The first thing after you open Jupyter Notebook

Activate the index number

File Edit View Run Kernel Tabs Settings Help

CA_RISCV.IPYNB

CA_RISCV.ipynb

Markdown

Python 3 (ipykernel)

1. Openlane backend flow with OpenRAM macros

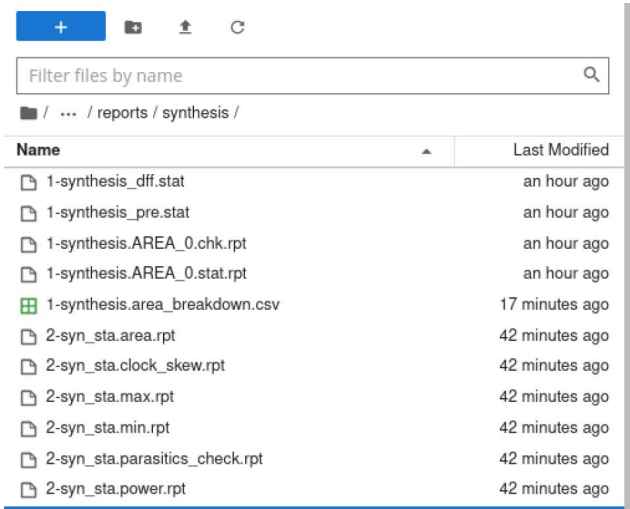
1.1. Objectives

- Synthesize the simulated RTL project.
- Replace and harden the behaviour SRAM modules (your data/instruction memory) in Verilog with the real memory macros generated by OpenRAM in floorplan.
- Finish the Openlane hardware backend flow.
- Analyze the results step-by-step and have a basic understanding of the RTL-to-GDSII procedure.

1.2. Tips

- The entire CA exercise session requires ESAT IT infrastructure to achieve full functionalities, such as using NC Verilog (Cadence) for simulation.
- However, this **Backend** session is fully open-source software based. Hence, you can also set up the environment on any linux system.
  - The `conda-env` build takes up ~**3 GB**. Each full-flow execution will generate **4~6 GB** output data under `runs/`.
  - Hence, if your ESAT disk quota is not sufficient, you can conduct this part of experiments on your own device after you have completed the simulation step.
- Suggestion when encounter fail: try to finish the Full Flow **even if the synthesis or floorplan run fails** (e.g., cannot meet timing). Sometimes it can be "synthesis fail , floorplan fail , full flow success".

Timing [lots of steps]



2-syn_sta.rpt

```
Startpoint: control_fw_EX_MEM._19_
            (rising edge-triggered flip-flop clocked by clk)
Endpoint: data_memory.process_for_mem[0].spad_inst
          (falling edge-triggered flip-flop clocked by clk')
Path Group: clk
Path Type: max

Fanout     Cap    Slew   Delay    Time   Description
---------------------------------------------------------------------------
                  0.15    0.00    0.00   clock clk (rise edge)
                          0.00    0.00   clock network delay (ideal)
                  0.15    0.00    0.00 ^ control_fw_EX_MEM._19_/CLK (sky130_fd_sc_hd__dfrtp_2)
                  0.06    0.47    0.47 v control_fw_EX_MEM._19_/Q (sky130_fd_sc_hd__dfrtp_2)
     3    0.01                           control_fw_EX_MEM.dout[3] (net)
                  0.06    0.00    0.47 v data_memory._4_/A (sky130_fd_sc_hd__or2b_2)
                  0.05    0.30    0.77 v data_memory._4_/X (sky130_fd_sc_hd__or2b_2)
     1    0.00                           data_memory._1_ (net)
                  0.05    0.00    0.77 v data_memory._5_/A (sky130_fd_sc_hd__buf_1)
                  0.05    0.11    0.88 v data_memory._5_/X (sky130_fd_sc_hd__buf_1)
     1    0.01                           data_memory.web0 (net)
                  0.07    0.00    0.88 v data_memory.process_for_mem[0].spad_inst/web0 (sky130_sram_2rw_64x128_64)
                                  0.88   data arrival time

                  0.15   50.00   50.00   clock clk' (rise edge)
                          0.00   50.00   clock network delay (ideal)
                         -0.25   49.75   clock uncertainty
                          0.00   49.75   clock reconvergence pessimism
                                 49.75 ^ data_memory.process_for_mem[0].spad_inst/clk0 (sky130_sram_2rw_64x128_64)
                         -0.10   49.65   library setup time
                                 49.65   data required time
---------------------------------------------------------------------------
                                 49.65   data required time
                                 -0.88   data arrival time
---------------------------------------------------------------------------
                                 48.76   slack (MET)
```

We only focus on this group here.

