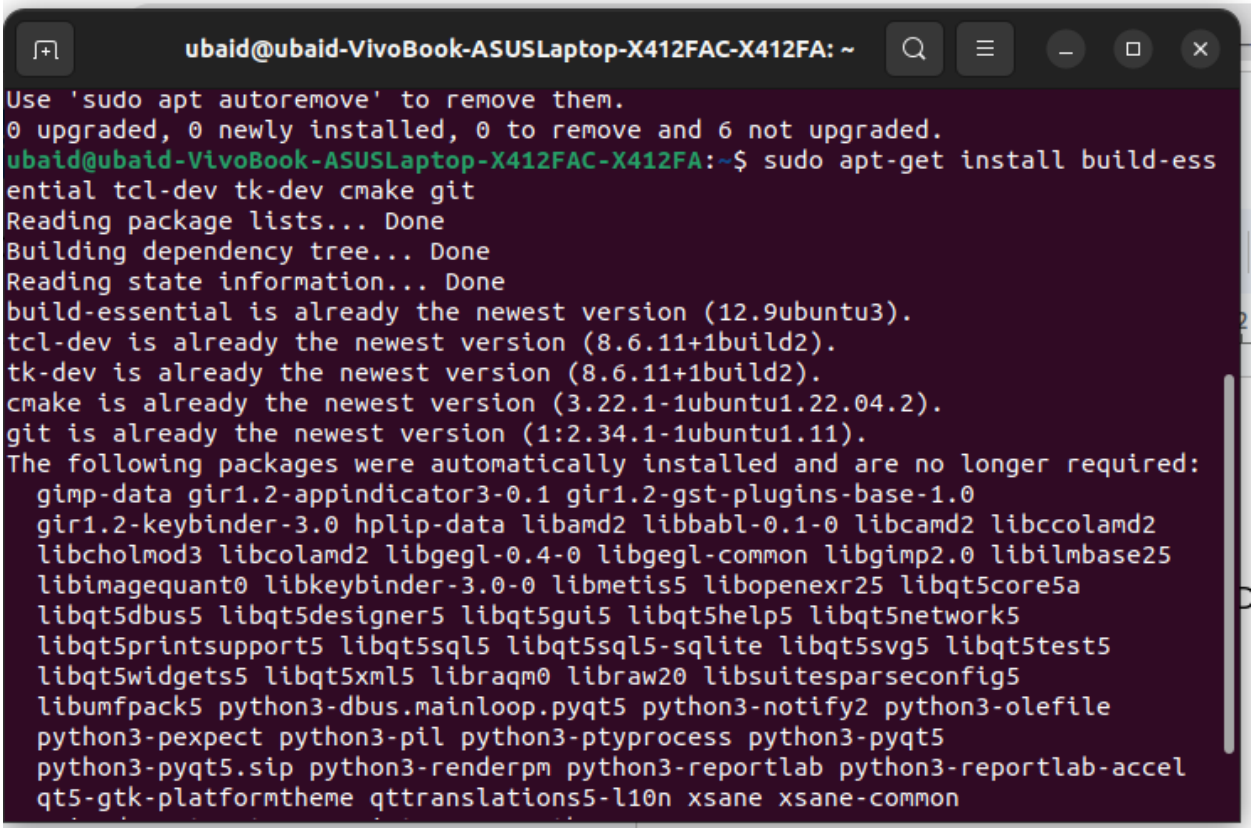
VLSI Design Flow: RTL To GDS (NPTEL Course)

Tutorial 7

To Install OpenSTA

Before installing OpenSTA, make sure you have the necessary development tools and libraries installed. You can install them using the following commands:

- On your Home Directory type the following command
  sudo apt-get update
  sudo apt-get install build-essential tcl-dev tk-dev cmake git

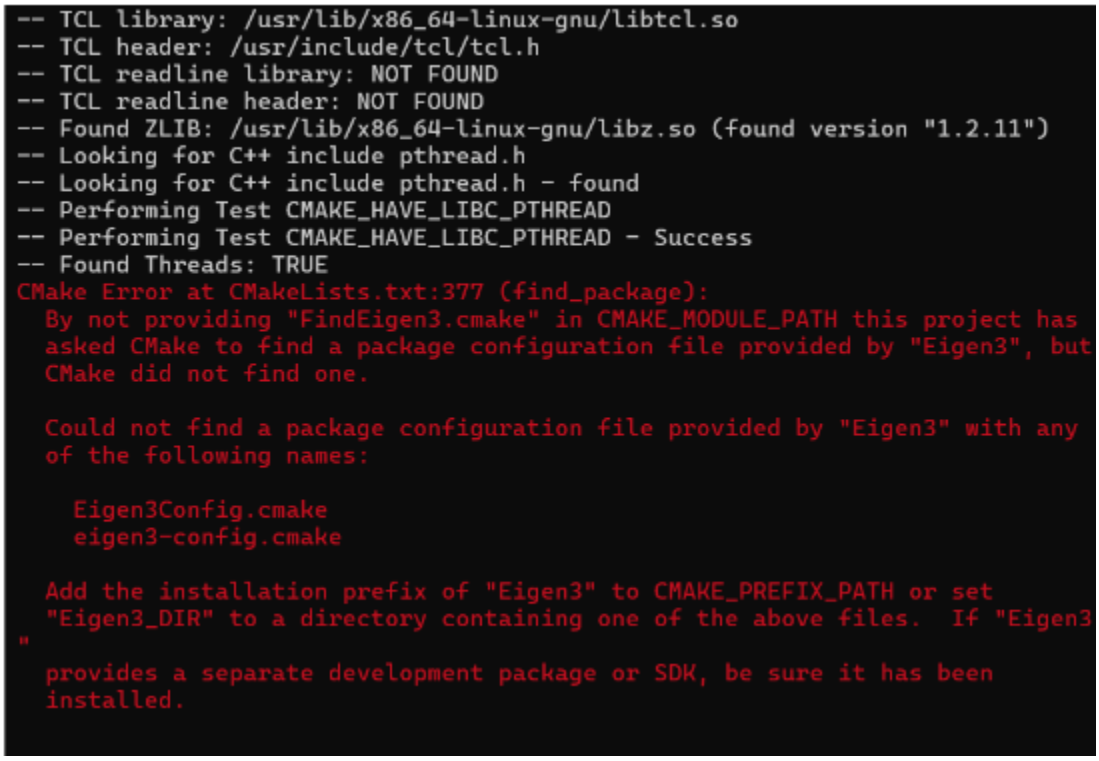


- Clone the OpenSTA repository by executing the following command:
  git clone https://github.com/The-OpenROAD-Project/OpenSTA.git

