**OpenFPGA**

Consider [fpga\_verilog](https://github.com/lnis-uofu/OpenFPGA/tree/master/openfpga_flow/tasks/fpga_verilog) tasks from git repo (https://github.com/lnis-uofu/OpenFPGA/)--

/home/ubuntu/Desktop/OpenFPGA/openfpga\_flow/tasks/fpga\_verilog/verilog\_netlist\_formats/synthesizable\_verilog

My repo for this thesis - <https://github.com/baruaeee/FPGA-OpenFPGA>

Copy ‘synthesizable\_verilog’ to ‘FPGA-OpenFPGA/Fabric’

All command need to be run from the OpenFPGA root folder

Run task command—

run-task /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog --debug --show\_thread\_logs

the task should be successful. But as I have to change the openfpga architecture file a lot, lets copy the architecture file from “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/arch” to ‘/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/’’

cp -r /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/arch /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/

make a copy of “k6\_frac\_N10\_stdcell\_mux\_40nm\_openfpga\_synthesizable.xml” as “k6\_frac\_N10\_stdcell\_mux\_sky130\_openfpga\_synthesizable.xml”

as the architecture location have changed update it in the task file

line 20/21: openfpga\_arch\_file=${PATH:OPENFPGA\_PATH}/openfpga\_flow/openfpga\_arch/k6\_frac\_N10\_stdcell\_mux\_40nm\_openfpga\_synthesizable.xml

replace with

openfpga\_arch\_file= openfpga\_arch\_file=${PATH:TASK\_DIR}/arch/k6\_frac\_N10\_stdcell\_mux\_sky130\_openfpga\_synthesizable.xml

line 28/29:

arch0=${PATH:OPENFPGA\_PATH}/openfpga\_flow/vpr\_arch/k6\_frac\_N10\_tileable\_40nm.xml

replace with

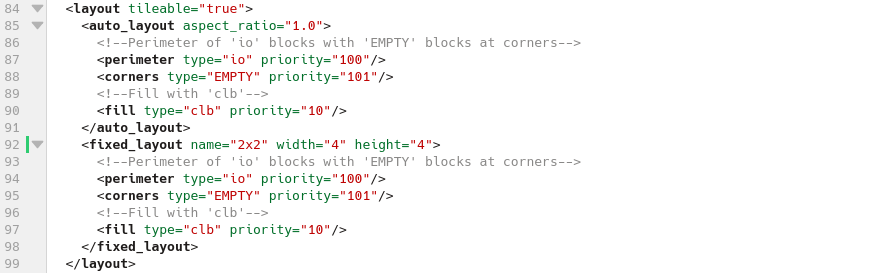
arch0=${PATH:TASK\_DIR}/arch/k6\_frac\_N10\_tileable\_40nm.xml

Run task command again to chack the file location update is corrects—

run-task /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog --debug --show\_thread\_logs

to get larger grid size the vpr architecture file (k6\_frac\_N10\_tileable\_40nm.xml) need to be edited. Lets open the architecture file—

the layout definition is like the following—



There are 2 layout defined, 1 is auto and another is fixed (2x2). Lets define another fixed layout—

<fixed\_layout name="8x8" width="10" height="10">

<!--Perimeter of 'io' blocks with 'EMPTY' blocks at corners-->

<perimeter type="io" priority="100"/>

<corners type="EMPTY" priority="101"/>

<!--Fill with 'clb'-->

<fill type="clb" priority="10"/>

</fixed\_layout>

Keep width and length moderate, otherwise the compilation will take very long time.

Make a copy of “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/arch/k6\_frac\_N10\_stdcell\_mux\_40nm\_openfpga\_synthesizable.xml”

cp /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/arch/k6\_frac\_N10\_stdcell\_mux\_40nm\_openfpga\_synthesizable.xml /home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/arch/k6\_frac\_N10\_stdcell\_mux\_sky130\_openfpga\_synthesizable.xml

lets open and edit the new copied file—

all standarded cell should be replace with the desired technology library (in this case Sky130)

the library can be found from https://github.com/google/skywater-pdk.git

from OpenFPGA root folder run—

git clone https://github.com/google/skywater-pdk.git

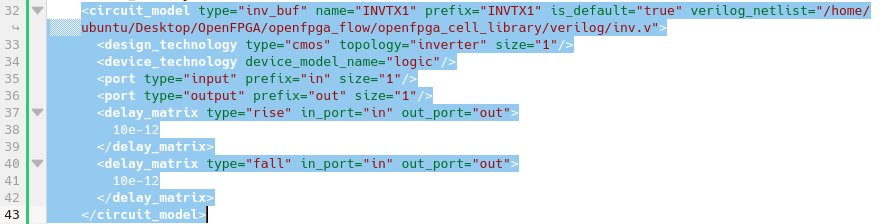
cd skywater-pdk

SUBMODULE\_VERSION=latest make submodules -j3 || make submodules -j1

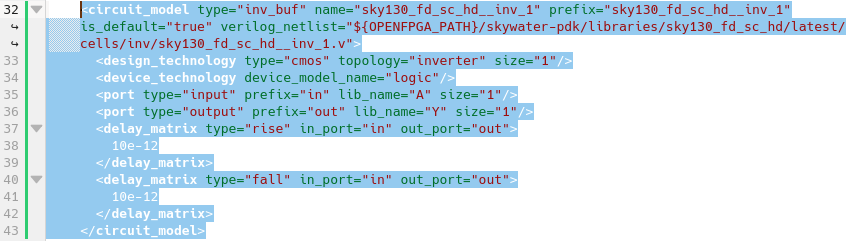
The library location should be ” /home/ubuntu/Desktop/OpenFPGA/skywater-pdk/libraries/”

Replace e.g.

INVTX1:

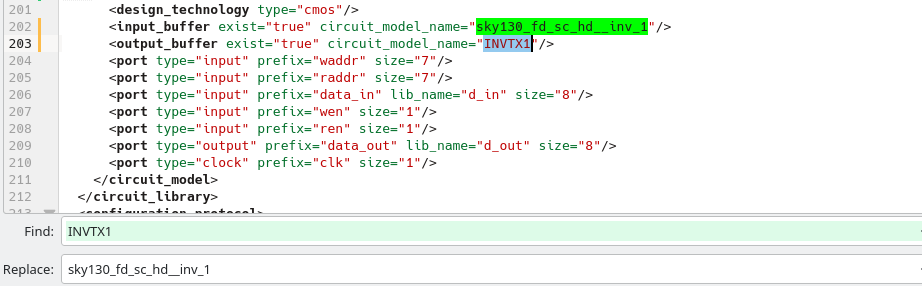


After modification—



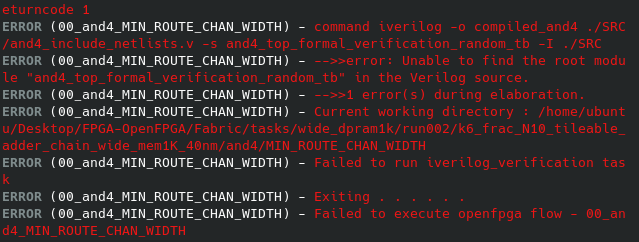
In line 35, 36 the ‘lib name’ should be corresponding to the definition in “sky130\_fd\_sc\_hd\_\_inv\_1.v”

Now look for INVTX1 in the architecture file to replace the remaining definition related to ‘INVTX1’ and rplace it with “sky130\_fd\_sc\_hd\_\_inv\_1”



Now run the task again to check if everything ok.

It should throw some error related to iverilog.



To fix the error navigate to the directory “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/” and open “iverilog\_output.txt”

The line with “…… ……. -I ./SRC” need to be modified as it cannot find the relevant files from sky130 library. The libray location already mentioned in the next line—

“/home/ubuntu/Desktop/OpenFPGA/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv/sky130\_fd\_sc\_hd\_\_inv\_1.v:34: Include file sky130\_fd\_sc\_hd\_\_inv.v not found

error: Unable to find the root module "and4\_top\_formal\_verification\_random\_tb" in the Verilog source.

