UNIC-CASS Mentoring Session 02

Introduction to the Digital Design Flow

Rodrigo N. Wuerdig
Joint PhD Student @ UFRGS (Brazil) and KU Leuven (Belgium)

August 28th, 2024

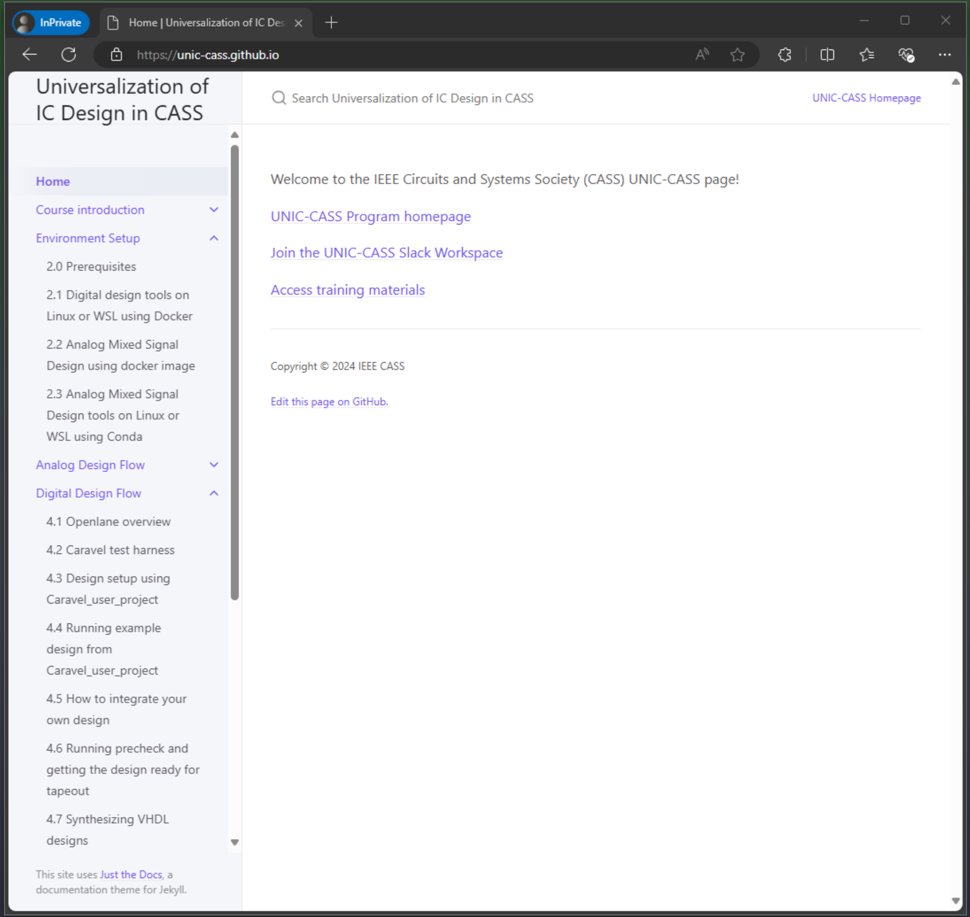
CAS
IEEE CIRCUITS AND SYSTEMS SOCIETY

IEEE

UNIC-CASS Page

Do not miss the educational material at

https://unic-cass.github.io/



InPrivate
Home | Universalization of IC Des
https://unic-cass.github.io

Universalization of IC Design in CASS

Search Universalization of IC Design in CASS

UNIC-CASS Homepage

- Home
- Course introduction
- Environment Setup
  - 2.0 Prerequisites
  - 2.1 Digital design tools on Linux or WSL using Docker
  - 2.2 Analog Mixed Signal Design using docker image
  - 2.3 Analog Mixed Signal Design tools on Linux or WSL using Conda
- Analog Design Flow
- Digital Design Flow
  - 4.1 Openlane overview
  - 4.2 Caravel test harness
  - 4.3 Design setup using Caravel_user_project
  - 4.4 Running example design from Caravel_user_project
  - 4.5 How to integrate your own design
  - 4.6 Running precheck and getting the design ready for tapeout
  - 4.7 Synthesizing VHDL designs

This site uses Just the Docs, a documentation theme for Jekyll.

Welcome to the IEEE Circuits and Systems Society (CASS) UNIC-CASS page!

UNIC-CASS Program homepage

Join the UNIC-CASS Slack Workspace

Access training materials

Copyright © 2024 IEEE CASS

Edit this page on GitHub.

Join the Open-EDA Community

UNIC-CASS Slack

https://join.slack.com/t/unic-cass/shared_invite/zt-1xxifr0ow-n8dpt0qNBxb4J50g8MEvmw

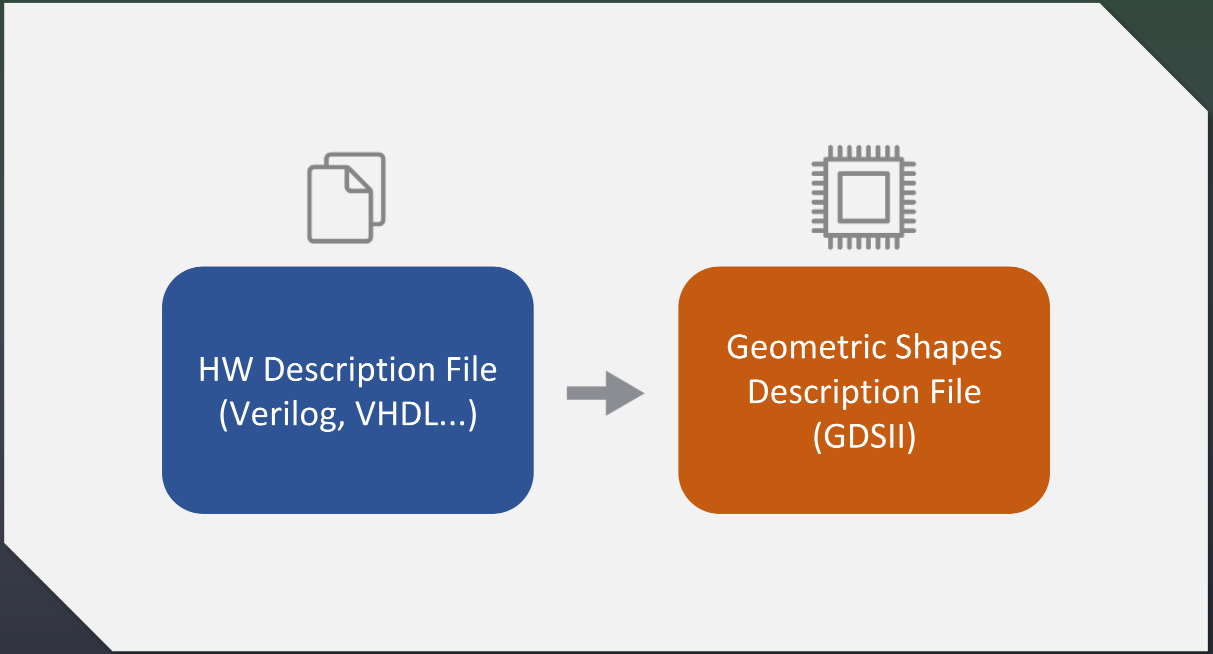
open-source-silicon Slack

https://join.slack.com/t/open-source-silicon/shared_invite/zt-1zopfd1gk-uI2eSlNXB54xN9RCowGa4g

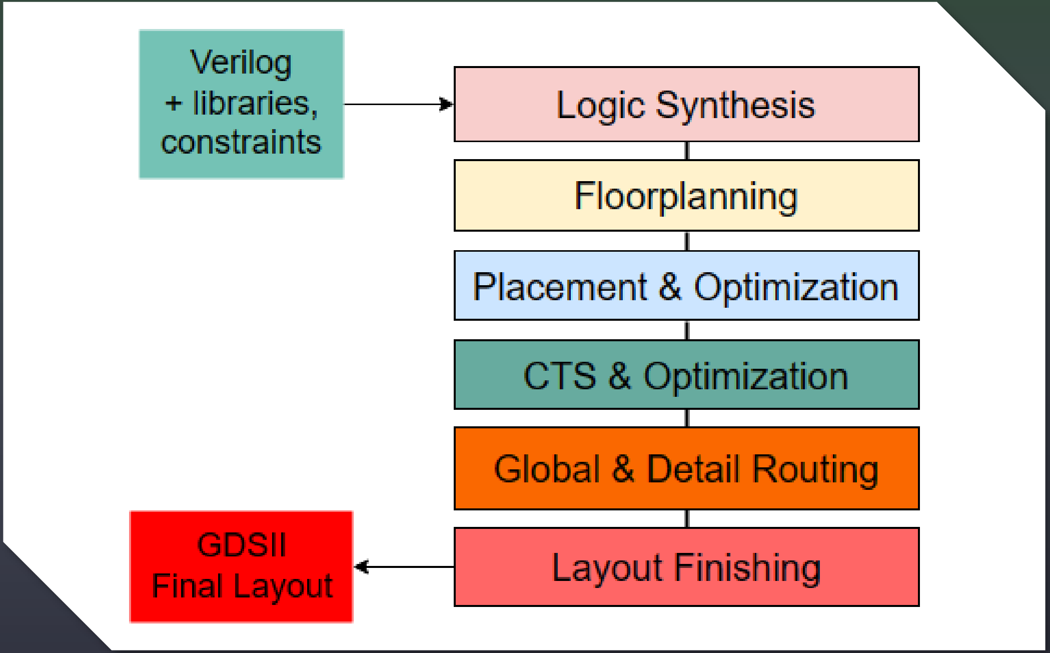
Outline

1. Introduction to the RTL2GDSII Flow
    - Logical Synthesis
    - Physical Synthesis
2. Open-PDK Ecosystem
3. Open-EDA Ecosystem
4. Caravel
5. How the top-level integration works

RTL2GDSII


HW Description File
(Verilog, VHDL...)
Geometric Shapes Description File
(GDSII)