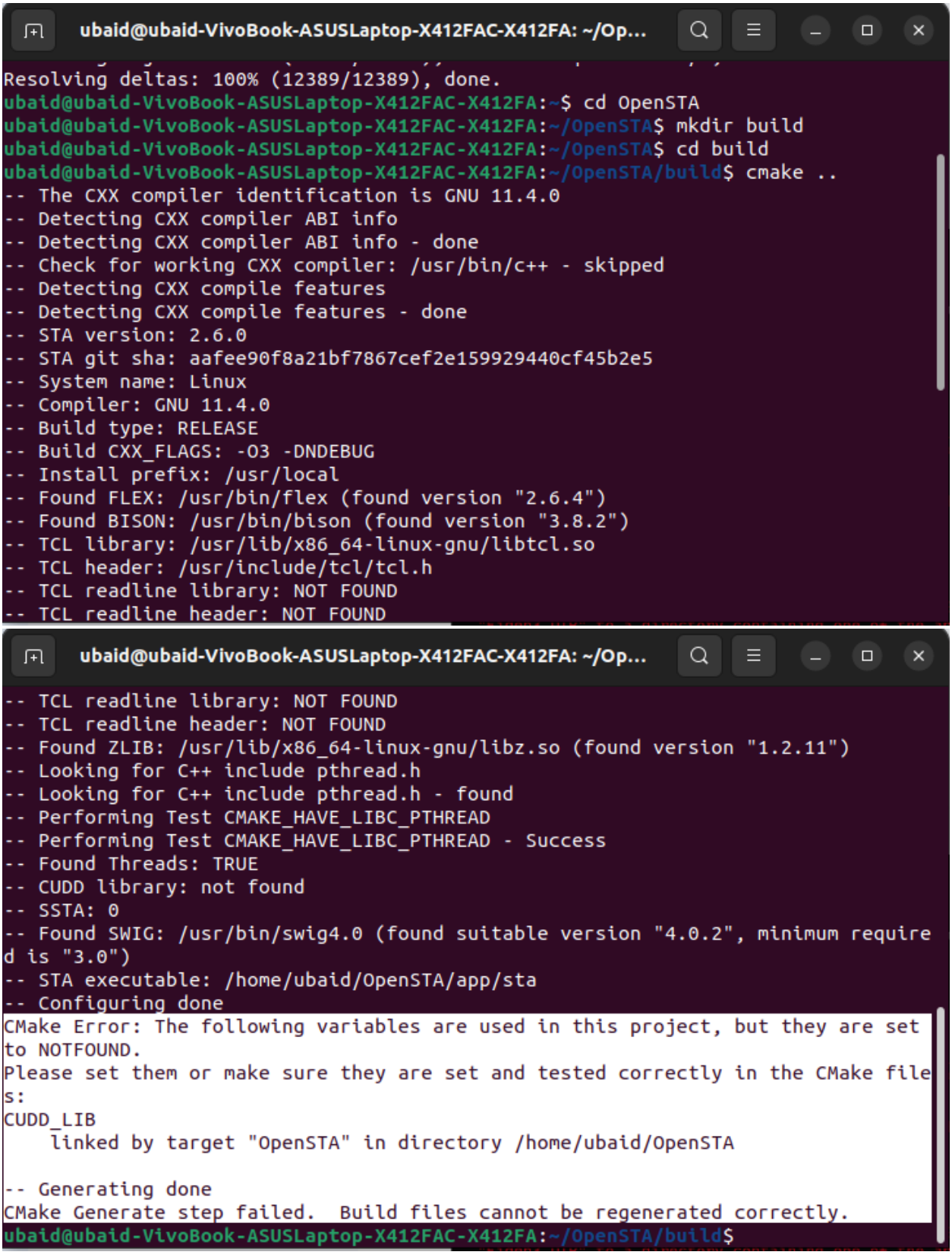
- Move into the OpenSTA directory that was created during the cloning process using the following command:
  cd OpenSTA
- Create a build directory using the following command:
  mkdir build
- Move into the build directory using the following command:
  cd build

- Configure the build by executing the following command:
  cmake ..

  This command configures the build process for OpenSTA, generating the necessary build files based on the project's configuration.



