Digital System Design
IAIK Open Flow

DSD Team

06.03.2024

Digital System Design
Graz University of Technology

Introduction

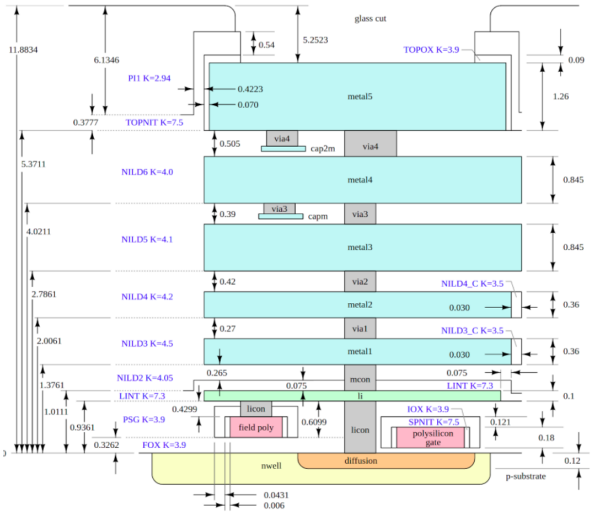
Motivation

With the recent release of several open source Process Design Kits (PDKs) such as SKY130 and advances in open source EDA tools, it is now possible to design manufacturable chips without the need to sign NDAs.

This document shows how to use the IAIK Open Flow to develop, test and integrate your cipher into a chip.

The goal of the IAIK Open Flow is to provide students with the tools and framework to develop their cipher locally on their own computer.

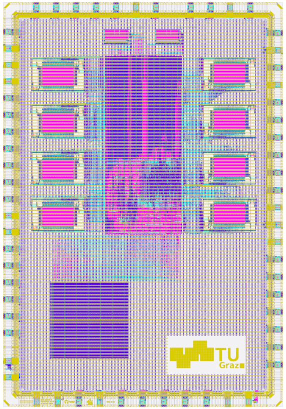
Sky130 Stackup



Layer Stackup Sky130

Open Flow ASIC

- Acts as a harness for your cipher
- SoC with Ibex RISC-V Core
    - 8 kB ROM
    - 16 kB SRAM
- Various peripherals
- Repository: *open-flow-asic*



Open Flow ASIC

Open Flow Docker

- Container with open source tools
- Make sure to *install docker*
- List of tools: *open-flow-docker*
- Pull the image via:

```
$ docker pull \
extgit.iaik.tugraz.at:8443/\
sesys/iaik-open-flow/open-flow-docker
```

Open Flow Template

- Your working template for the DSD course
- Tasks are split into
    - ex1/ - Files for exercise 1: cipher_core
    - ex2/ - Files for exercise 2: cipher_peripheral
    - open-flow/tapeout/ - Integration into harness
- Repository: *open-flow-template*

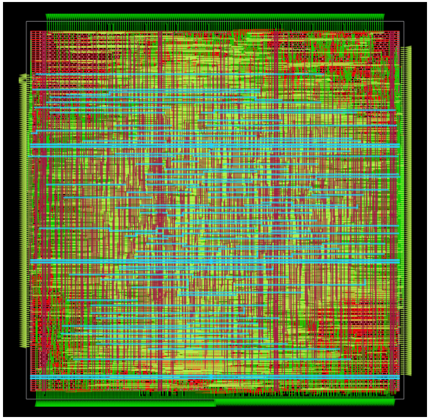
General Procedure

- The Open Flow ASIC has already been pre-hardened
- For now, a placeholder-macro is used instead of your cipher
  - Your goal: Create your cipher in 1000x1000 um
  - Communicate with the SoC over bus interfaces
  - Harden your cipher as a hard macro
  - Replace the empty wrapper in the final tapeout step
- This solution was chosen so that development of your cipher is possible on lightweight hardware such as laptops

Exercise 1

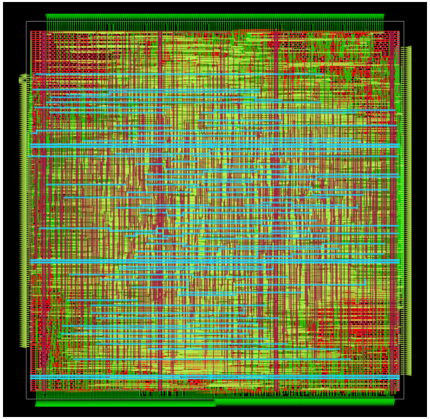
Cipher Core - Sources

- All source files under ex1/
  - ex1/src/ – RTL files
  - ex1/tb/ – testbench
  - open-flow/ex1_aux/config.json – config for OL2
  - open-flow/ex1_aux/pins.cfg – pin config for OL2



Cipher Core - Output

- Output directories
  - results/ex1_runs/
    - OL2 output files
  - results/ex1_macro/
    - Most recent OL2 output
  - ex1/tb/sim_build/
    - Simulation output



Make Targets

- ex1-lint
- ex1-openlane
- ex1-openroad
- ex1-klayout
- ex1-cocotb
- ex1-cocotb-gl

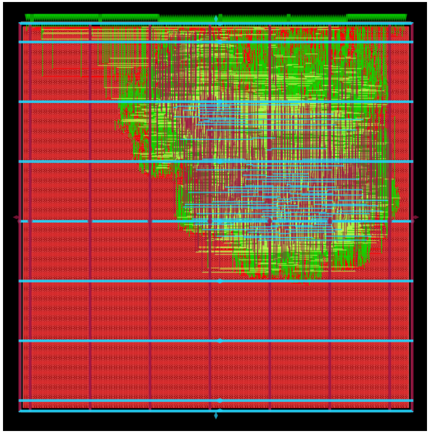
To add additional source files to your cipher, just place them into the src/ folder! All files ending in .sv are automatically picked up for the implementation and the simulation.