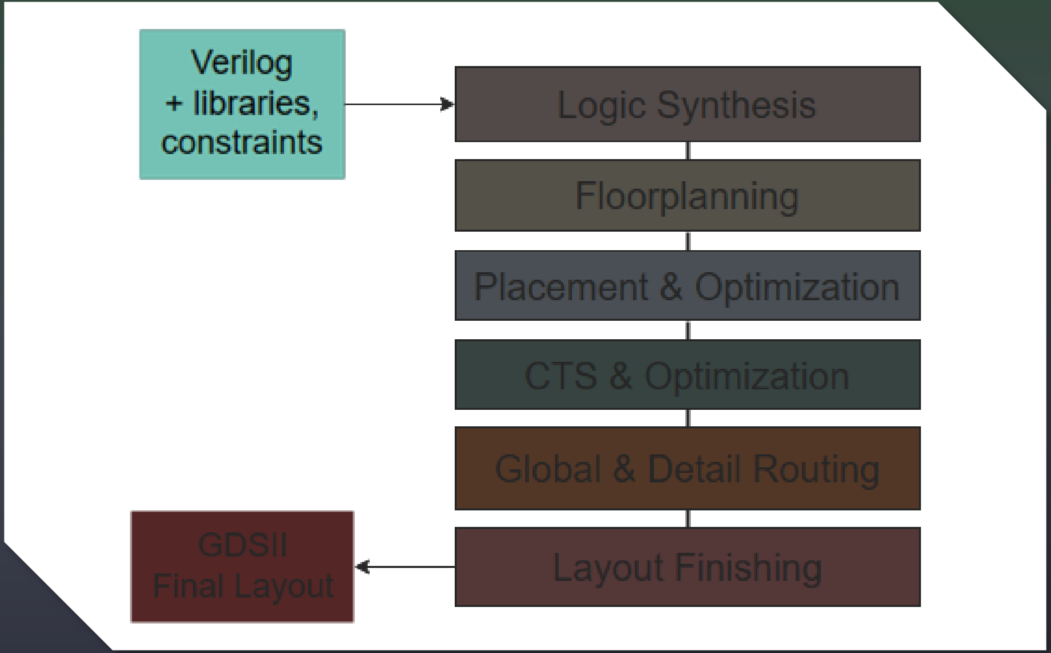
RTL2GDSII



Source: https://openroad.readthedocs.io/en/latest/main/README.html

RTL2GDSII

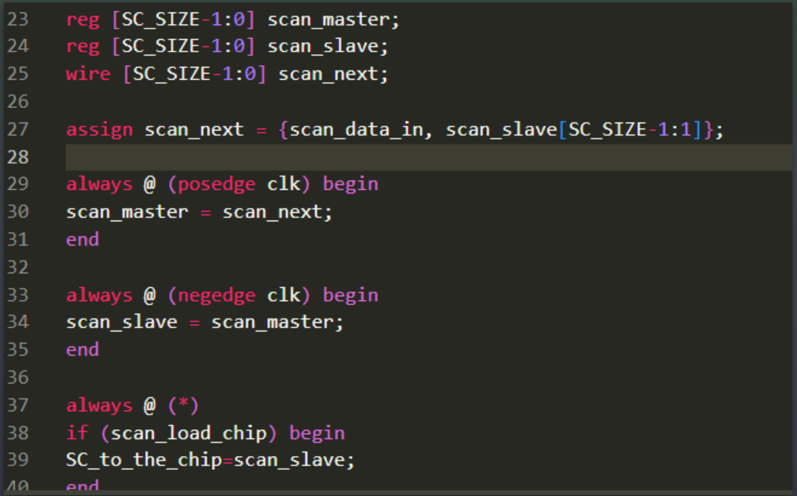
If using VHDL, follow section 4.7:
https://unic-cass.github.io/training/4.7-synthesis-vhdl-design.html



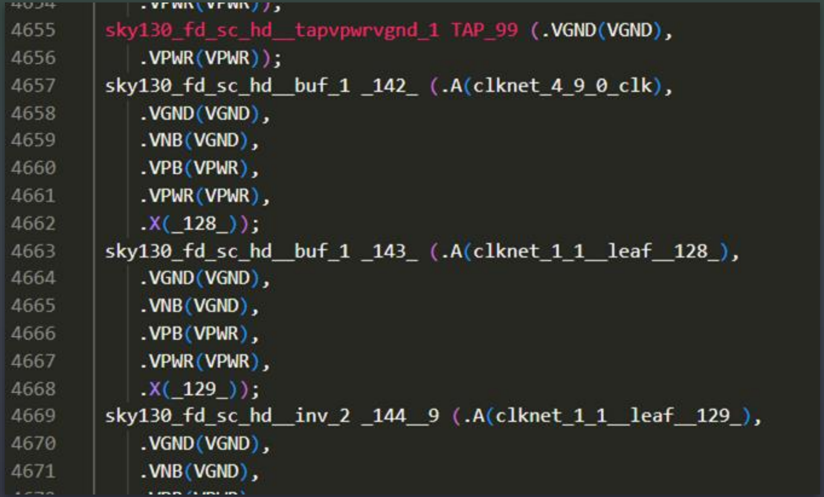
Source: https://openroad.readthedocs.io/en/latest/main/README.html

Logic Synthesis

Transform a **HW Description** into a “mapped description” / netlist.



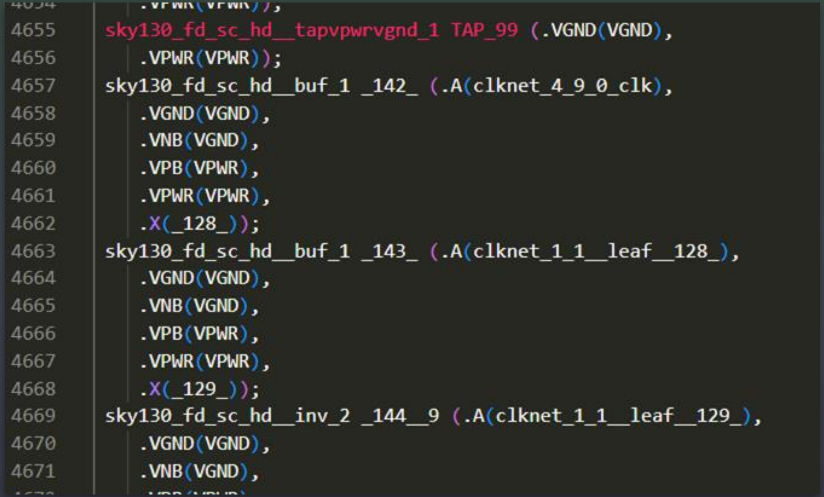
```
reg [SC_SIZE-1:0] scan_master;
reg [SC_SIZE-1:0] scan_slave;
wire [SC_SIZE-1:0] scan_next;

assign scan_next = {scan_data_in, scan_slave[SC_SIZE-1:1]};

always @ (posedge clk) begin
scan_master = scan_next;
end

always @ (negedge clk) begin
scan_slave = scan_master;
end

always @ (*)
if (scan_load_chip) begin
SC_to_the_chip=scan_slave;
end
```



```
    .VPWR(VPWR));
sky130_fd_sc_hd__tapvpwrvgnd_1 TAP_99 (.VGND(VGND),
    .VPWR(VPWR));
sky130_fd_sc_hd__buf_1 _142_ (.A(clknet_4_9_0_clk),
    .VGND(VGND),
    .VNB(VGND),
    .VPB(VPWR),
    .VPWR(VPWR),
    .X(_128_));
sky130_fd_sc_hd__buf_1 _143_ (.A(clknet_1_1__leaf__128_),
    .VGND(VGND),
    .VNB(VGND),
    .VPB(VPWR),
    .VPWR(VPWR),
    .X(_129_));
sky130_fd_sc_hd__inv_2 _144__9 (.A(clknet_1_1__leaf__129_),
    .VGND(VGND),
    .VNB(VGND),
    .VPB(VPWR),
```

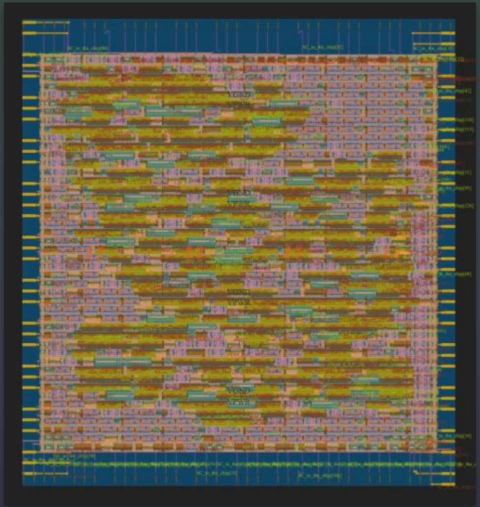
Physical Synthesis

Transform the mapped netlist into a geometric shapes description (<*.gds>).



```
    .VPWR(VPWR));
sky130_fd_sc_hd__tapvpwrvgnd_1 TAP_99 (.VGND(VGND),
    .VPWR(VPWR));
sky130_fd_sc_hd__buf_1 _142_ (.A(clknet_4_9_0_clk),
    .VGND(VGND),
    .VNB(VGND),
    .VPB(VPWR),
    .VPWR(VPWR),
    .X(_128_));
sky130_fd_sc_hd__buf_1 _143_ (.A(clknet_1_1__leaf__128_),
    .VGND(VGND),
    .VNB(VGND),
    .VPB(VPWR),
    .VPWR(VPWR),
    .X(_129_));
sky130_fd_sc_hd__inv_2 _144__9 (.A(clknet_1_1__leaf__129_),
    .VGND(VGND),
    .VNB(VGND),
```

