



Introduction to the Digital Design Flow

Rodrigo N. Wuerdig

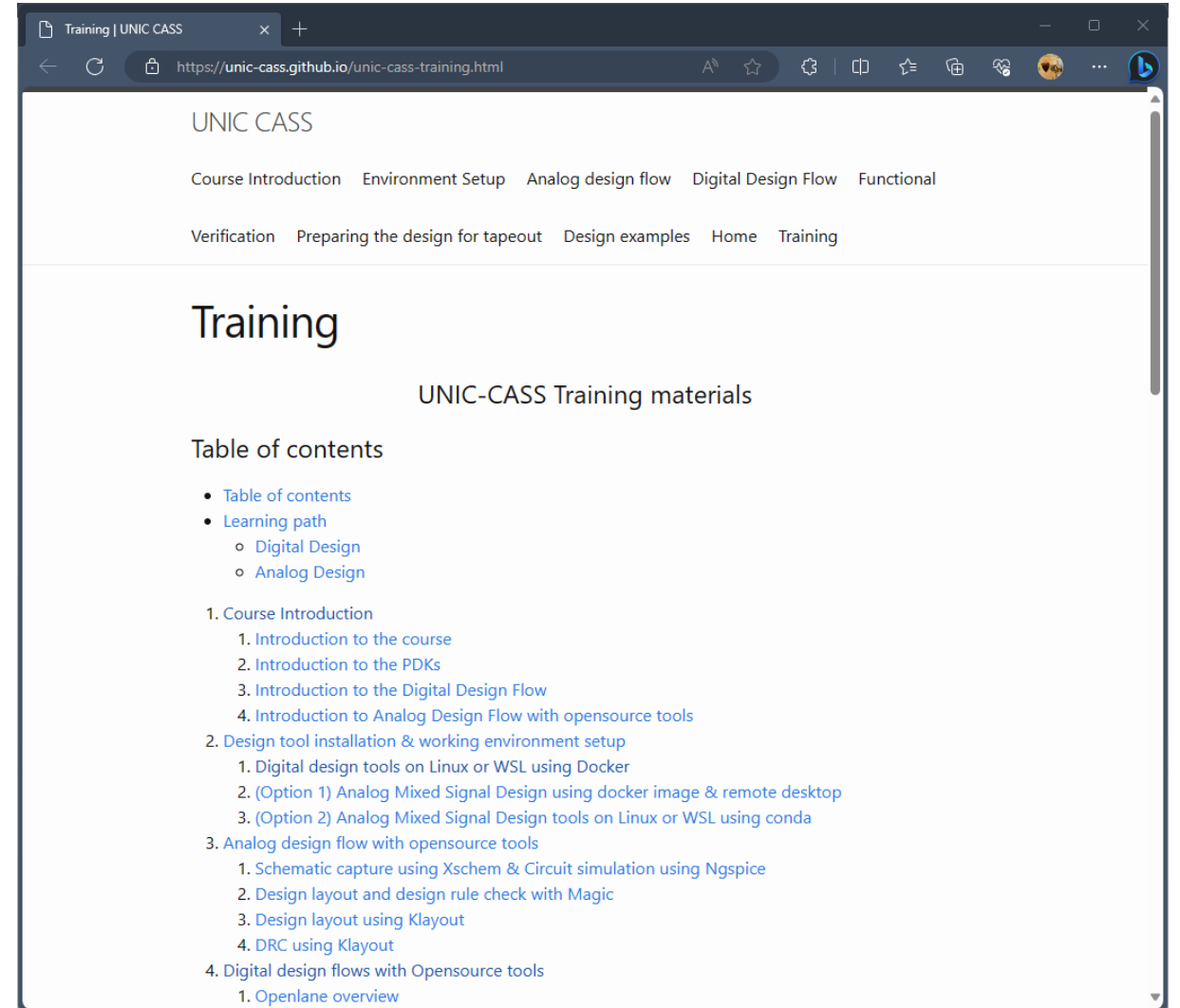
PhD Student @ Universidade Federal do Rio Grande do Sul (UFRGS)
ASIC Design Engineer @ HaiLa

August 30th, 2023

UNIC CASS Page

Do not miss the page created by Prof. Duy-Hieu Bui:

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



The screenshot shows a web browser window with the URL <https://unic-cass.github.io/unic-cass-training.html>. The page title is "UNIC CASS". The navigation menu includes: Course Introduction, Environment Setup, Analog design flow, Digital Design Flow, Functional, Verification, Preparing the design for tapeout, Design examples, Home, and Training. The main heading is "Training" with the subtitle "UNIC-CASS Training materials". Below this is a "Table of contents" section with the following items:

- [Table of contents](#)
- [Learning path](#)
 - [Digital Design](#)
 - [Analog Design](#)
- 1. [Course Introduction](#)
 - 1. [Introduction to the course](#)
 - 2. [Introduction to the PDKs](#)
 - 3. [Introduction to the Digital Design Flow](#)
 - 4. [Introduction to Analog Design Flow with opensource tools](#)
- 2. [Design tool installation & working environment setup](#)
 - 1. [Digital design tools on Linux or WSL using Docker](#)
 - 2. [\(Option 1\) Analog Mixed Signal Design using docker image & remote desktop](#)
 - 3. [\(Option 2\) Analog Mixed Signal Design tools on Linux or WSL using conda](#)
- 3. [Analog design flow with opensource tools](#)
 - 1. [Schematic capture using Xschem & Circuit simulation using Ngspice](#)
 - 2. [Design layout and design rule check with Magic](#)
 - 3. [Design layout using Klayout](#)
 - 4. [DRC using Klayout](#)
- 4. [Digital design flows with Opensource tools](#)
 - 1. [Openlane overview](#)

