Open-source ASIC Design

K.Gavaskar

Postdoc Researcher, Heidelberg University

ZITI, INF 368,

69120, Heidelberg, Germany.

Contents

Contents

- Introduction

Contents

- Introduction
- Contents of PDK

Contents

- Introduction
- Contents of PDK
- SkyWater Foundry Provided Cell Libraries

Contents

- Introduction
- Contents of PDK
- SkyWater Foundry Provided Cell Libraries
- Design Stages and Tools

Contents

- Introduction
- Contents of PDK
- SkyWater Foundry Provided Cell Libraries
- Design Stages and Tools
- OpenLane
  - Installing the OpenLane
  - Open Lane Architecture
  - Design Stages and Tools of OpenLane
  - Hardening Macros and Hardening the Core
  - Chip Level Integration
  - Results Directory Structure
  - Screenshots of Execution Flow and KLayout View

Contents

- Introduction
- Contents of PDK
- SkyWater Foundry Provided Cell Libraries
- Design Stages and Tools
- OpenLane
  - Installing the OpenLane
  - Open Lane Architecture
  - Design Stages and Tools of OpenLane
  - Hardening Macros and Hardening the Core
  - Chip Level Integration
  - Results Directory Structure
  - Screenshots of Execution Flow and KLayout View
- References

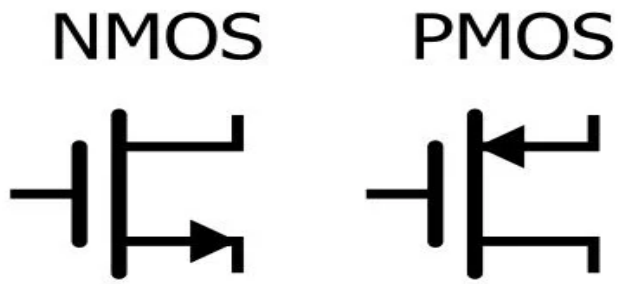
MOSFET

MOSFET

- Metal Oxide Semiconductor Field Effect Transistor
- Voltage controlled device
- To control connectivity
- High scalability
- Higher density
- High input impedance
- Low power consumption
- It is used to switch or amplify voltages
- Easy to manufacture

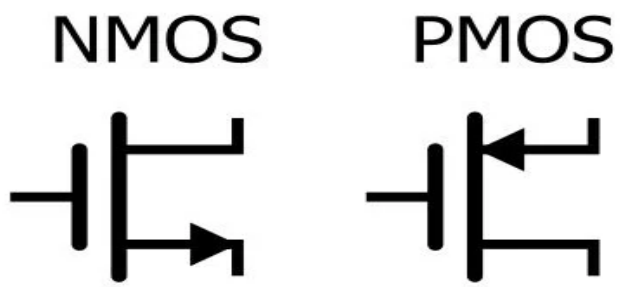
MOSFET

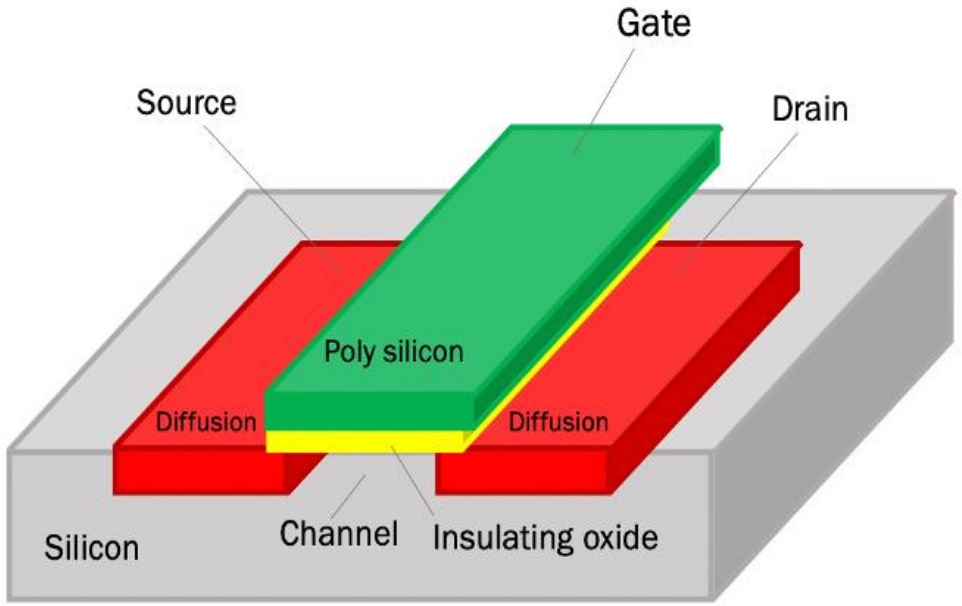
- Metal Oxide Semiconductor Field Effect Transistor
- Voltage controlled device
- To control connectivity
- High scalability
- Higher density
- High input impedance
- Low power consumption
- It is used to switch or amplify voltages
- Easy to manufacture