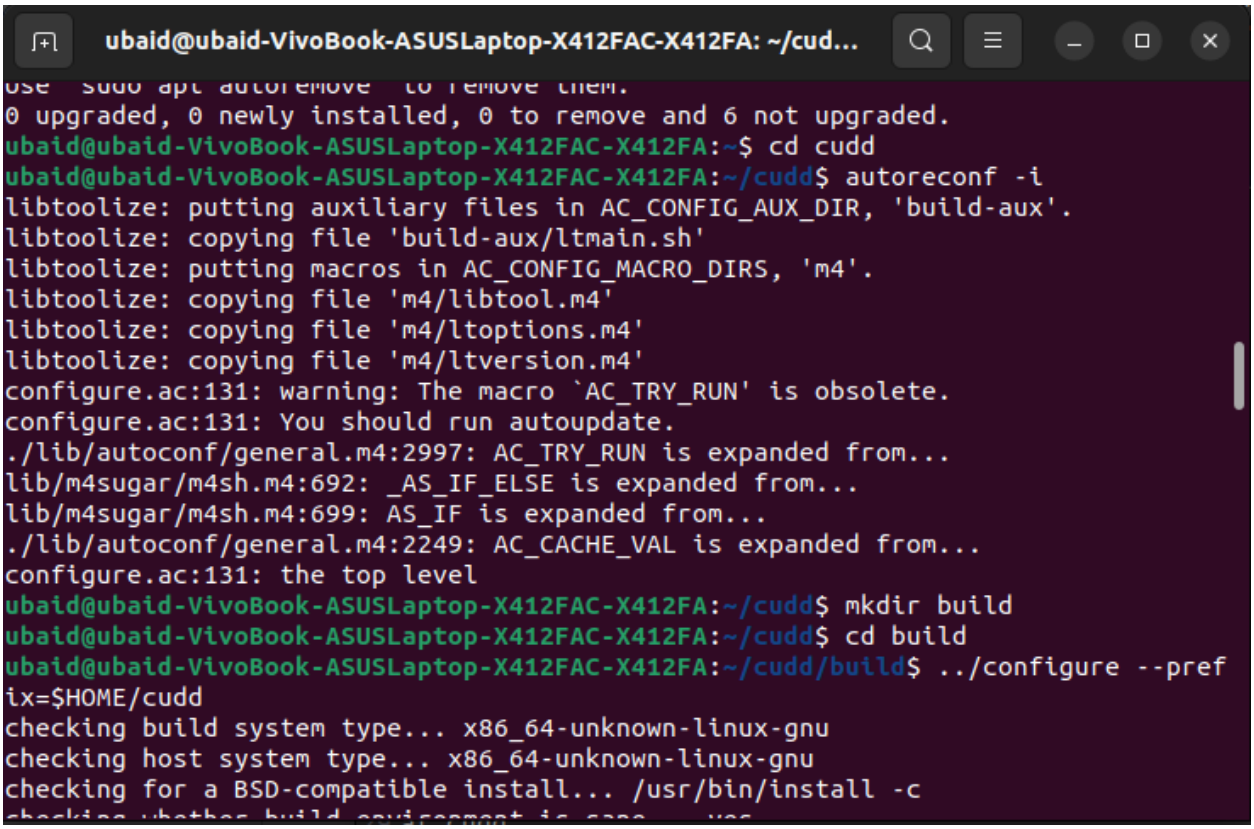
  If it shows an error like this then you have to install Eigen3 by using this command.
- Move to the home directory using following command:
  cd
- Install the eigen by using following command:
  sudo apt-get install libeigen3-dev
  again move to the build directory in OpenSTA and Configure the build by executing the following command:
  cmake ..



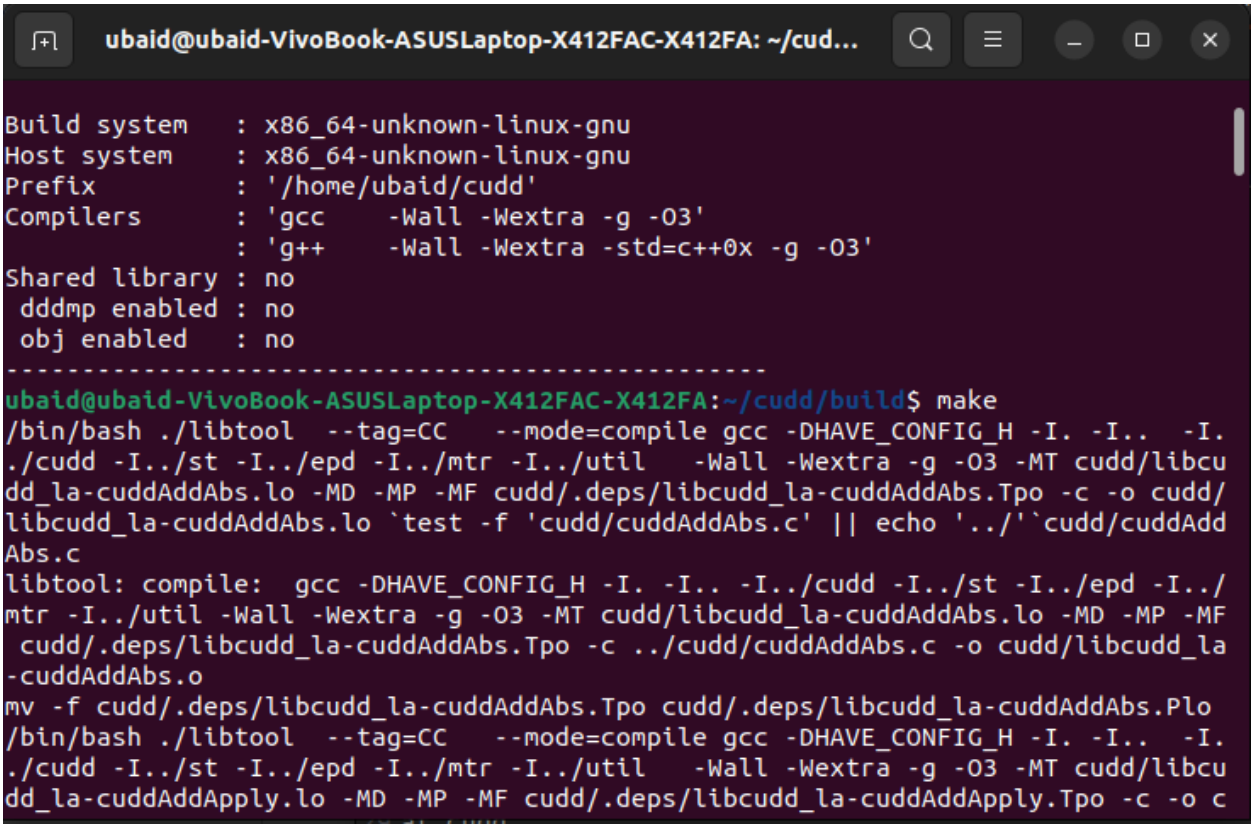
If the above error occurred then you have to install CUDD

- move to your home directory and Clone the CUDD repository by executing the following command:
  git clone https://github.com/ivmai/cudd.git