Join the Open-EDA Community

UNIC-CASS Slack

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

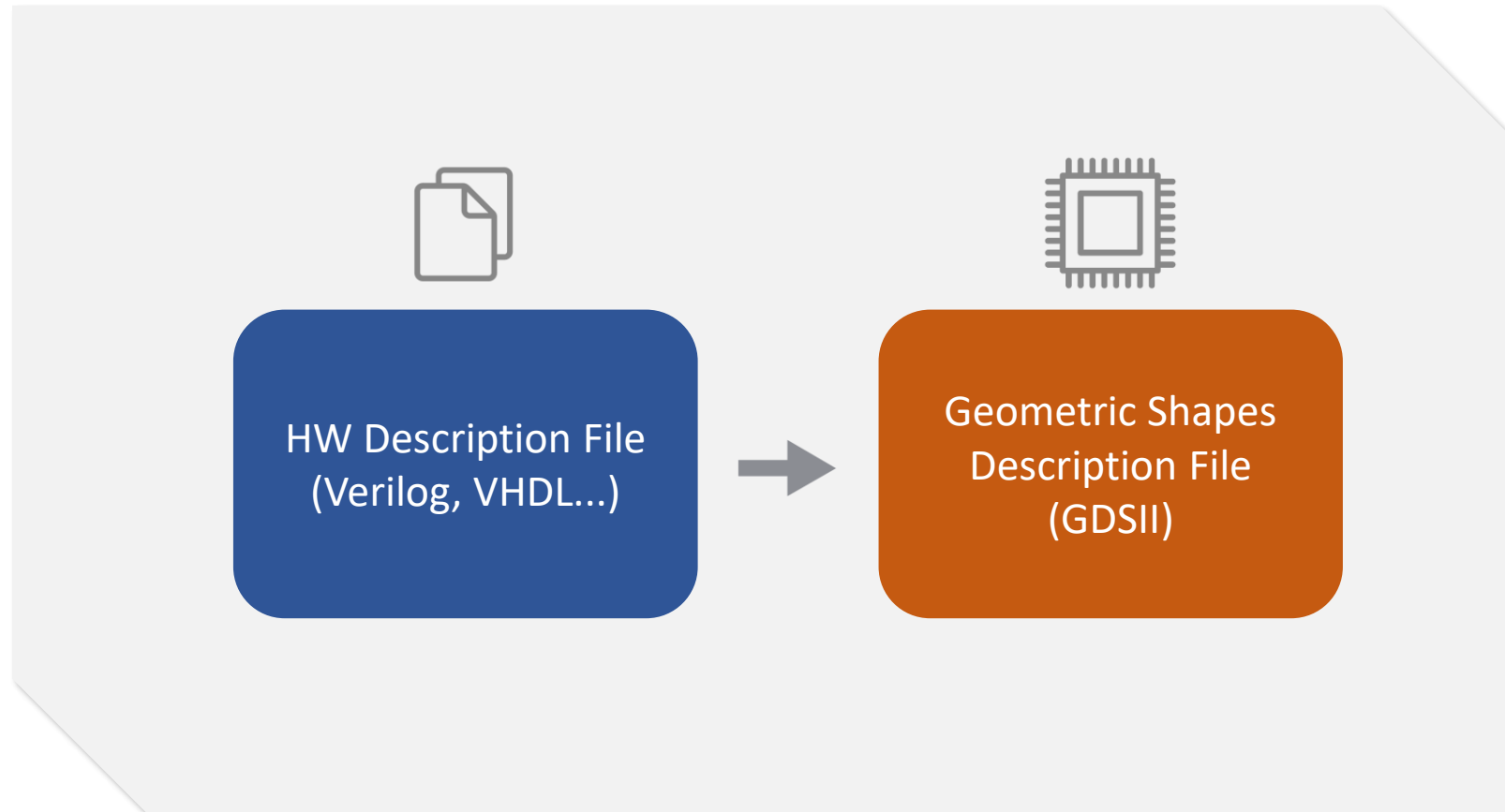
open-source-silicon Slack

https://join.slack.com/t/open-source-silicon/shared_invite/zt-1zopfd1gk-uI2eSINXB54xN9RCowGa4g

Outline

1. Introduction to the RTL2GDSII Flow
 - Logical Synthesis
 - Physical Synthesis
2. Open-EDA Ecosystem
3. OpenLane
4. Digital Design Environment Setup
5. Running Example

RTL2GDSII



RTL2GDSII

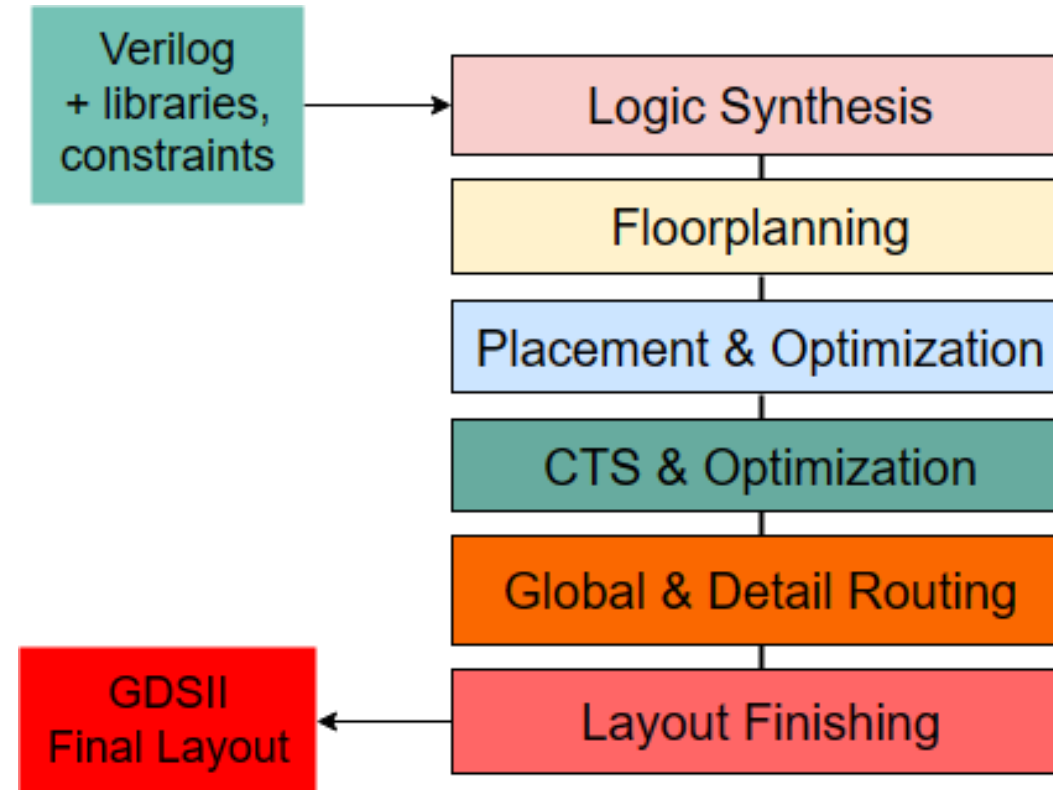


Figure Source: <https://openroad.readthedocs.io/en/latest/main/README.html>

Logic Synthesis

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

```
23 reg [SC_SIZE-1:0] scan_master;
24 reg [SC_SIZE-1:0] scan_slave;
25 wire [SC_SIZE-1:0] scan_next;
26
27 assign scan_next = {scan_data_in, scan_slave[SC_SIZE-1:1]};
28
29 always @ (posedge clk) begin
30 scan_master = scan_next;
31 end
32
33 always @ (negedge clk) begin
34 scan_slave = scan_master;
35 end
36
37 always @ (*)
38 if (scan_load_chip) begin
39 SC_to_the_chip=scan_slave;
40 end
```