MOSFET

- Metal Oxide Semiconductor Field Effect Transistor
- Voltage controlled device
- To control connectivity
- High scalability
- Higher density
- High input impedance
- Low power consumption
- It is used to switch or amplify voltages
- Easy to manufacture





Introduction

Introduction

- Industry EDA tools

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design
  - PDKs

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design
  - PDKs (OSU (Oklahoma State University), Skywater, X-FAB and Global Foundries)

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design
  - PDKs (OSU (Oklahoma State University), Skywater, X-FAB and Global Foundries)
  - Electronic Design Automation (EDA) Tools and IPs

Introduction

- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design
  - PDKs (OSU (Oklahoma State University), Skywater, X-FAB and Global Foundries)
  - Electronic Design Automation (EDA) Tools and IPs
- Various VLSI stages

Introduction

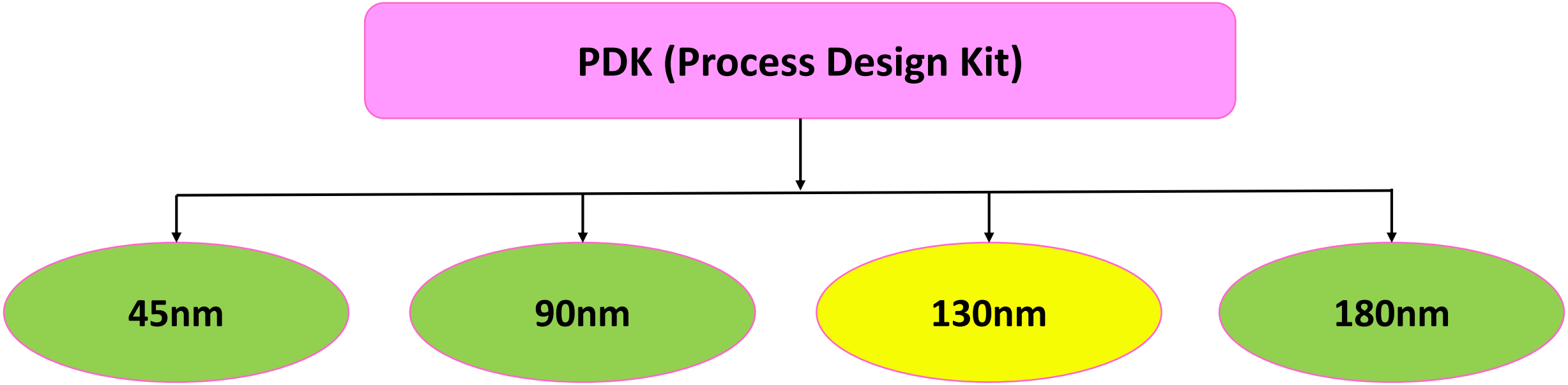
- Industry EDA tools (Tanner, Synopsys, Cadence, OrCAD & Siemens)
- Open Source - shadow of industry grade EDA tools
  - Software
  - Hardware
- **Needs:**
  - Design
  - PDKs (OSU (Oklahoma State University), Skywater, X-FAB and Global Foundries)
  - Electronic Design Automation (EDA) Tools and IPs
- Various VLSI stages
- ASIC with eFPGA