- ex1/src/*.sv

Exercise 2

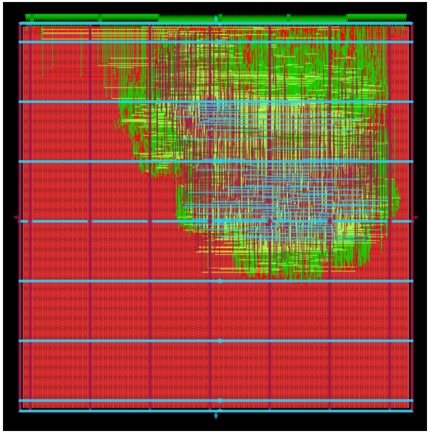
Cipher Peripheral - Sources

- All source files under ex2/
  - ex2/src/ – RTL files
  - ex2/tb/ – testbench
  - ex2/sw/ – software for the ibex core
  - open-flow/ex2_aux/config.json – config for OL2



Cipher Peripheral - Output

- Output directories
  - results/ex2_runs/
    - OL2 output files
  - results/ex2_macro/
    - Most recent OL2 output
  - ex2/tb/sim_build/
    - Simulation output



Make Targets

- ex2-lint
- ex2-openlane
- ex2-openroad
- ex2-klayout
- ex2-cocotb
- ex2-cocotb-gl

To add additional source files to your cipher, just place them into the src/ folder! All files ending in .sv are automatically picked up for the implementation and the simulation.

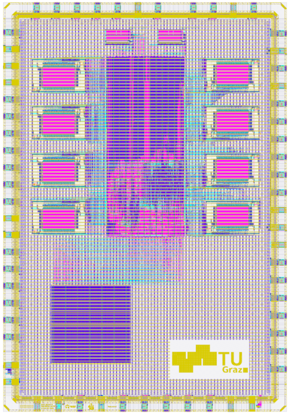
- ex2/src/*.sv

Tapeout

Final Step

To complete the tapeout:

- Harden your cipher_peripheral
- Generate the ROMs for your program
- (Optionally) Update the chip art
    - open-flow/tapeout/chip_art/chip_art.png
- Merge everything!



Make Targets

- tapeout-chip_art – generate the macro for your chip art
- tapeout-rom – generate both ROM macros
    - Set $PROGRAM env variable to active program
- tapeout-final – perform the final merging of the layouts
    - Cipher, ROM and chip art must be hardened first
- tapeout-klayout – open the final layout using KLayout

Set the active $PROGRAM

To change the program that is compiled, set the $PROGRAM environment variable.

To create a new program, just copy the whole folder ex2/sw/hello-world and rename it, for example ex2/sw/cipher-test.

This is necessary for:

- ex2-sw
- ex2-cocotb
- ex2-cocotb-gl
- tapeout-rom

Execute make targets like this:

```
$ PROGRAM=cipher-test ex2-cocotb
```

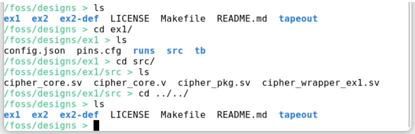
General Makefile Targets

$ make interactive

This command starts the docker container in interactive mode.

You will find yourself in a command prompt and have access to all the tools installed in the docker container.

Normally this should not be necessary for the course, but can be used for troubleshooting.

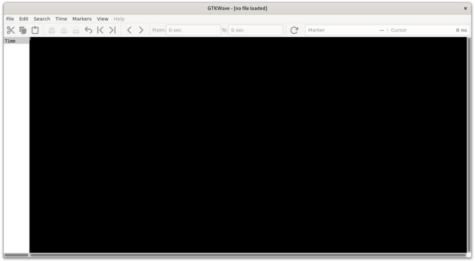


```
/foss/designs > ls
ex1  ex2  ex2-def  LICENSE  Makefile  README.md  tapeout
/foss/designs > cd ex1/
/foss/designs/ex1 > ls
config.json  pins.cfg  runs  src  tb
/foss/designs/ex1 > cd src/
/foss/designs/ex1/src > ls
cipher_core.sv  cipher_core.v  cipher_pkg.sv  cipher_wrapper_ex1.sv
/foss/designs/ex1/src > cd ../../
/foss/designs > ls
ex1  ex2  ex2-def  LICENSE  Makefile  README.md  tapeout
/foss/designs >
```

$ make gtkwave

This command starts GTKWave using the docker container without a waveform loaded.

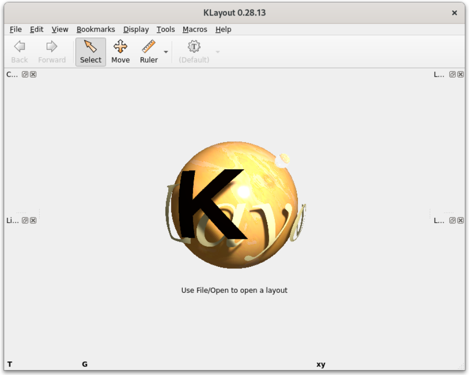
With it you can view the simulation results for your cipher.


GTKWave - [no file loaded]
File Edit Search Time Markers View Help
From: 0 sec
To: 0 sec
Marker
Cursor
0 ns
Time

$ make klayout

This command starts KLayout using the docker container without a design loaded.

Klayout is a capable layout viewer and will be used to visualize the layout of your cipher and the final chip.


KLayout 0.28.13
File Edit View Bookmarks Display Tools Macros Help
Back Forward Select Move Ruler (Default)
C...
L...
Li...
L...
T
G
xy

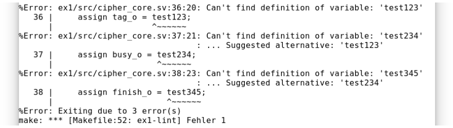
Use File/Open to open a layout

Makefile Targets - Exercise 1

$ make ex1-lint

Runs verilator in linting mode over your design to detect issues that can lead to problems.

Make sure to fix all warnings. Otherwise this might lead to problems later on.