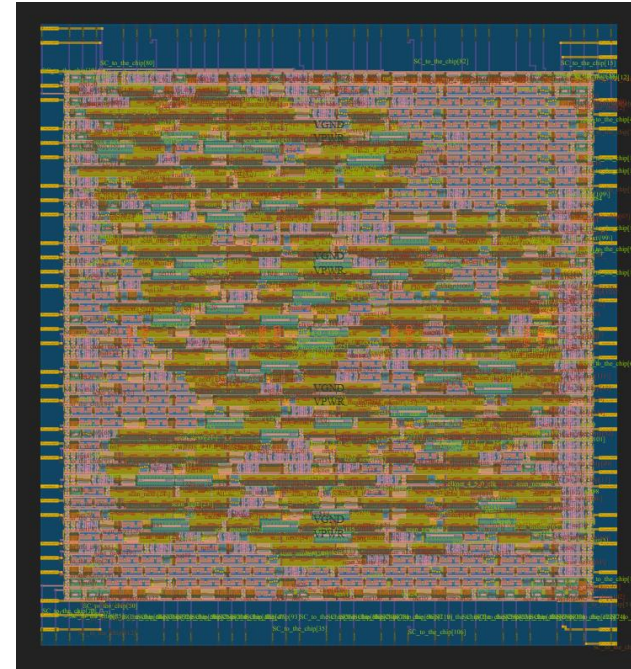
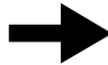


```
4654 .VPWR(VPWR)),
4655 sky130_fd_sc_hd_tapvpwrvgnd_1 TAP_99 (.VGND(VGND),
4656 .VPWR(VPWR));
4657 sky130_fd_sc_hd_buf_1_142_ (.A(clknet_4_9_0_clk),
4658 .VGND(VGND),
4659 .VNB(VGND),
4660 .VPB(VPWR),
4661 .VPWR(VPWR),
4662 .X(_128_));
4663 sky130_fd_sc_hd_buf_1_143_ (.A(clknet_1_1_leaf_128_),
4664 .VGND(VGND),
4665 .VNB(VGND),
4666 .VPB(VPWR),
4667 .VPWR(VPWR),
4668 .X(_129_));
4669 sky130_fd_sc_hd_inv_2_144_9 (.A(clknet_1_1_leaf_129_),
4670 .VGND(VGND),
4671 .VNB(VGND),
4672 .VPB(VPWR),
```

Physical Synthesis

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

```
4654 .VPWR(VPWR),
4655 sky130_fd_sc_hd__tapvpwrvgnd_1 TAP_99 (.VGND(VGND),
4656 .VPWR(VPWR));
4657 sky130_fd_sc_hd__buf_1_142_ (.A(clknet_4_9_0_clk),
4658 .VGND(VGND),
4659 .VNB(VGND),
4660 .VPB(VPWR),
4661 .VPWR(VPWR),
4662 .X(_128_));
4663 sky130_fd_sc_hd__buf_1_143_ (.A(clknet_1_1_leaf_128_),
4664 .VGND(VGND),
4665 .VNB(VGND),
4666 .VPB(VPWR),
4667 .VPWR(VPWR),
4668 .X(_129_));
4669 sky130_fd_sc_hd__inv_2_144_9 (.A(clknet_1_1_leaf_129_),
4670 .VGND(VGND),
4671 .VNB(VGND),
4672 .VPB(VPWR),
```



Open-EDA Ecosystem

70+ open/free tools for IC Design

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

Frameworks/Wrappers

OpenLane

OpenROAD Flow Scripts

SiliconCompiler

...

Frameworks/Wrappers

OpenLane

OpenROAD Flow Scripts

SiliconCompiler

...