→



Open-PDK Ecosystem

Open PDKs

- GlobalFoundries 180
- Skywater 130
- IHP 130 BiCMOS

alpha release:
- SKY90-FD

Predictive PDKs

- ASAP 7
- PTM-MG
- FreePDK45
- …

Open-PDK Ecosystem

Open PDKs

- GlobalFoundries 180
- Skywater 130
- IHP 130 BiCMOS

alpha release:
- SKY90-FD

Predictive PDKs

- ASAP 7
- PTM-MG
- FreePDK45
- …

Open-EDA Ecosystem

70+ open/free tools for IC Design

Source: https://semiwiki.com/wikis/industry-wikis/eda-open-source-tools-wiki/

Frameworks / Software Wrappers

OpenROAD Flow Scripts

OpenLane

SiliconCompiler

…

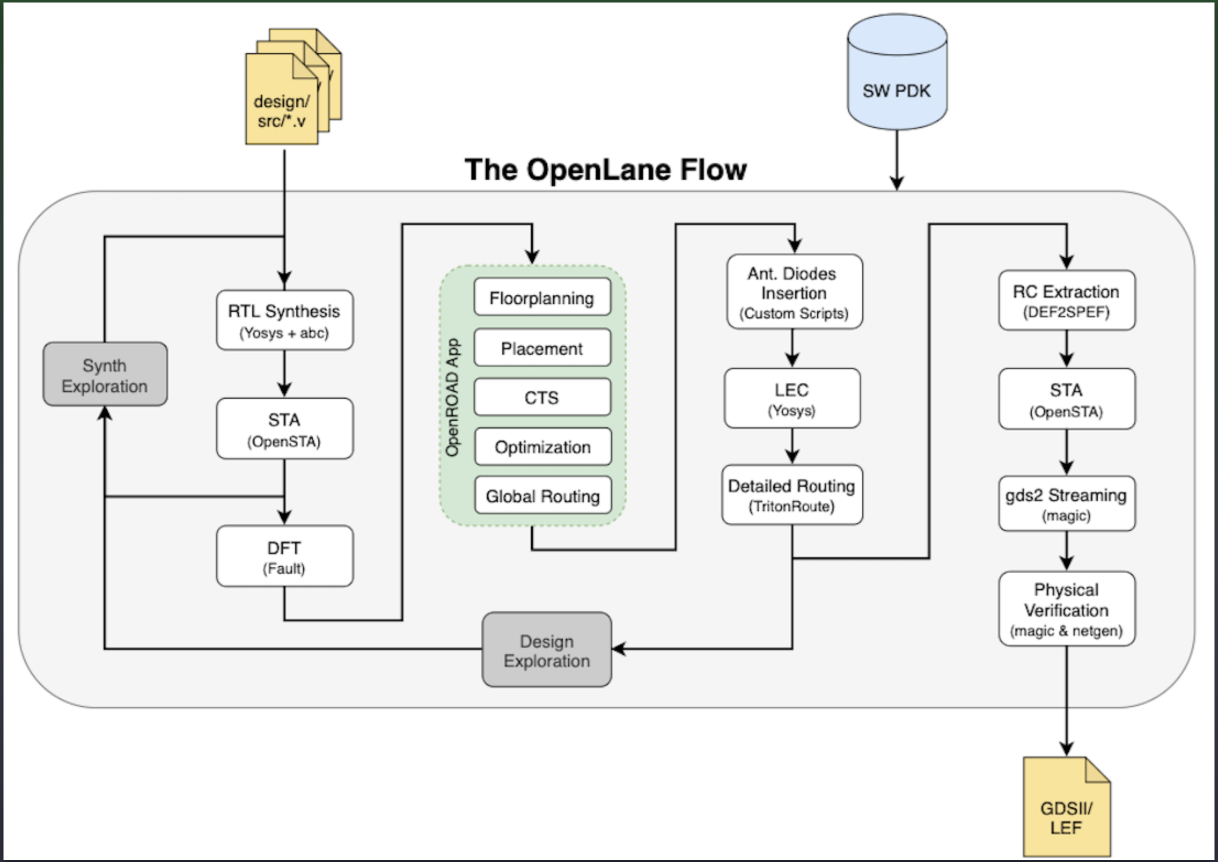
Frameworks / Software Wrappers

OpenROAD Flow Scripts

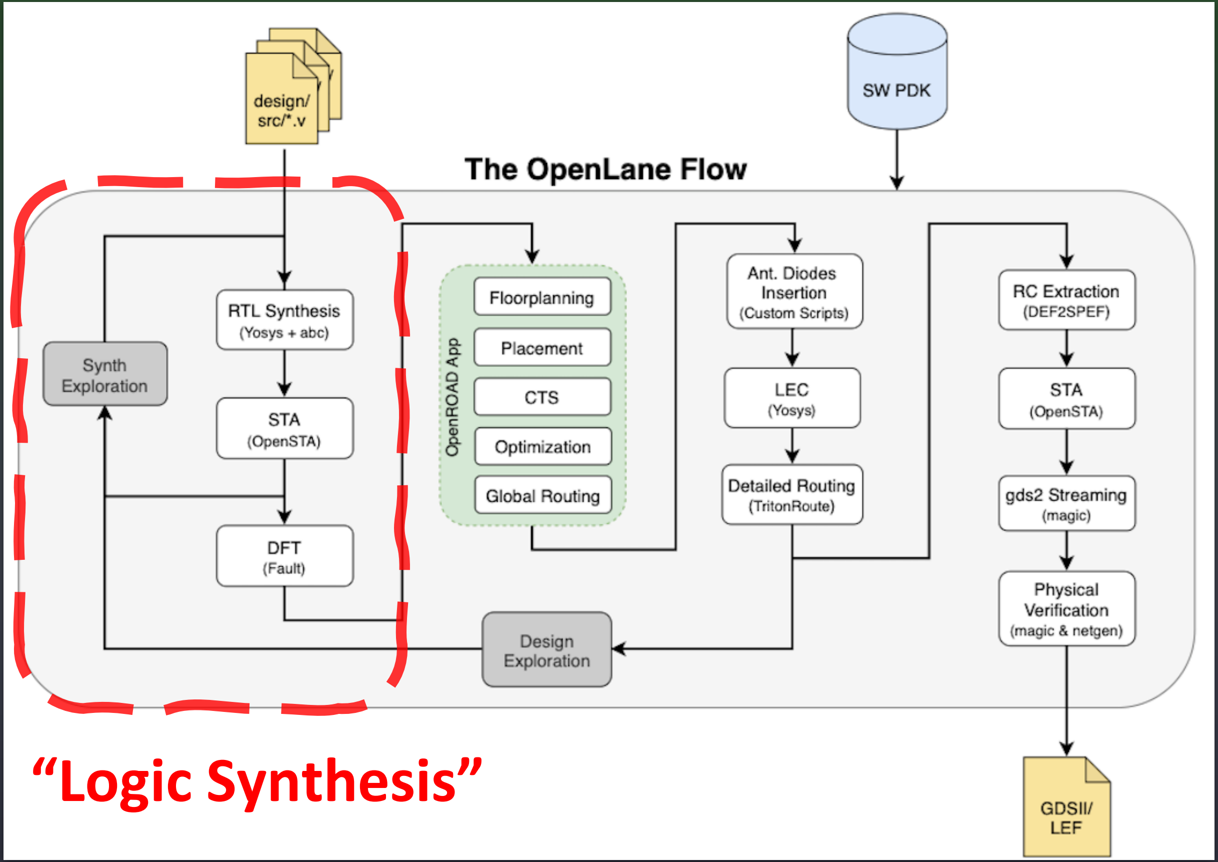
OpenLane

SiliconCompiler

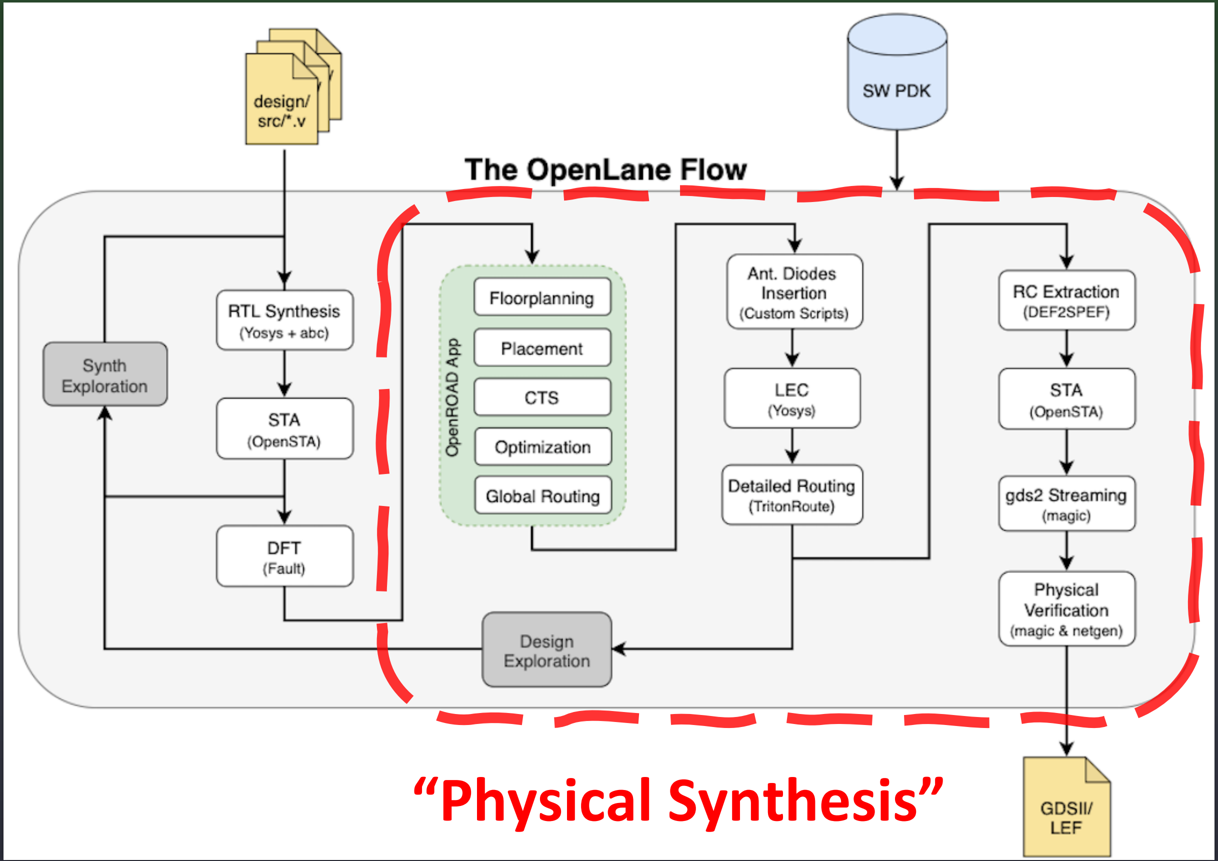
...



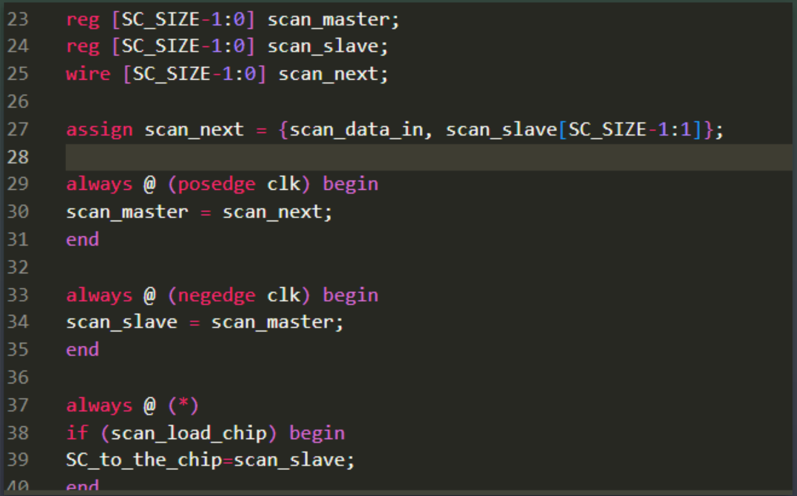
Source: https://openlane.readthedocs.io/en/latest/flow_overview.html



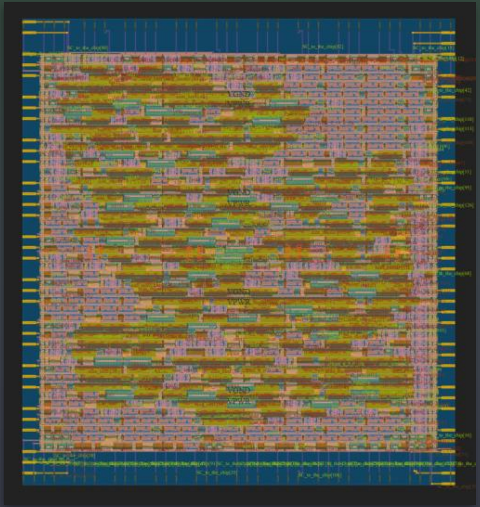
Source: https://openlane.readthedocs.io/en/latest/flow_overview.html