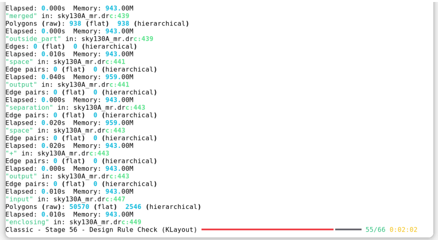


```
%Error: ex1/src/cipher_core.sv:36:20: Can't find definition of variable: 'test123'
   36 |     assign tag_o = test123;
      |                    ^~~~~~~
%Error: ex1/src/cipher_core.sv:37:21: Can't find definition of variable: 'test234'
                                    : ... Suggested alternative: 'test123'
   37 |     assign busy_o = test234;
      |                     ^~~~~~~
%Error: ex1/src/cipher_core.sv:38:23: Can't find definition of variable: 'test345'
                                    : ... Suggested alternative: 'test234'
   38 |     assign finish_o = test345;
      |                       ^~~~~~~
%Error: Exiting due to 3 error(s)
make: *** [Makefile:52: ex1-lint] Fehler 1
```

$ make ex1-openlane

Starts the physical process of hardening cipher_core using the sky130A PDK.

Actually, OpenLane 2 is used to harden your ciper. OpenLane 2 is the successor to OpenLane.

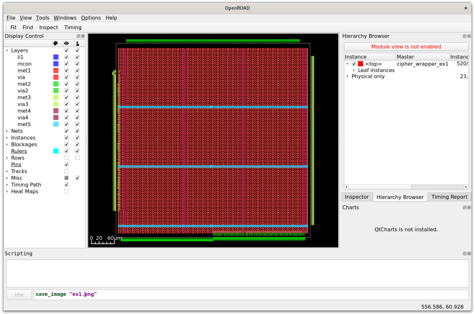


```
Elapsed: 0.000s  Memory: 943.00M
"merged" in: sky130A_mr.drc:439
Polygons (raw): 938 (flat)  938 (hierarchical)
Elapsed: 0.000s  Memory: 943.00M
"outside_part" in: sky130A_mr.drc:439
Edges: 0 (flat)  0 (hierarchical)
Elapsed: 0.010s  Memory: 943.00M
"space" in: sky130A_mr.drc:441
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.040s  Memory: 959.00M
"output" in: sky130A_mr.drc:441
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.000s  Memory: 943.00M
"separation" in: sky130A_mr.drc:443
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.020s  Memory: 959.00M
"space" in: sky130A_mr.drc:443
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.020s  Memory: 943.00M
"+" in: sky130A_mr.drc:443
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.000s  Memory: 943.00M
"output" in: sky130A_mr.drc:443
Edge pairs: 0 (flat)  0 (hierarchical)
Elapsed: 0.010s  Memory: 943.00M
"input" in: sky130A_mr.drc:447
Polygons (raw): 50570 (flat)  2546 (hierarchical)
Elapsed: 0.010s  Memory: 943.00M
"enclosing" in: sky130A_mr.drc:449
Classic - Stage 56 - Design Rule Check (KLayout) ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 55/66 0:02:02
```

$ make ex1-openroad

Loads the latest stage from „ex1-openlane“ into OpenROAD to visualize the design. This can be useful if the hardening process fails due to routing conqestion etc.

To save an image of your design execute „save_image image.png“ in the tcl command line.



Module view is not enabled

QtCharts is not installed.

```
save_image "ex1.png"
```

$ make ex1-klayout

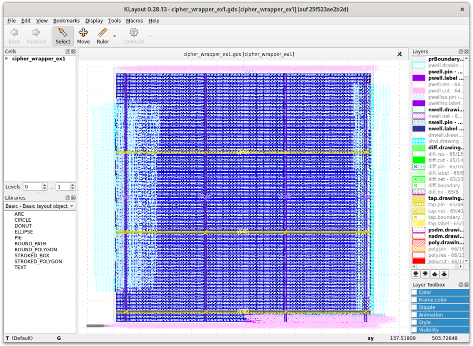
Opens the layout of your design from ex1/results in Klayout.

To save a high-resolution image, open:

Macros → Macro Development

Then execute in the console:

```
RBA::Application.instance
.main_window.current_view
.save_image("image.png",2000,2000)
```



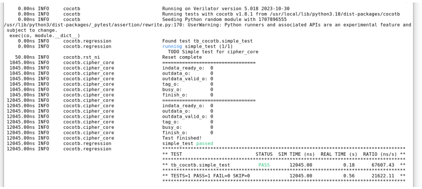
KLayout 0.28.13 - cipher_wrapper_ex1.gds [cipher_wrapper_ex1] (auf 25f523ae2b2d)
File Edit View Bookmarks Display Tools Macros Help
Back Forward Select Move Ruler (Default)
Cells
cipher_wrapper_ex1
cipher_wrapper_ex1.gds [cipher_wrapper_ex1]
Layers
Levels 0 .. 1
Libraries
Basic - Basic layout object
ARC
CIRCLE
DONUT
ELLIPSE
PIE
ROUND_PATH
ROUND_POLYGON
STROKED_BOX
STROKED_POLYGON
TEXT
Layer Toolbox
Color
Frame color
Stipple
Animation
Style
Visibility
T (Default) G
xy 137.51809 503.72648

$ make ex1-cocotb

Runs RTL simulation of your design using cocotb as the testbench environment.

You can write your testbench in Python under ex1/tb/.

For RTL simulation, Verilator is used as the simulator.