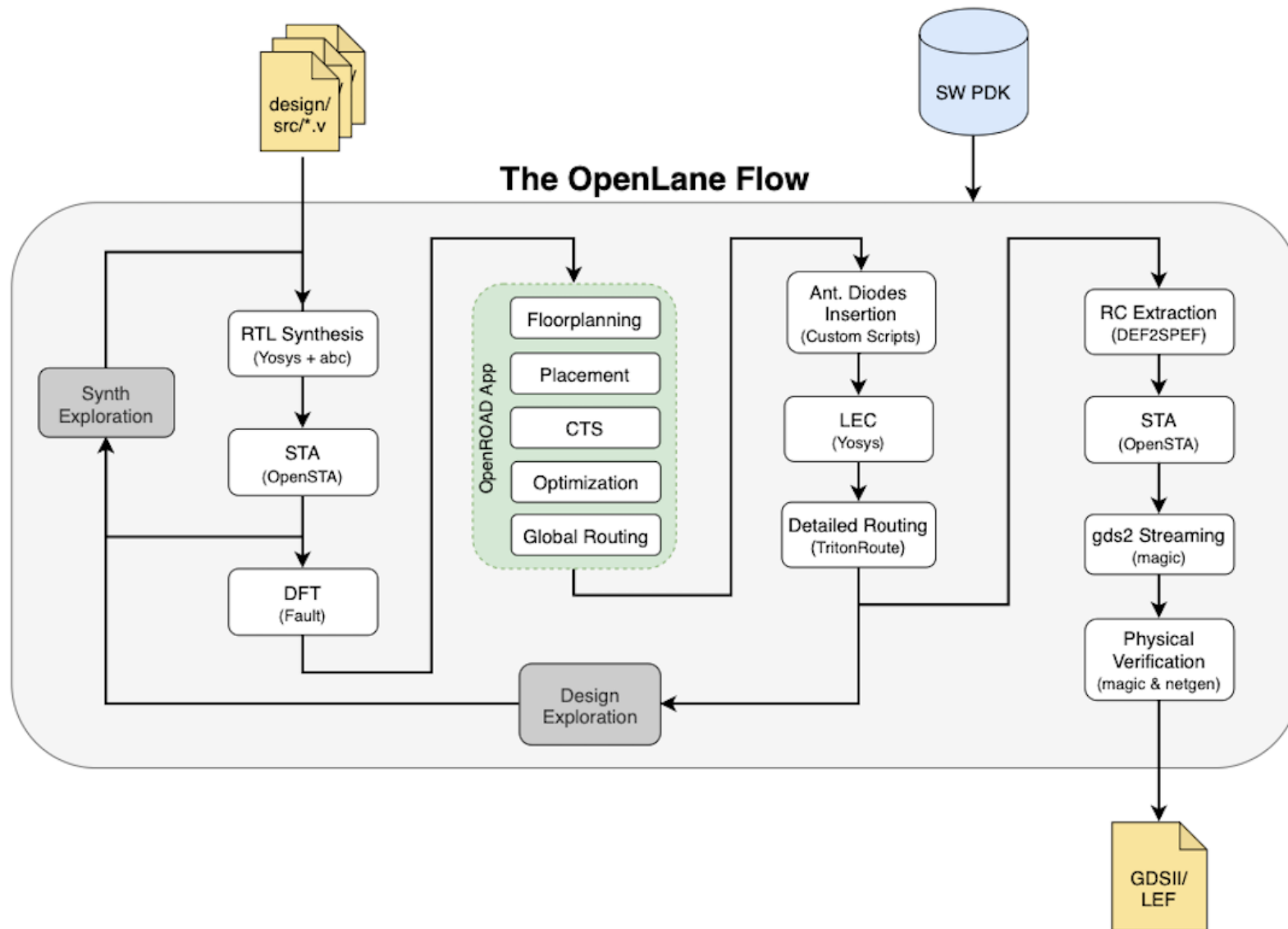
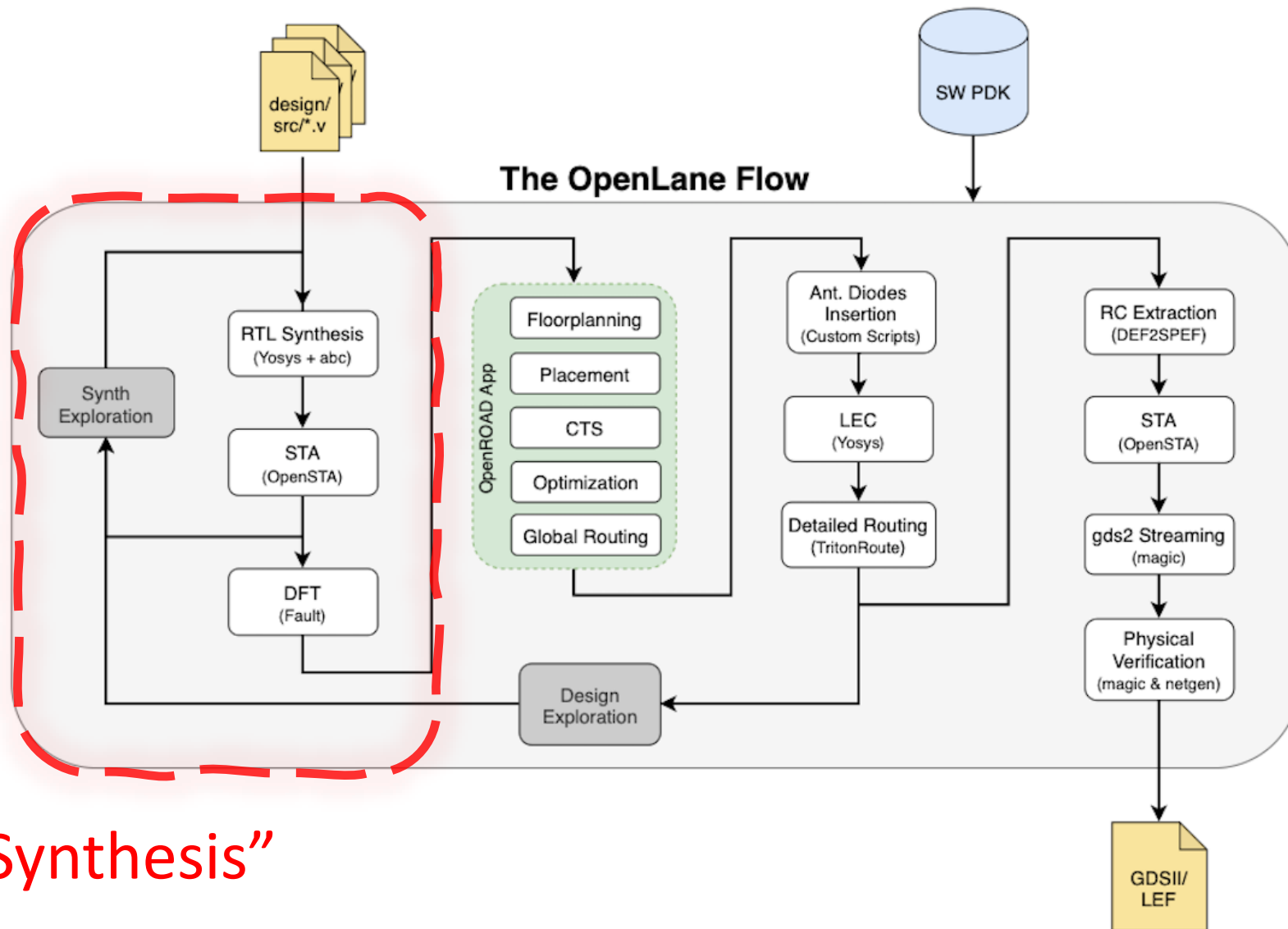
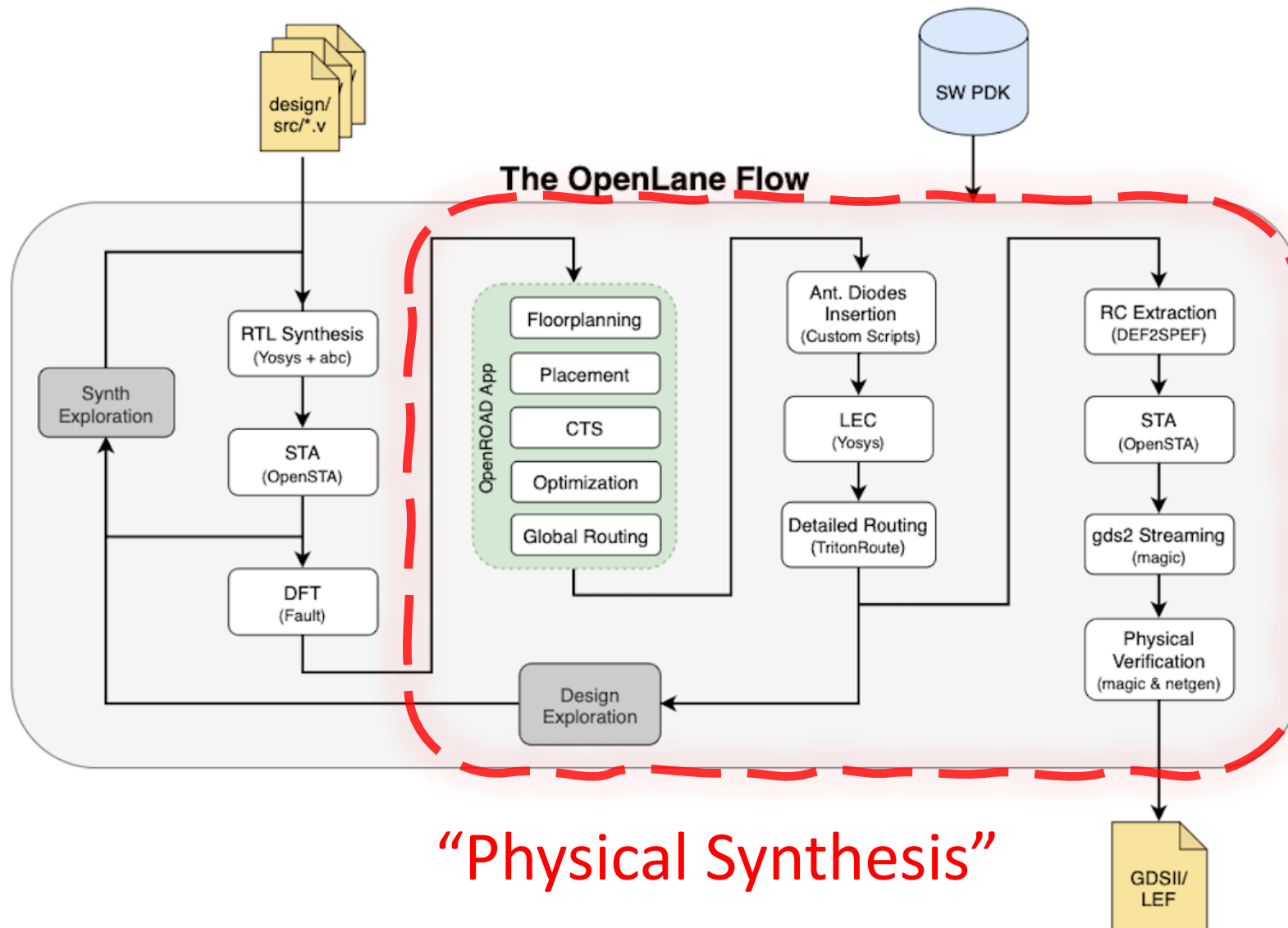