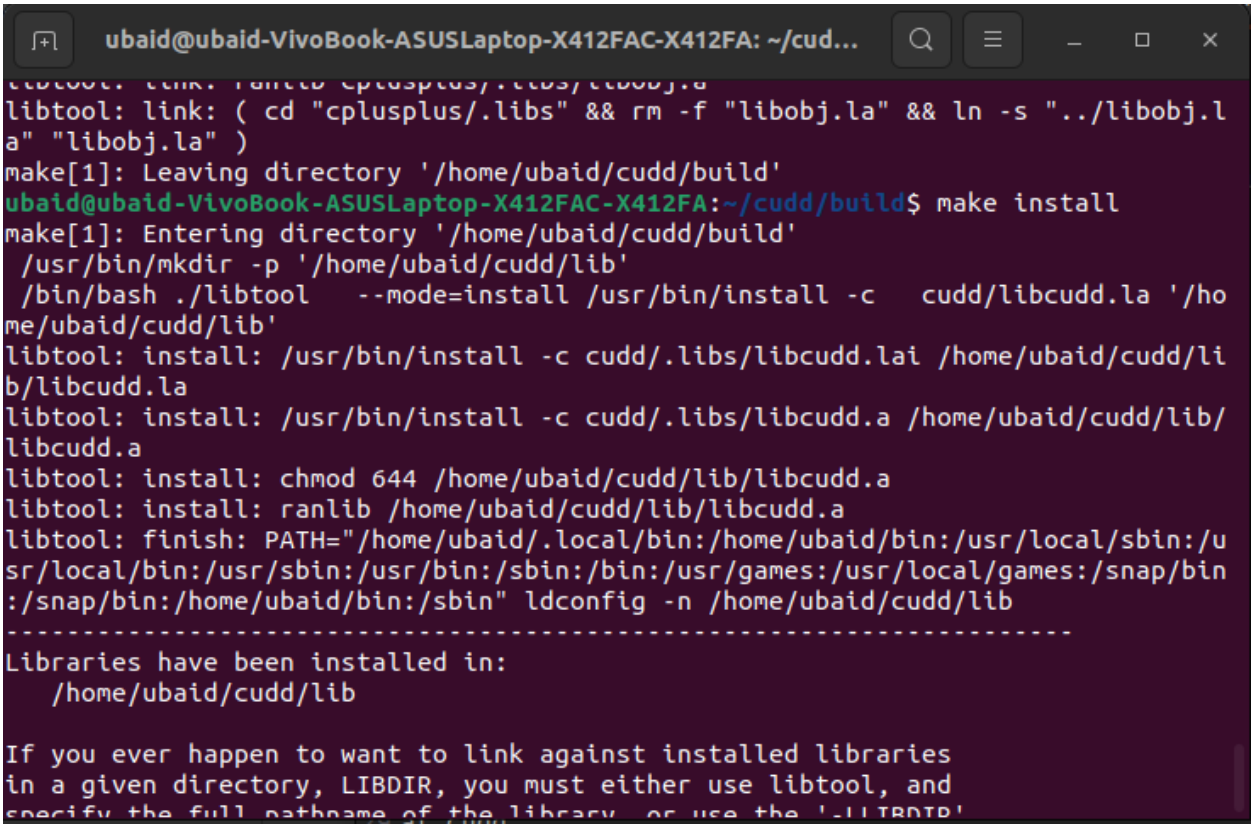
- And you have to install these 2 dependencies by following commands:
  sudo apt-get install automake
  sudo apt-get install autoconf m4 perl
- Then move into the directory cudd by typing command
  cd cudd
- After that run following commands to configure CUDD:
  autoreconf -i
- Create a build directory using the following command:
  mkdir build
- Move into the build directory using the following command:
  cd build
  ../configure --prefix=$HOME/cudd

