Tutorial Number 2

Introduction to Synthesis

LEARNING OUTCOMES:

At the end of this lab, participants will be able to:

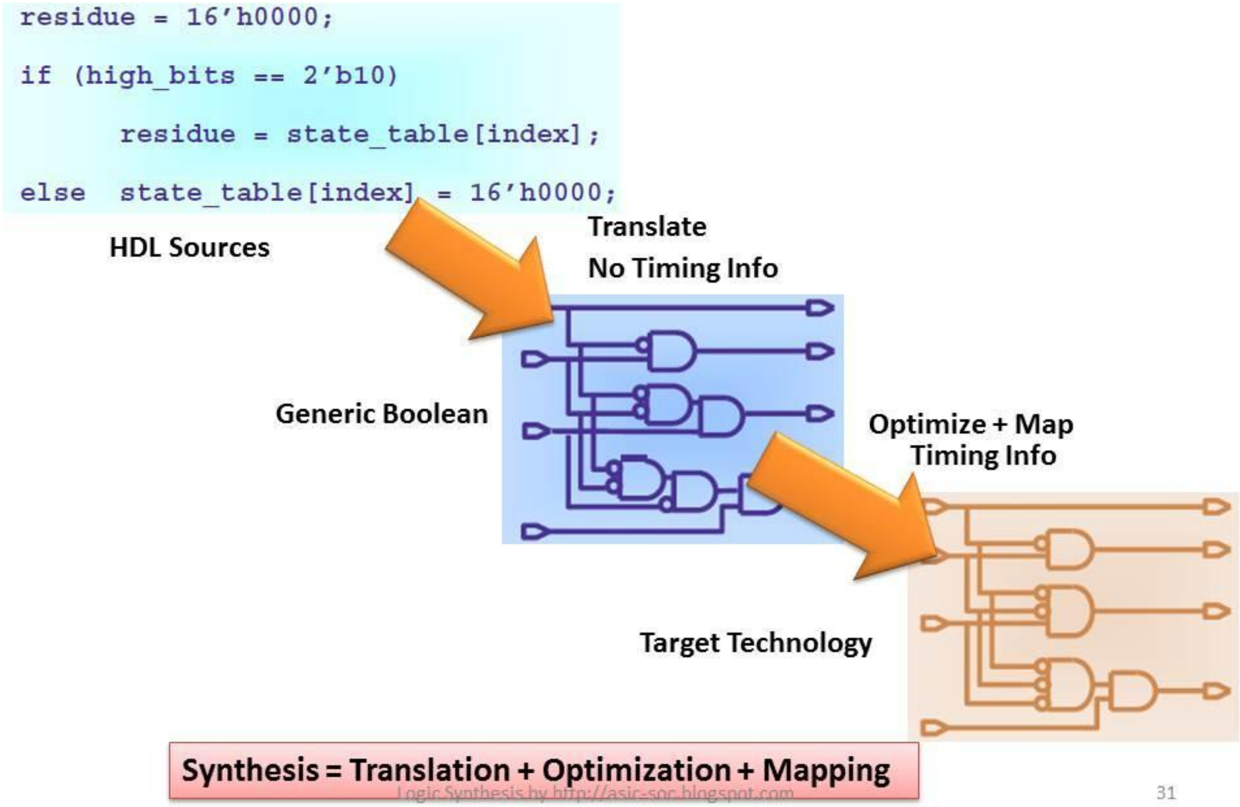
1. Initialize the design prior to running prep -design <design_name>
2. Run the whole flow for synthesis
3. Experiment with the different variables for synthesis configuration scripts.
4. View the statistics for a complete synthesis flow.
5. Meet the timing requirements for the design using Static Timing Analysis (STA).

Introduction to RTL Synthesis

Synthesis is a process by which the RTL or the logical design is converted into a circuit using the components of standard cell library (SCL). This results in a gate level netlist. The other factors responsible for specifying design intent includes the timing, clock definition, false path, power and area constraints for the synthesis step are defined in the Synopsys Design Constraint (SDC) file.