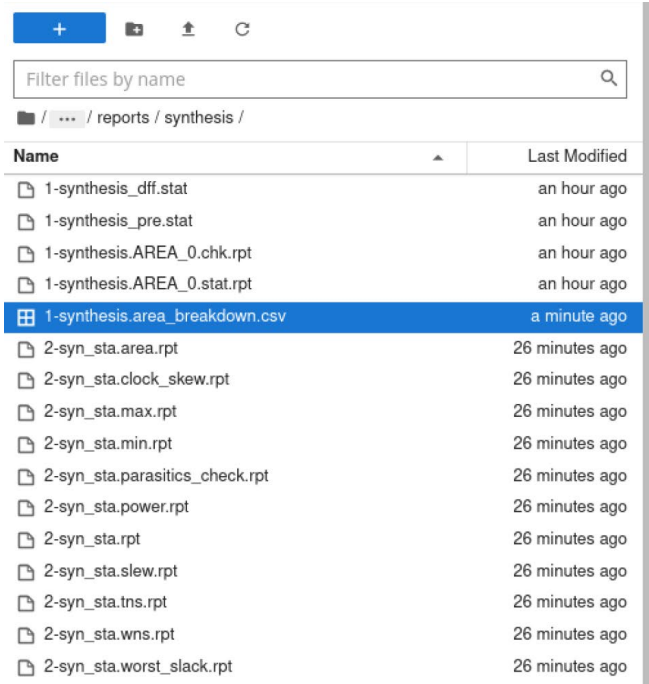
Critical path

Clock margin (ns)

Q: How to derive the fastest possible CLK frequency?

in reports/<the step>/x-<step name>_sta.rpt

Design Area Break down [Synthesis]



1-synthesis.area_breakdown

Delimiter: ,

Area breakdown

		Line #	Module Name	Area(um^2)	Area(%)
1	0	8080:	'$paramod$6136cc84484...	2653.795	0.415
2	1	8168:	'$paramod\alu\DATA_W=...	247360.989	38.718
3	2	8223:	'$paramod\branch_unit\D...	5463.990	0.855
4	3	8238:	'$paramod\mux_2\DATA_...	972.182	0.152
5	4	8274:	'$paramod\pc\DATA_W=s...	1501.440	0.235
6	5	8290:	'$paramod\reg_arstn_en\...	82.579	0.013
7	6	8306:	'$paramod\reg_arstn_en\...	165.158	0.026
8	7	8322:	'$paramod\reg_arstn_en\...	206.448	0.032
9	8	8338:	'$paramod\reg_arstn_en\...	412.896	0.065
10	9	8354:	'$paramod\reg_arstn_en\...	1325.021	0.207
11	10	8370:	'$paramod\reg_arstn_en\...	2653.795	0.415
12	11	8417:	'$paramod\register_file\D...	119121.747	18.646
13	12	8435:	'$paramod\sram_BW32\A...	138703.362	21.711
14	13	8453:	'$paramod\sram_BW64\A...	227869.314	35.667
15	14	8478:	'\alu_control':	137.632	0.022
16	15	8504:	'\control_unit':	137.632	0.022
17	16	8551:	'\cpu':	127.622	0.020
18	17	8577:	'\immediate_extend_unit':	520.499	0.081
19	18	9373:	'\cpu':	638872.583	100.000

Total design area (after synthesis)

in reports/synthesis/1-synthesis.area_breakdown.csv

**You might need to rerun block 3.3 in the notebook to regenerate the table.*

Die Area [Floorplan]