: Perhaps ``-s and4\_top\_formal\_verification\_random\_tb'' is incorrect?

1 error(s) during elaboration.

1”

Replace everything with

“iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv

Now run open a terminal in the same folder and run—

source source iverilog\_output.txt

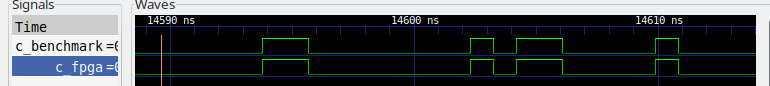
it should run success and then run—

vvp compiled\_and2

the compiled file name can be different depending on the benchmark name. it should work and then run—

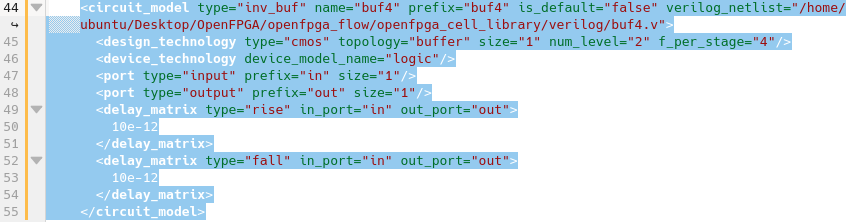
gtkwave and2\_formal.vcd

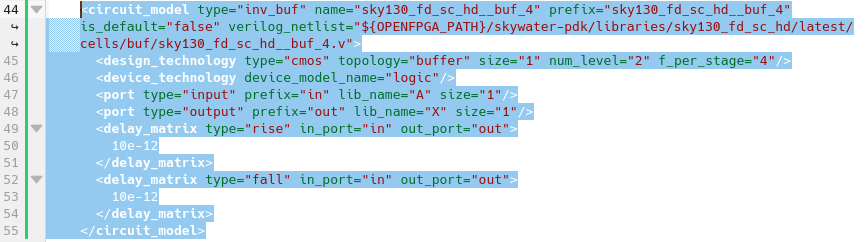
it should open the GTKWave windows where the fpga output can be checked and compared with the bench output.



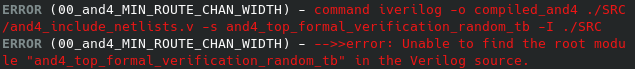
If the both outputs are same, then the std cell binding is working fine.

buf4(inv\_buf):



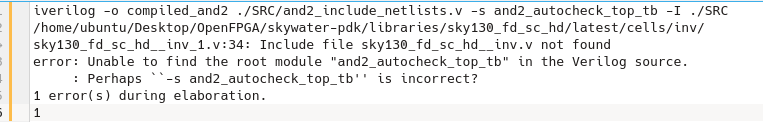


Find and replace “buf4”,” sky130\_fd\_sc\_hd\_\_buf\_4”



Navigate to—

“/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/”

è

“iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ./SRC \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/buf”

è

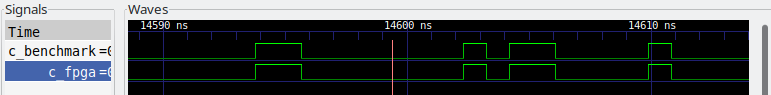
source iverilog\_output.txt

è

vvp compiled\_and2

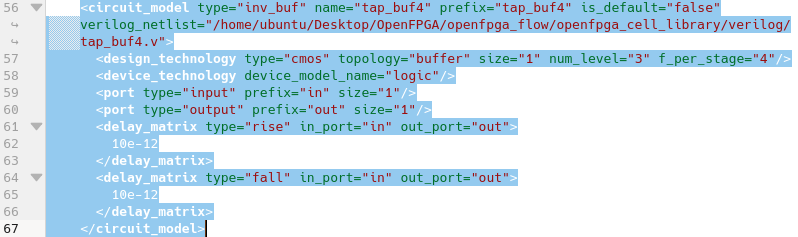
è

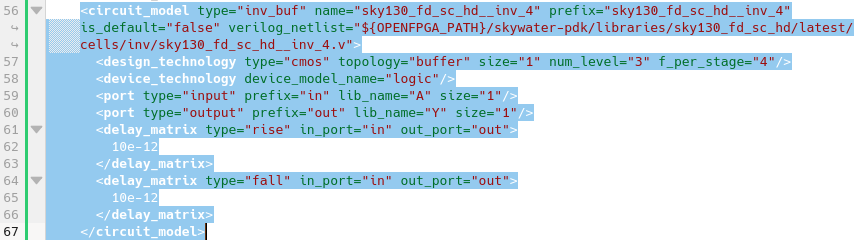
gtkwave and2\_formal.vcd



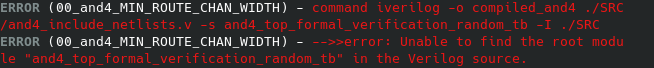
tap\_buf4:

**tap\_buf4** can be interpreted as an inverter with a fanout of 4

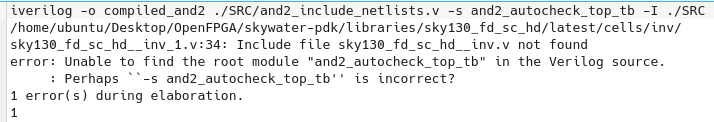




Find and replace “tap\_buf4”, ”sky130\_fd\_sc\_hd\_\_inv\_4”



Navigate to “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/”



è

iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ./SRC \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/buf

è

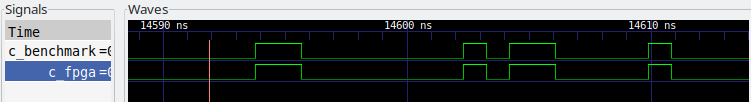
source iverilog\_output.txt

è

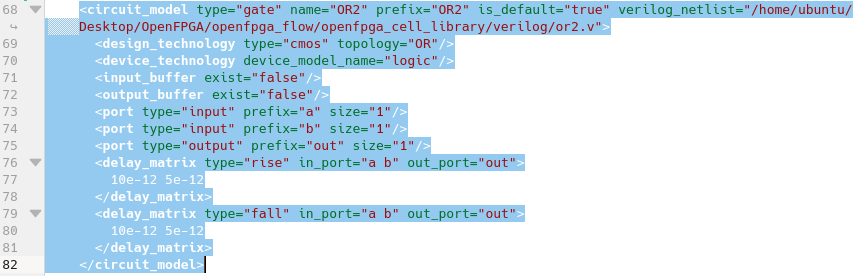
vvp compiled\_and2

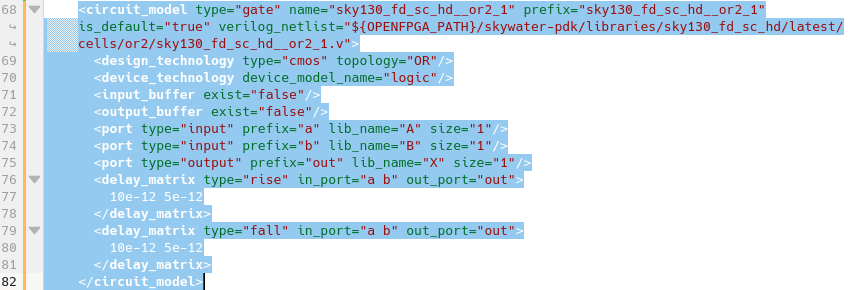
è

gtkwave and2\_formal.vcd



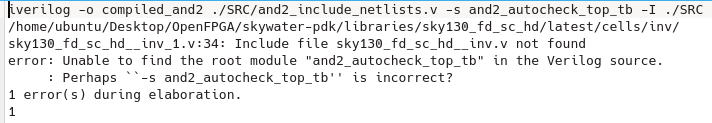
OR2:





Replace “OR2” with " sky130\_fd\_sc\_hd\_\_or2\_1"