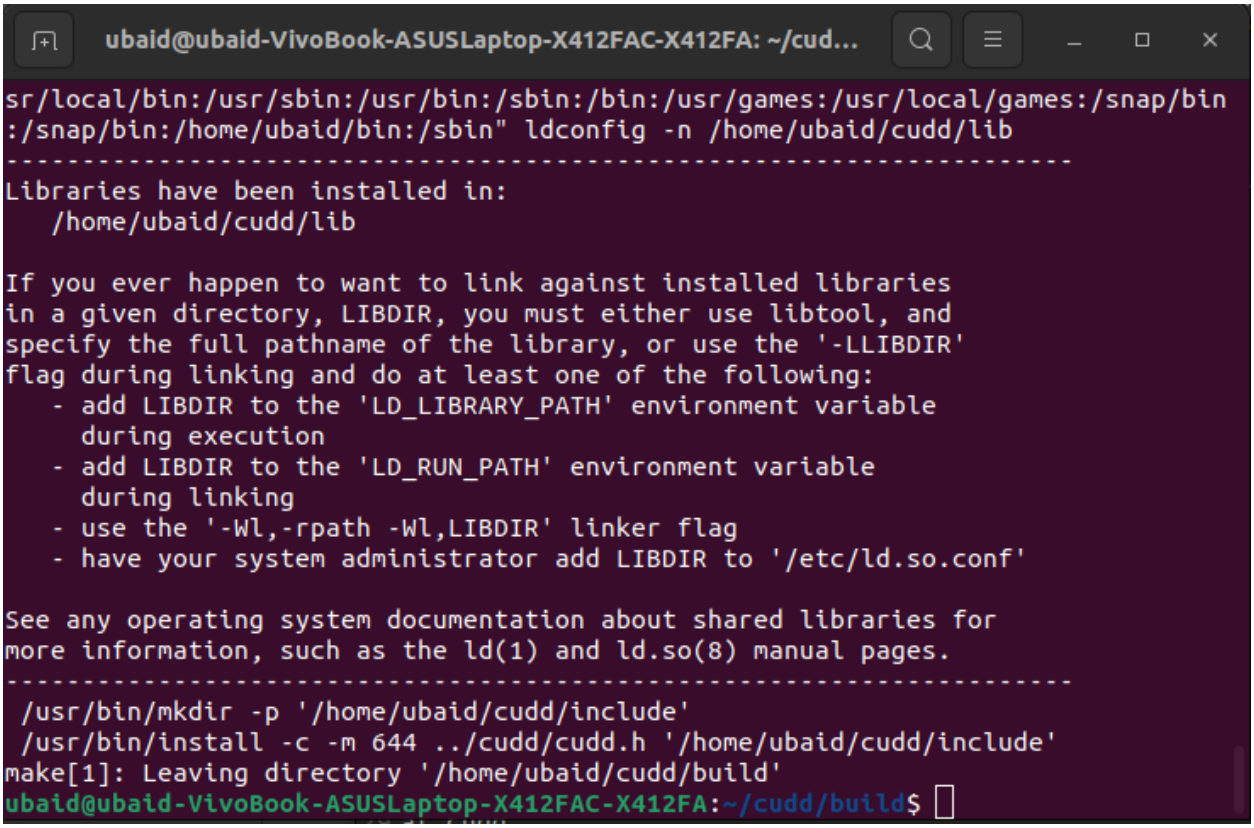


- Build CUDD by running the following command:
  make



- Install CUDD by executing the following command:
  make install

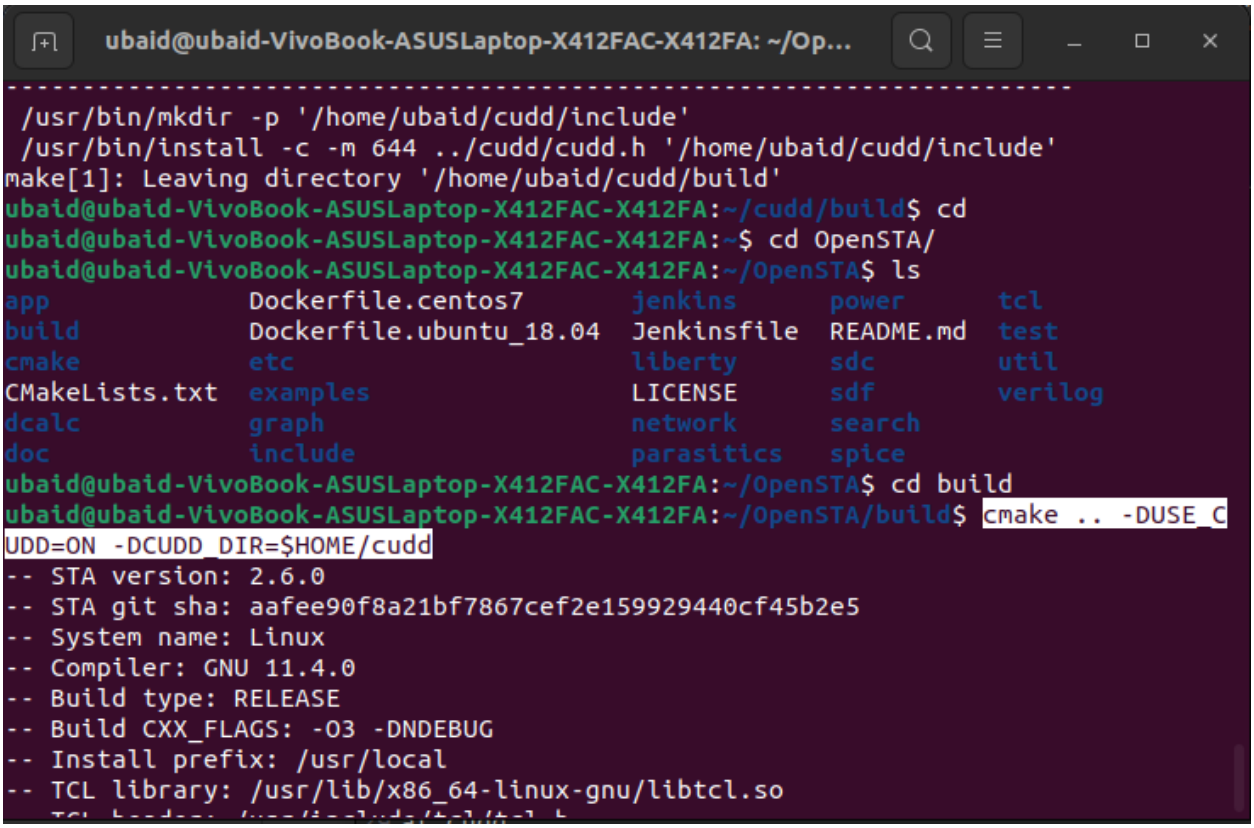
- 

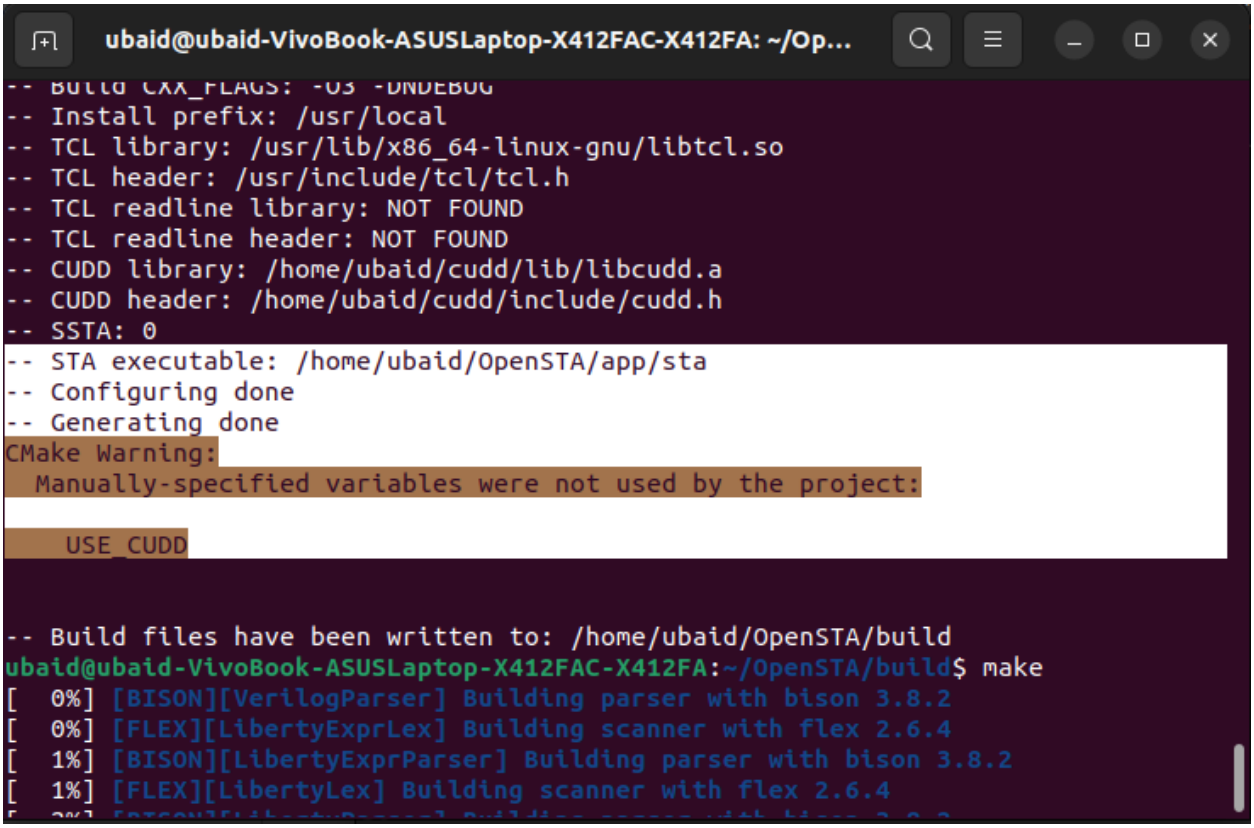
- 

  Now CUDD is installed Successfully.
- Move into the OpenSTA directory that was created during the cloning process using the