RTL synthesis is defined in two steps

- Translation from RTL description into gates
- Optimization of Logic

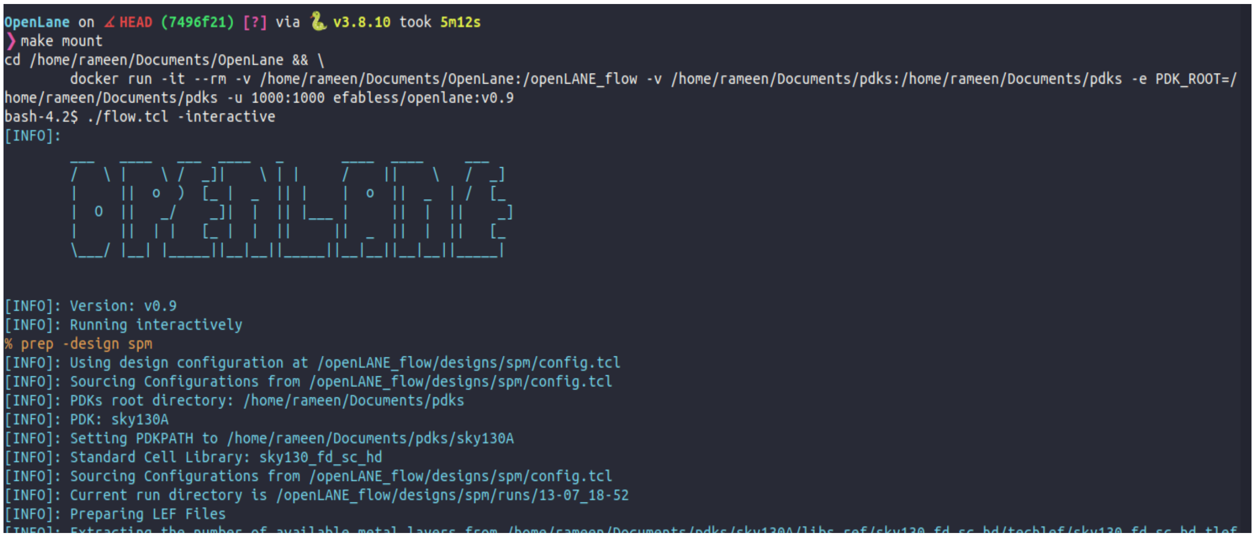


```
residue = 16'h0000;
if (high_bits == 2'b10)
      residue = state_table[index];
else  state_table[index] = 16'h0000;
```

Synthesis = Translation + Optimization + Mapping

For this lab we will be exploring the design picorv32a.

Interactive Mode:

