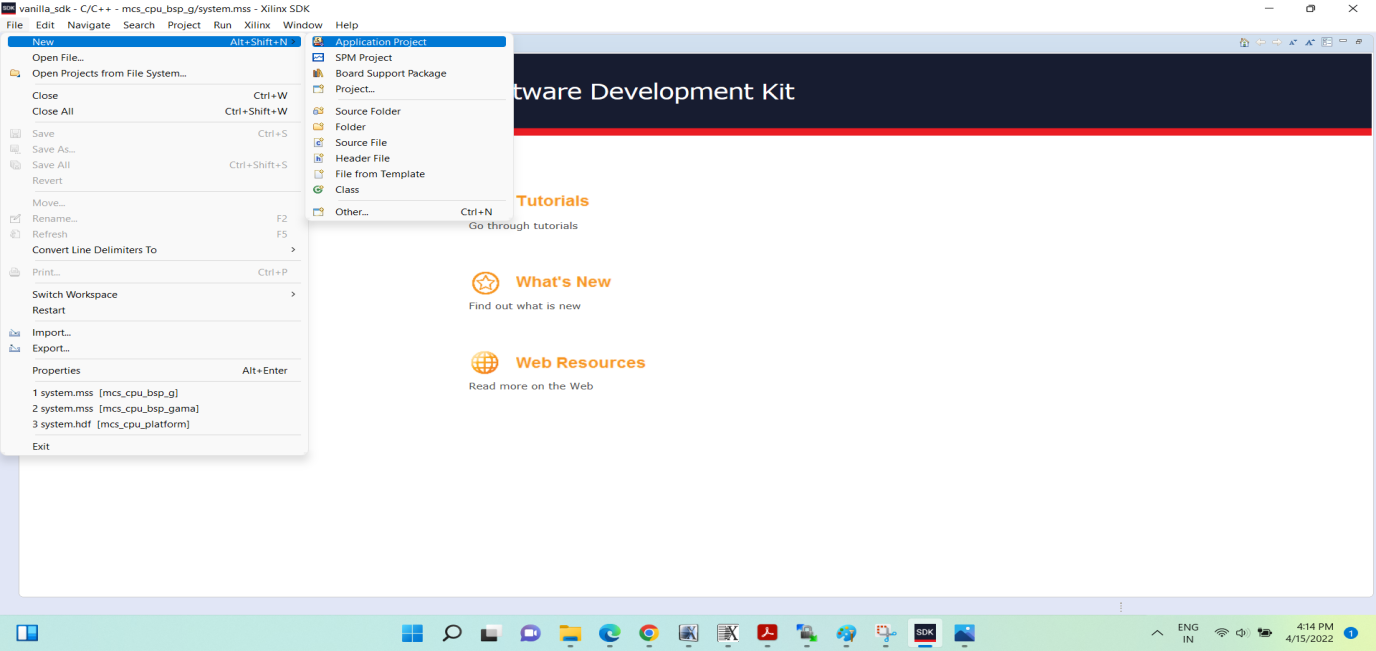


1. Secondly, Go to File--> New--> Board Support Package and Select OK without making any changes in the settings.





1. Thirdly, go to File-->New--> Application Project



Here,

Project name : User defined name