Contents of PDK

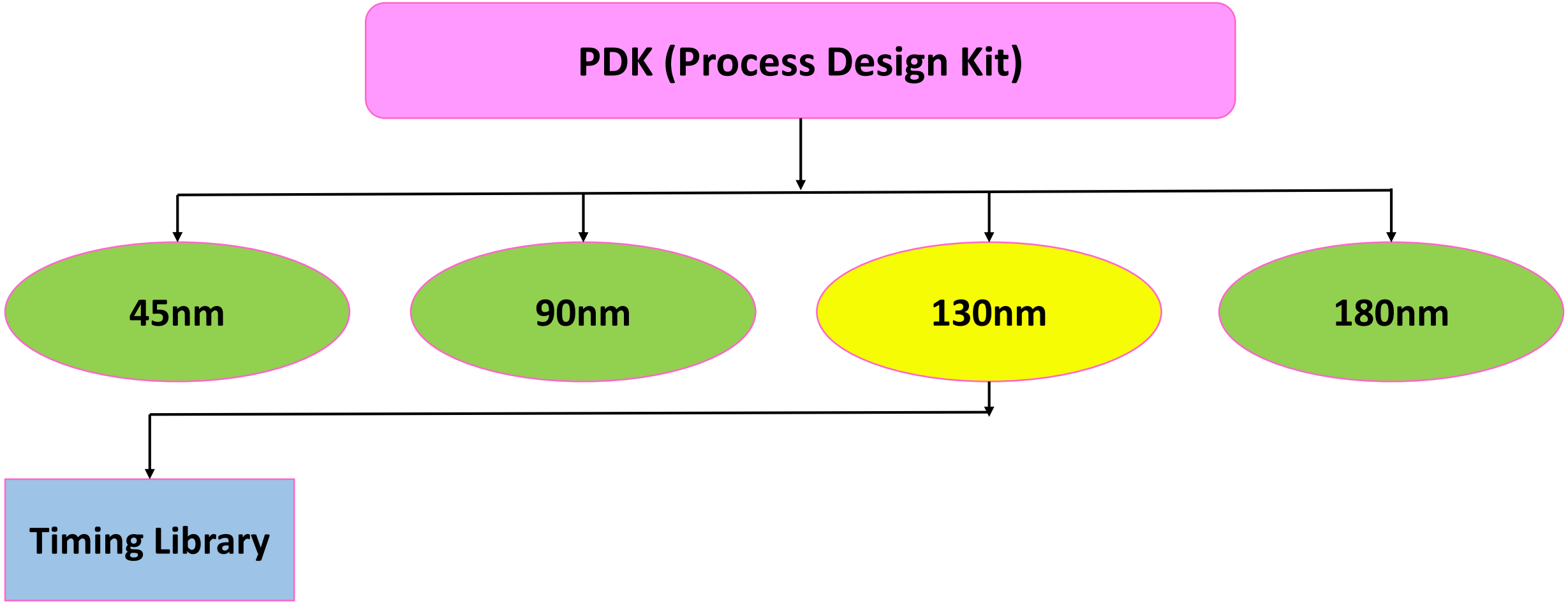
Contents of PDK

PDK (Process Design Kit)