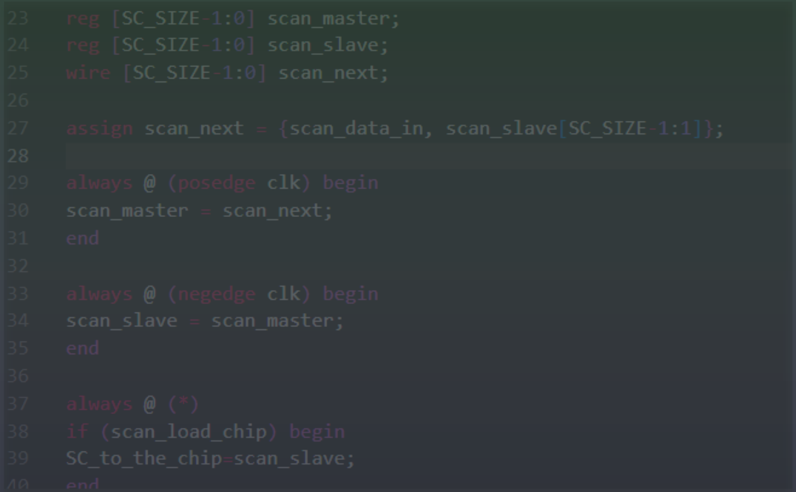
Source: https://openlane.readthedocs.io/en/latest/flow_overview.html



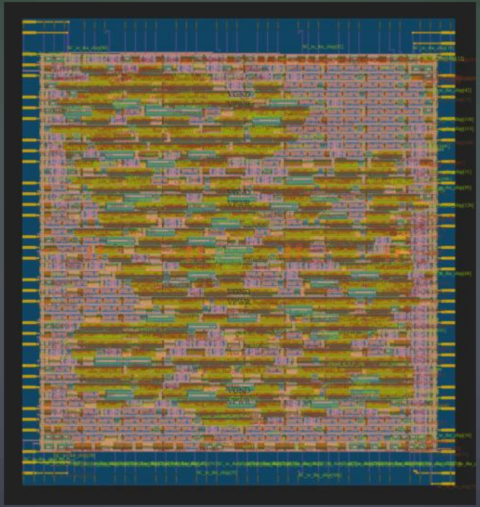
```
reg [SC_SIZE-1:0] scan_master;
reg [SC_SIZE-1:0] scan_slave;
wire [SC_SIZE-1:0] scan_next;

assign scan_next = {scan_data_in, scan_slave[SC_SIZE-1:1]};

always @ (posedge clk) begin
scan_master = scan_next;
end

always @ (negedge clk) begin
scan_slave = scan_master;
end

always @ (*)
if (scan_load_chip) begin
SC_to_the_chip=scan_slave;
end
```



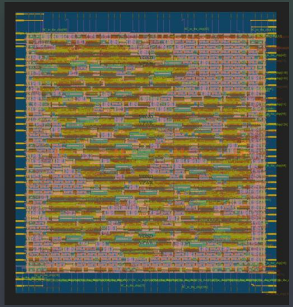
Ok! We can synthesize a block using OpenLane. But this block lacks context

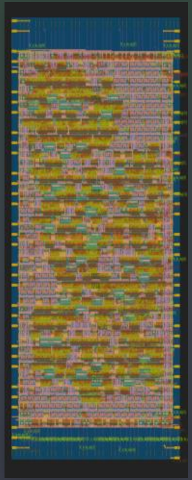


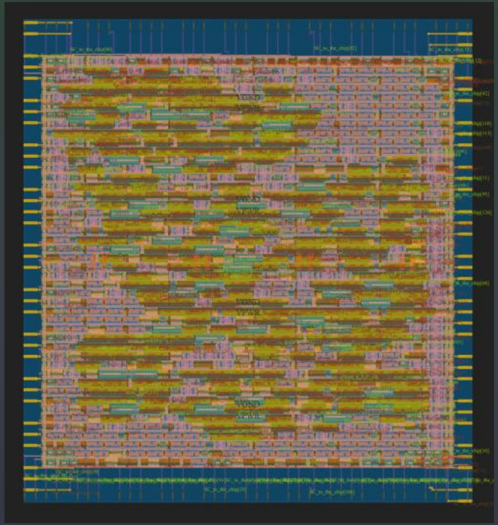
```
reg [SC_SIZE-1:0] scan_master;
reg [SC_SIZE-1:0] scan_slave;
wire [SC_SIZE-1:0] scan_next;

assign scan_next = {scan_data_in, scan_slave[SC_SIZE-1:1]};

always @ (posedge clk) begin
scan_master = scan_next;
end

always @ (negedge clk) begin
scan_slave = scan_master;
end

always @ (*)
if (scan_load_chip) begin
SC_to_the_chip=scan_slave;
end
```



OpenLane gives us freedom for making our block with several shapes and sizes

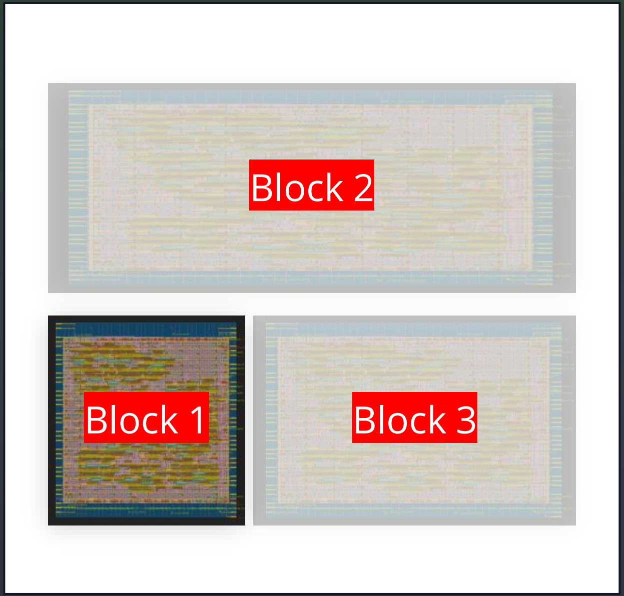




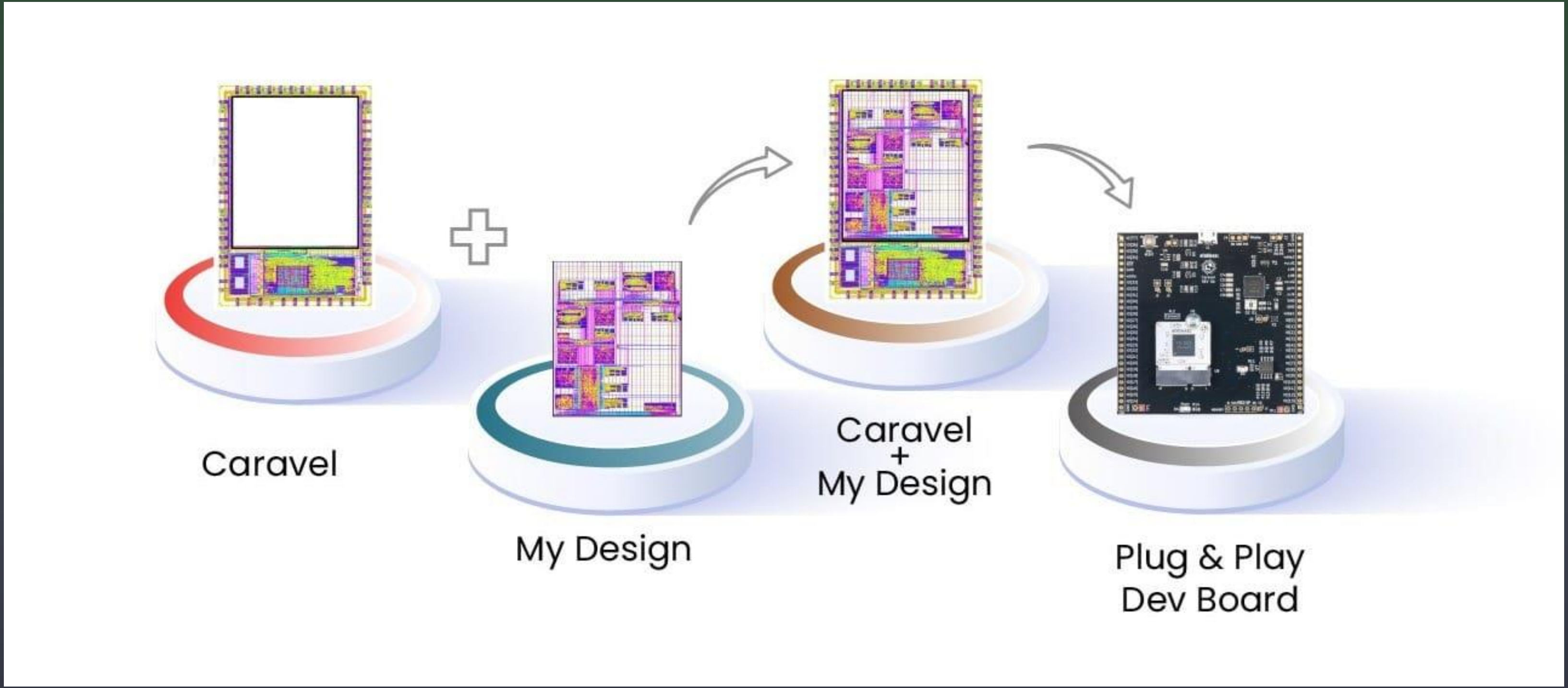


But in a real application (ic) this block will be placed under some context

position, size, pin-connection…



Our context is placing the multiple blocks under the Caravel Wrapper

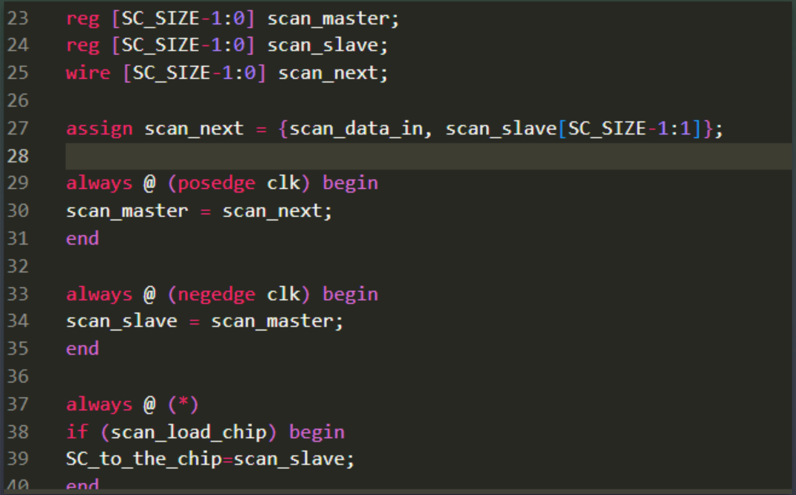


Source: https://efabless.com/chipignite

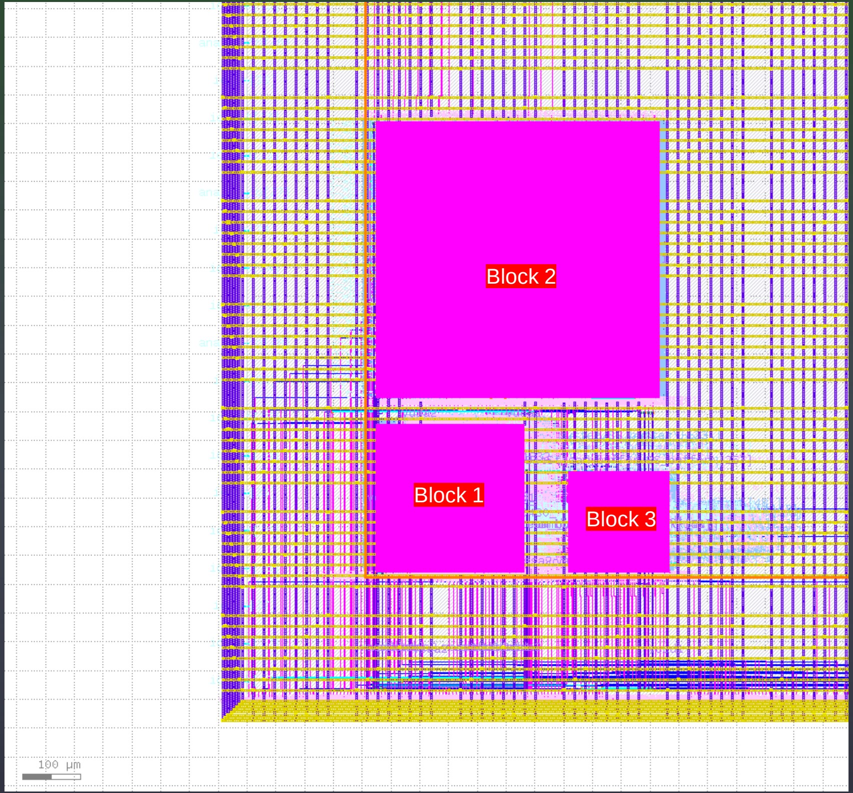
So it is important to try both:

1. The standard OpenLane block synthesis https://unic-cass.github.io/training/2.1-digital-design-tool-docker.html

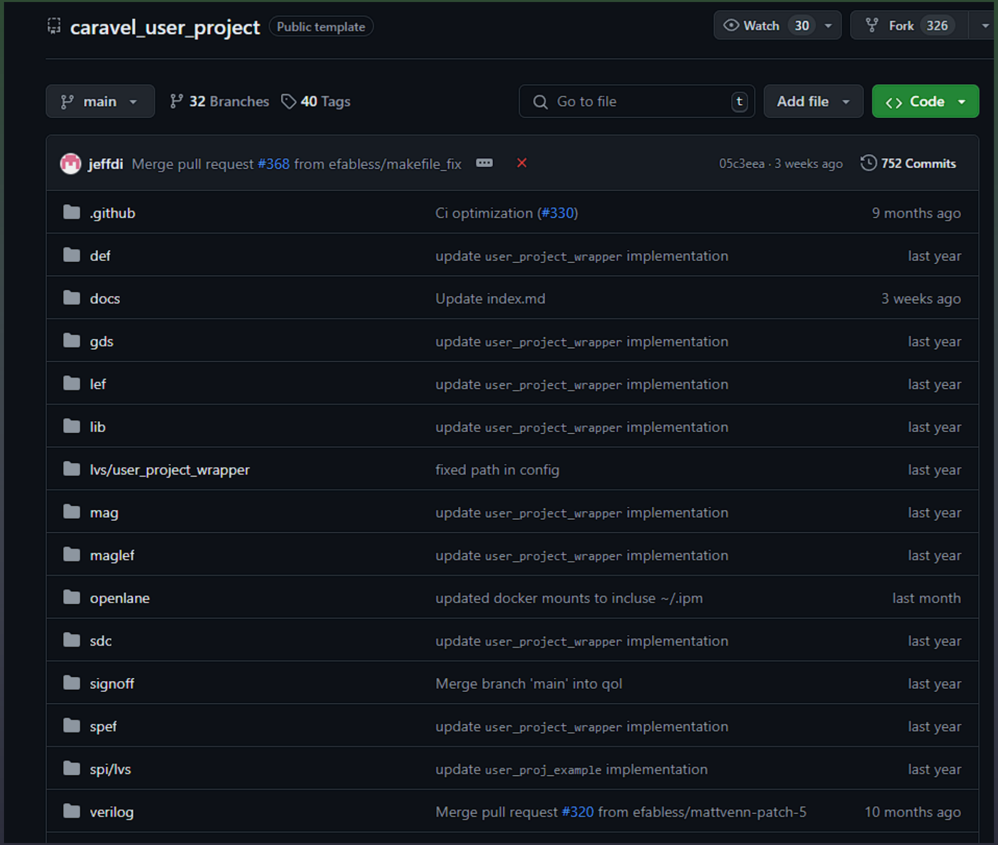
2. And especially the caravel synthesis flow https://unic-cass.github.io/training/4.3-design-setup-caravel-user-project.html



```
reg [SC_SIZE-1:0] scan_master;
reg [SC_SIZE-1:0] scan_slave;
wire [SC_SIZE-1:0] scan_next;

assign scan_next = {scan_data_in, scan_slave[SC_SIZE-1:1]};

always @ (posedge clk) begin
scan_master = scan_next;
end

always @ (negedge clk) begin
scan_slave = scan_master;
end

always @ (*)
if (scan_load_chip) begin
SC_to_the_chip=scan_slave;
end
```



How the top-level integration works


caravel_user_project Public template

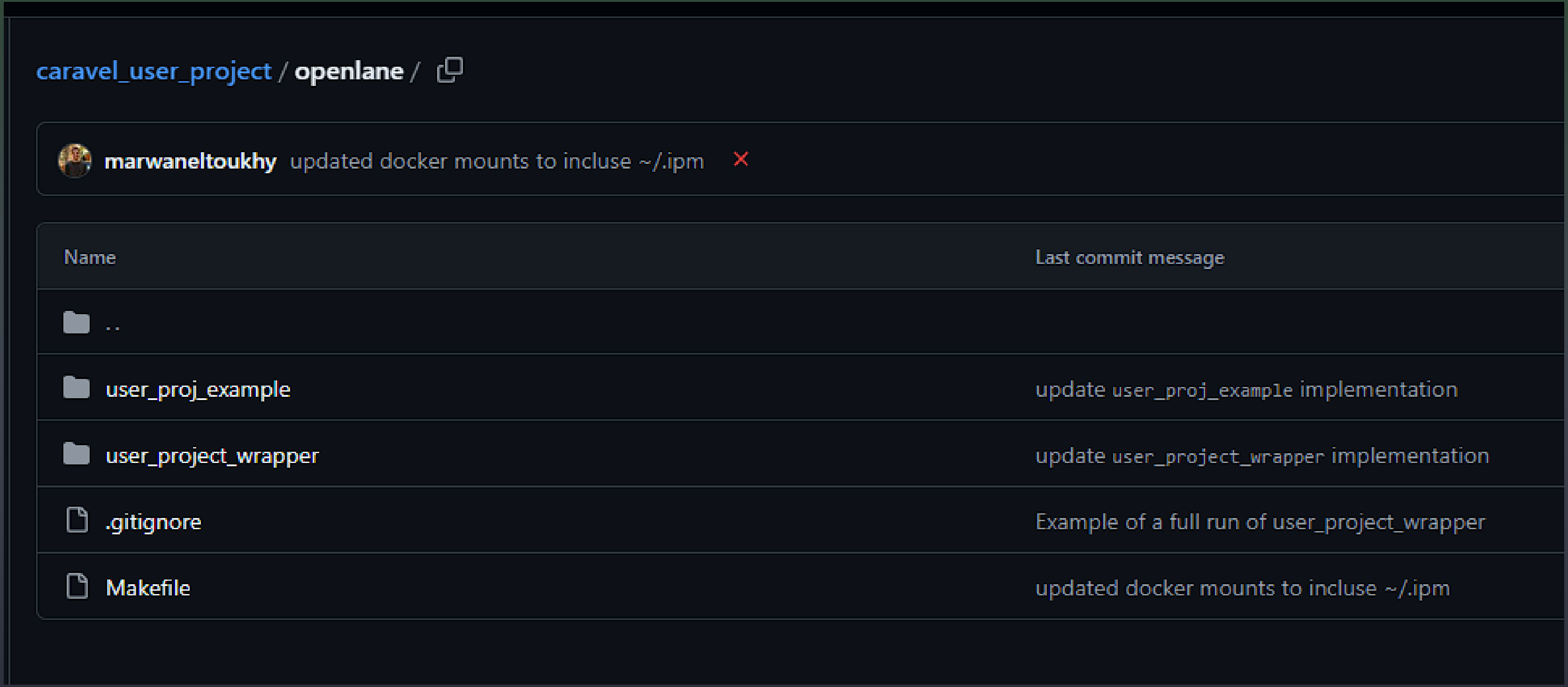
Watch 30 · Fork 326

main · 32 Branches · 40 Tags · Go to file · Add file · Code

jeffdi Merge pull request #368 from efabless/makefile_fix · 05c3eea · 3 weeks ago · 752 Commits

.github	Ci optimization (#330)	9 months ago
def	update `user_project_wrapper` implementation	last year
docs	Update index.md	3 weeks ago
gds	update `user_project_wrapper` implementation	last year
lef	update `user_project_wrapper` implementation	last year
lib	update `user_project_wrapper` implementation	last year
lvs/user_project_wrapper	fixed path in config	last year
mag	update `user_project_wrapper` implementation	last year
maglef	update `user_project_wrapper` implementation	last year
openlane	updated docker mounts to incluse ~/.ipm	last month
sdc	update `user_project_wrapper` implementation	last year
signoff	Merge branch 'main' into qol	last year
spef	update `user_project_wrapper` implementation	last year
spi/lvs	update `user_proj_example` implementation	last year
verilog	Merge pull request #320 from efabless/mattvenn-patch-5	10 months ago