Figure Source: https://openlane.readthedocs.io/en/latest/flow_overview.html



“Logic Synthesis”

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

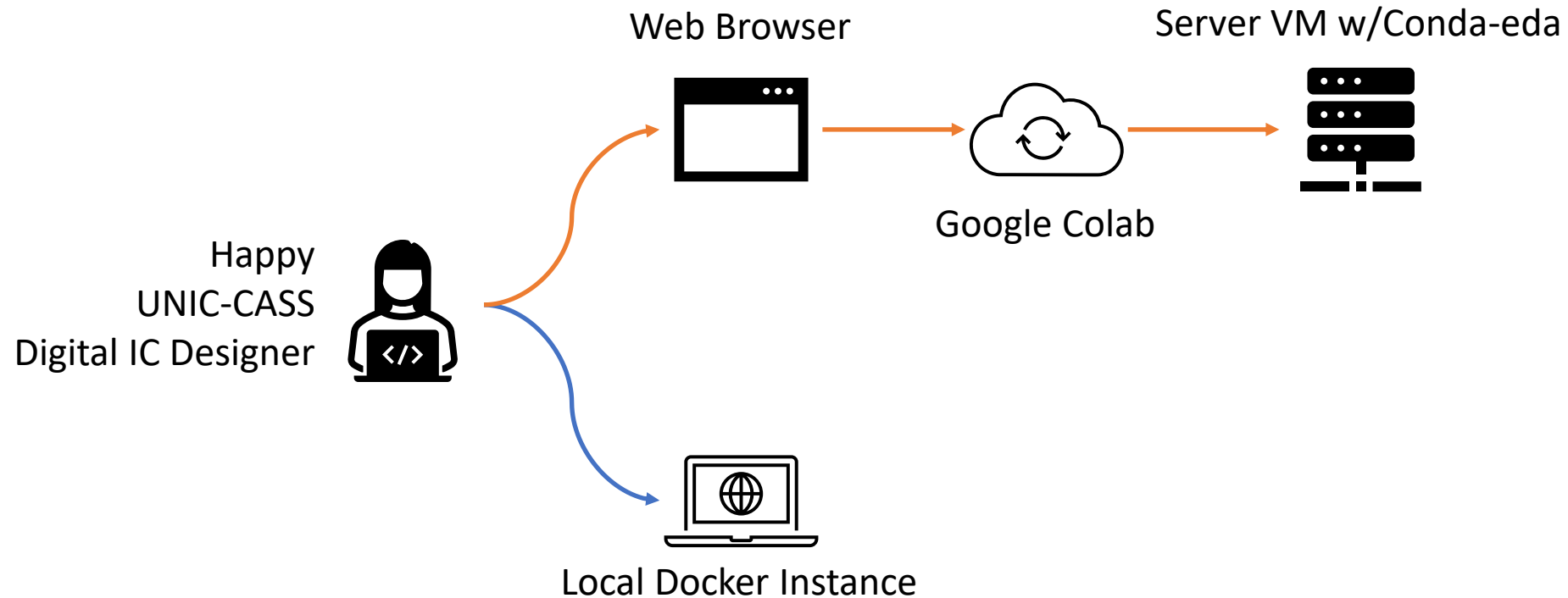


“Physical Synthesis”

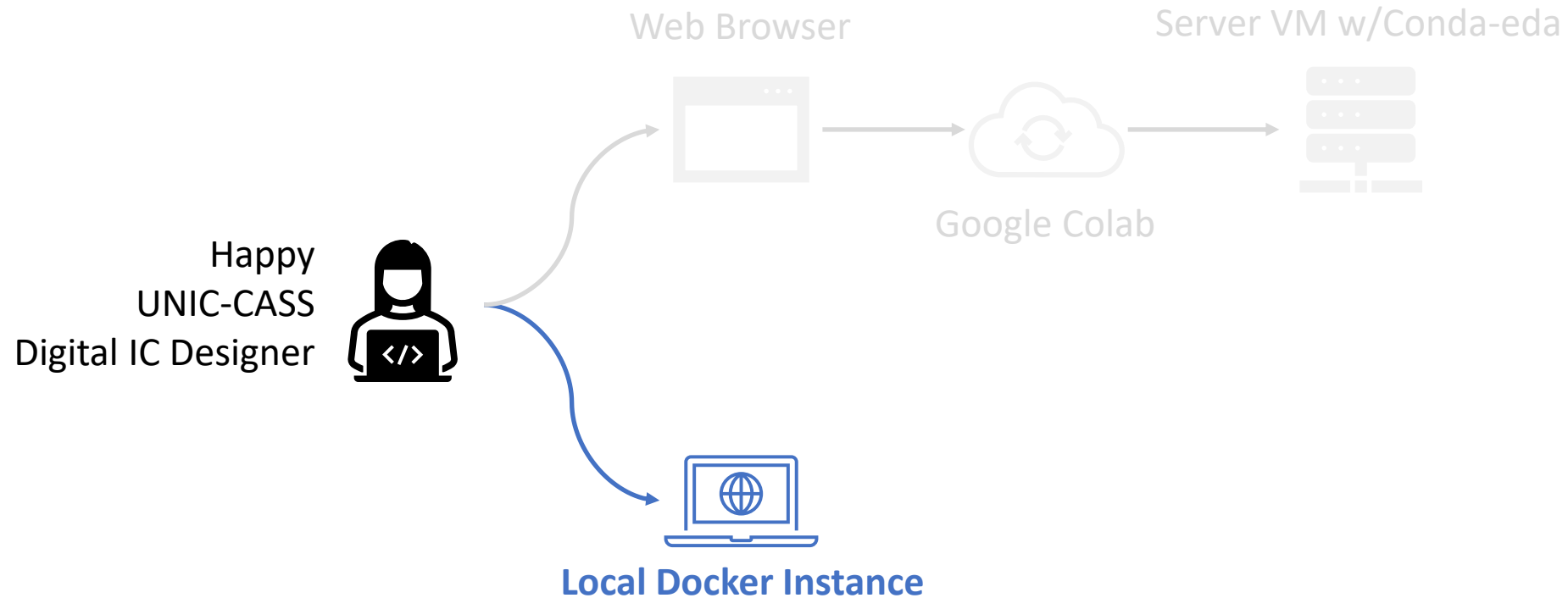
Figure Source: https://openlane.readthedocs.io/en/latest/flow_overview.html

Digital Design Environment Setup

Docker/Local vs Jupyter Notebook



Docker/Local vs Jupyter Notebook



How to Start with the Docker Download and Installation

Prerequisites

To follow this lesson, you need a working linux system either of the followings:

- A Linux-based system (preferably Ubuntu 22.04)
- Windows users can install WSL (Windows Subsystem Linux), then install Ubuntu 22.04

The commands in this guide are prepared for Ubuntu/Debian-based systems. However, you can easily adapt them to other linux distributions such as RHEL or Centos.