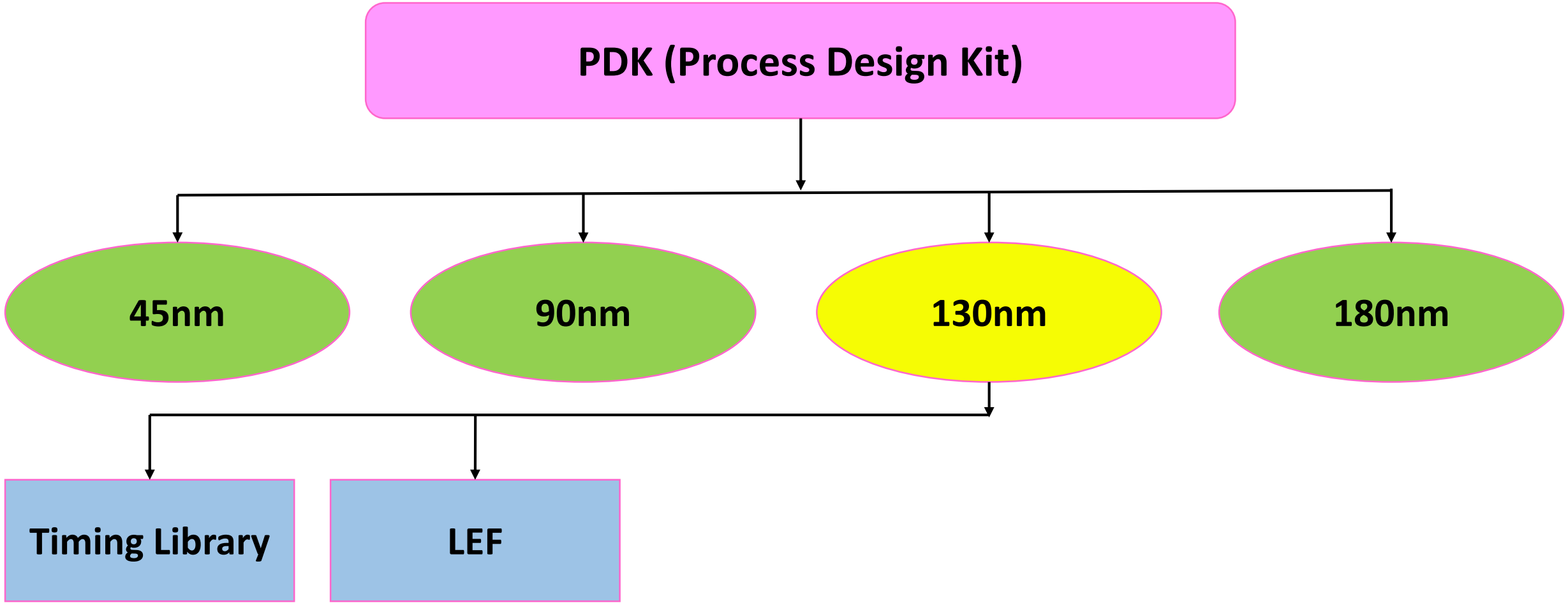
Contents of PDK



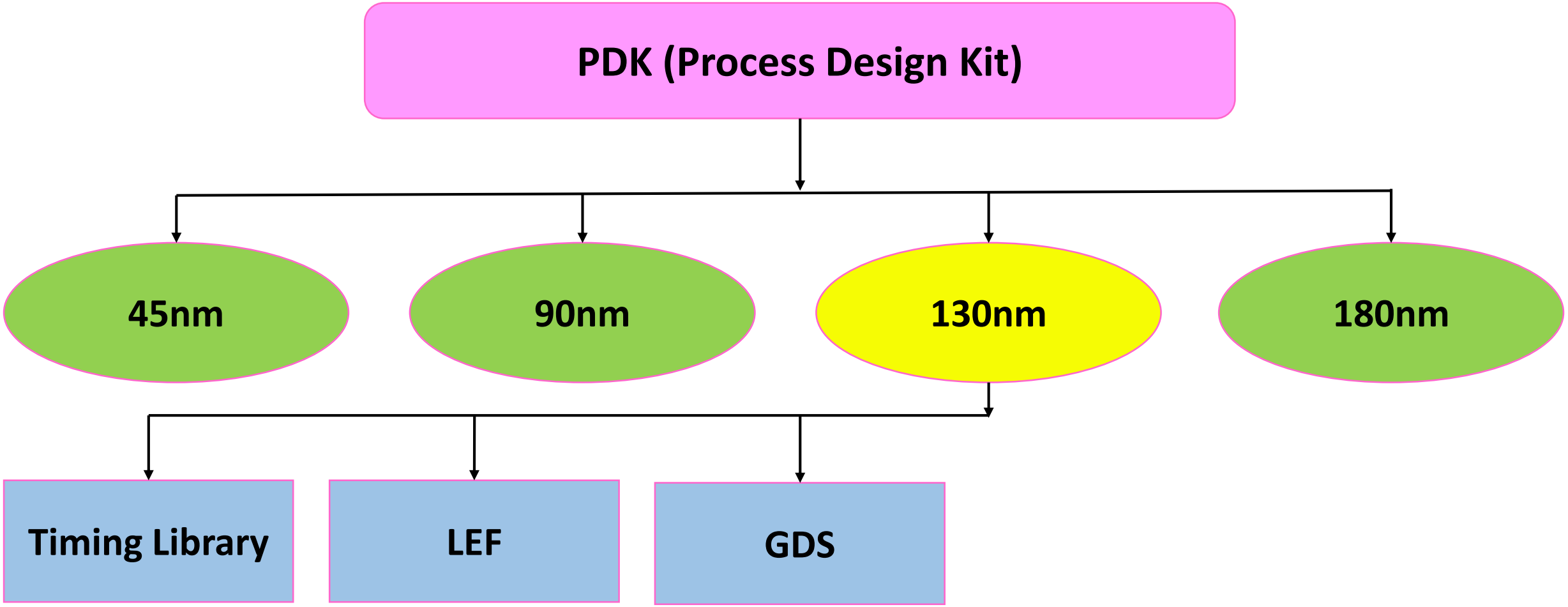
Contents of PDK



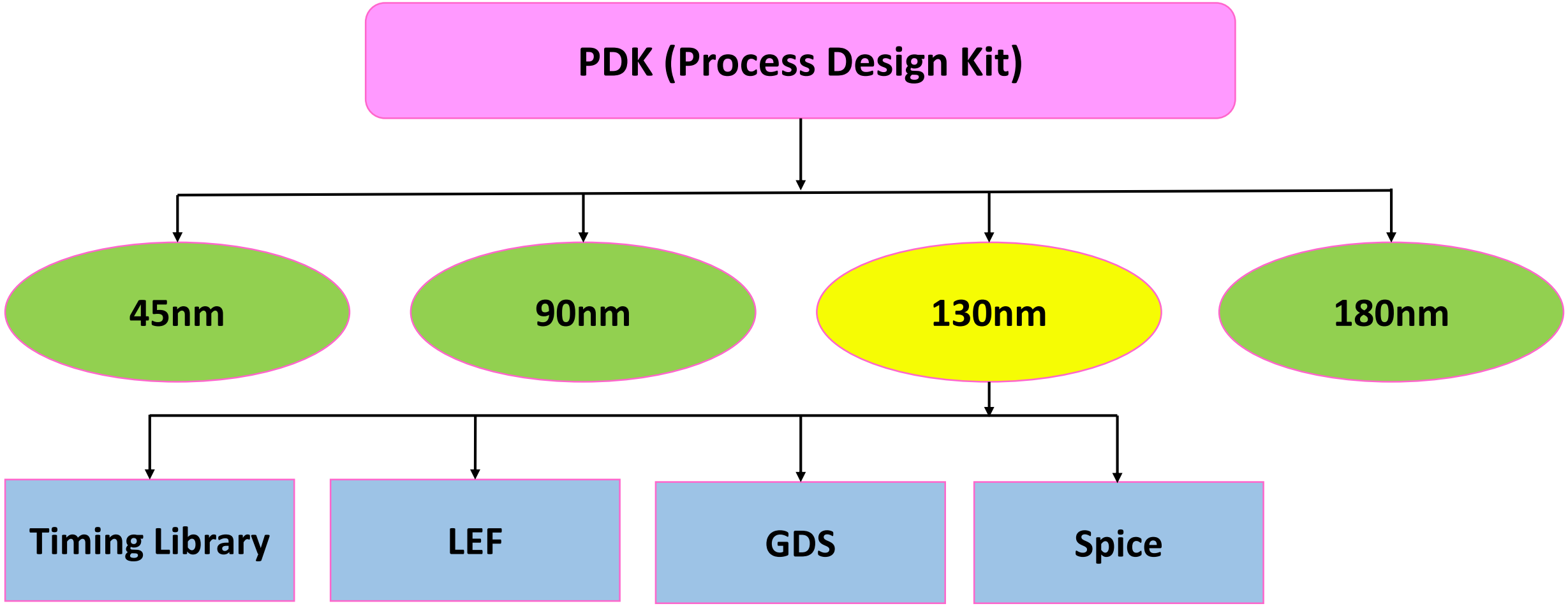
Contents of PDK



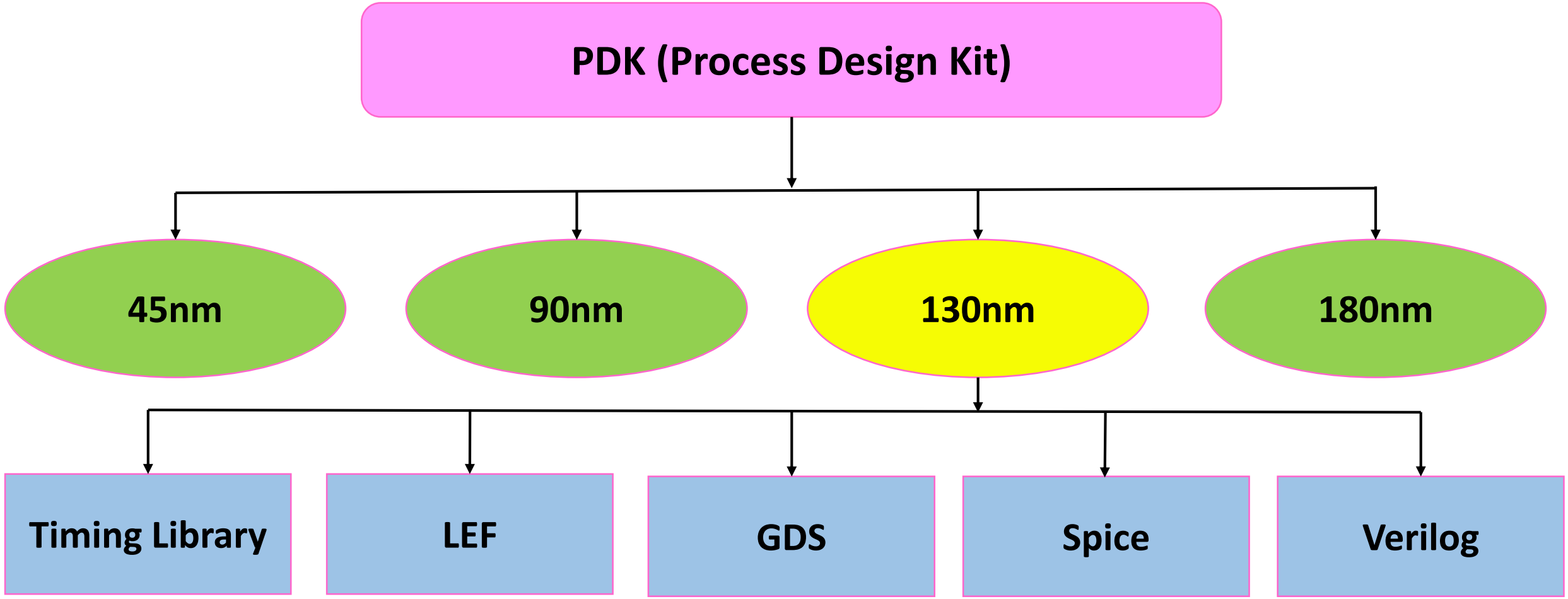
Contents of PDK



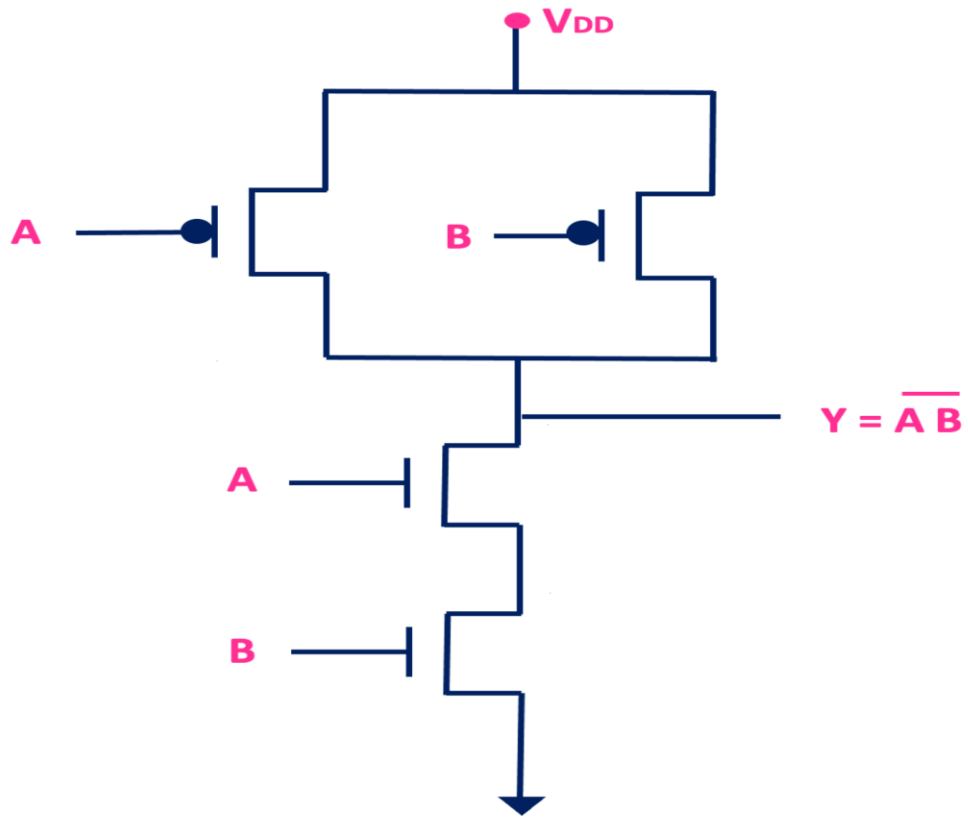
Contents of PDK



Contents of PDK



Example: Standard Cell (NAND Logic Gate)



Truth Table

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