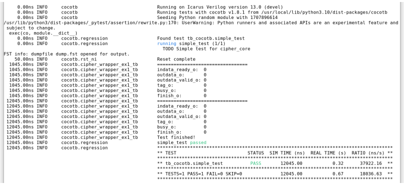
```
     0.00ns INFO     cocotb                            Running on Verilator version 5.018 2023-10-30
     0.00ns INFO     cocotb                            Running tests with cocotb v1.8.1 from /usr/local/lib/python3.10/dist-packages/cocotb
     0.00ns INFO     cocotb                            Seeding Python random module with 1707896555
/usr/lib/python3/dist-packages/_pytest/assertion/rewrite.py:170: UserWarning: Python runners and associated APIs are an experimental feature and
 subject to change.
  exec(co, module.__dict__)
     0.00ns INFO     cocotb.regression                 Found test tb_cocotb.simple_test
     0.00ns INFO     cocotb.regression                 running simple_test (1/1)
                                                         TODO Simple test for cipher_core

    50.00ns INFO     cocotb.rst_ni                     Reset complete
  1045.00ns INFO     cocotb.cipher_core                ==================================
  1045.00ns INFO     cocotb.cipher_core                indata_ready_o:  0
  1045.00ns INFO     cocotb.cipher_core                outdata_o:       0
  1045.00ns INFO     cocotb.cipher_core                outdata_valid_o: 0
  1045.00ns INFO     cocotb.cipher_core                tag_o:           0
  1045.00ns INFO     cocotb.cipher_core                busy_o:          0
  1045.00ns INFO     cocotb.cipher_core                finish_o:        0
 12045.00ns INFO     cocotb.cipher_core                ==================================
 12045.00ns INFO     cocotb.cipher_core                indata_ready_o:  0
 12045.00ns INFO     cocotb.cipher_core                outdata_o:       0
 12045.00ns INFO     cocotb.cipher_core                outdata_valid_o: 0
 12045.00ns INFO     cocotb.cipher_core                tag_o:           0
 12045.00ns INFO     cocotb.cipher_core                busy_o:          0
 12045.00ns INFO     cocotb.cipher_core                finish_o:        0
 12045.00ns INFO     cocotb.cipher_core                Test finished!
 12045.00ns INFO     cocotb.regression                 simple_test passed
 12045.00ns INFO     cocotb.regression                 ****************************************************************************************
                                                       ** TEST                            STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
                                                       ****************************************************************************************
                                                       ** tb_cocotb.simple_test            PASS       12045.00           0.18      67607.43  **
                                                       ****************************************************************************************
                                                       ** TESTS=1 PASS=1 FAIL=0 SKIP=0                12045.00           0.56      21622.11  **
                                                       ****************************************************************************************
```

$ make ex1-cocotb-gl

Runs GL simulation of your design using cocotb as the testbench environment.

The same testbench under ex1/tb/ is used with $GL set to 1.

For GL simulation, Icarus Verilog is used as the simulator.

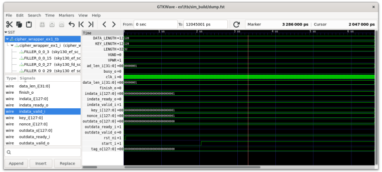


```
     0.00ns INFO     cocotb                             Running on Icarus Verilog version 13.0 (devel)
     0.00ns INFO     cocotb                             Running tests with cocotb v1.8.1 from /usr/local/lib/python3.10/dist-packages/cocotb
     0.00ns INFO     cocotb                             Seeding Python random module with 1707896614
/usr/lib/python3/dist-packages/_pytest/assertion/rewrite.py:170: UserWarning: Python runners and associated APIs are an experimental feature and
 subject to change.
  exec(co, module.__dict__)
     0.00ns INFO     cocotb.regression                  Found test tb_cocotb.simple_test
     0.00ns INFO     cocotb.regression                  running simple_test (1/1)
                                                          TODO Simple test for cipher_core

FST info: dumpfile dump.fst opened for output.
    50.00ns INFO     cocotb.rst_ni                      Reset complete
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       =================================
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       indata_ready_o:  0
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       outdata_o:       0
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       outdata_valid_o: 0
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       tag_o:           0
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       busy_o:          0
  1045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       finish_o:        0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       =================================
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       indata_ready_o:  0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       outdata_o:       0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       outdata_valid_o: 0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       tag_o:           0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       busy_o:          0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       finish_o:        0
 12045.00ns INFO     cocotb.cipher_wrapper_ex1_tb       Test finished!
 12045.00ns INFO     cocotb.regression                  simple_test passed
 12045.00ns INFO     cocotb.regression                  **************************************************************************************
                                                        ** TEST                            STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
                                                        **************************************************************************************
                                                        ** tb_cocotb.simple_test            PASS       12045.00           0.32      37922.16  **
                                                        **************************************************************************************
                                                        ** TESTS=1 PASS=1 FAIL=0 SKIP=0                12045.00           0.67      18036.63  **
                                                        **************************************************************************************
```

$ make ex1-gtkwave

After running RTL or GL simulation for exercise 1 the resulting waveform „dump.fst“ is saved under ex1/tb/sim_build.

This make target starts GTKWave with „dump.fst“ loaded.

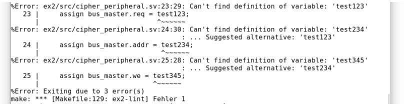


Makefile Targets - Exercise 2

$ make ex2-lint

Runs verilator in linting mode over your design to detect issues that can lead to problems.

Make sure to fix all warnings. Otherwise this might lead to problems later on.

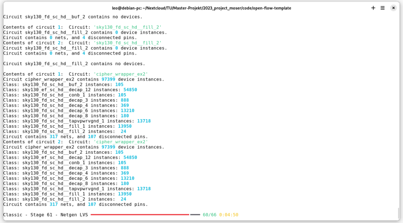


```
%Error: ex2/src/cipher_peripheral.sv:23:29: Can't find definition of variable: 'test123'
   23 |     assign bus_master.req = test123;
      |                             ^~~~~~~
%Error: ex2/src/cipher_peripheral.sv:24:30: Can't find definition of variable: 'test234'
                                          : ... Suggested alternative: 'test123'
   24 |     assign bus_master.addr = test234;
      |                              ^~~~~~~
%Error: ex2/src/cipher_peripheral.sv:25:28: Can't find definition of variable: 'test345'
                                          : ... Suggested alternative: 'test234'
   25 |     assign bus_master.we = test345;
      |                            ^~~~~~~
%Error: Exiting due to 3 error(s)
make: *** [Makefile:129: ex2-lint] Fehler 1
```

$ make ex2-openlane

Starts the physical process of hardening cipher_core using the sky130A PDK.

Actually, OpenLane 2 is used to harden your ciper. OpenLane 2 is the successor to OpenLane.