Navigate to “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/” and open “iverilog\_output.txt”



è

iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ./SRC \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/buf \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/or2

è

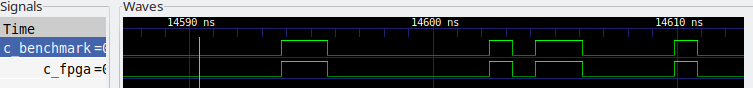
source iverilog\_output.txt

è

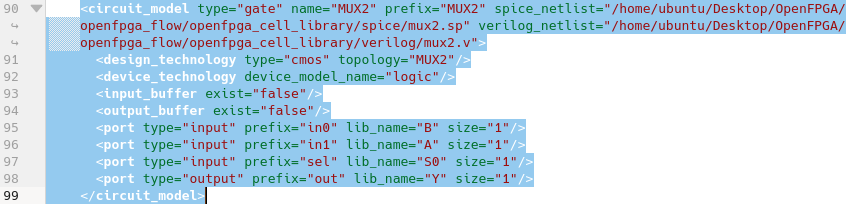
vvp compiled\_and2

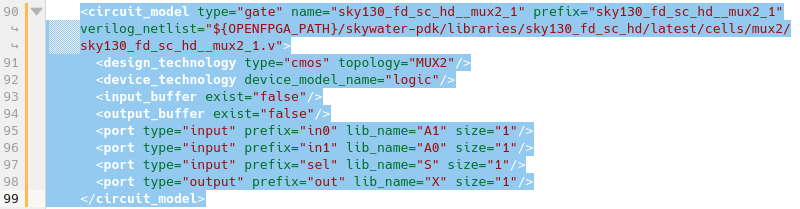
è

gtkwave and2\_formal.vcd



MUX2 (gate):

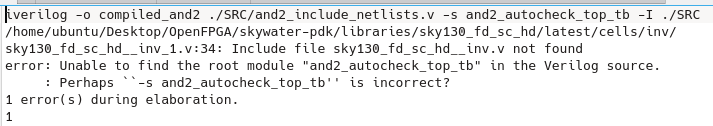




replace-"MUX2"==>"sky130\_fd\_sc\_hd\_\_mux2\_1"

as the topology is also “MUX2”, don’t replace it.

Navigate to “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/” and open “iverilog\_output.txt”



è

iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ./SRC \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/buf \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/or2 \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/mux2

è

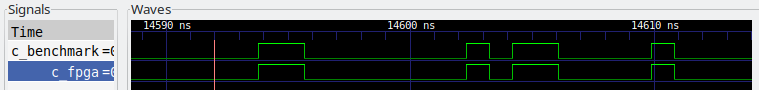
source iverilog\_output.txt

è

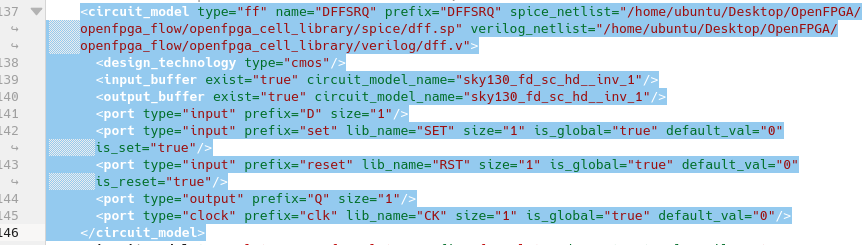
vvp compiled\_and2

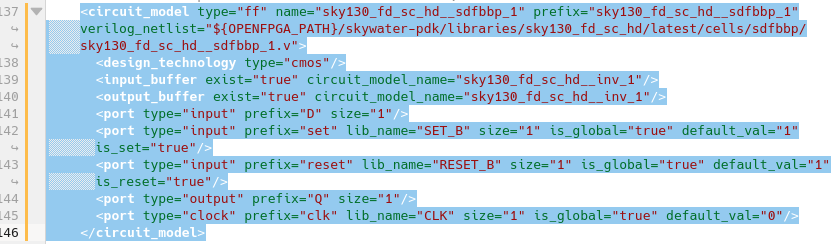
è

gtkwave and2\_formal.vcd



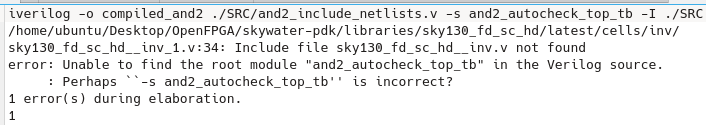
DFFSRQ:





replace-DFFSRQ==>sky130\_fd\_sc\_hd\_\_sdfbbp\_1

Navigate to “/home/ubuntu/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/latest/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/” and open “iverilog\_output.txt”



è

iverilog -o compiled\_and2 ./SRC/and2\_include\_netlists.v -s and2\_autocheck\_top\_tb \

-I ./SRC \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/inv \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/buf \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/or2 \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/mux2 \

-I ${OPENFPGA\_PATH}/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/sdfbbp

è

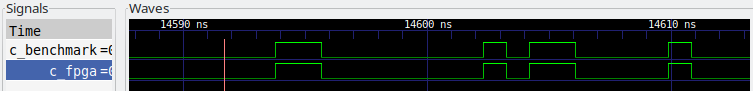
source iverilog\_output.txt

è

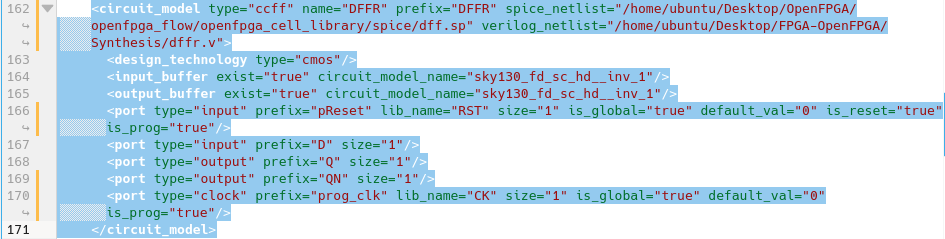
vvp compiled\_and2

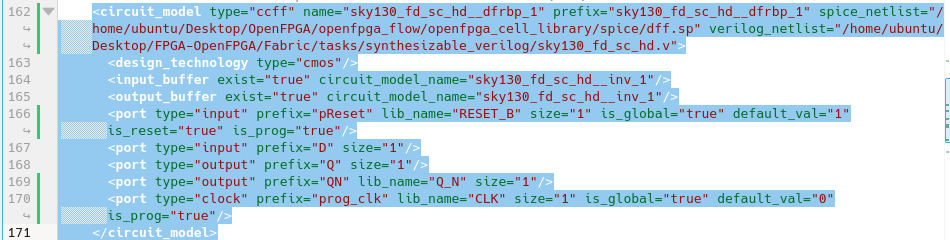
è

gtkwave and2\_formal.vcd



DFFR:





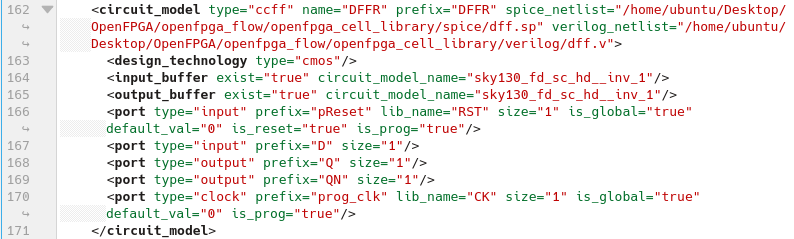
replace-DFFR==>sky130\_fd\_sc\_hd\_\_dfrbp\_1

this part cannot be compiled with iverilog because of limitation of iverilog (probably it cann’t handle delayed d input and throw some internal error that cannot be identified from OpenFPGA iverilog wrapped)

to confirm it I have to run the dff.v (from OpenFPGA/openfpga\_flow/openfpga\_cell\_library/verilog/) with a testbench with a separately synthesized netlist.