How the top-level integration works

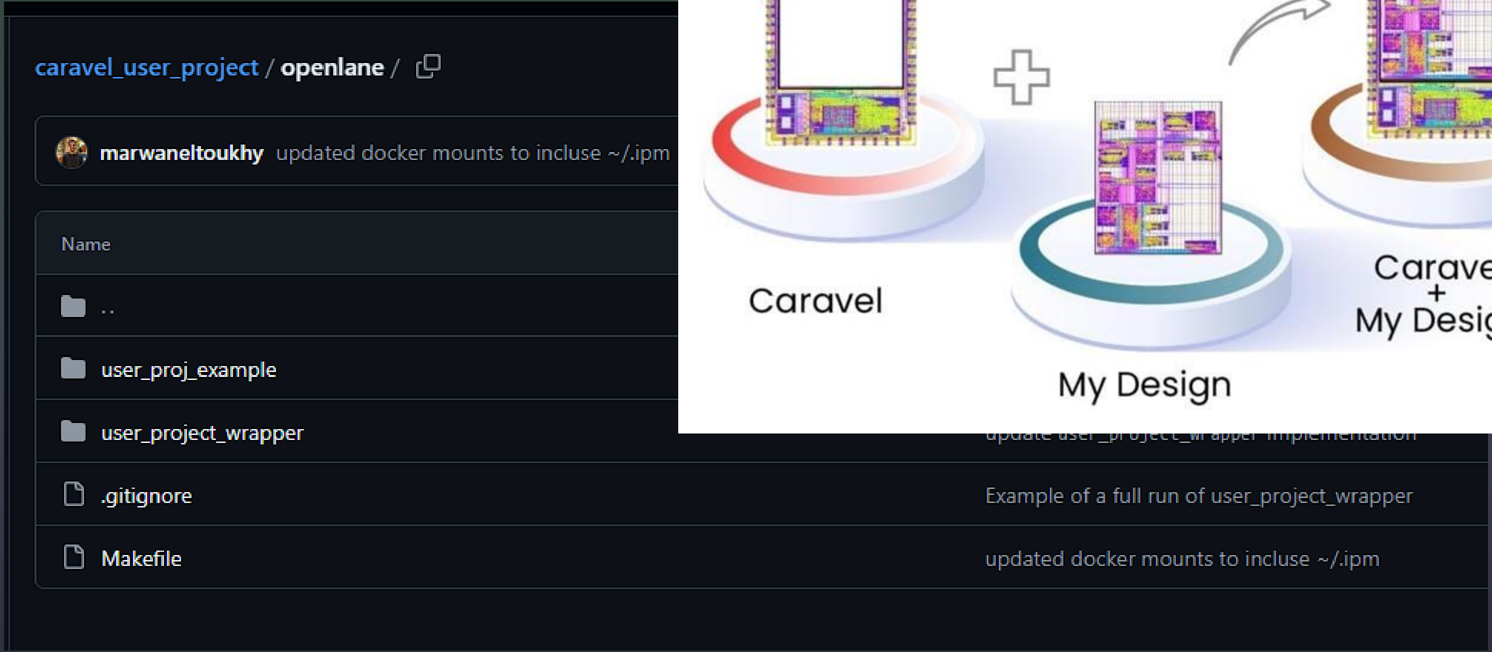


caravel_user_project / openlane /

marwaneltoukhy updated docker mounts to incluse ~/.ipm ✗

Name	Last commit message
..	
user_proj_example	update `user_proj_example` implementation
user_project_wrapper	update `user_project_wrapper` implementation
.gitignore	Example of a full run of user_project_wrapper
Makefile	updated docker mounts to incluse ~/.ipm

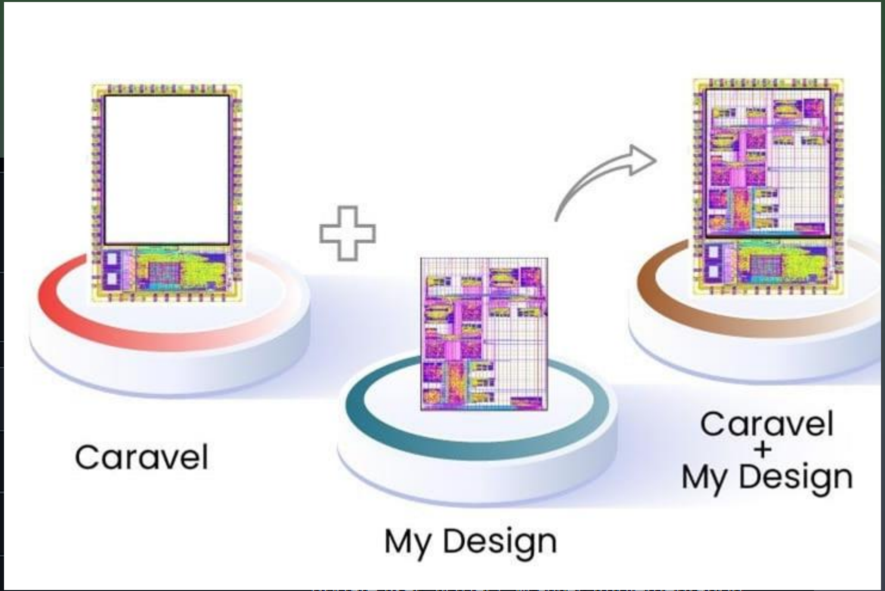
How the top-level integration works



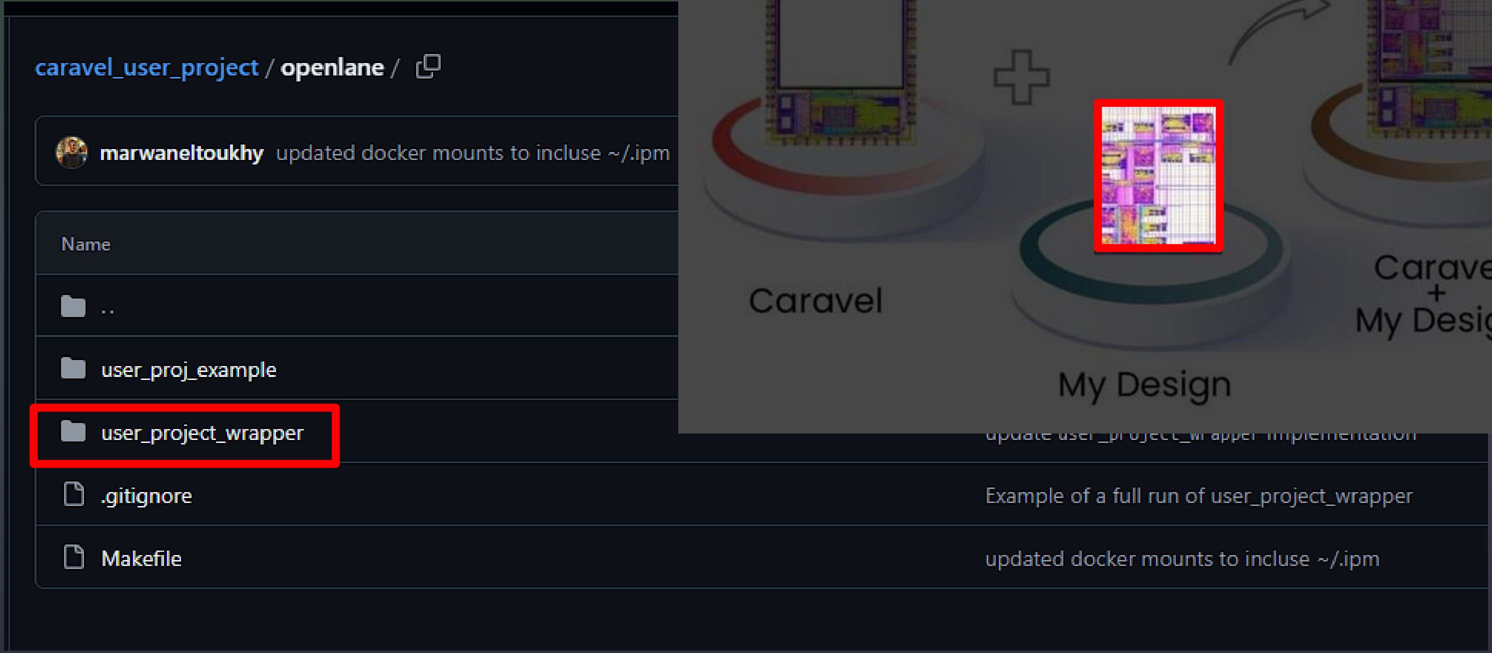
caravel_user_project / openlane /

marwaneltoukhy updated docker mounts to incluse ~/.ipm

Name	
..	
user_proj_example	
user_project_wrapper	update user_project_wrapper implementation
.gitignore	Example of a full run of user_project_wrapper
Makefile	updated docker mounts to incluse ~/.ipm



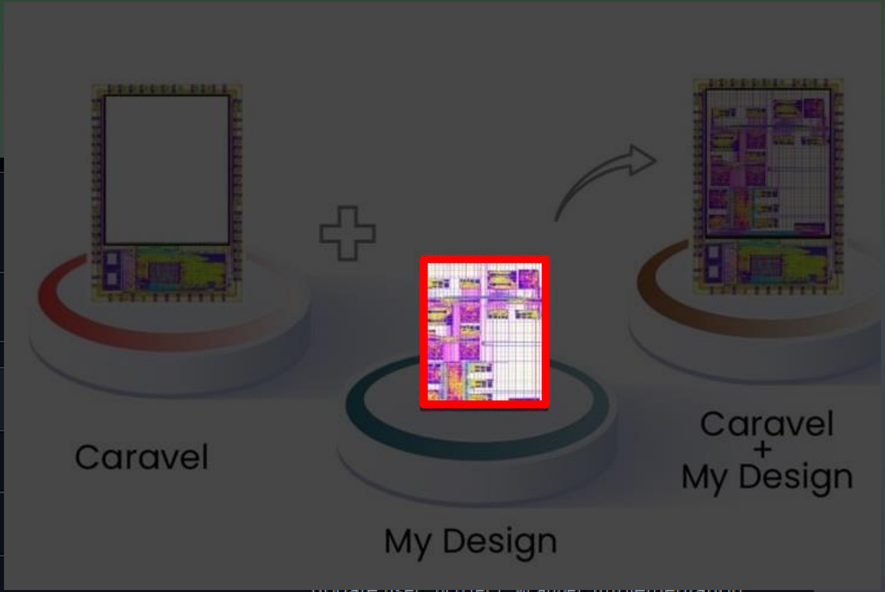
How the top-level integration works



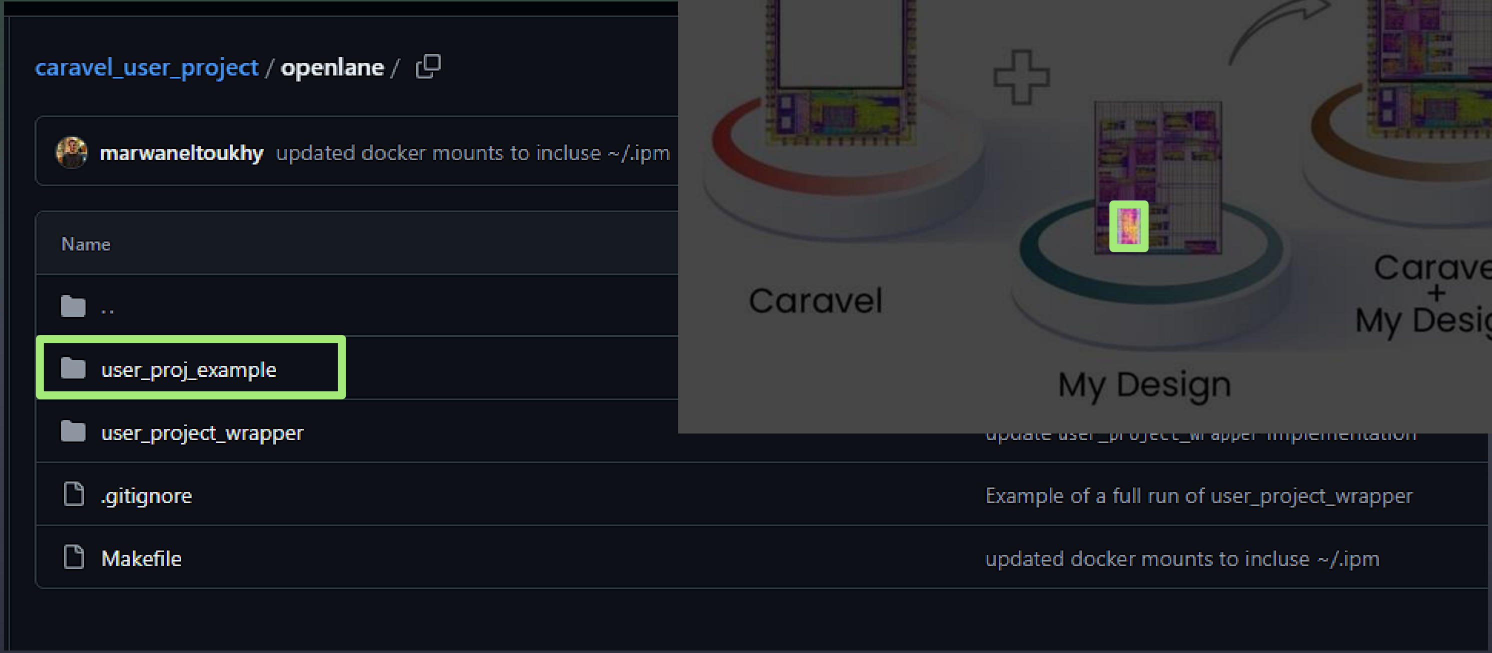
caravel_user_project / openlane /

marwaneltoukhy updated docker mounts to incluse ~/.ipm

Name	
..	
user_proj_example	
user_project_wrapper	update user_project_wrapper implementation
.gitignore	Example of a full run of user_project_wrapper
Makefile	updated docker mounts to incluse ~/.ipm