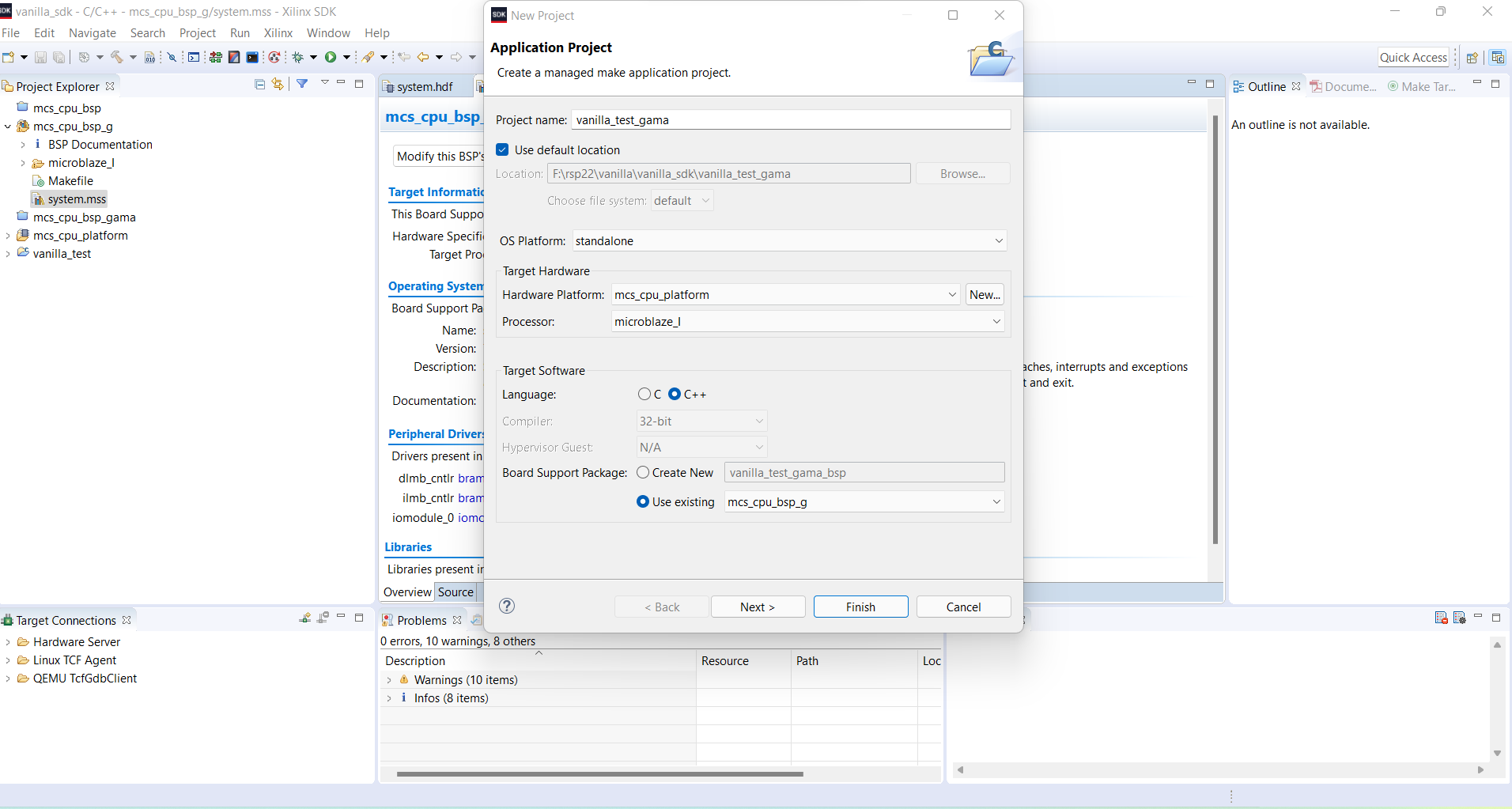
OS Platform: standalone

Hardware Platform: /\*Previosly given name\*/

Processor: microblaze\_1

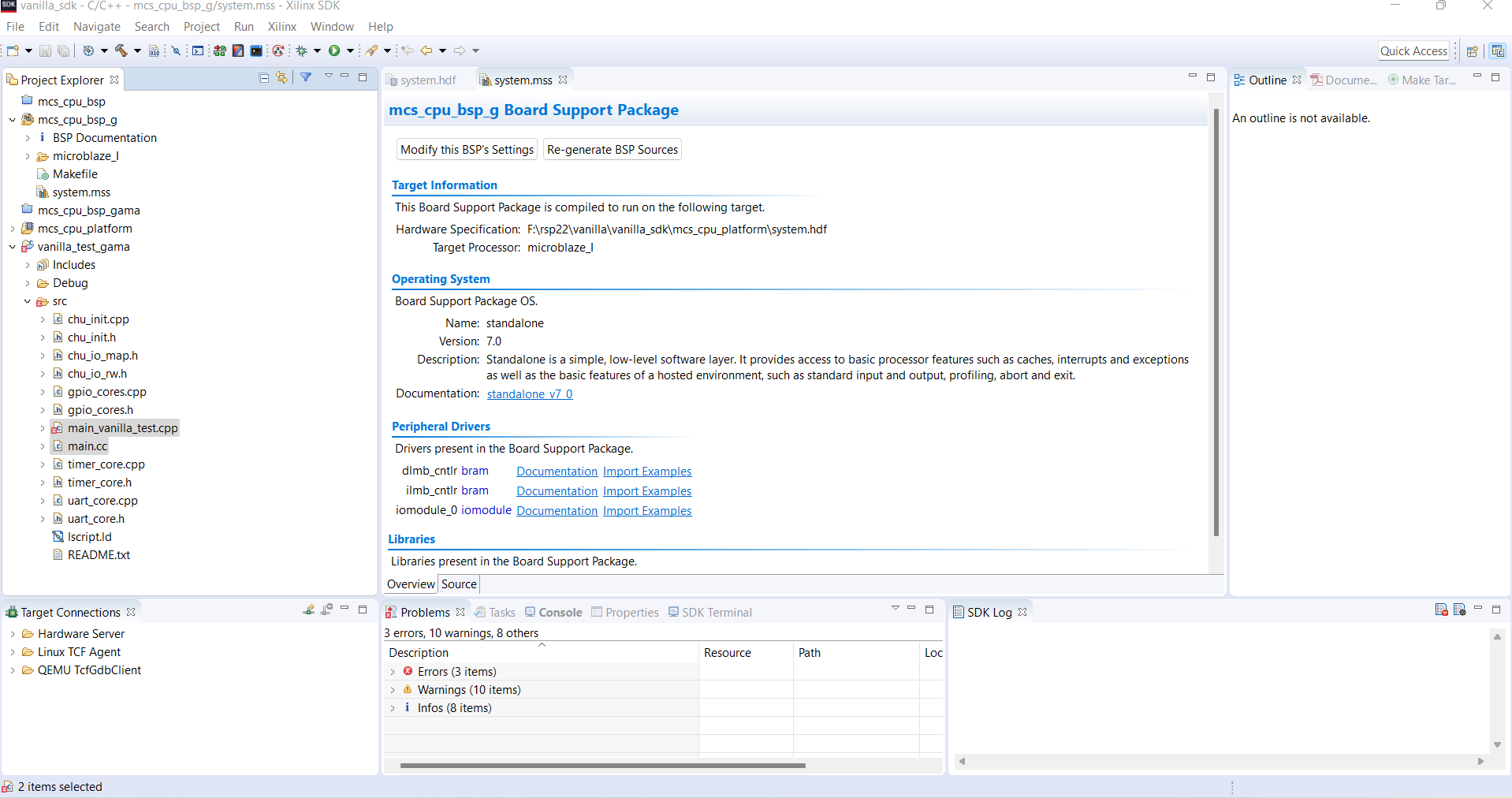
Language: C++

Board Support Package : (Use existing) - Choose the existing option available)