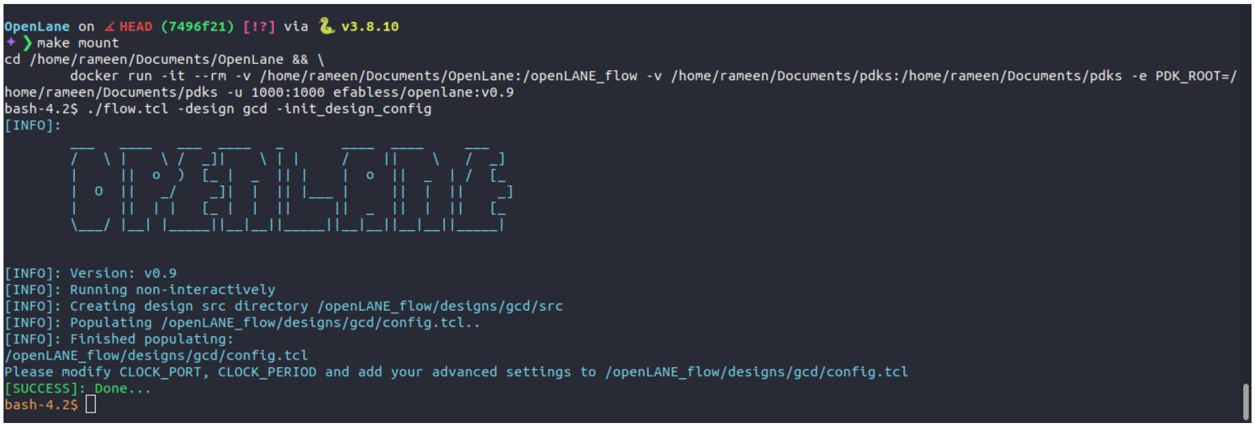


```
OpenLane on ∡HEAD (7496f21) [?] via 🐍 v3.8.10 took 5m12s
❯ make mount
cd /home/rameen/Documents/OpenLane && \
        docker run -it --rm -v /home/rameen/Documents/OpenLane:/openLANE_flow -v /home/rameen/Documents/pdks:/home/rameen/Documents/pdks -e PDK_ROOT=/
home/rameen/Documents/pdks -u 1000:1000 efabless/openlane:v0.9
bash-4.2$ ./flow.tcl -interactive
[INFO]:
         ___   ____   ___  ____   _       ____  ____     ___
        /   \ |    \ /  _]|    \ | |     /    ||    \   /  _]
        |     ||  o  )  [_ |  _  || |    |  o  ||  _  | /  [_
        |  O  ||   _/    _]|  |  || |___ |     ||  |  ||    _]
        |     ||  |  |   [_ |  |  ||     ||  _  ||  |  ||   [_
         \___/ |__|  |_____||__|__||_____||__|__||__|__||_____|


[INFO]: Version: v0.9
[INFO]: Running interactively
% prep -design spm
[INFO]: Using design configuration at /openLANE_flow/designs/spm/config.tcl
[INFO]: Sourcing Configurations from /openLANE_flow/designs/spm/config.tcl
[INFO]: PDKs root directory: /home/rameen/Documents/pdks
[INFO]: PDK: sky130A
[INFO]: Setting PDKPATH to /home/rameen/Documents/pdks/sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Sourcing Configurations from /openLANE_flow/designs/spm/config.tcl
[INFO]: Current run directory is /openLANE_flow/designs/spm/runs/13-07_18-52
[INFO]: Preparing LEF Files
[INFO]: Extracting the number of available metal layers from /home/rameen/Documents/pdks/sky130A/libs.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd.tlef
```

For initialize a new design from RTL coding we have to run init_design_config command