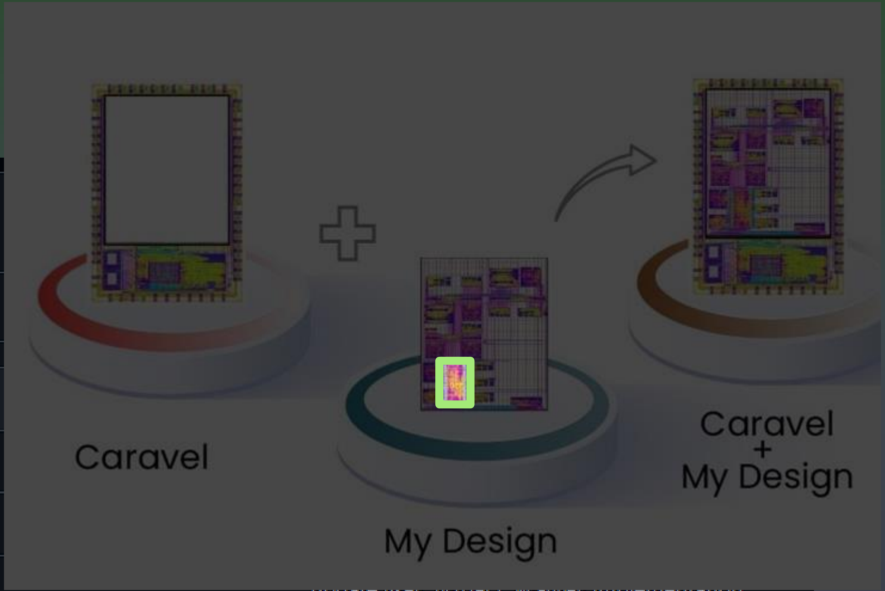
How the top-level integration works



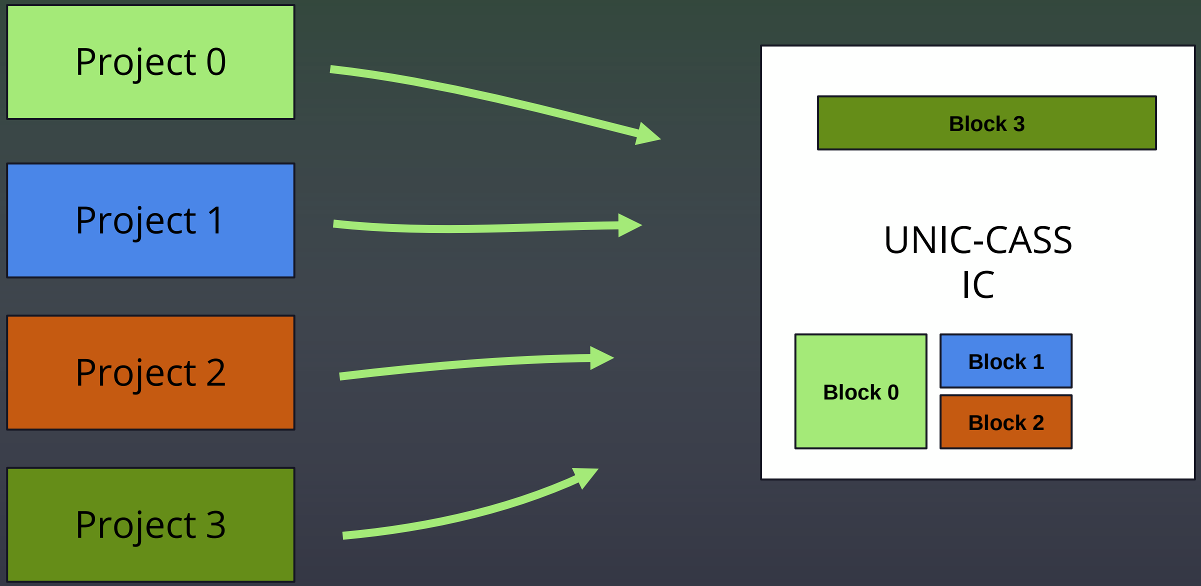
caravel_user_project / openlane /

marwaneltoukhy updated docker mounts to incluse ~/.ipm

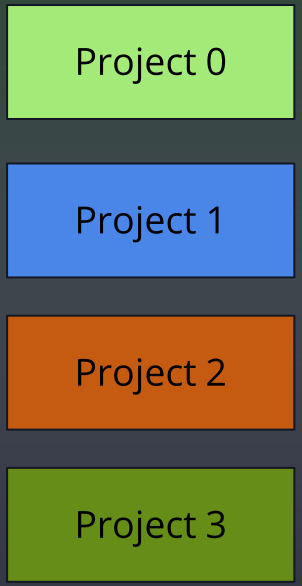
Name	
..	
user_proj_example	
user_project_wrapper	update user_project_wrapper implementation
.gitignore	Example of a full run of user_project_wrapper
Makefile	updated docker mounts to incluse ~/.ipm