Once all the options are verified, choose “Finish”.

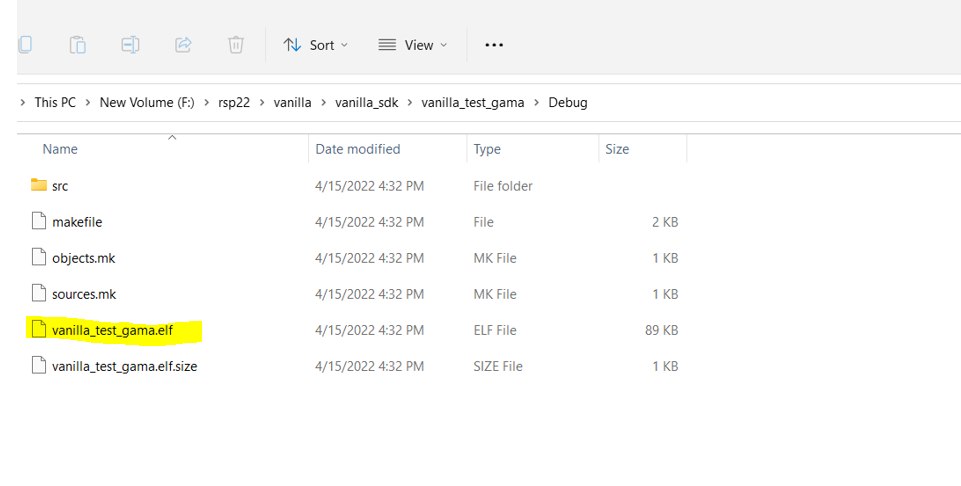
1. Once the SDK window is launched, start adding the C++ files to run the script. We can drag and the files in /\*project\*/src folder as below.