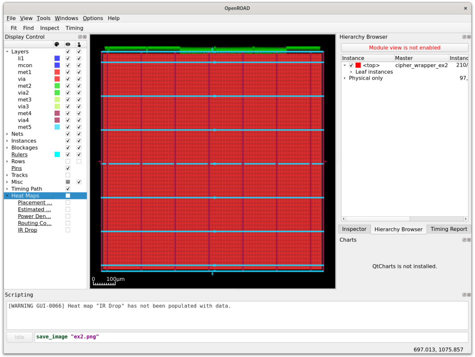
leo@debian-pc: ~/Nextcloud/TU/Master-Projekt/2023_project_moser/code/open-flow-template

```
Circuit sky130_fd_sc_hd__buf_2 contains no devices.

Contents of circuit 1:  Circuit: 'sky130_fd_sc_hd__fill_2'
Circuit sky130_fd_sc_hd__fill_2 contains 0 device instances.
Circuit contains 0 nets, and 4 disconnected pins.
Contents of circuit 2:  Circuit: 'sky130_fd_sc_hd__fill_2'
Circuit sky130_fd_sc_hd__fill_2 contains 0 device instances.
Circuit contains 0 nets, and 4 disconnected pins.

Circuit sky130_fd_sc_hd__fill_2 contains no devices.

Contents of circuit 1:  Circuit: 'cipher_wrapper_ex2'
Circuit cipher_wrapper_ex2 contains 97399 device instances.
Class: sky130_fd_sc_hd__buf_2 instances: 105
Class: sky130_ef_sc_hd__decap_12 instances: 54850
Class: sky130_fd_sc_hd__conb_1 instances: 105
Class: sky130_fd_sc_hd__decap_3 instances: 888
Class: sky130_fd_sc_hd__decap_4 instances: 369
Class: sky130_fd_sc_hd__decap_6 instances: 13210
Class: sky130_fd_sc_hd__decap_8 instances: 180
Class: sky130_fd_sc_hd__tapvpwrvgnd_1 instances: 13718
Class: sky130_fd_sc_hd__fill_1 instances: 13950
Class: sky130_fd_sc_hd__fill_2 instances:  24
Circuit contains 317 nets, and 107 disconnected pins.
Contents of circuit 2:  Circuit: 'cipher_wrapper_ex2'
Circuit cipher_wrapper_ex2 contains 97399 device instances.
Class: sky130_fd_sc_hd__buf_2 instances: 105
Class: sky130_ef_sc_hd__decap_12 instances: 54850
Class: sky130_fd_sc_hd__conb_1 instances: 105
Class: sky130_fd_sc_hd__decap_3 instances: 888
Class: sky130_fd_sc_hd__decap_4 instances: 369
Class: sky130_fd_sc_hd__decap_6 instances: 13210
Class: sky130_fd_sc_hd__decap_8 instances: 180
Class: sky130_fd_sc_hd__tapvpwrvgnd_1 instances: 13718
Class: sky130_fd_sc_hd__fill_1 instances: 13950
Class: sky130_fd_sc_hd__fill_2 instances:  24
Circuit contains 317 nets, and 107 disconnected pins.

Classic - Stage 61 - Netgen LVS ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60/66 0:04:50
```

$ make ex2-openroad

Loads the latest stage from „ex2-openlane“ into OpenROAD to visualize the design. This can be useful if the hardening process fails due to routing conqestion etc.

To save an image of your design execute „save_image image.png“ in the tcl command line.



```
[WARNING GUI-0066] Heat map "IR Drop" has not been populated with data.
```

```
save_image "ex2.png"
```

$ make ex2-klayout

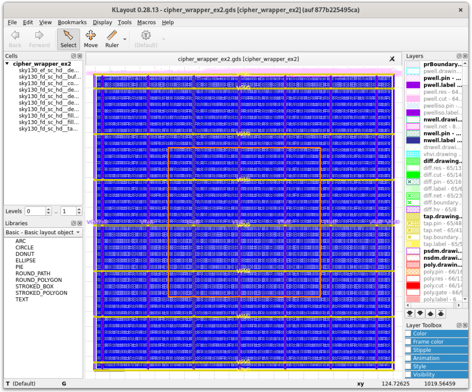
Opens the layout of your design from ex2/results in Klayout.

To save a high-resolution image, open:

Macros → Macro Development

Then execute in the console:

```
RBA::Application.instance
.main_window.current_view
.save_image("image.png",2000,2000)
```

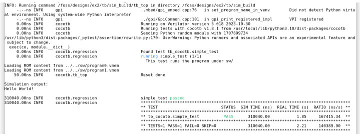

KLayout 0.28.13 - cipher_wrapper_ex2.gds [cipher_wrapper_ex2] (auf 877b225495ca)
File Edit View Bookmarks Display Tools Macros Help
Back Forward Select Move Ruler (Default)
Cells
cipher_wrapper_ex2
cipher_wrapper_ex2.gds [cipher_wrapper_ex2]
Layers
Levels 0 .. 1
Libraries
Basic - Basic layout object
ARC
CIRCLE
DONUT
ELLIPSE
PIE
ROUND_PATH
ROUND_POLYGON
STROKED_BOX
STROKED_POLYGON
TEXT
Layer Toolbox
Color
Frame color
Stipple
Animation
Style
Visibility
T (Default) G
xy 124.72625 1019.56459

$ make ex2-cocotb

Runs RTL simulation of your design using cocotb as the testbench environment.

You have to write your own program under /ex2/sw/. Set $PROGRAM to the active program e.g. hello-world.

For RTL simulation, Verilator is used as the simulator.