**Synthesis**

As the synthesis already done within the OpenFPGA process no synthesis is necessary. But during binding the OpenFPGA architecture file with Sky130 I had to leave the “DFFR” scan chain flip-flop as there is no suitable Standard cell in the cell library. The nearest equivalent was with inverted reset did’nt passed the compilation.



Now I have to synthesize (with sky130 librery) the ‘dff.v’ only to include in the fabric\_netlists.v file. As I am going to use this for test purpose, time constraints are not considered and running through interactive mode in Cadence Genus. For the lib files follow the steps in <https://github.com/baruaeee/ASIC_ADC/blob/main/MyProject/Installation_Details.docx>

genus -gui

read\_libs sky130\_fd\_sc\_hd\_tt\_025C\_1v80.lib

read\_hdl dff.v

elaborate DFFR

set\_db syn\_generic\_effort medium

set\_db syn\_map\_effort medium

set\_db syn\_opt\_effort medium

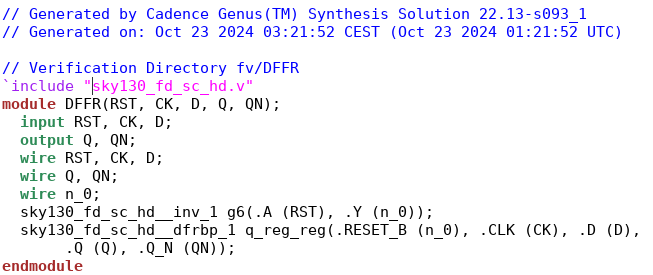
syn\_generic

syn\_map

syn\_opt

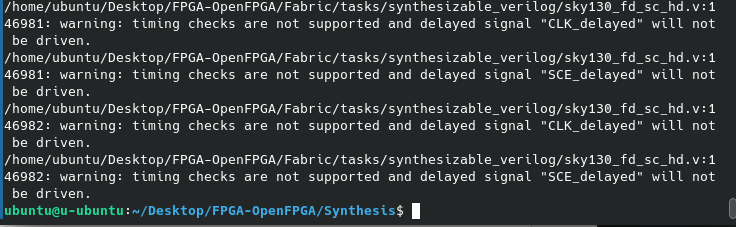
write\_hdl > dffr.v

the content of the synthesized file--



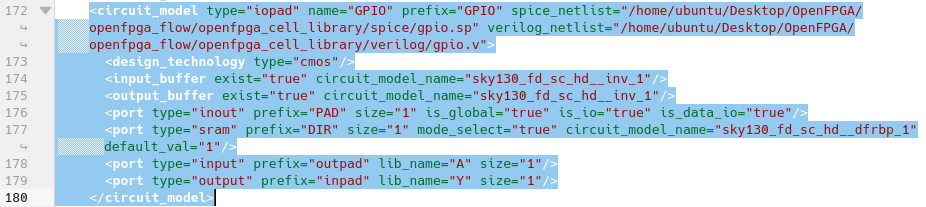
The testbench can be found at <https://github.com/baruaeee/FPGA-OpenFPGA/blob/main/Synthesis/dffr_tb.v>

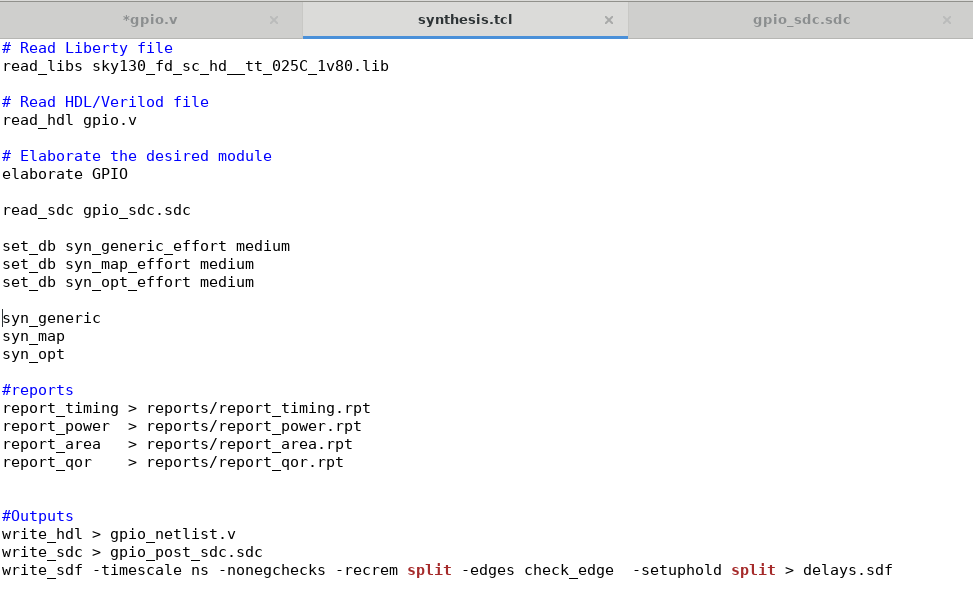
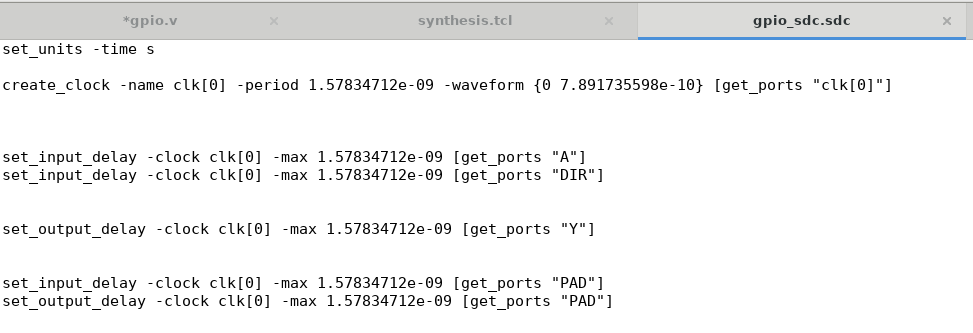
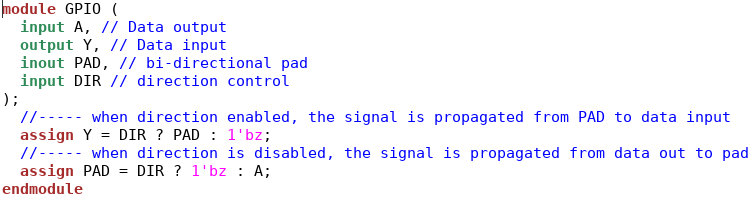
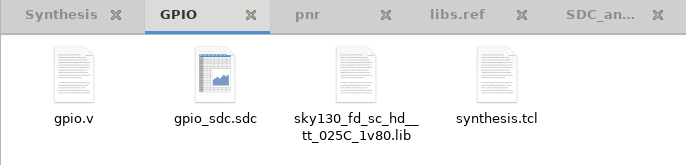
If I run the testbench with iverilog it throws bunch of errors which can’t be seen during compilation from OpenFPGA processes.

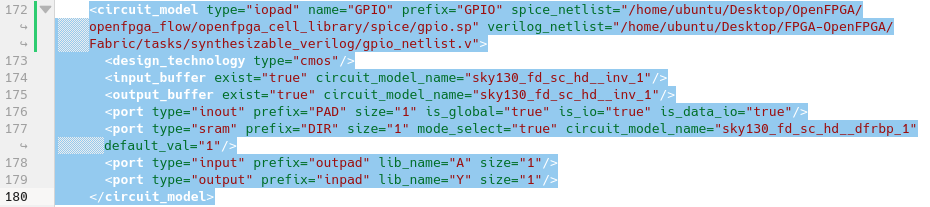
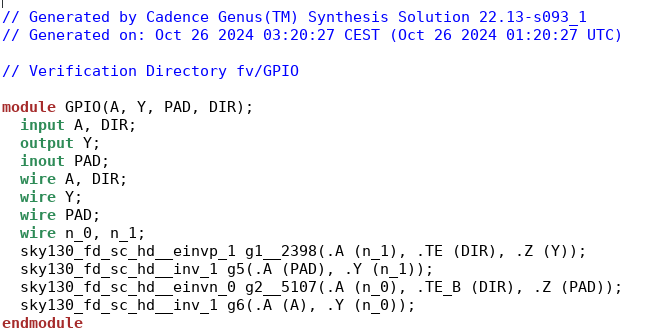


But the fabric netlist generated by OpenFPGA is correct.