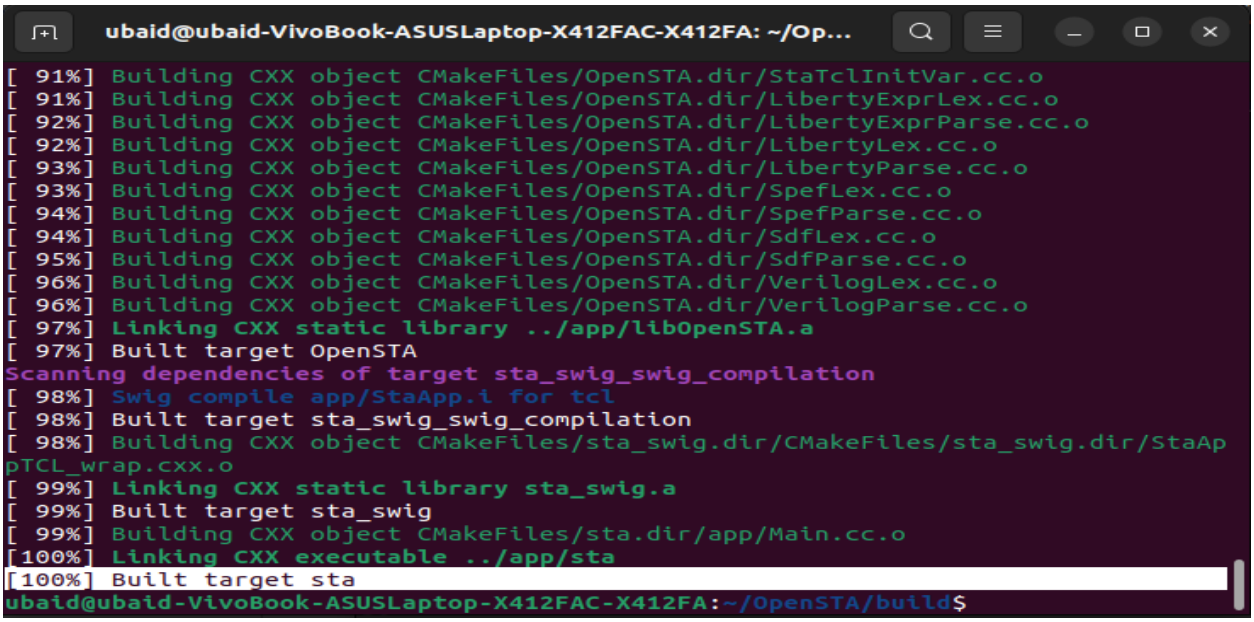
following command:
cd OpenSTA

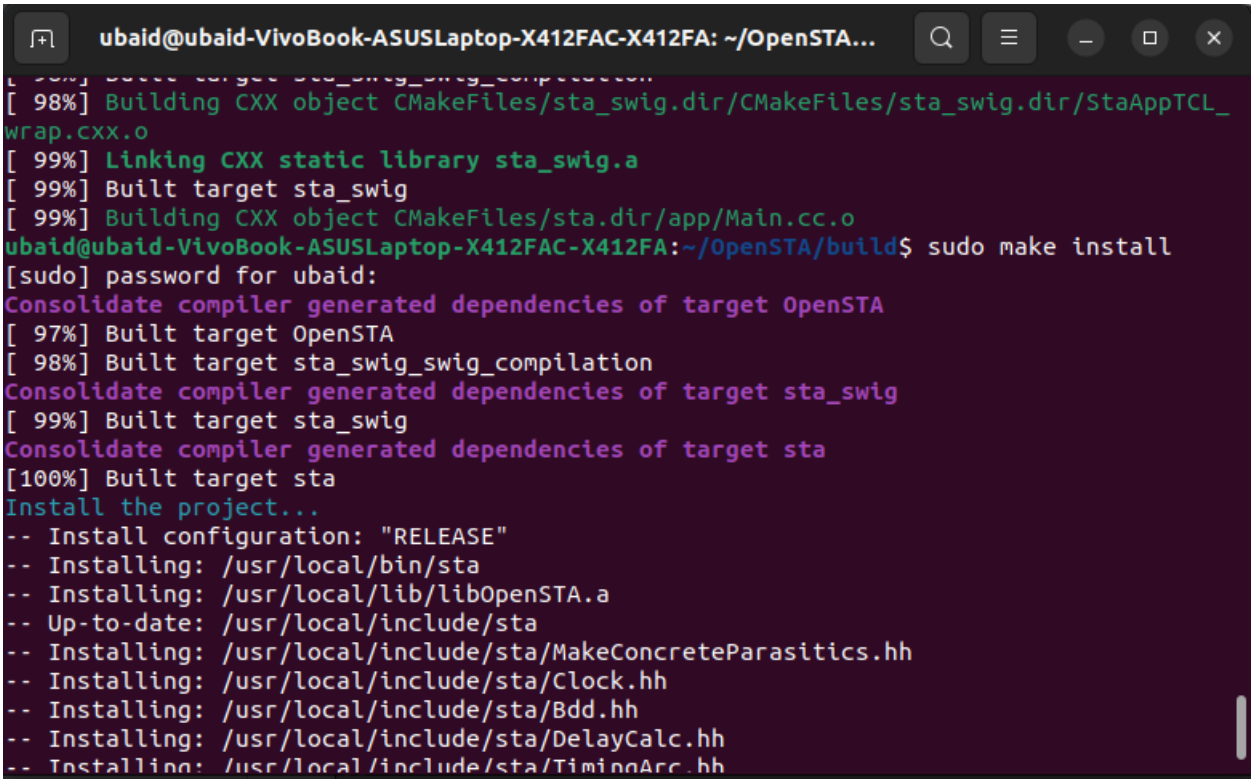
- Move into the build directory using the following command in the OpenSTA :
  cd build
- Configure the build by executing the following command:
  cmake .. -DUSE_CUDD=ON -DCUDD_DIR=$HOME/cudd
  This command configures the build process for OpenSTA, generating the necessary build files based on the project's configuration.