Environment Setup

2 Design tool installation & working environment setup

2.1 Digital design tools on Linux or WSL using Docker

Digital design tools on Linux or WSL using Docker

Prerequisites

To follow this lesson, you need a working linux system either of the followings:

- A Linux-based system (preferably Ubuntu 22.04)
- Windows users can install WSL (Windows Subsystem Linux), then install Ubuntu 22.04

The commands in this guide are prepared for Ubuntu/Debian-based systems. However, you can easily adapt them to other linux distributions such as RHEL or Centos.

Note for Windows users:

If you are a MS windows user, you need to install WSL2 and Ubuntu 22.04 as in this

Source: <https://unic-cass.github.io/02-env-setup.html>

Installing Docker

Terminal

```
sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

echo "deb [arch=$(dpkg --print-architecture)] signed-  
by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu "$(. /etc/os-release && echo  
"$VERSION_CODENAME")" stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Source: <https://unic-cass.github.io/02-env-setup.html>

If this happens...

```
Terminal  
E: Malformed entry 1 in list file /etc/apt/sources.list.d/docker.list (Suite)
```

Edit “/etc/apt/sources.list.d/docker.list” and remove line break

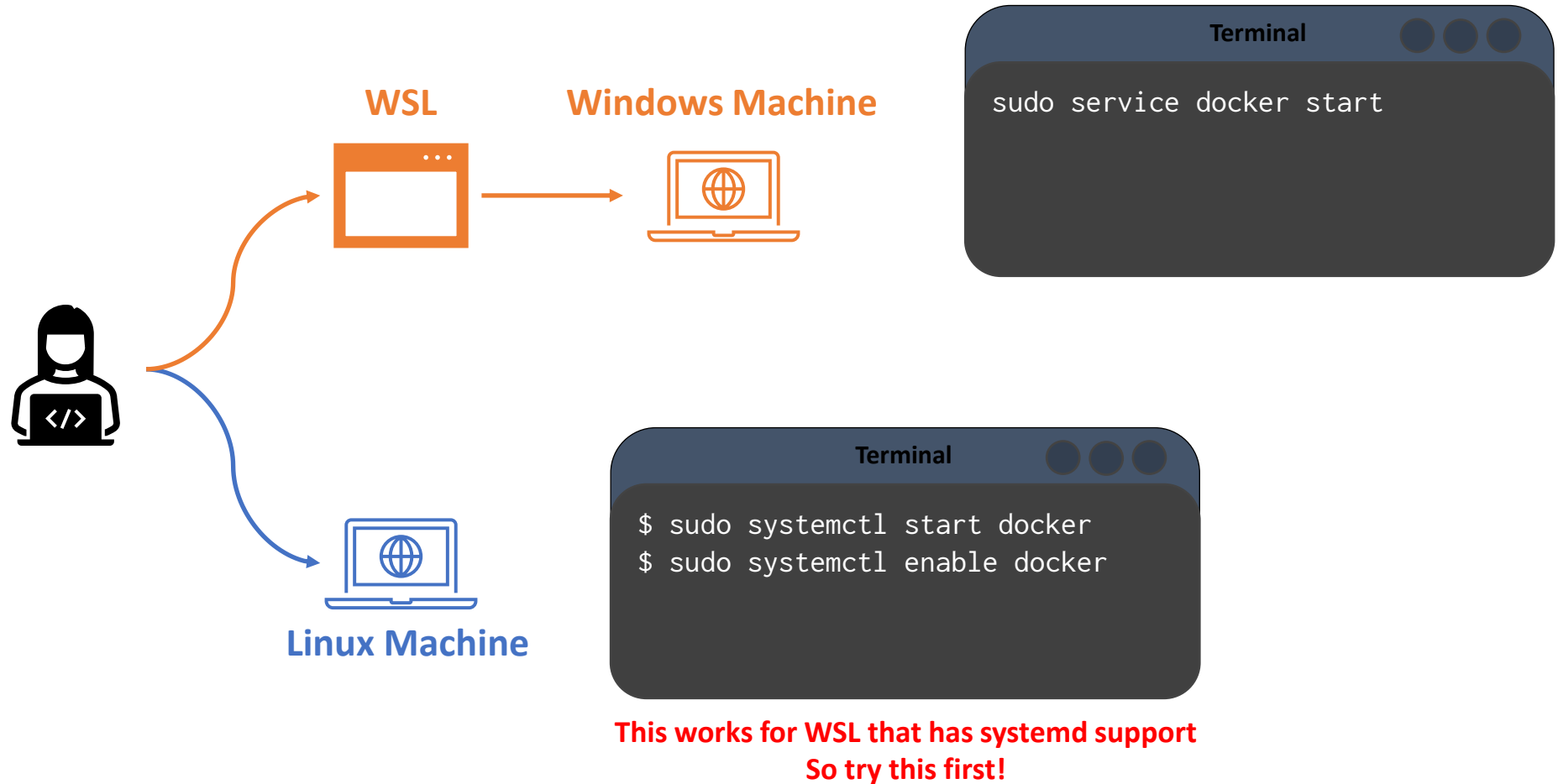
```
rodrigowue@JamesWebb: ~  
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu  
jammy stable  
~  
~  
~  
~
```



```
rodrigowue@JamesWebb: ~  
deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable  
~  
~  
~
```

Source: <https://unic-cass.github.io/02-env-setup.html>

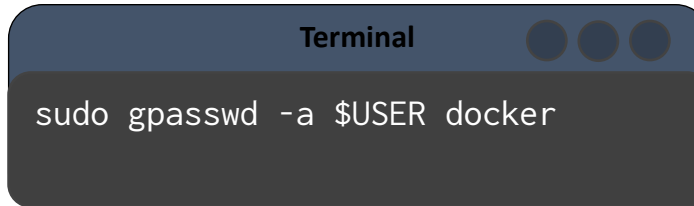
Starting Docker



Source: <https://unic-cass.github.io/02-env-setup.html>

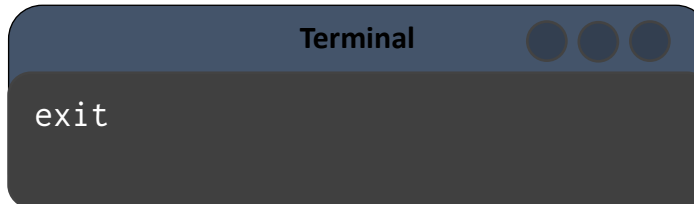
Add User to Docker Group

You will also need to add your user to docker group so that you have permission to pull the docker image as normal users:



```
Terminal
sudo gpasswd -a $USER docker
```

To get the above command taking effect, you have to log out:



```
Terminal
exit
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Pulling OpenLane Docker Image

```
Terminal  
docker pull efabless/openlane:latest
```



```
rodrigo@JamesWebb: ~  
e41d83eb9365: Pull complete  
cc5c8eb7fd73: Pull complete  
4a36667334cf: Pull complete  
094d582dc00c: Pull complete  
845d9a9766ee: Pull complete  
1fa0b1845dff: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:b7c6b828c3b95c673321c9be4da048a94fa4cb8b18bb99ef0e610aed38aede7f  
Status: Downloaded newer image for efabless/openlane:latest  
docker.io/efabless/openlane:latest  
rodrigo@JamesWebb: ~$
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Installing the Skywater130 PDK

Terminal

```
python3 -m pip install --upgrade --no-cache-dir volare  
echo 'export PATH=$HOME/.local/bin:$PATH' >> .bashrc  
source .bashrc
```

If pip is not found, then:

`sudo apt install python3-pip`

and redo the command sequence.

Source: <https://unic-cass.github.io/02-env-setup.html>

Test if Volare is Working

```
Terminal  
volare --version
```



```
rodrigowue@JamesWebb: ~  
rodrigowue@JamesWebb:~$ volare --version  
volare, version 0.12.3  
rodrigowue@JamesWebb:~$
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Installing the Skywater130 PDK

Terminal

```
export PDK_ROOT=$HOME/unic-cass/pdks
export PDK=sky130A
volare ls-remote --pdk sky130
volare enable --pdk sky130 78b7bc32ddb4b6f14f76883c2e2dc5b5de9d1cbc
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Running Example

Create Directory and Generate OpenLane Template

Terminal

```
mkdir gcd_example
cd gcd_example
docker run -it -v $(pwd):/gcd_example/ -v $PDK_ROOT:$PDK_ROOT -e PDK_ROOT=$PDK_ROOT -u $(id -u $USER):$(id -g $USER)
efabless/openlane:latest
cd /gcd_example
#flow.tcl -design <design name> -init_design_config - this will create a template structure and files for OpenLane
flow.tcl -design gcd -init_design_config
```

Source: <https://unic-cass.github.io/02-env-setup.html>



Enter Directory and Download RTL Files

Terminal

```
cd /gcd_example/openlane/gcd
#Download from Berkley the RTL files
wget https://inst.eecs.berkeley.edu/~cs250/fa20/files/gcd.tar.xz
#Uncompress, remove compressed file, and exit
tar xvf gcd.tar.xz
rm gcd.tar.xz
exit
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Edit Template to Fit Design Specs

After exiting the Docker run, we need to modify the `config.json` and add a `pin_order.cfg` to implement the design. These two files can be created using `gedit` commands as follows:

```
Terminal
gedit openlane/gcd/config.json
```

```
config.json
~/gcd_example/openlane/gcd
Save

1 {
2     "DESIGN_NAME": "gcd",
3     "VERILOG_FILES": "dir::src/*.v",
4     "CLOCK_PORT": "clk",
5     "CLOCK_NET": "ref::$CLOCK_PORT",
6     "FP_PIN_ORDER_CFG": "dir::pin_order.cfg",
7     "FP_PDN_VOFFSET": 7,
8     "FP_PDN_HOFFSET": 7,
9     "FP_PDN_SKIPTRIM": true,
10    "FP_CORE_UTIL" : 45,
11    "CLOCK_PERIOD": 10.0,
12    "DESIGN_IS_CORE": true
13 }
```

Source: <https://unic-cass.github.io/02-env-setup.html>

About the Configuration File (config.json)

Here is where the design freedom lives.
Check the documentation to know what each variable does.

```
config.json
~/gcd_example/openlane/gcd

1 {
2   "DESIGN_NAME": "gcd",
3   "VERILOG_FILES": "dir::src/*.v",
4   "CLOCK_PORT": "clk",
5   "CLOCK_NET": "ref:$CLOCK_PORT",
6   "FP_PIN_ORDER_CFG": "dir::pin_order.cfg",
7   "FP_PDN_VOFFSET": 7,
8   "FP_PDN_HOFFSET": 7,
9   "FP_PDN_SKIPTRIM": true,
10  "FP_CORE_UTIL" : 45,
11  "CLOCK_PERIOD": 10.0,
12  "DESIGN_IS_CORE": true
13 }
```

The screenshot shows a web browser displaying the 'Flow Configuration Variables' page from the OpenLane documentation. The page title is 'Flow Configuration Variables' and the URL is <https://openlane.readthedocs.io/en/latest/reference/configuration.html>. The page content includes a navigation sidebar on the left with categories like 'Getting Started', 'OpenLane Architecture', 'Usage guides', 'Tutorials', and 'Reference Manual'. The main content area features a 'General' section with a 'Warning' box stating that the `include` directive is not supported in Verilog files. Below the warning is a table with two columns: 'Variable' and 'Description'. The table lists variables such as PDK, DESIGN_NAME, VERILOG_FILES, and CLOCK_PERIOD with their respective descriptions.

Variable	Description
PDK	Specifies the process design kit (PDK). (Default: sky130A)
DESIGN_NAME	The name of the top level module of the design
VERILOG_FILES	The path of the design's Verilog files, provided as an array of files in JSON or a delimited list of files in Tcl. The files are evaluated in order, i.e., if file B depends A must be listed first.
CLOCK_PERIOD	The clock period used for clocks in the design, in nanoseconds.

<https://openlane.readthedocs.io/en/latest/reference/configuration.html>

Edit Template to Fit Design Specs

After exiting the Docker run, we need to modify the `config.json` and add a `pin_order.cfg` to implement the design. These two files can be created using `gedit` commands as follows:

```
Terminal
gedit openlane/gcd/pin_order.cfg
```

```
pin_order.cfg
~/gcd_example/openlane/gcd
Save
config.json x pin_order.cfg x
1 #N
2 clk
3 reset
4 operands_val
5 #S
6 operands_bits_B.*
7 #W
8 operands_bits_A.*
9 operands_rdy
10 result_rdy
11 #E
12 result_bits_data.*
13 result_val
Plain Text Tab Width: 8 Ln 1, Col 1 INS
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Run Script

Terminal

#Start Docker

```
docker run -it -v $(pwd):/gcd_example -v $PDK_ROOT:$PDK_ROOT -e PDK=$PDK -e PDK_ROOT=$PDK_ROOT -u $(id -u $USER):$(id -g $USER) efabless/openlane:latest
```

#Enter Directory

```
cd /gcd_example/openlane
```

#Run OpenLane

```
flow.tcl -design gcd
```

Source: <https://unic-cass.github.io/02-env-setup.html>

If this happens... Its not the end of the world

```
Terminal
OpenLane Container (00caae2):/gcd_example/openlane$ flow.tcl -design gcd
OpenLane 00caae2b3186bb6b24a0b7db7ee1a6f056ad1702
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.