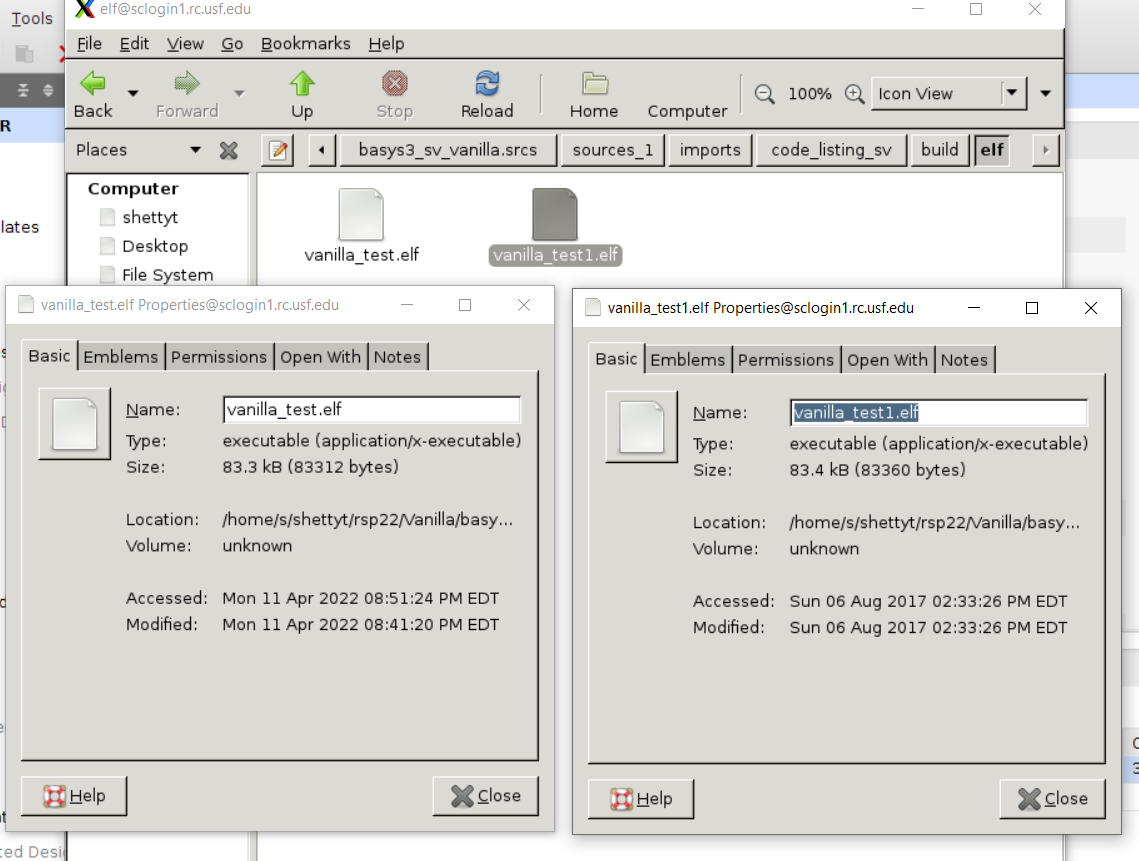
The software compiles he files as when the files are added and we don’t have to do it manually.

Note : delete the “main.cc” generated as it is an automatically generated script and is not a part of our main script.