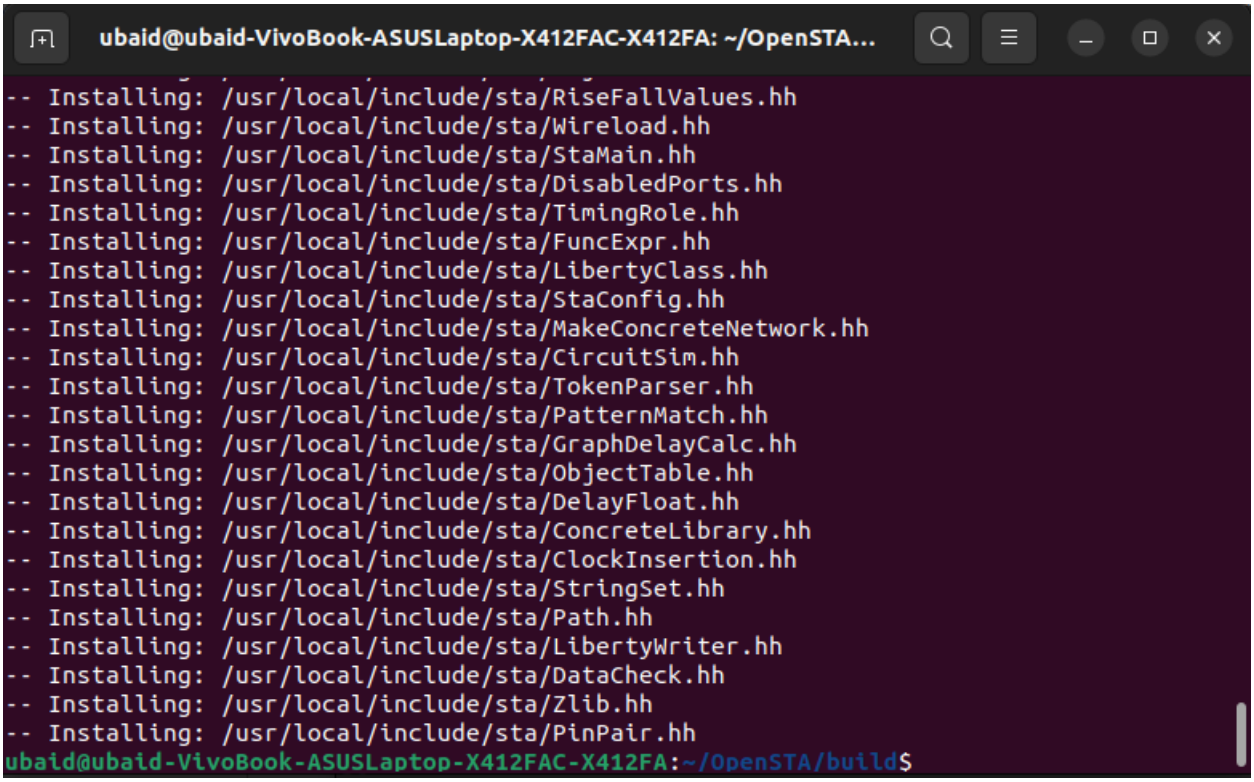




- Build OpenSTA by running the following command:
  make
  This command initiates the build process for OpenSTA. It compiles the source code and generates the executable files.
- Install OpenSTA by executing the following command:
  sudo make install