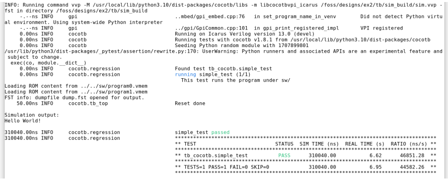
```
INFO: Running command /foss/designs/ex2/tb/sim_build/tb_top in directory /foss/designs/ex2/tb/sim_build
     -.--ns INFO     gpi                                ..mbed/gpi_embed.cpp:76   in set_program_name_in_venv        Did not detect Python virtu
al environment. Using system-wide Python interpreter
     -.--ns INFO     gpi                                ../gpi/GpiCommon.cpp:101  in gpi_print_registered_impl       VPI registered
     0.00ns INFO     cocotb                             Running on Verilator version 5.018 2023-10-30
     0.00ns INFO     cocotb                             Running tests with cocotb v1.8.1 from /usr/local/lib/python3.10/dist-packages/cocotb
     0.00ns INFO     cocotb                             Seeding Python random module with 1707899734
/usr/lib/python3/dist-packages/_pytest/assertion/rewrite.py:170: UserWarning: Python runners and associated APIs are an experimental feature and
 subject to change.
  exec(co, module.__dict__)
     0.00ns INFO     cocotb.regression                  Found test tb_cocotb.simple_test
     0.00ns INFO     cocotb.regression                  running simple_test (1/1)
                                                          This test runs the program under sw/

Loading ROM content from ../../sw/program0.vmem
Loading ROM content from ../../sw/program1.vmem
    50.00ns INFO     cocotb.tb_top                      Reset done

Simulation output:
Hello World!

310040.00ns INFO     cocotb.regression                  simple_test passed
310040.00ns INFO     cocotb.regression                  ************************************************************************************
                                                        ** TEST                          STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
                                                        ************************************************************************************
                                                        ** tb_cocotb.simple_test          PASS      310040.00           1.85      167415.34  **
                                                        ************************************************************************************
                                                        ** TESTS=1 PASS=1 FAIL=0 SKIP=0             310040.00           2.21      140389.90  **
                                                        ************************************************************************************
```

$ make ex2-cocotb-gl

Runs GL simulation of your design using cocotb as the testbench environment.

Same as ex2-cocotb, but $GL is set to 1 and the gate level files for cipher_peripheral are used.

For GL simulation, Icarus Verilog is used as the simulator.

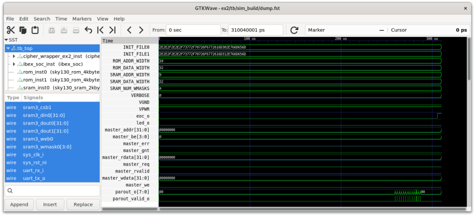


```
INFO: Running command vvp -M /usr/local/lib/python3.10/dist-packages/cocotb/libs -m libcocotbvpi_icarus /foss/designs/ex2/tb/sim_build/sim.vvp -
fst in directory /foss/designs/ex2/tb/sim_build
     -.--ns INFO     gpi                                ..mbed/gpi_embed.cpp:76   in set_program_name_in_venv        Did not detect Python virtu
al environment. Using system-wide Python interpreter
     -.--ns INFO     gpi                                ../gpi/GpiCommon.cpp:101  in gpi_print_registered_impl       VPI registered
     0.00ns INFO     cocotb                             Running on Icarus Verilog version 13.0 (devel)
     0.00ns INFO     cocotb                             Running tests with cocotb v1.8.1 from /usr/local/lib/python3.10/dist-packages/cocotb
     0.00ns INFO     cocotb                             Seeding Python random module with 1707899801
/usr/lib/python3/dist-packages/_pytest/assertion/rewrite.py:170: UserWarning: Python runners and associated APIs are an experimental feature and
 subject to change.
  exec(co, module.__dict__)
     0.00ns INFO     cocotb.regression                  Found test tb_cocotb.simple_test
     0.00ns INFO     cocotb.regression                  running simple_test (1/1)
                                                          This test runs the program under sw/

Loading ROM content from ../../sw/program0.vmem
Loading ROM content from ../../sw/program1.vmem
FST info: dumpfile dump.fst opened for output.
    50.00ns INFO     cocotb.tb_top                      Reset done

Simulation output:
Hello World!

310040.00ns INFO     cocotb.regression                  simple_test passed
310040.00ns INFO     cocotb.regression                  **************************************************************************************
                                                        ** TEST                          STATUS  SIM TIME (ns)  REAL TIME (s)  RATIO (ns/s) **
                                                        **************************************************************************************
                                                        ** tb_cocotb.simple_test          PASS      310040.00           6.62      46851.28  **
                                                        **************************************************************************************
                                                        ** TESTS=1 PASS=1 FAIL=0 SKIP=0             310040.00           6.95      44582.26  **
                                                        **************************************************************************************
```

$ make ex2-gtkwave

After running RTL or GL simulation for exercise 2 the resulting waveform „dump.fst“ is saved under ex2/tb/sim_build.

This make target starts GTKWave with „dump.fst“ loaded.



```
tb_top
  cipher_wrapper_ex2_inst  (ciphe
  ibex_soc_inst  (ibex_soc)
  rom_inst0  (sky130_rom_4kbyte
  rom_inst1  (sky130_rom_4kbyte
  sram_inst0  (sky130_sram_2kby

wire   sram3_csb1
wire   sram3_din0[31:0]
wire   sram3_dout0[31:0]
wire   sram3_dout1[31:0]
wire   sram3_web0
wire   sram3_wmask0[3:0]
wire   sys_clk_i
wire   sys_rst_ni
wire   uart_rx_i
wire   uart_tx_o
```

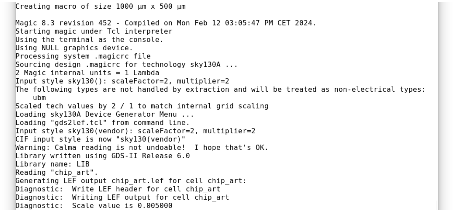
```
Time
         INIT_FILE0  2E2E2F2E2E2F73772F70726F6772616D302E766D656D
         INIT_FILE1  2E2E2F2E2E2F73772F70726F6772616D312E766D656D
     ROM_ADDR_WIDTH  10
     ROM_DATA_WIDTH  32
    SRAM_ADDR_WIDTH  9
    SRAM_DATA_WIDTH  32
    SRAM_NUM_WMASKS  4
            VERBOSE  0
               VGND
               VPWR
              eoc_o
              led_o
  master_addr[31:0]  00000000
    master_be[3:0]  0
         master_err
         master_gnt
 master_rdata[31:0]  00000000
         master_req
      master_rvalid
 master_wdata[31:0]  00000000
          master_we
      parout_o[7:0]  00                                    00
     parout_valid_o
```

Makefile Targets - Tapeout

$ make tapeout-chip_art

Starts a Klayout Python script to convert the PNG image under tapeout/chip_art/chip_art.png to .gds. A tcl script for Magic creates a .lef file.

