VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh

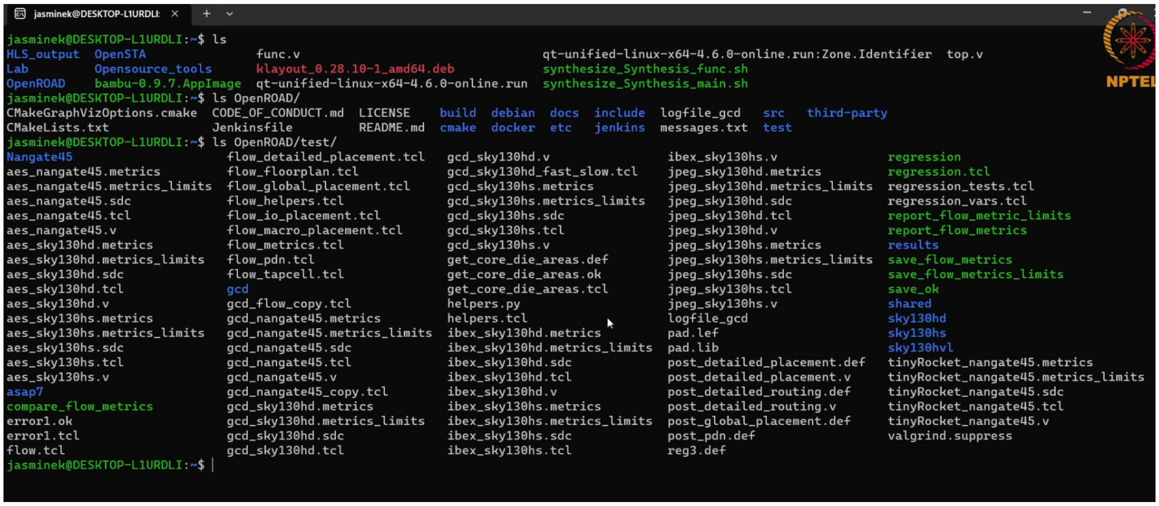
Department of Electronics and Communication Engineering
IIIT-Delhi

Lecture 46
Installation of OpenRoad

Hello everyone. My name is Jasmine Kaur and I am a PhD student at IIIT Delhi and I am your TA for the course VLSI Design Flow RTL to GDS. In the previous tutorials, we have explored different tools for the various stages of VLSI design flow from high level synthesis to logic synthesis. Now in this tutorial, we will move on to physical design for which we will be using an open road app which is an open source tool. And what is an open road? It is an integrated circuit physical design tool that takes the design from logic synthesized every log that we obtained from the logic synthesis step to the final routed layout. So what are the different steps that will be carried out using the open road? That is chip planning which includes floor planning, power planning, then placement, then we will do clock tree synthesis and then finally we will do routing.

So in this tutorial, we will see the steps to build an open road locally in your machine. So for this, we will open a WSL window and in that we will clone the repository from GitHub. So git clone minus minus recursive and this is the link from GitHub to download the repository. Enter.