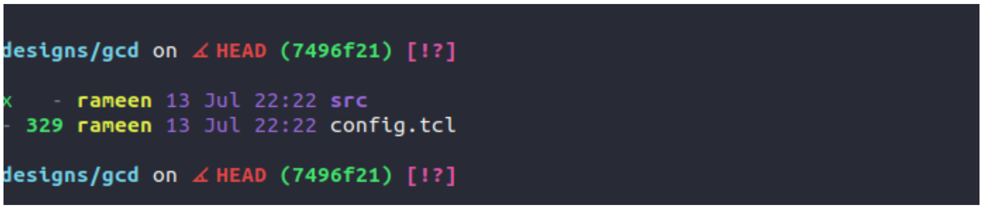
Download gcd RTL design file from here

1. Initialize your design



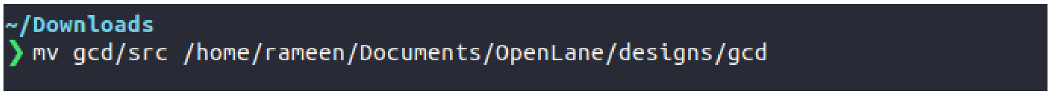
```
OpenLane on ∡HEAD (7496f21) [!?] via 🐍 v3.8.10
✦ ❯ make mount
cd /home/rameen/Documents/OpenLane && \
        docker run -it --rm -v /home/rameen/Documents/OpenLane:/openLANE_flow -v /home/rameen/Documents/pdks:/home/rameen/Documents/pdks -e PDK_ROOT=/
home/rameen/Documents/pdks -u 1000:1000 efabless/openlane:v0.9
bash-4.2$ ./flow.tcl -design gcd -init_design_config
[INFO]:
         ___   ____   ___  ____   _       ____  ____     ___
        /   \ |    \ /  _]|    \ | |     /    ||    \   /  _]
        |     ||  o  )  [_ |  _  || |    |  o  ||  _  | /  [_
        |  O  ||   _/    _]|  |  || |___ |     ||  |  ||    _]
        |     ||  |  |   [_ |  |  ||     ||  _  ||  |  ||   [_
         \___/ |__|  |_____||__|__||_____||__|__||__|__||_____|


[INFO]: Version: v0.9
[INFO]: Running non-interactively
[INFO]: Creating design src directory /openLANE_flow/designs/gcd/src
[INFO]: Populating /openLANE_flow/designs/gcd/config.tcl..
[INFO]: Finished populating:
/openLANE_flow/designs/gcd/config.tcl
Please modify CLOCK_PORT, CLOCK_PERIOD and add your advanced settings to /openLANE_flow/designs/gcd/config.tcl
[SUCCESS]: Done...
bash-4.2$ ▯
```

2. You can check there are two folders in design/gcd



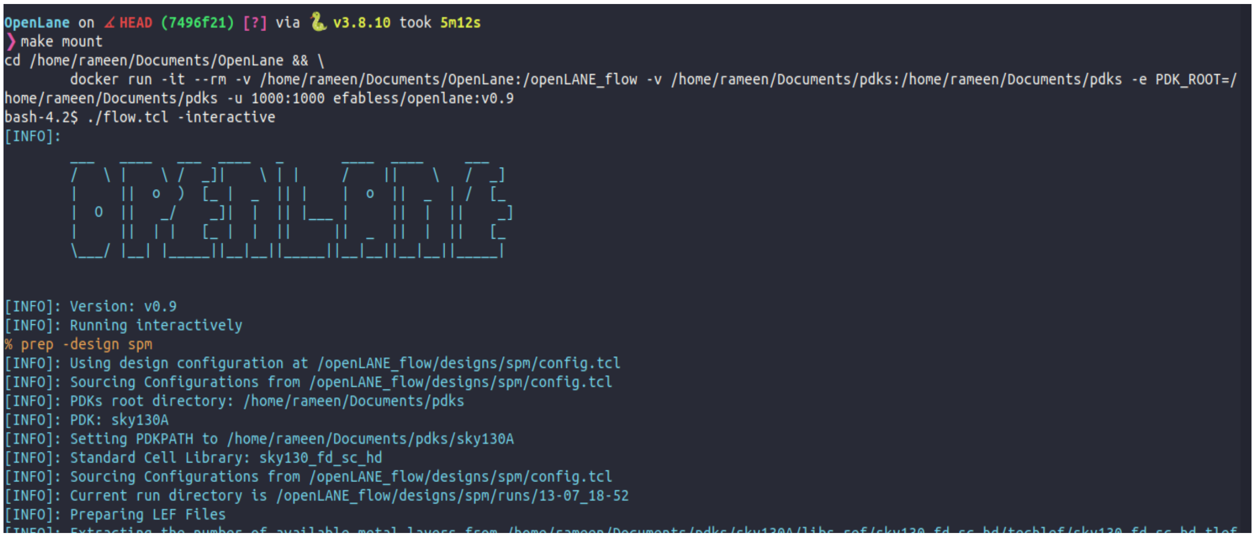
```
designs/gcd on ∡HEAD (7496f21) [!?]

x    - rameen 13 Jul 22:22 src
- 329 rameen 13 Jul 22:22 config.tcl

designs/gcd on ∡HEAD (7496f21) [!?]
```