SkyWater Foundry provided Cell Libraries

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms
- Sky130_fd_sc_ls

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms
- Sky130_fd_sc_ls
- Sky130_fd_sc_lp

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms
- Sky130_fd_sc_ls
- Sky130_fd_sc_lp
- Sky130_fd_sc_hd

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms
- Sky130_fd_sc_ls
- Sky130_fd_sc_lp
- Sky130_fd_sc_hd
- Sky130_fd_sc_hdll

SkyWater Foundry provided Cell Libraries

- Sky130_fd_sc_hs
- Sky130_fd_sc_ms
- Sky130_fd_sc_ls
- Sky130_fd_sc_lp
- Sky130_fd_sc_hd
- Sky130_fd_sc_hdll
- Sky130_fd_sc_hvl

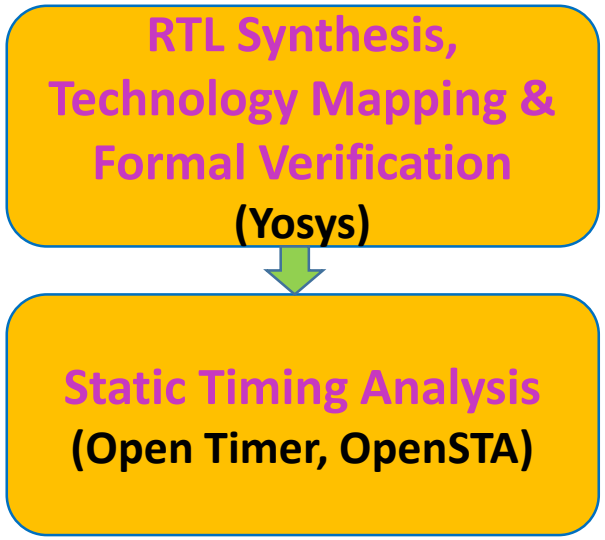
Link

Design Stages and Tools

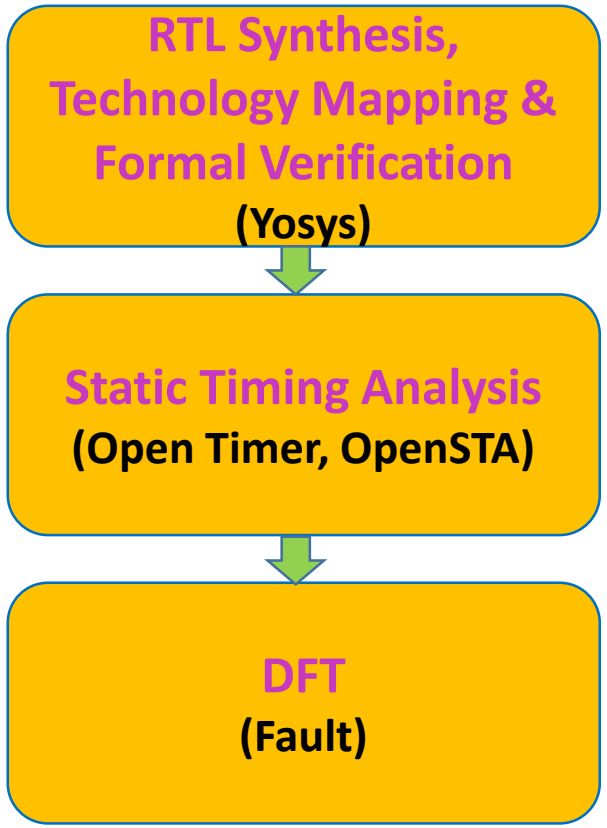
Design Stages and Tools

RTL Synthesis, Technology Mapping & Formal Verification
(Yosys)