**GPIO:**

The gpio.v need to be synthesized to properly bind with sky130.

the sdc file is not useful, as the verilog design has no clock defined.



Lets try the compilation in Cadence XCELIUM.

**Compile in Cadence XCELIUM:**

This is the functional netlist verification. If the synthesis could be done with Genus, a Logical Equivalent Check (LEC) can be done using CONFRML. As synthesis has been already done inside the OpenFPGA process and we have data for functional verification, for now skip the LEC.

The XCELIUM tools (e.g. xmvlog) do not work properly if the ‘include’ files are out of the working directory. For sky130 the complete PDK directory is about 12GB. To avoid copying all library files I will only copy the necessary files/folders to the working directory.

Here are the view of the task directory.



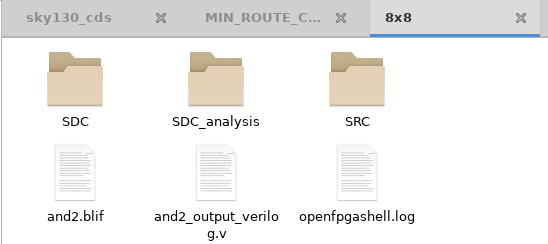
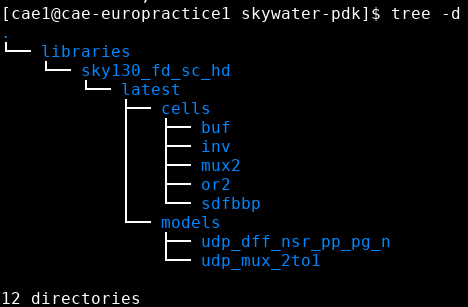
Lets make ‘8x8’ directory for compilation. I will delete all unnecessary files for simplicity.

Without modification, “/home/cae1/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/8x8/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/” is the working directory. Lets move everything to ‘8x8’ directory and delete everything.

è

mv /home/cae1/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/8x8/k6\_frac\_N10\_tileable\_40nm/and2/MIN\_ROUTE\_CHAN\_WIDTH/\* .

after modifying it should look like following—



Copy ‘skywater-pdk’ from ‘/home/cae1/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/8x8/skywater-pdk/libraries/sky130\_fd\_sc\_hd/latest/cells/’. It should look like following—

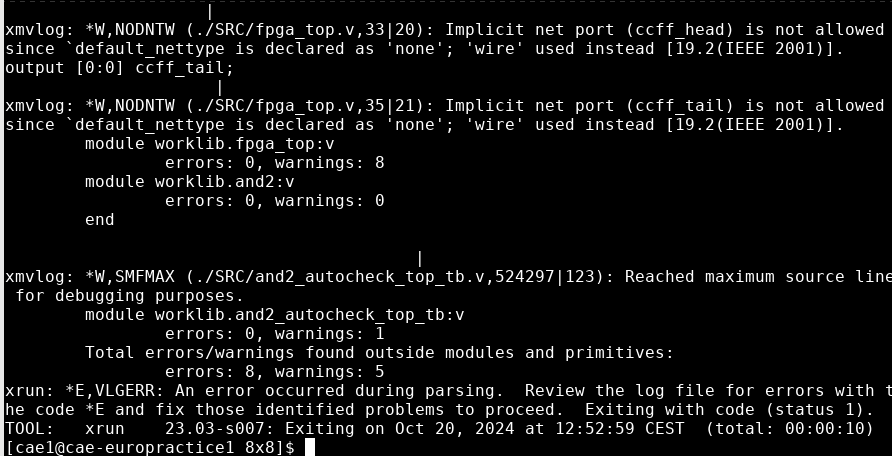


Open terminal from ‘8x8’. Let run the ‘xrun’ without any modification of the verilog files.

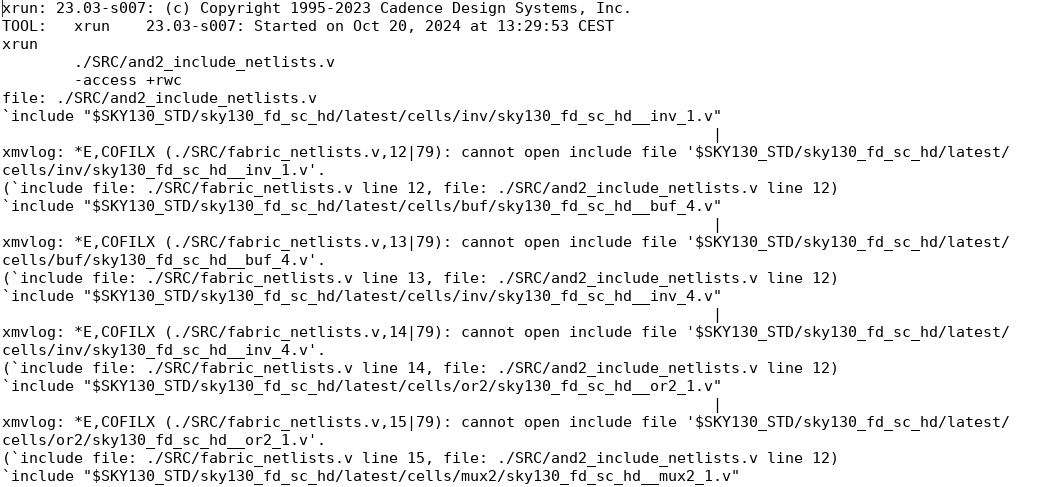
è

xrun ./SRC/and2\_include\_netlists.v -access +rwc

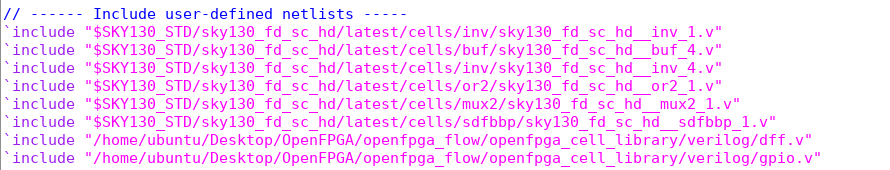
as the ‘and2\_include\_netlists.v’ file has include reference to sky130-pdk, it will throw a lots of errors.



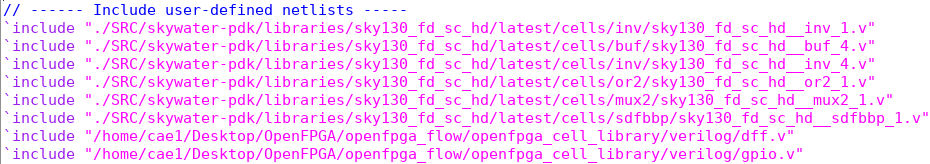
The details on these errors can be found in “/home/cae1/Desktop/FPGA-OpenFPGA/Fabric/tasks/synthesizable\_verilog/8x8/xrun.log”



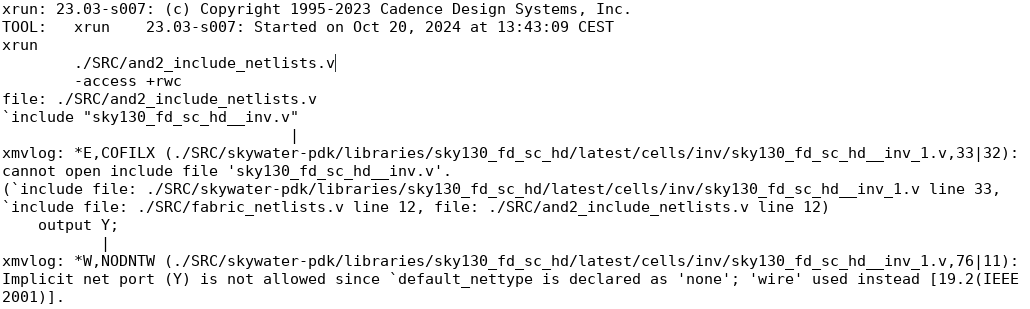
Now I have to fix all the errors and run the ‘xrun’ command on and on.