[ERROR]: The version of open_pdk used in building the PDK does not match the version OpenLane was tested on (installed:
78b7bc32ddb4b6f14f76883c2e2dc5b5de9d1cbc, tested: e3b630d9b7c0e23615367d52c4f78b2d2ede58ac)
This may introduce some issues. You may want to re-install the PDK by invoking `make pdk`.
[ERROR]: Please update your environment. OpenLane will now quit.
```

Exit docker and redo the following command with the correct version:

```
Terminal
//volare enable --pdk sky130 <version that appears on "tested">
volare enable --pdk sky130 e3b630d9b7c0e23615367d52c4f78b2d2ede58ac
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Run Again and Voilà

```
rodrigowue@JamesWebb: ~/ / x + v
OpenLane Container (00caae2):/gcd_example/openlane$ flow.tcl -design gcd
OpenLane 00caae2b3186bb6b24a0b7db7ee1a6f056ad1702
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 'gcd/config.json'...
[INFO]: PDK Root: /home/rodrigowue/unic-cass/pdks
[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: /gcd_example/openlane/gcd/runs/RUN_2023.08.30_00.53.30
[INFO]: Saving runtime environment...
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
[INFO]: Running linter (Verilator) (log: gcd/runs/RUN_2023.08.30_00.53.30/logs/synthesis/linter.log)...
[INFO]: 0 errors found by linter
[WARNING]: 2 warnings found by linter
[STEP 1]
[INFO]: Running Synthesis (log: gcd/runs/RUN_2023.08.30_00.53.30/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: gcd/runs/RUN_2023.08.30_00.53.30/logs/synthesis/2-sta.log)...[STEP 3
]
[INFO]: Running Initial Floorplanning (log: gcd/runs/RUN_2023.08.30_00.53.30/logs/floorplan/3-initial_fp.log)...
[INFO]: Floorplanned with width 235.52 and height 233.92.
[STEP 4]
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Installing Klayout for Viewing the .GDS

(exit docker before continuing)

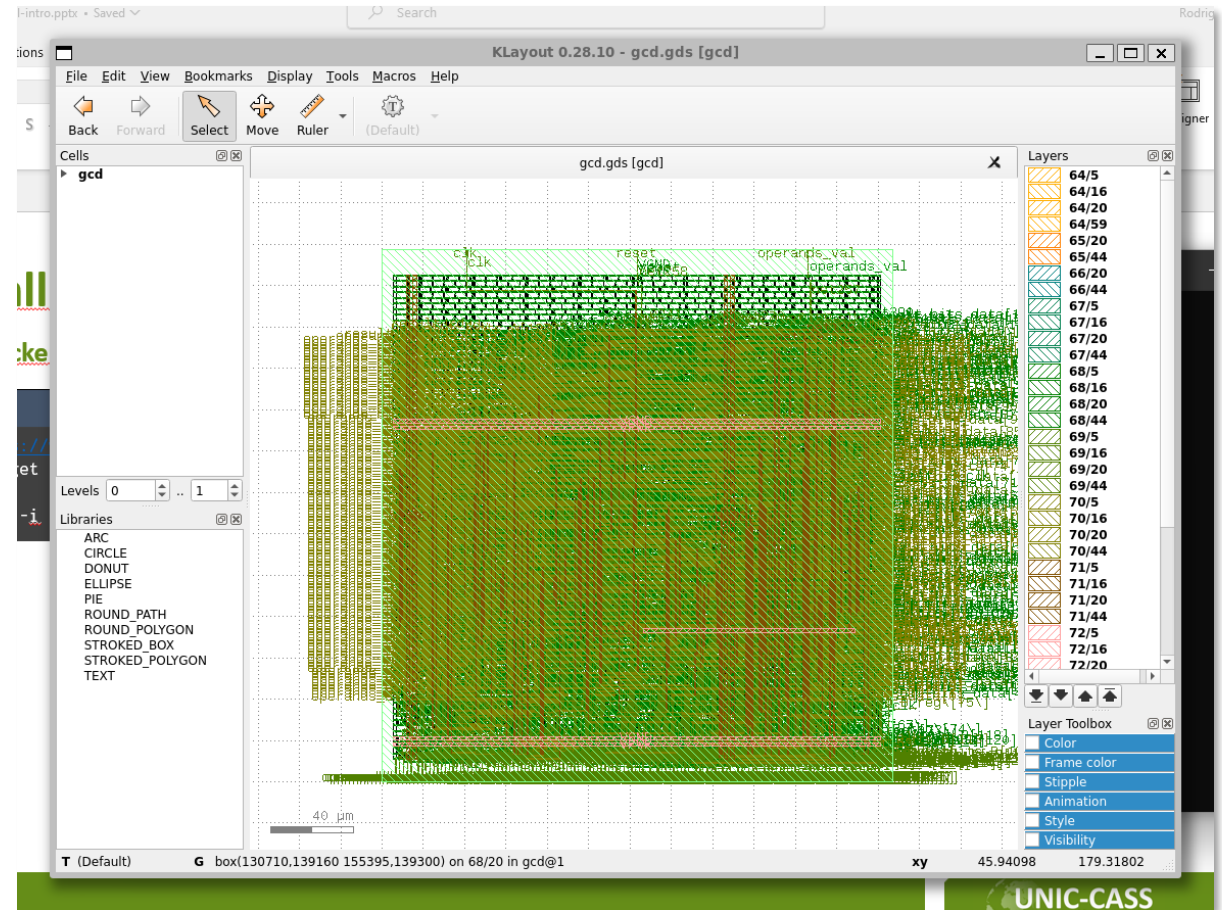
```
Terminal
wget https://www.klayout.org/downloads/Ubuntu-22/klayout_0.28.10-1_amd64.deb
sudo apt-get install libqt5designer5 libqt5multimedia5 libqt5opengl5 libqt5multimediawidgets5 \
    libqt5printsupport5 libqt5sql5 libqt5sql5 libqt5xmlpatterns5 libruby3.0
sudo dpkg -i klayout_0.28.10-1_amd64.deb
```

Source: <https://unic-cass.github.io/02-env-setup.html>

Opening <*.gds>

Terminal

```
klayout openlane/gcd/runs/RUN_*/results/signoff/gcd.gds
```



Source: <https://unic-cass.github.io/02-env-setup.html>

Thanks, any questions? :)

feel free to reach me out
rnwuerdig@inf.ufrgs.br