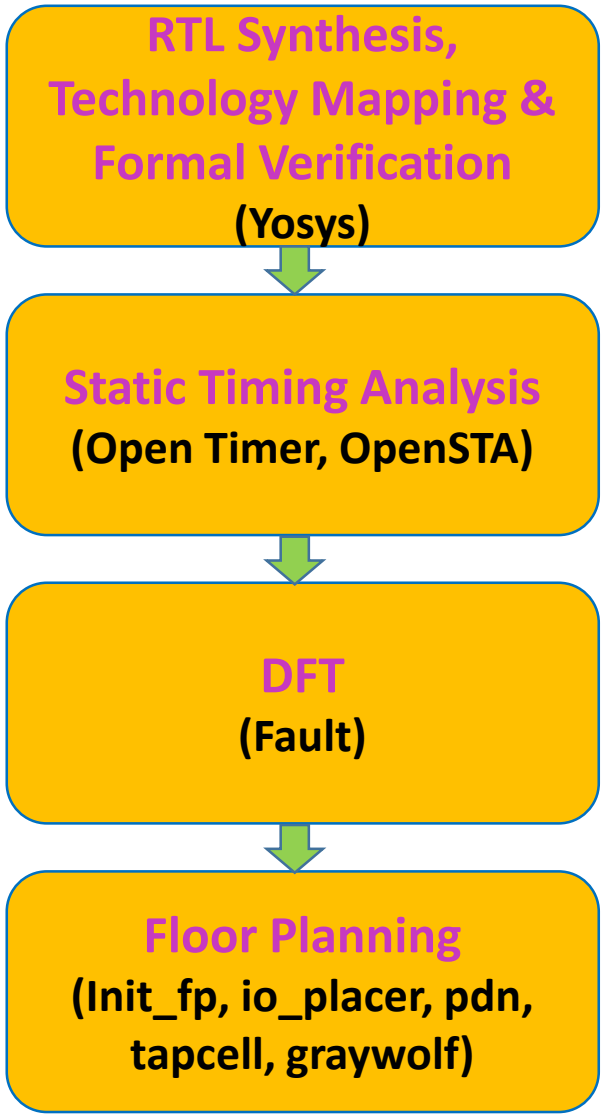
Design Stages and Tools



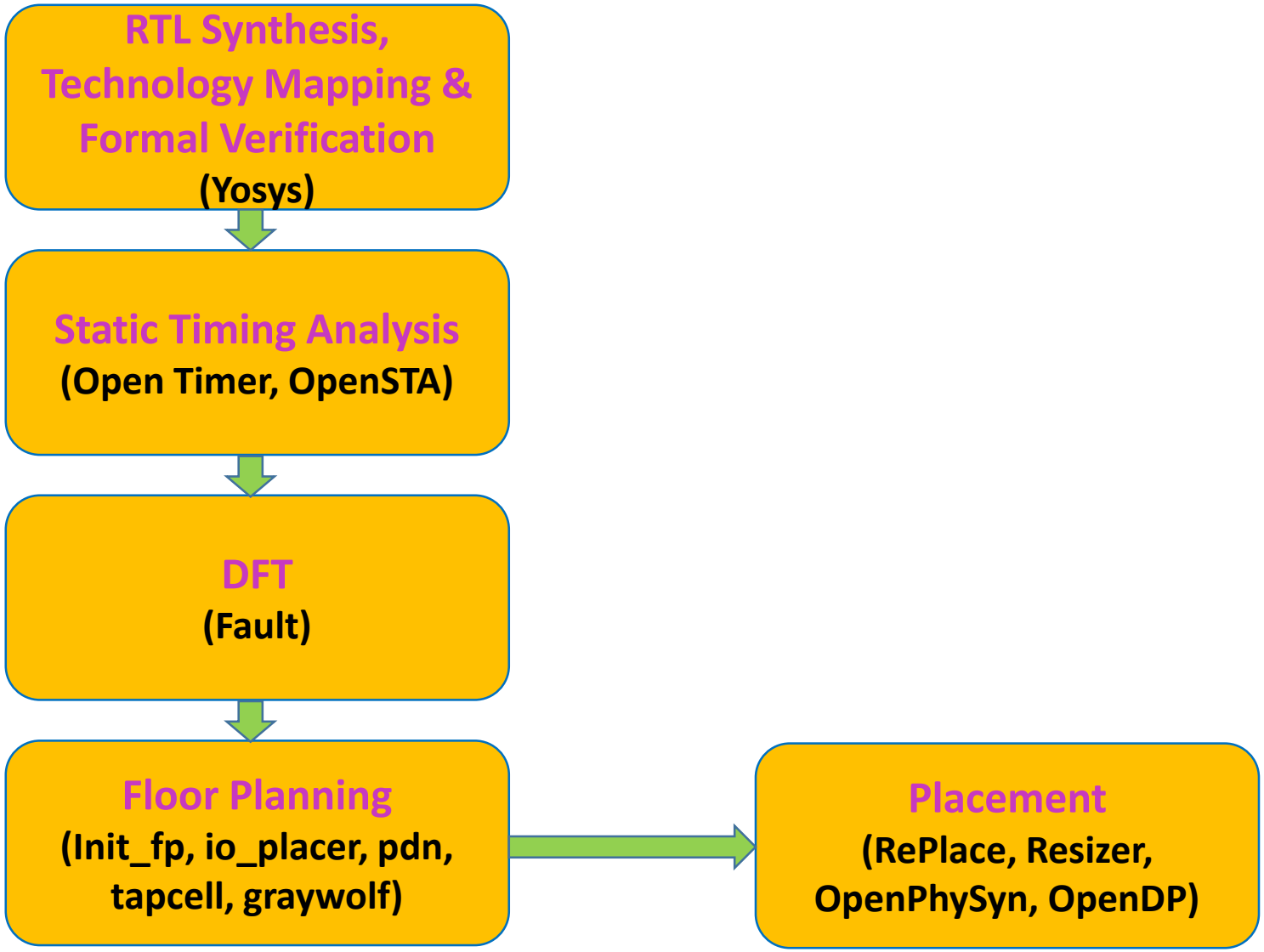