Once there are no errors after compilation, the .elf file will be generated in the below path.

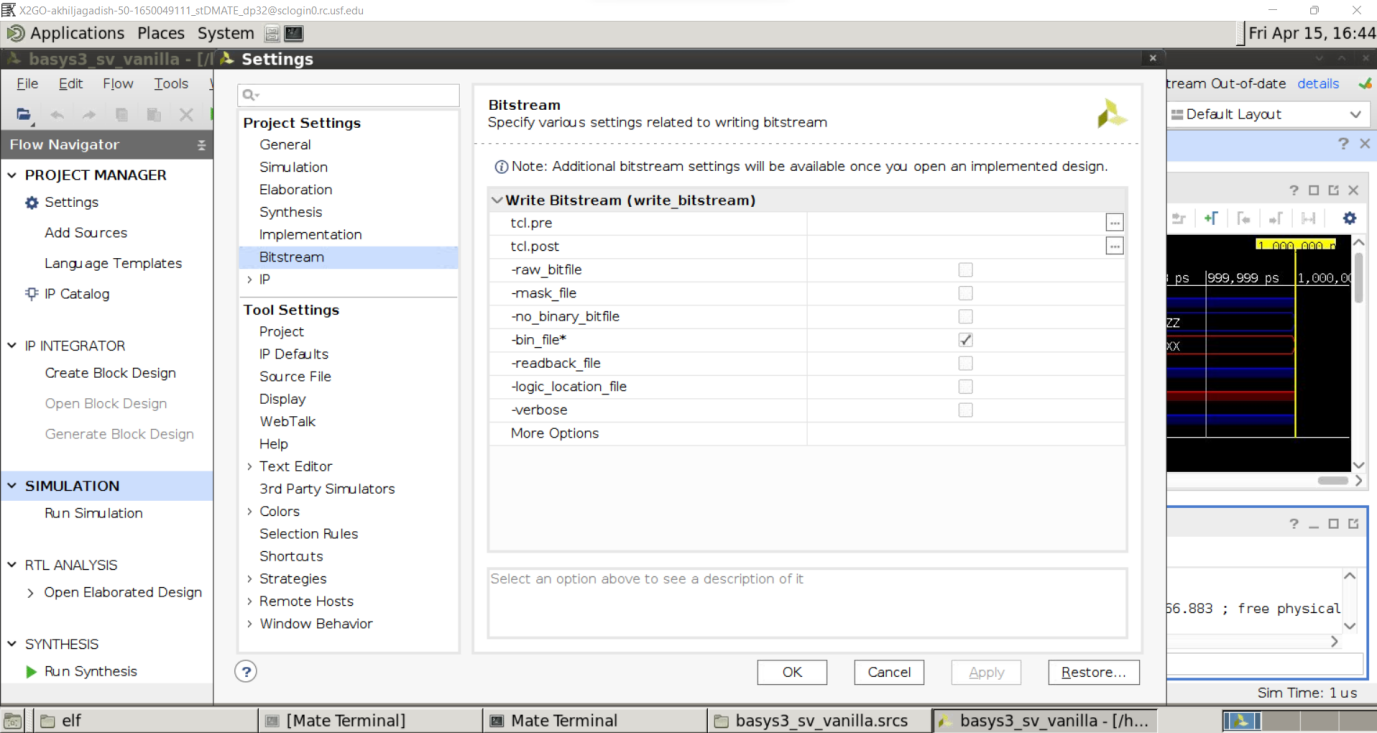


1. Rename the file as “Vanilla\_test.elf” and copy it in “basys3\_sv\_vanilla.srcs/sources\_1/imports/code\_listing\_sv/build/elf” folder. And rename the previously exiting one as “Vanilla\_test1.elf”.
2. When compared, we can see that there is a difference in both the files.