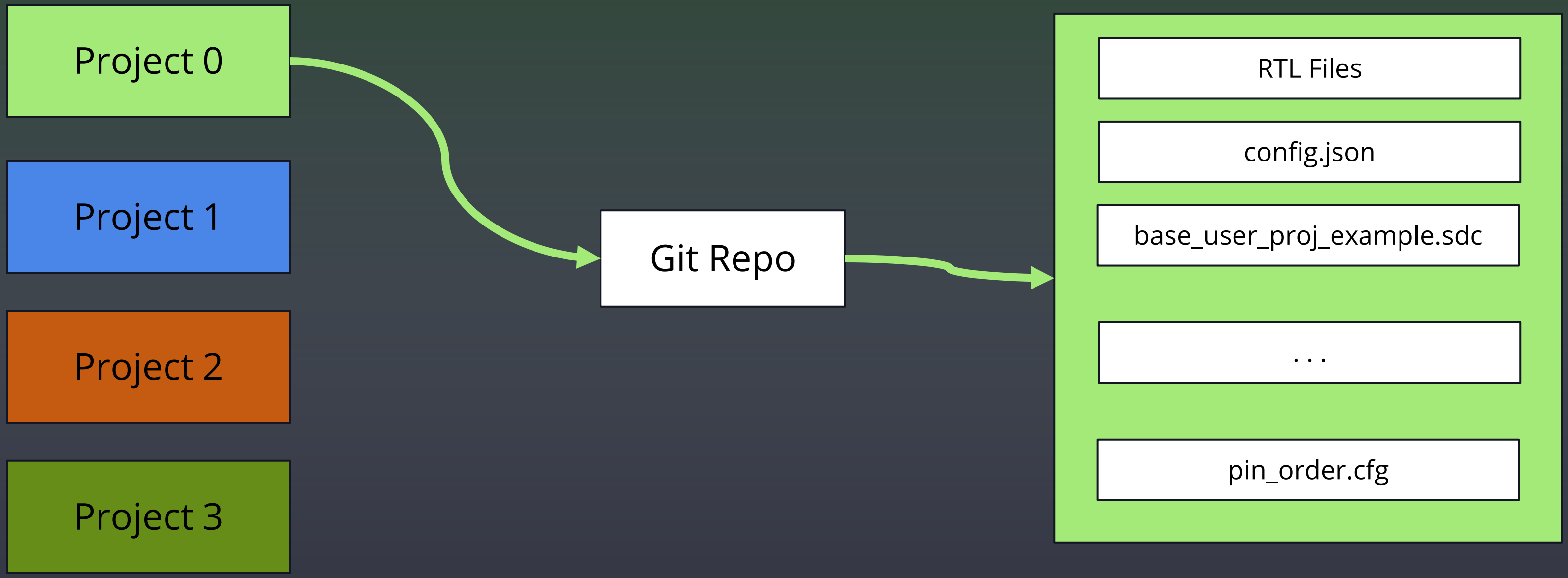
How the top-level integration works



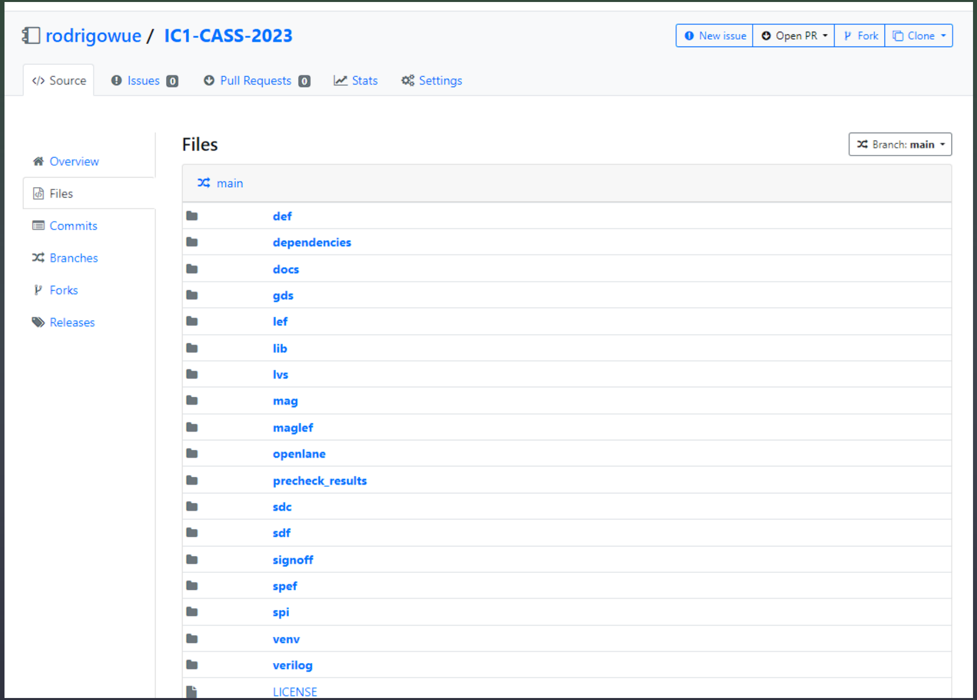
How the top-level integration works



How the top-level integration works



How the top-level integration works



Files

main

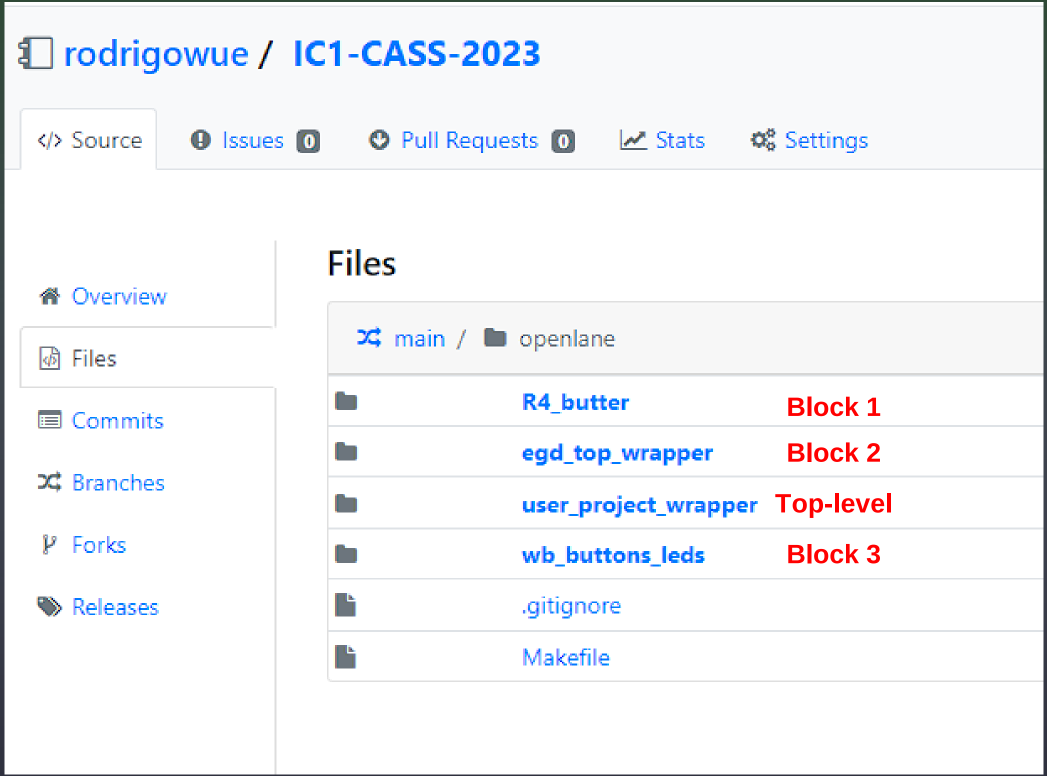
- def
- dependencies
- docs
- gds
- lef
- lib
- lvs
- mag
- maglef
- openlane
- precheck_results
- sdc
- sdf
- signoff
- spef
- spi
- venv
- verilog
- LICENSE

Efabless git repo for last year IC

A “top-level” git will be created

by me :)

How the top-level integration works



rodrigowue / IC1-CASS-2023

Files

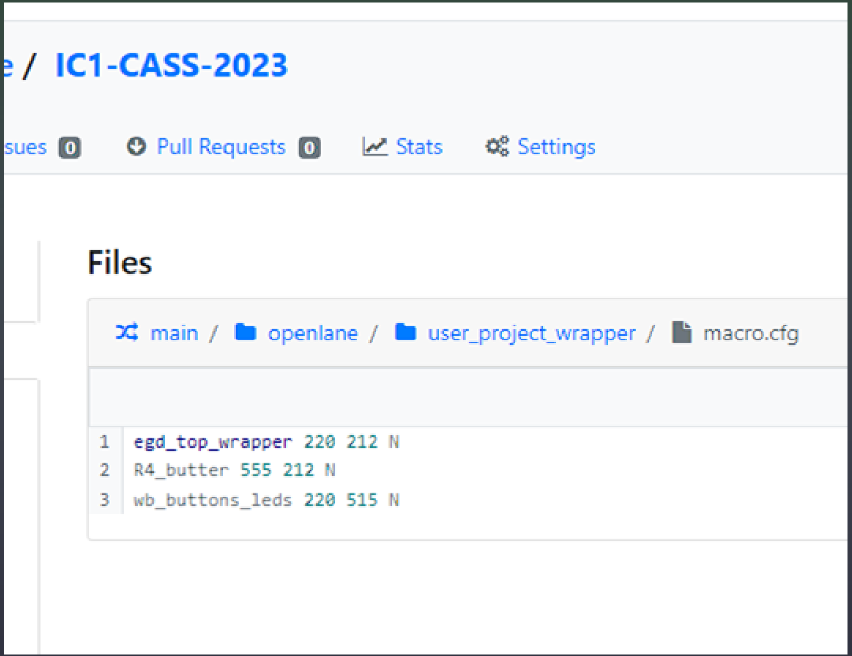
main / openlane

R4_butter	Block 1
egd_top_wrapper	Block 2
user_project_wrapper	Top-level
wb_buttons_leds	Block 3
.gitignore	
Makefile	

Efabless git repo for last year IC

And information from each block composing the wrapper will be inserted there.

How the top-level integration works



e / IC1-CASS-2023

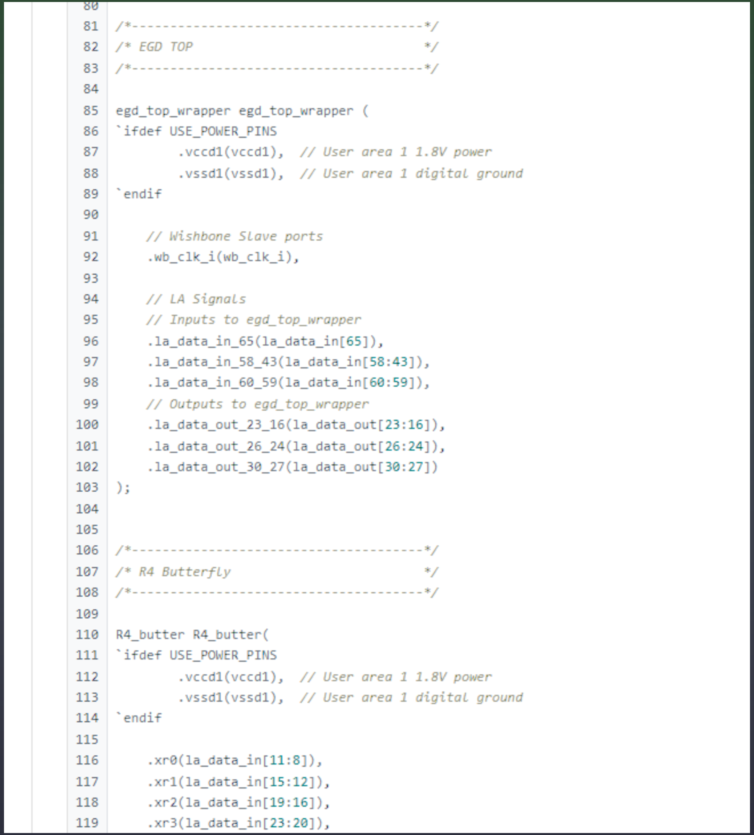
main / openlane / user_project_wrapper / macro.cfg

```
egd_top_wrapper 220 212 N
R4_butter 555 212 N
wb_buttons_leds 220 515 N
```

Efabless git repo for last year IC

The placement for each block will be defined

How the top-level integration works



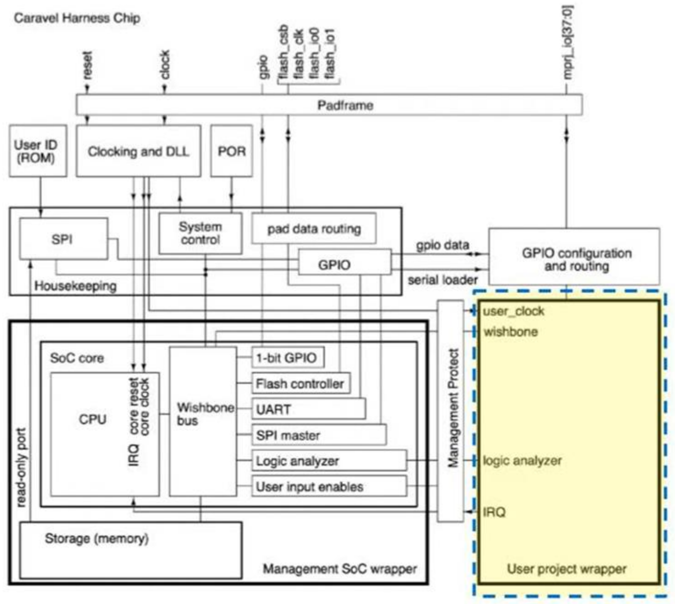
```
/*-------------------------------------*/
/* EGD TOP                             */
/*-------------------------------------*/

egd_top_wrapper egd_top_wrapper (
`ifdef USE_POWER_PINS
        .vccd1(vccd1),  // User area 1 1.8V power
        .vssd1(vssd1),  // User area 1 digital ground
`endif

    // Wishbone Slave ports
    .wb_clk_i(wb_clk_i),

    // LA Signals
    // Inputs to egd_top_wrapper
    .la_data_in_65(la_data_in[65]),
    .la_data_in_58_43(la_data_in[58:43]),
    .la_data_in_60_59(la_data_in[60:59]),
    // Outputs to egd_top_wrapper
    .la_data_out_23_16(la_data_out[23:16]),
    .la_data_out_26_24(la_data_out[26:24]),
    .la_data_out_30_27(la_data_out[30:27])
);


/*-------------------------------------*/
/* R4 Butterfly                        */
/*-------------------------------------*/

R4_butter R4_butter(
`ifdef USE_POWER_PINS
        .vccd1(vccd1),  // User area 1 1.8V power
        .vssd1(vssd1),  // User area 1 digital ground
`endif

    .xr0(la_data_in[11:8]),
    .xr1(la_data_in[15:12]),
    .xr2(la_data_in[19:16]),
    .xr3(la_data_in[23:20]),
```

Efabless git repo for last year IC

The blocks will be also logically connected to the user_project_wrapper to the assigned pinout. By their instantiation in the wrapper Verilog.

Think about the context! How do you want to connect it with the outside world?!
https://unic-cass.github.io/training/6.1-design-for-tapeout-Caravel-overview.html

- 10 mm^2 silicon area
- 38 User's I/O Pads
- User I/O Pad Configuration
  - FW running on the Management SoC
  - Housekeeping SPI
  - Integration-time confuiguration
- Access to the Management SoC wishbone bus
- Clocking and Reseting Signals
- 128 R/W Logic Probes



Thanks, any questions? :)

feel free to reach me out

rnwuerdig@inf.ufrgs.br

IEEE CIRCUITS AND SYSTEMS SOCIETY