So it is downloading the repository from GitHub. Now we will go into the directory open road, cd open road. In this, now we need to install the dependencies for that we will sudo dot slash etc dependency installer dot sh. So this file contains all the dependent files and applications and programs that are required for open road. So it is downloading these dependencies.



```
jasminek@DESKTOP-L1URDLI:~$ ls
HLS_output  OpenSTA               func.v                                 qt-unified-linux-x64-4.6.0-online.run:Zone.Identifier  top.v
Lab         Opensource_tools      klayout_0.28.10-1_amd64.deb            synthesize_Synthesis_func.sh
OpenROAD    bambu-0.9.7.AppImage  qt-unified-linux-x64-4.6.0-online.run  synthesize_Synthesis_main.sh
jasminek@DESKTOP-L1URDLI:~$ ls OpenROAD/
CMakeGraphVizOptions.cmake  CODE_OF_CONDUCT.md  LICENSE    build  debian  docs  include  logfile_gcd   src   third-party
CMakeLists.txt              Jenkinsfile         README.md  cmake  docker  etc   jenkins  messages.txt  test
jasminek@DESKTOP-L1URDLI:~$ ls OpenROAD/test/
Nangate45                     flow_detailed_placement.tcl   gcd_sky130hd.v                ibex_sky130hs.v               regression
aes_nangate45.metrics         flow_floorplan.tcl            gcd_sky130hd_fast_slow.tcl    jpeg_sky130hd.metrics         regression.tcl
aes_nangate45.metrics_limits  flow_global_placement.tcl     gcd_sky130hs.metrics          jpeg_sky130hd.metrics_limits  regression_tests.tcl
aes_nangate45.sdc             flow_helpers.tcl              gcd_sky130hs.metrics_limits   jpeg_sky130hd.sdc             regression_vars.tcl
aes_nangate45.tcl             flow_io_placement.tcl         gcd_sky130hs.sdc              jpeg_sky130hd.tcl             report_flow_metric_limits
aes_nangate45.v               flow_macro_placement.tcl      gcd_sky130hs.tcl              jpeg_sky130hd.v               report_flow_metrics
aes_sky130hd.metrics          flow_metrics.tcl              gcd_sky130hs.v                jpeg_sky130hs.metrics         results
aes_sky130hd.metrics_limits   flow_pdn.tcl                  get_core_die_areas.def        jpeg_sky130hs.metrics_limits  save_flow_metrics
aes_sky130hd.sdc              flow_tapcell.tcl              get_core_die_areas.ok         jpeg_sky130hs.sdc             save_flow_metrics_limits
aes_sky130hd.tcl              gcd                           get_core_die_areas.tcl        jpeg_sky130hs.tcl             save_ok
aes_sky130hd.v                gcd_flow_copy.tcl             helpers.py                    jpeg_sky130hs.v               shared
aes_sky130hs.metrics          gcd_nangate45.metrics         helpers.tcl                   logfile_gcd                   sky130hd
aes_sky130hs.metrics_limits   gcd_nangate45.metrics_limits  ibex_sky130hd.metrics         pad.lef                       sky130hs
aes_sky130hs.sdc              gcd_nangate45.sdc             ibex_sky130hd.metrics_limits  pad.lib                       sky130hvl
aes_sky130hs.tcl              gcd_nangate45.tcl             ibex_sky130hd.sdc             post_detailed_placement.def   tinyRocket_nangate45.metrics
aes_sky130hs.v                gcd_nangate45.v               ibex_sky130hd.tcl             post_detailed_placement.v     tinyRocket_nangate45.metrics_limits
asap7                         gcd_nangate45_copy.tcl        ibex_sky130hd.v               post_detailed_routing.def     tinyRocket_nangate45.sdc
compare_flow_metrics          gcd_sky130hd.metrics          ibex_sky130hs.metrics         post_detailed_routing.v       tinyRocket_nangate45.tcl
error1.ok                     gcd_sky130hd.metrics_limits   ibex_sky130hs.metrics_limits  post_global_placement.def     tinyRocket_nangate45.v
error1.tcl                    gcd_sky130hd.sdc              ibex_sky130hs.sdc             post_pdn.def                  valgrind.suppress
flow.tcl                      gcd_sky130hd.tcl              ibex_sky130hs.tcl             reg3.def
jasminek@DESKTOP-L1URDLI:~$ |
```

So it will take quite some time. So after these dependencies are downloaded, now we will make a directory build and now moving on into this directory, now c make. So this will take a few seconds. Then the next command is make command. So after you run this command, it will take around two hours for installation.

So after the installation is complete, the final command is sudo make install. It will ask for a password and it will build all the files. Now the open road is installed and we can see that this open road folder is created after this and let us see what are the contents of this open road folder. So in this we can see these are the different files and folders. Then let us look at the test folder inside this open road.

So here we can see there are Nangate 45, ASAP7 and Skywater. So these are the different libraries that are there that come as you install open road. So these are already there. And then we have these scripts, TCL files, Verilog files and constraint files. So we have example scripts here.

So in our tutorials we will be looking at GCD examples using Nangate 45 library. And now we can see how we can run this open road tool. So we type open road, enter and we can see that we can run this tool. So we will look at the various examples and various commands and various scripts in the next tutorial. Thank you.