Even though the files are in the denoted location, still ‘xrun/xmvlog’ can’t detect. Let’s modify these lines—



After running the ‘xrun’ command—



It cannot find the include verilog files annoted in the ‘sky130\_fd\_sc\_hd\_\_inv\_1.v’ and similarly other corresponding files. To fix this add use ‘-include’ in the xrun command to include the relevant directories. Without 64bit mode the compilation throws an error related to virtual memory.

xrun ./SRC/and2\_include\_netlists.v -access +rwc \

-incdir ./SRC/sky130/cells/inv \

-incdir ./SRC/sky130/cells/buf \

-incdir ./SRC/sky130/cells/or2 \

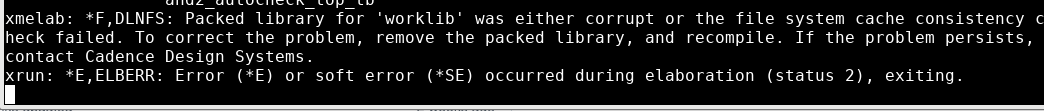
-incdir ./SRC/sky130/cells/mux2 \

-incdir ./SRC/sky130/cells/sdfbbp \

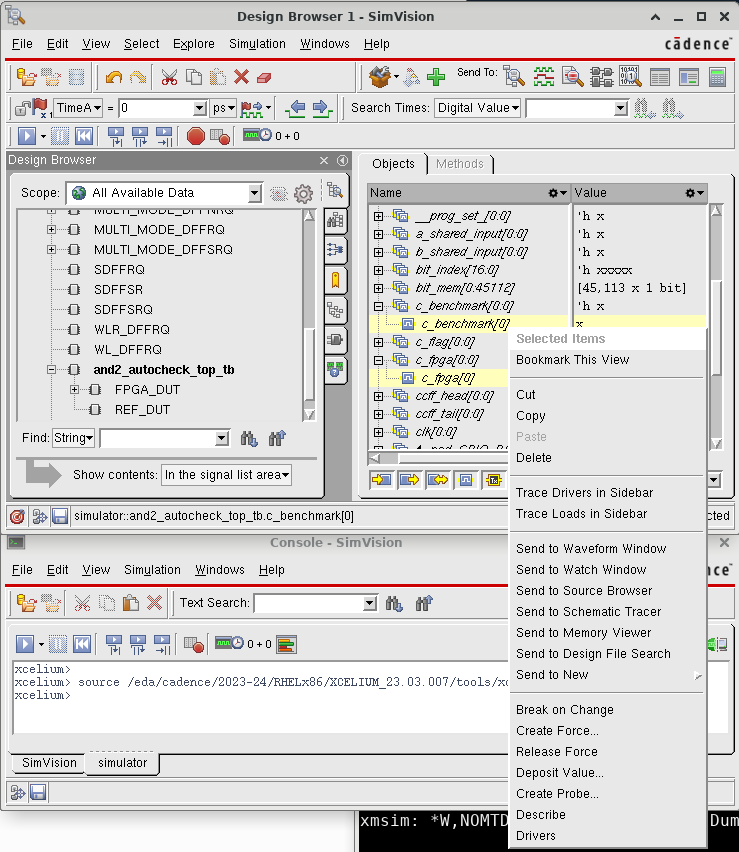
-64bit \

-gui &

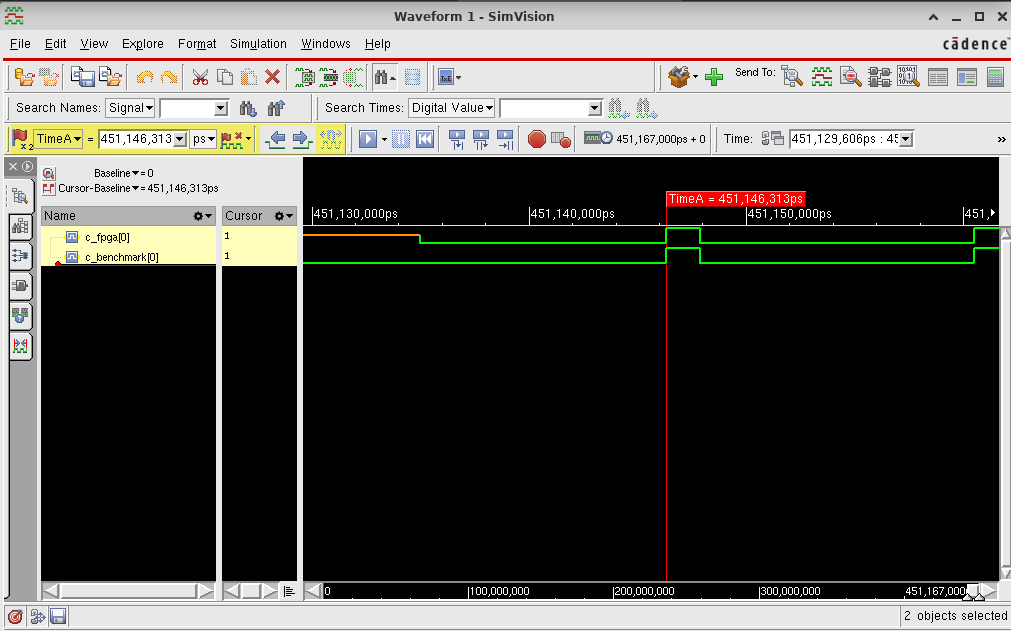
It throwing some error without details specifics.



If similar procedure done in ‘6x6’ è 8x8 grid fabric it works.



Use ‘Send to Waveform Window’—



**Place and Route (Physical Design)**

Input/output files—

Synthesized gate level netlist (\*.v file, from OpenFPGA or Genus)

Physical library (\*.lef file for Innovus)

IO pad assignment file (\*.io -- optional)

Timing library (\*.lib file)

Constraint file (\*.sdc file)

Innovus either can be run with TCL script or in gui.

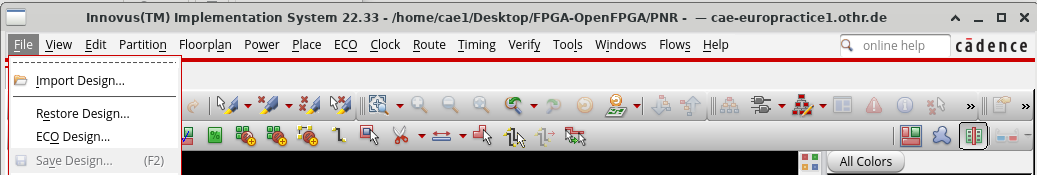
Design import—

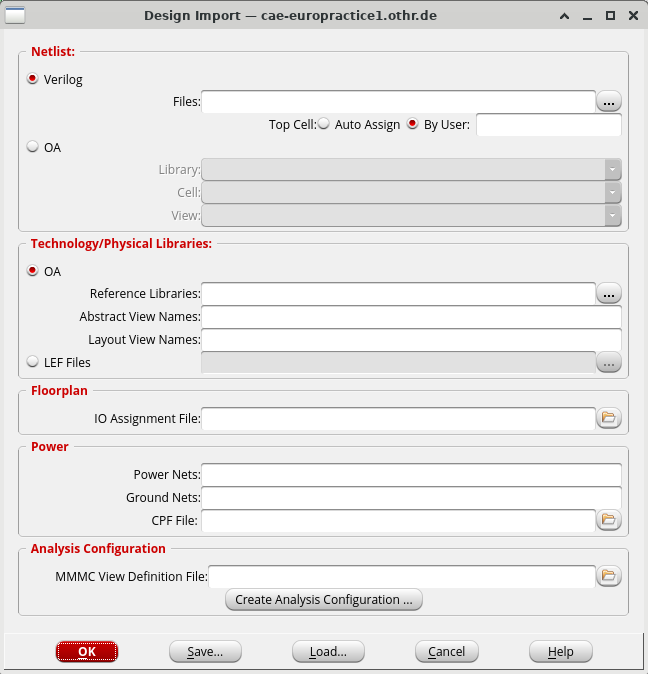
1. Netlist (from synthesis)
2. Physical library (\*.lef)
   1. Technology lef – metals/via (e.g. sky130\_ef\_sc\_hd.lef)
   2. Standard cell lef -- (e.g. sky130\_fd\_sc\_hd.lef)
3. Floorplan –
4. Power—VDD, VSS
5. Analysis configuration—MMMC setup