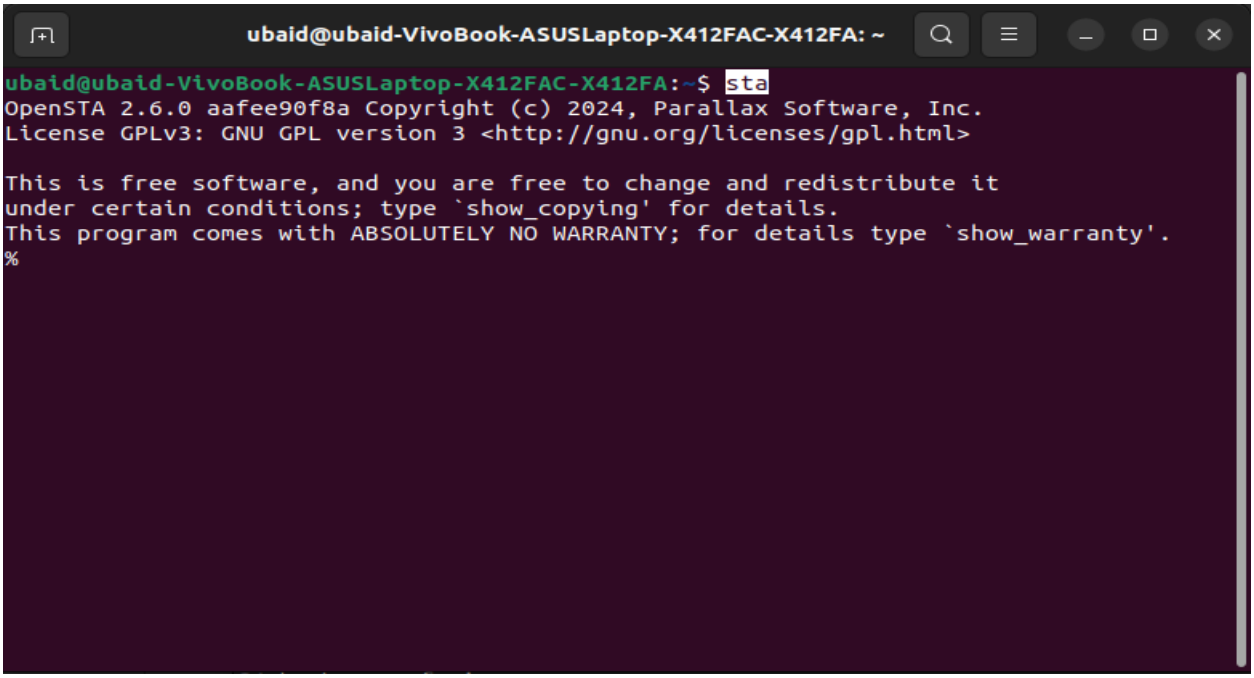






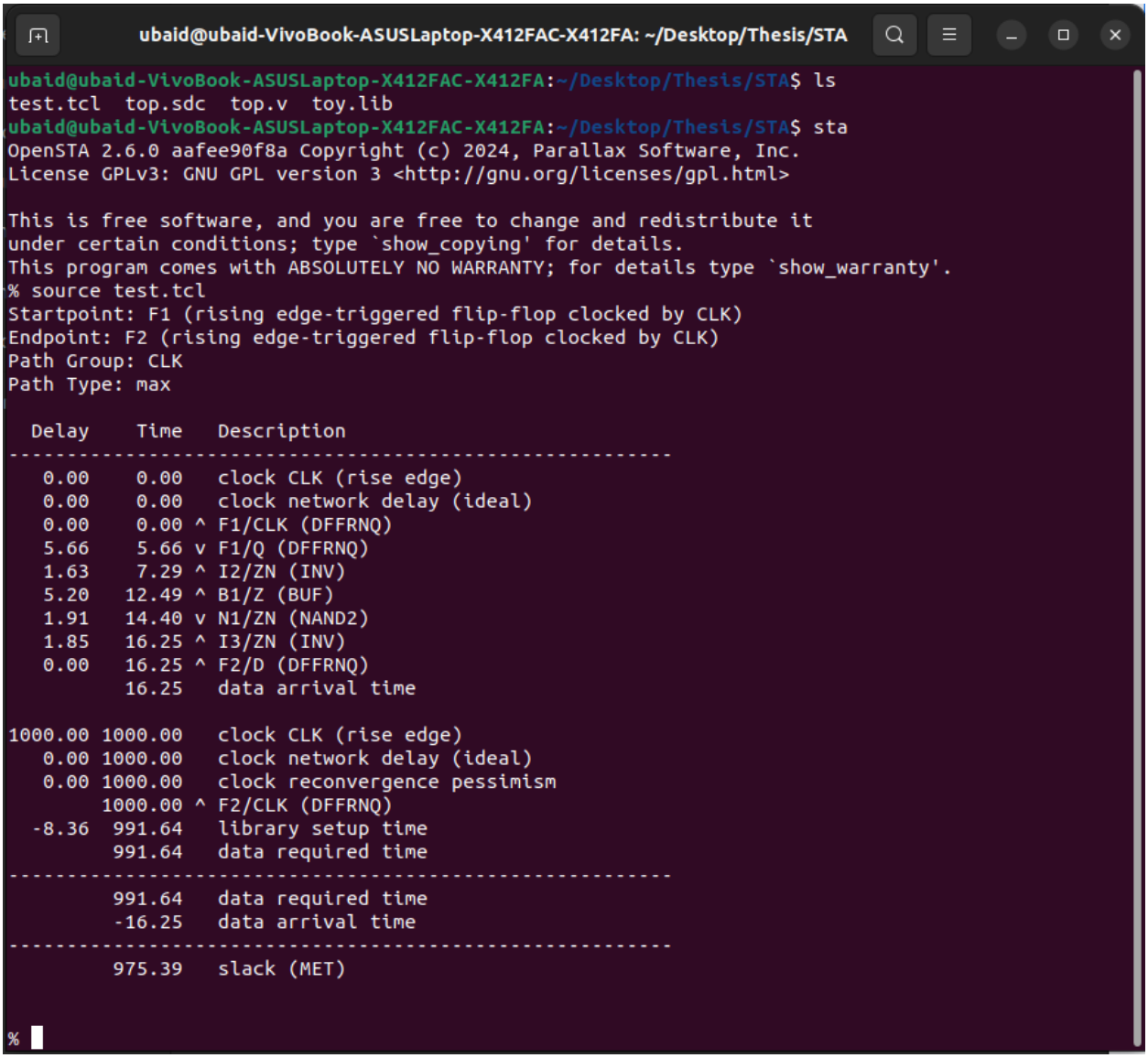
OpenSTA is installed successfully.

- To check that type sta on your terminal
  sta



- After installation process is successful.
  move the folder where you have your Netlist(top.v), constraint file(top.sdc), liberty file(toy.lib), test.tcl

  Type the following command to invoke OpenSTA
  sta
  you can check if OpenSTA is properly installed. If the installation was successful, the OpenSTA tool interface should launch without any errors or issues
  source test.tcl
  This command executes the commands specified in the "test.tcl" script file.


ubaid@ubaid-VivoBook-ASUSLaptop-X412FAC-X412FA: ~/Desktop/Thesis/STA
ubaid@ubaid-VivoBook-ASUSLaptop-X412FAC-X412FA:~/Desktop/Thesis/STA$ ls
test.tcl top.sdc top.v toy.lib
ubaid@ubaid-VivoBook-ASUSLaptop-X412FAC-X412FA:~/Desktop/Thesis/STA$ sta
OpenSTA 2.6.0 aafee90f8a Copyright (c) 2024, Parallax Software, Inc.
License GPLv3: GNU GPL version 3 <http://gnu.org/licenses/gpl.html>
This is free software, and you are free to change and redistribute it
under certain conditions; type `show_copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
% source test.tcl
Startpoint: F1 (rising edge-triggered flip-flop clocked by CLK)
Endpoint: F2 (rising edge-triggered flip-flop clocked by CLK)
Path Group: CLK
Path Type: max
Delay Time Description
0.00 0.00 clock CLK (rise edge)
0.00 0.00 clock network delay (ideal)
0.00 0.00 ^ F1/CLK (DFFRNQ)
5.66 5.66 v F1/Q (DFFRNQ)
1.63 7.29 ^ I2/ZN (INV)
5.20 12.49 ^ B1/Z (BUF)
1.91 14.40 v N1/ZN (NAND2)
1.85 16.25 ^ I3/ZN (INV)
0.00 16.25 ^ F2/D (DFFRNQ)
16.25 data arrival time
1000.00 1000.00 clock CLK (rise edge)
0.00 1000.00 clock network delay (ideal)
0.00 1000.00 clock reconvergence pessimism
1000.00 ^ F2/CLK (DFFRNQ)
-8.36 991.64 library setup time
991.64 data required time
991.64 data required time
-16.25 data arrival time
975.39 slack (MET)
%