Trigger the Openlane Execution

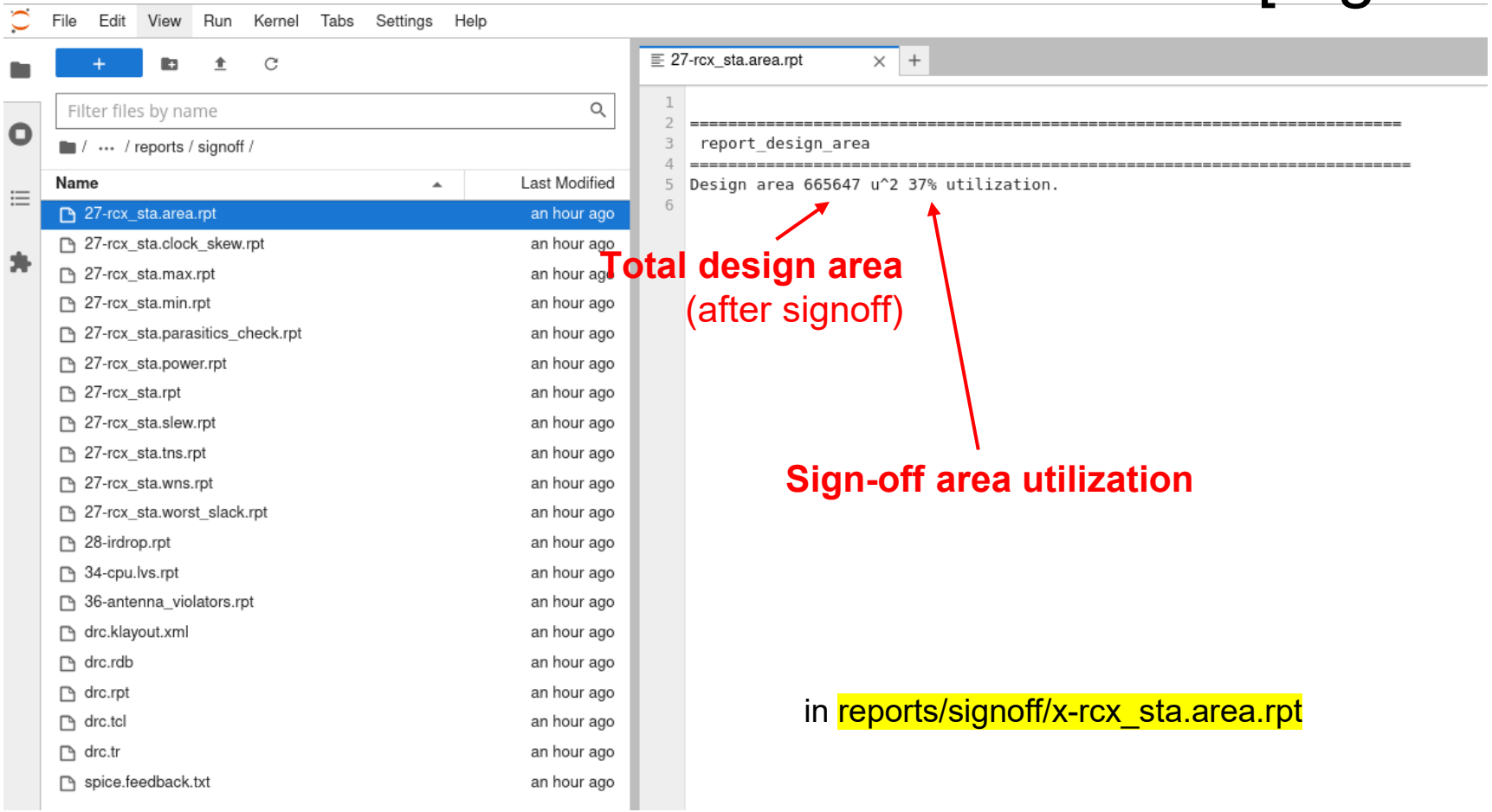
```
[14]: !flow.tcl -design . -ignore_mismatches -tag $TAG -overwrite
Last executed at 2023-03-31 17:27:37 in 30m 59.49s
OpenLane 2023.03.01_0_ge10820ec-conda
All rights reserved. (c) 2020-2022 Efabless Corporation and contributors.
Available under the Apache License, version 2.0. See the LICENSE file for more details.

[INFO]: Using configuration in 'config.tcl'...
[INFO]: PDK Root: /users/micas/lmei/Documents/CA_project_2023/Backend/conda-env/share/pdk
[INFO]: Process Design Kit: sky130A
[INFO]: Standard Cell Library: sky130_fd_sc_hd
[INFO]: Optimization Standard Cell Library: sky130_fd_sc_hd
[INFO]: Run Directory: /users/micas/lmei/Documents/CA_project_2023/Backend/runs/230331-165113
[INFO]: Removing existing /users/micas/lmei/Documents/CA_project_2023/Backend/runs/230331-165113...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[STEP 1]
[INFO]: Running Synthesis (log: runs/230331-165113/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: runs/230331-165113/logs/synthesis/2-sta.log)...
[STEP 3] (Floorplan step)
[INFO]: Running Initial Floorplanning (log: runs/230331-165113/logs/floorplan/3-initial_fp.log)...
[INFO]: Floorplanned with width 1347.34 and height 1346.4.
[STEP 4]
[INFO]: Running IO Placement...
[STEP 5]
[INFO]: Running Global Placement (log: runs/230331-165113/logs/placement/5-global.log)...
[STEP 6]
[INFO]: Running basic macro placement (log: runs/230331-165113/logs/placement/6-basic_mp.log)...
[STEP 7]
[INFO]: Running Tap/Decap Insertion (log: runs/230331-165113/logs/floorplan/7-tap.log)...
[INFO]: Power planning with power {vccd1} and ground {vssd1}...
[STEP 8]
[INFO]: Generating PDN (log: runs/230331-165113/logs/floorplan/8-pdn.log)...
[STEP 9]
[INFO]: Running Global Placement (log: runs/230331-165113/logs/placement/9-global.log)...
```

Die area

You can also get it in reports/floorplan/ x-initial_fp_die_area.rpt

Area Utilization [Sign-off]



in reports/signoff/x-rcx_sta.area.rpt