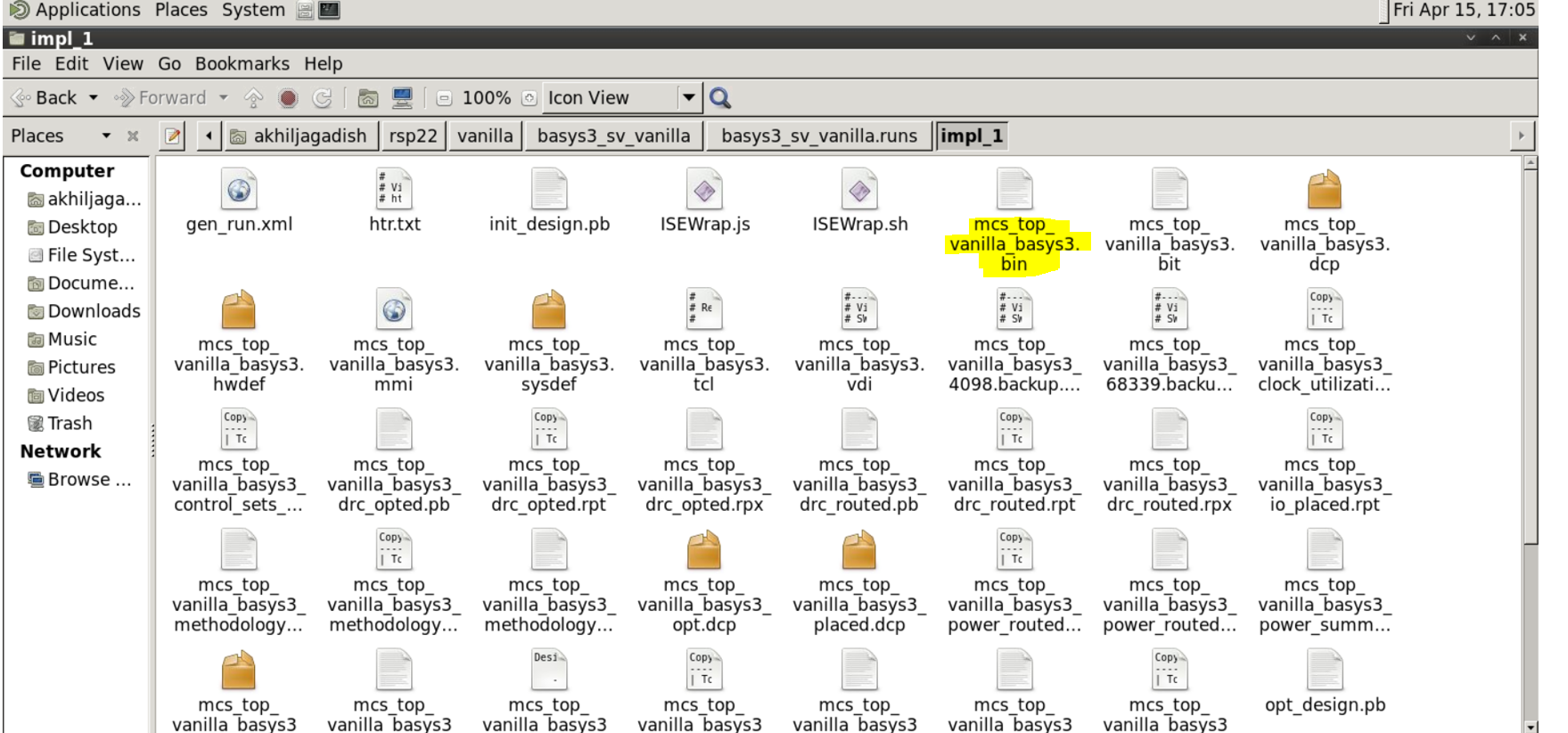


To generate .bin file for flashing:

1. In the Vivado window, go to Project Settings--> Select Bitstream--> choose bin\_file.



1. Now again run the Simulation, Synthesis and Implementation and generate bitstream. This time, .bin file will be generated along with the .bit file.



1. Now, open Vivado lab in the system and connect the board to be flashed.
2. Keep the jumper in ‘QSPI’ position(1st position) to program he board in flash memory. Once the board is detected, upload the bin file generated to the board.
3. Restart the board as it programmed in flash memory and check the output for Timer check, LED check, switch check and UART check.