Feel free to change the image, but make sure to keep the resolution.

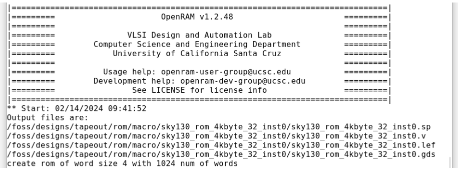


```
Creating macro of size 1000 µm x 500 µm

Magic 8.3 revision 452 - Compiled on Mon Feb 12 03:05:47 PM CET 2024.
Starting magic under Tcl interpreter
Using the terminal as the console.
Using NULL graphics device.
Processing system .magicrc file
Sourcing design .magicrc for technology sky130A ...
2 Magic internal units = 1 Lambda
Input style sky130(): scaleFactor=2, multiplier=2
The following types are not handled by extraction and will be treated as non-electrical types:
    ubm
Scaled tech values by 2 / 1 to match internal grid scaling
Loading sky130A Device Generator Menu ...
Loading "gds2lef.tcl" from command line.
Input style sky130(vendor): scaleFactor=2, multiplier=2
CIF input style is now "sky130(vendor)"
Warning: Calma reading is not undoable!  I hope that's OK.
Library written using GDS-II Release 6.0
Library name: LIB
Reading "chip_art".
Generating LEF output chip_art.lef for cell chip_art:
Diagnostic:  Write LEF header for cell chip_art
Diagnostic:  Writing LEF output for cell chip_art
Diagnostic:  Scale value is 0.005000
```

$ make tapeout-rom

Starts OpenRAM (or you could say OpenROM) to create a ROM macro for your program.

Set $PROGRAM to the active program e.g. hello-world.



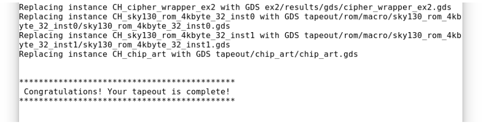
```
|==============================================================================|
|=========                      OpenRAM v1.2.48                      =========|
|=========                                                           =========|
|=========                VLSI Design and Automation Lab             =========|
|=========          Computer Science and Engineering Department      =========|
|=========             University of California Santa Cruz           =========|
|=========                                                           =========|
|=========           Usage help: openram-user-group@ucsc.edu         =========|
|=========          Development help: openram-dev-group@ucsc.edu     =========|
|=========                See LICENSE for license info               =========|
|==============================================================================|
** Start: 02/14/2024 09:41:52
Output files are:
/foss/designs/tapeout/rom/macro/sky130_rom_4kbyte_32_inst0/sky130_rom_4kbyte_32_inst0.sp
/foss/designs/tapeout/rom/macro/sky130_rom_4kbyte_32_inst0/sky130_rom_4kbyte_32_inst0.v
/foss/designs/tapeout/rom/macro/sky130_rom_4kbyte_32_inst0/sky130_rom_4kbyte_32_inst0.lef
/foss/designs/tapeout/rom/macro/sky130_rom_4kbyte_32_inst0/sky130_rom_4kbyte_32_inst0.gds
create rom of word size 4 with 1024 num of words
```

$ make tapeout-final

The final step to complete tapeout!

This target calls a Klayout Python script to merge the layouts of your cipher, the ROMs and the chip_art with the pre-hardened chip.

Congratulations, your design is finished!



```
Replacing instance CH_cipher_wrapper_ex2 with GDS ex2/results/gds/cipher_wrapper_ex2.gds
Replacing instance CH_sky130_rom_4kbyte_32_inst0 with GDS tapeout/rom/macro/sky130_rom_4kb
yte_32_inst0/sky130_rom_4kbyte_32_inst0.gds
Replacing instance CH_sky130_rom_4kbyte_32_inst1 with GDS tapeout/rom/macro/sky130_rom_4kb
yte_32_inst1/sky130_rom_4kbyte_32_inst1.gds
Replacing instance CH_chip_art with GDS tapeout/chip_art/chip_art.gds


********************************************
 Congratulations! Your tapeout is complete!
********************************************
```

$ make tapeout-klayout

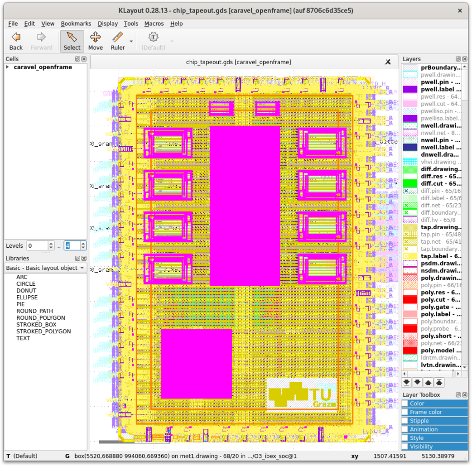
Opens the layout of your design chip_tapeout.gds in Klayout.

To save a high-resolution image, open:

Macros → Macro Development

Then execute in the console:

```
RBA::Application.instance
 .main_window.current_view
 .save_image("image.png",2000,2000)
```


KLayout 0.28.13 - chip_tapeout.gds [caravel_openframe] (auf 8706c6d35ce5)
File Edit View Bookmarks Display Tools Macros Help
Back Forward Select Move Ruler (Default)
Cells
caravel_openframe
chip_tapeout.gds [caravel_openframe]
Layers
Levels 0 .. 4
Libraries
Basic - Basic layout object
ARC
CIRCLE
DONUT
ELLIPSE
PIE
ROUND_PATH
ROUND_POLYGON
STROKED_BOX
STROKED_POLYGON
TEXT
Layer Toolbox
Color
Frame color
Stipple
Animation
Style
Visibility
TU Graz
box(5520,668880 994060,669360) on met1.drawing - 68/20 in .../O3_ibex_soc@1
xy 1507.41591 5130.38979

Open Source Tools

Verilator

Verilator is a free and open-source software tool which converts Verilog to a cycle-accurate behavioral model in C++ or SystemC. Verilator is the fastest Verilog/SystemVerilog simulator.