==>

Innovus

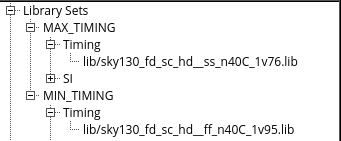
==>

Import design:

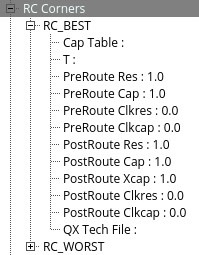


Select 

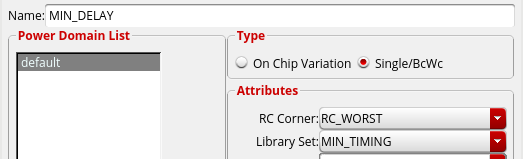
Create 2 library set, 1 for max and aonther for min. 

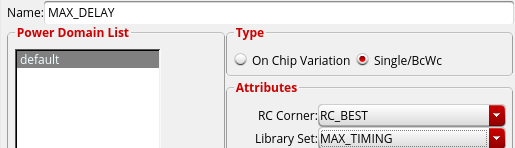


Create 2 RC Corner, 1 for best and another for worst. 

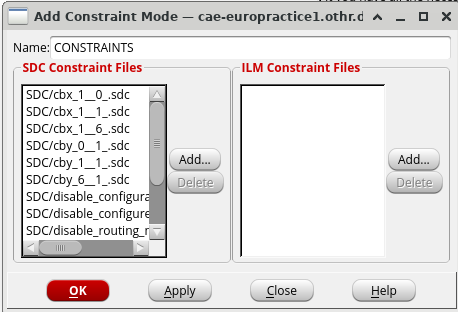
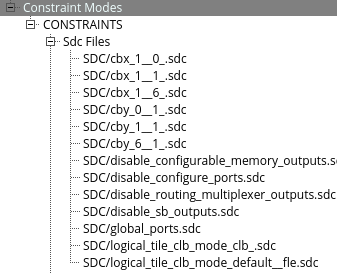


Create 2 Delay Corners 1 for max delay and another min dilay. 

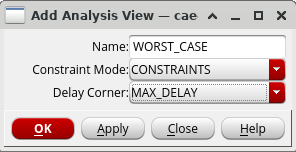
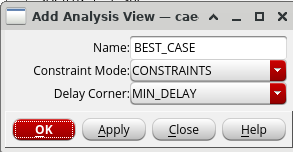




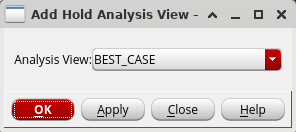
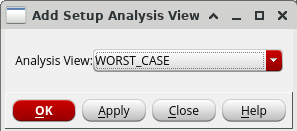
Create Constraint Mode. 



Create Analysis Views, 1 for best case and another for worst case. 



Create Setup and Analysis Views. 



Save & close.

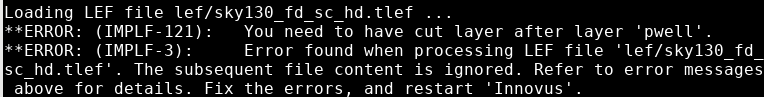
After including every files--

The working directory and its contents can be found <https://github.com/baruaeee/FPGA-OpenFPGA/tree/main/PNR>

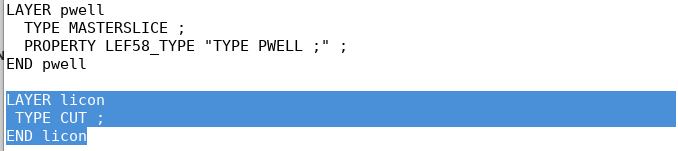
For innouvus the sky130\_fd\_sc\_hd.v file is not necessary. Including this in the fabric netlist can invite unwanted syntax errors. So it can be removed or commented in the fabric\_netlists.v file.

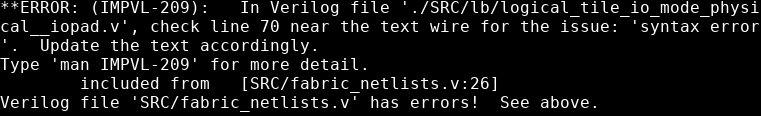


It can throw some error relater to ‘pwell’--

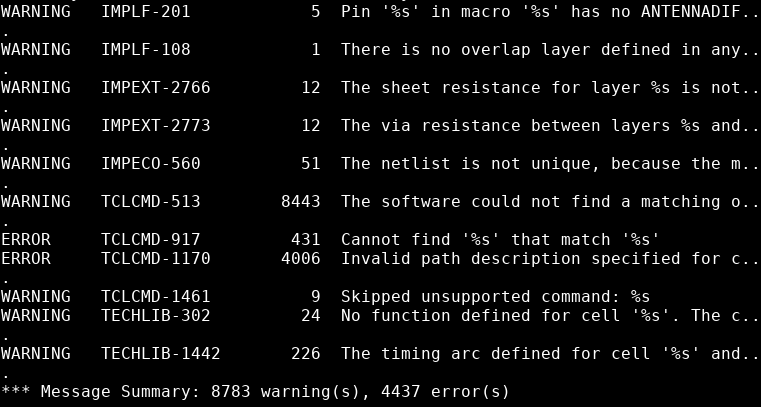
To fix the error open the ’sky130\_fd\_sc\_hd.tlef’ file and add the folloing after pwell layer dfinition.

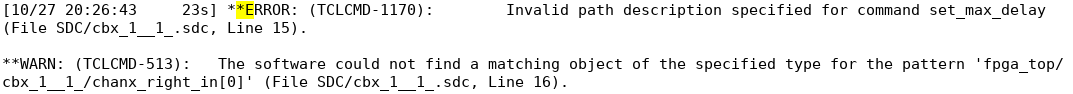
LAYER licon  
 TYPE CUT ;  
END licon

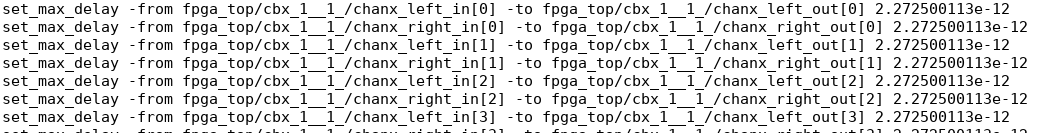
Next it can throw some syntax erro due to ”`default\_nettype wire” in the structural verilog file.