3. Copy your design into src folder



```
~/Downloads
❯ mv gcd/src /home/rameen/Documents/OpenLane/designs/gcd
```

4. Verify the Design name and Clock port name into src/<top_module_file> and config.tcl file
5. Now you can prep the design



```
OpenLane on ∡ HEAD (7496f21) [?] via 🐍 v3.8.10 took 5m12s
❯ make mount
cd /home/rameen/Documents/OpenLane && \
        docker run -it --rm -v /home/rameen/Documents/OpenLane:/openLANE_flow -v /home/rameen/Documents/pdks:/home/rameen/Documents/pdks -e PDK_ROOT=/
home/rameen/Documents/pdks -u 1000:1000 efabless/openlane:v0.9
bash-4.2$ ./flow.tcl -interactive
[INFO]:
         ___   ____   ___  ____   _       ____  ____     ___
        /   \ |    \ /  _]|    \ | |     /    ||    \   /  _]
        |     ||  o  )  [_ |  _  || |    |  o  ||  _  | /  [_
        |  O  ||   _/    _]|  |  || |___ |     ||  |  ||    _]
        |     ||  |  |   [_ |  |  ||     ||  _  ||  |  ||   [_
         \___/ |__|  |_____||__|__||_____||__|__||__|__||_____|


[INFO]: Version: v0.9
[INFO]: Running interactively
% prep -design spm
[INFO]: Using design configuration at /openLANE_flow/designs/spm/config.tcl
[INFO]: Sourcing Configurations from /openLANE_flow/designs/spm/config.tcl
[INFO]: PDKs root directory: /home/rameen/Documents/pdks
[INFO]: PDK: sky130A
[INFO]: Setting PDKPATH to /home/rameen/Documents/pdks/sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Sourcing Configurations from /openLANE_flow/designs/spm/config.tcl
[INFO]: Current run directory is /openLANE_flow/designs/spm/runs/13-07_18-52
[INFO]: Preparing LEF Files
[INFO]: Extracting the number of available metal layers from /home/rameen/Documents/pdks/sky130A/libs.ref/sky130_fd_sc_hd/techlef/sky130_fd_sc_hd.tlef
```

6. Run the command “run_synthesis” for synthesize the design.

LabTask: Create a copy of the design picorv32a with your name in the designs directory.

Step 1: Go to path “cd $OPENLANE_ROOT” and execute “make mount” command to enter the docker container;

Synthesis in interactive mode:

- ./flow.tcl -interactive
- prep –design picorv32a__<your_name>
- run_synthesis

You can interact with the synthesis parameters during flow execution:

To check the value of certain parameters when in docker:

echo $::env(variable_name)

for eg. *echo $::env(SYNTH_MAX_FANOUT)*

Similarly, we can update the values for the synthesis variables interactively, without having to leave the docker.

Steps to follow for updating values in variables:

- Prep –design <design_name> -tag <run_name>
- set ::env(SYNTH_MAX_FANOUT) 4

and then check if the value is updated: *echo $::env(SYNTH_MAX_FANOUT)*

- run_synthesis

The runs created after this step can be found on this path:

/home/document/openlane/designs/picorv32a_<your_name>/runs/<run_name>

The generated netlist can be found at:

/home/document/openlane/designs/picorv32a_<your_name>/runs/<run_name>/results/synthesis/<design_name>.synthesis.v

The logs can be found at:

/home/document/openlane/designs/picorv32a_<your_name>/runs/<run_name>/logs/synthesis/

The timing reports can be found at:

/home/document/openlane/designs/picorv32a_<your_name>/runs/<run_name>/reports/synthesis/

Constraints:

- The area and timing of your circuit are mainly determined by your circuit/design architecture and coding style.
- There is always a trade-off between the design timing and area.
- In fact, a super tight timing constraint may work while synthesis, but failed in the Place & Route (P&R) procedure.

TO DO TASK:

1. From the logs, i.e. synthesis.log identifies the flop ratio in the design.

(hint: flop ratio= total no. of flops/ total cells)

2. Try different synthesis strategies and identify the differences in area and timing for picorv32a_test after synthesis execution. Complete the table:

(hint: Vary the value of ::env(SYNTH_STRATEGY) in design config.tcl file)

SYNTH_STRATEGY	TNS	Area
DELAY 0		
DELAY 1		
DELAY 2		
AREA 0		
AREA 1		
AREA 2		

Useful repo : https://github.com/merledu/OpenLane_Workshop