Website: `https://www.veripool.org/verilator/`

Icarus Verilog

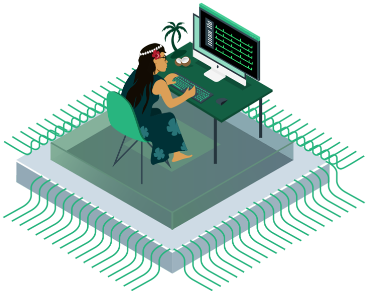
Icarus Verilog is an implementation of the Verilog hardware description language compiler that generates netlists in the desired format. It supports the 1995, 2001 and 2005 versions of the standard, portions of SystemVerilog, and some extensions.

Repository: `https://github.com/steveicarus/iverilog`

cocotb

cocotb is an open source coroutine-based cosimulation testbench environment for verifying VHDL and SystemVerilog RTL using Python.

Website: `https://www.cocotb.org/`



GTKWave

GTKWave is an open source waveform viewer and can read various formats such as fst and vcd files.

Repository:

`https://github.com/gtkwave/gtkwave`

OpenROAD

OpenROAD's unified application implementing an RTL-to-GDS Flow.

Repository: `https://github.com/The-OpenROAD-Project/OpenROAD`

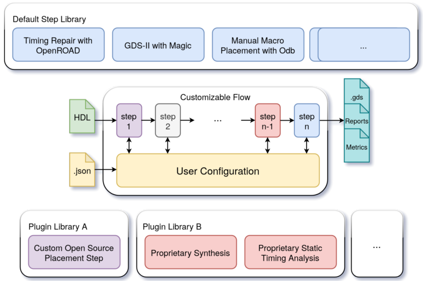
Documentation: `https://openroad.readthedocs.io/en/latest/`

OpenLane 2

The next generation of OpenLane, rewritten from scratch with a modular architecture

Repository: `https://github.com/efabless/openlane2`

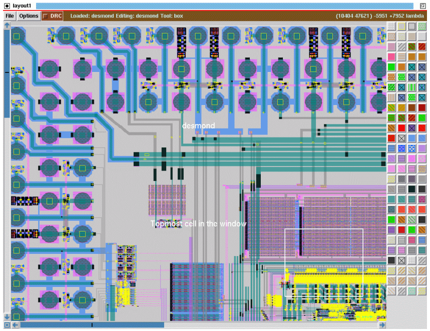
Documentation: `https://openlane2.readthedocs.io/en/latest/`



Magic VLSI Layout Tool

Magic is a venerable VLSI layout tool, written in the 1980's at Berkeley by John Ousterhout. With well thought-out core algorithms, Magic is a powerful yet simple tool for circuit layout and validation.

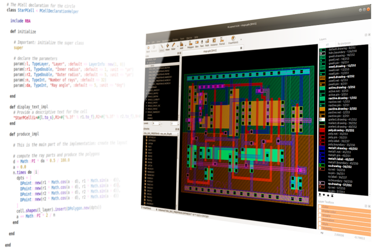
Repository: `https://github.com/RTimothyEdwards/magic`


layout1
File
Options
DRC
Loaded: desmond Editing: desmond Tool: box
(10404 47621) -5951 +7952 lambda
desmond
Topmost cell in the window

KLayout

KLayout chip mask layout viewing, editing and more. It provides an extensive Ruby and Python API.

Website: `https://www.klayout.de/`



```
# The PCell declaration for the circle
class StarPCell < PCellDeclarationHelper

  include RBA

  def initialize

    # Important: initialize the super class
    super

    # declare the parameters
    param(:l, TypeLayer, "Layer", :default => LayerInfo::new(1, 0))
    param(:r1, TypeDouble, "Inner radius", :default => 1, :unit => "µm")
    param(:r2, TypeDouble, "Outer radius", :default => 5, :unit => "µm")
    param(:n, TypeInt, "Number of rays", :default => 32)
    param(:da, TypeInt, "Ray angle", :default => 5, :unit => "deg")

  end

  def display_text_impl
    # Provide a descriptive text for the cell
    "StarPCell(L=#{l.to_s},R1=#{'%.3f' % r1.to_f},R2=#{'%.3f' % r2.to_f},N=#{
  end

  def produce_impl

    # This is the main part of the implementation: create the layout

    # compute the ray parts and produce the polygons
    d = Math::PI * da * 0.5 / 180.0
    a = 0.0
    n.times do |i|
      dpts = [
        DPoint::new(r1 * Math.cos(a - d), r1 * Math.sin(a - d)),
        DPoint::new(r1 * Math.cos(a + d), r1 * Math.sin(a + d)),
        DPoint::new(r2 * Math.cos(a + d), r2 * Math.sin(a + d)),
        DPoint::new(r2 * Math.cos(a - d), r2 * Math.sin(a - d))
      ]
      cell.shapes(l_layer).insert(DPolygon.new(dpts))
      a += Math::PI * 2 / n
    end

  end

end
```

OpenRAM

An open-source static random access memory (SRAM) compiler.

Website: `https://openram.org/`

Repository:

`https://github.com/VLSIDA/OpenRAM`

Further Links

- *IAIK Open Flow*
- *Skywater SKY130 PDK*
- *GlobalFoundries GF180MCU PDK*
- *IHP SG13G2 PDK*
- *Qflow*
- *Coriolis*
- *LunaPnR*
- *Place & route on silicon*
- *Tiny Tapeout*

Digital System Design
IAIK Open Flow

DSD Team

06.03.2024

Digital System Design
Graz University of Technology