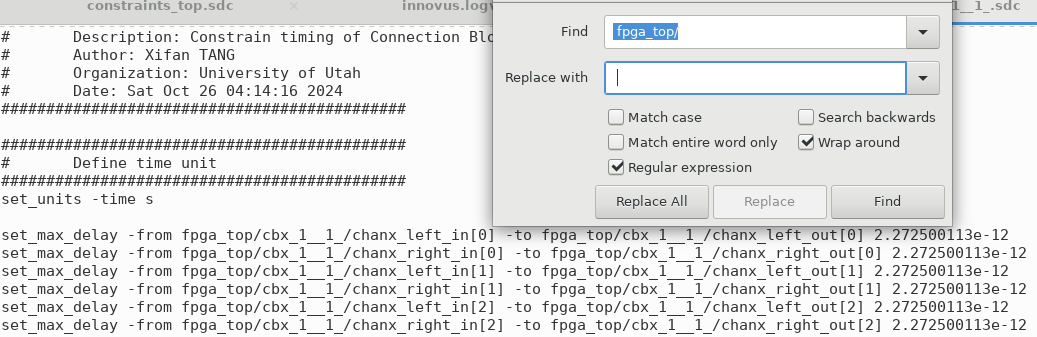
Lets remove or comment this line in all verilog netlist files where it is not necessary (at the end of module decription).

Now innovus the files provided, but with a lots of errors.

The details of these errors can be found in the innovus.log file in the working directory. Lets resolve one by one of these problems.

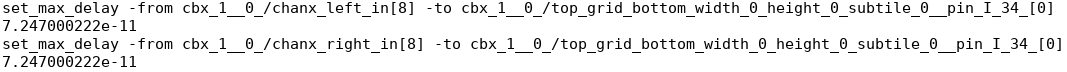
The sdc file cbx\_1\_\_1\_.sdc looks like--

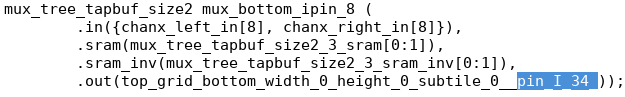
Remove fpga\_top/ --

after editing (same for other similar files)--

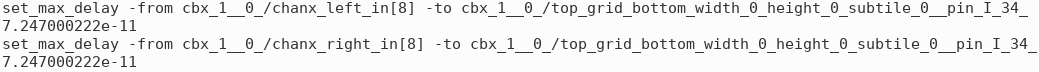
Some ports are mentioned with [0] at the end even though in the verilog files it defined differently.

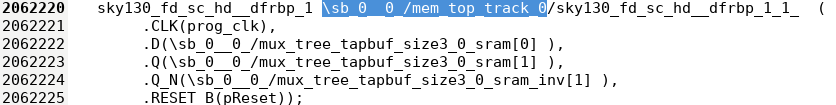
In the sdc file--

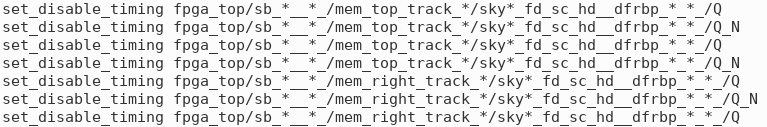
In the verilog file--



These “[0]” need to be removed to avoid errors. After editing--

Some instantiation has long names that should be fixed in sdc file--

This instance referenced in the sdc generated from OpenFPGA as below--

After correction--

As the sdc generated for hierarchical netlist and innovus is flattening the netlist the the ports/pins name become different. As a result innovus can’t identify the path description often and it difficult to figure it out the proper fix. To make the fix easier lets save the flatten netlist from innovus consol.

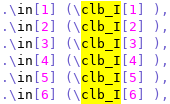
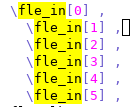
to get the netlists run the following command in the innovus consol.

==>

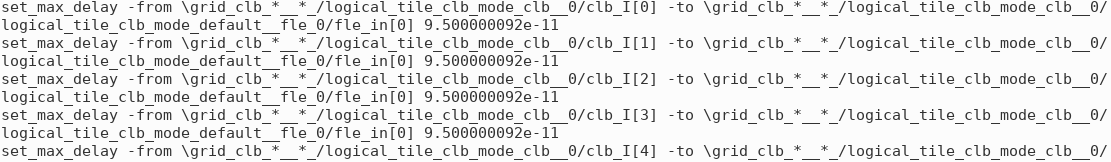
saveNetlist -flat fabric\_flat\_innovus.v

saveNetlist -flattenBus fabric\_innovus.v

==>

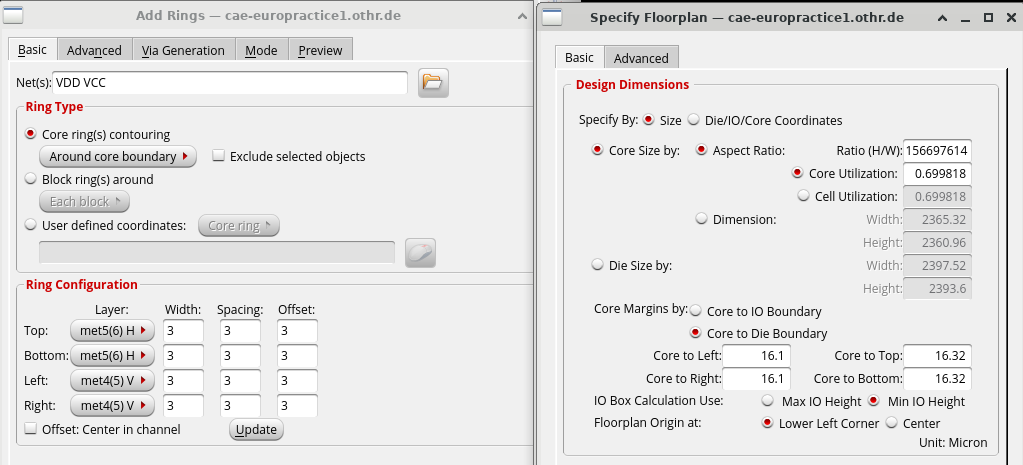
From the flattenBus netlist the path look like-- 

After correction--

**Floorplan**

Before planning IO pad, to get a rough overview I want make discreate design.

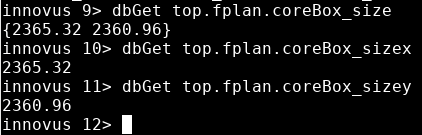
The Specify Floorplan and Add Rings are as below--

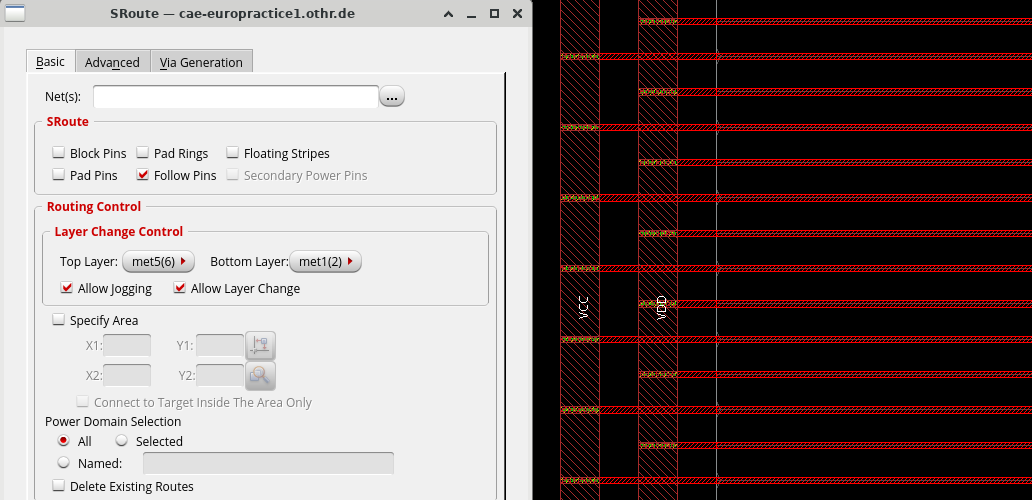
To get the size of the core area run the following--

==>

dbGet top.fPlan.coreBox\_size

dbGet top.fPlan.coreBox\_sizex

dbGet top.fPlan.coreBox\_sizey



Corner cell size: 200 x 203.665

PAD cell size: 75 x 200