Design Stages and Tools



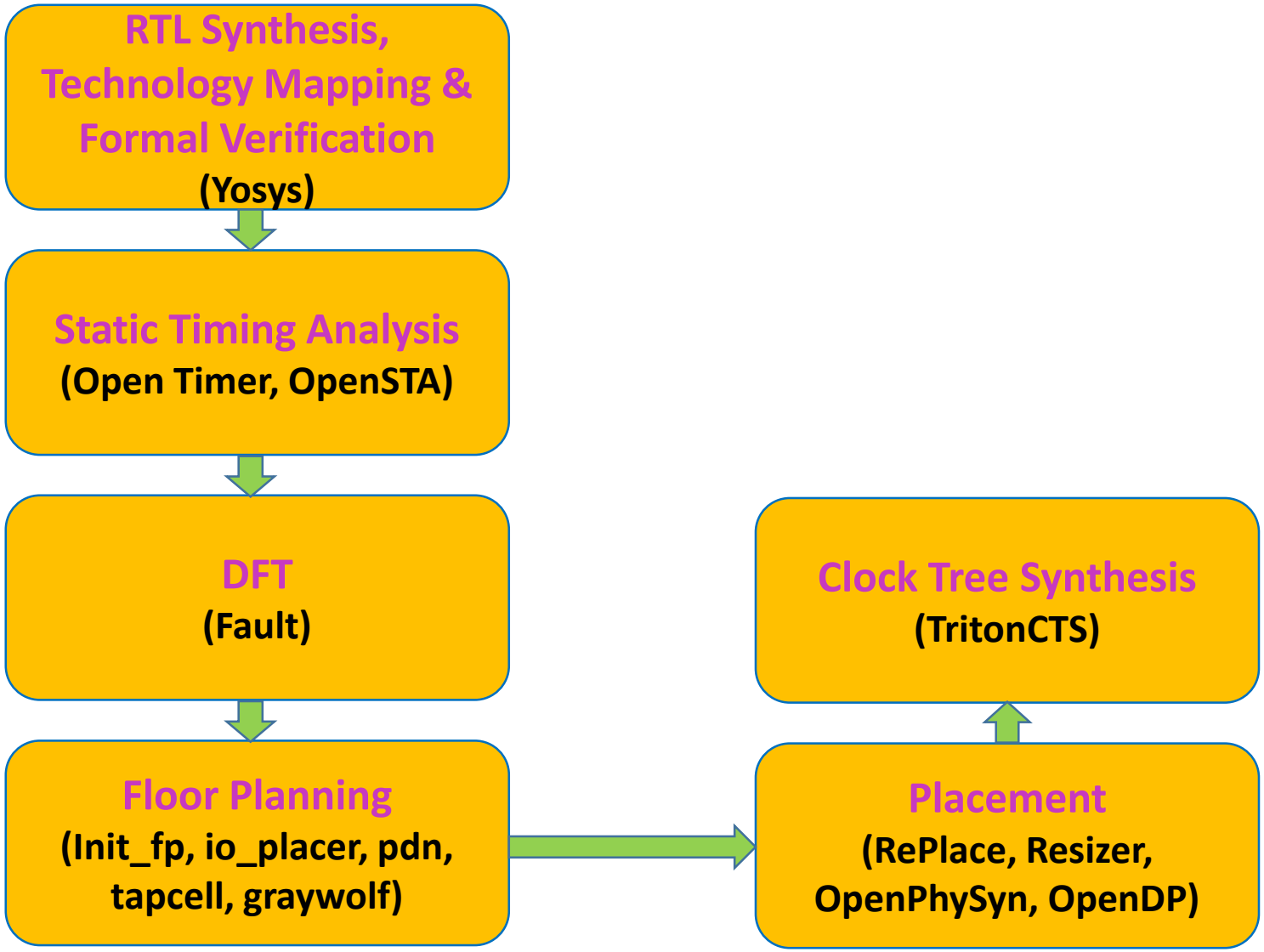
Design Stages and Tools



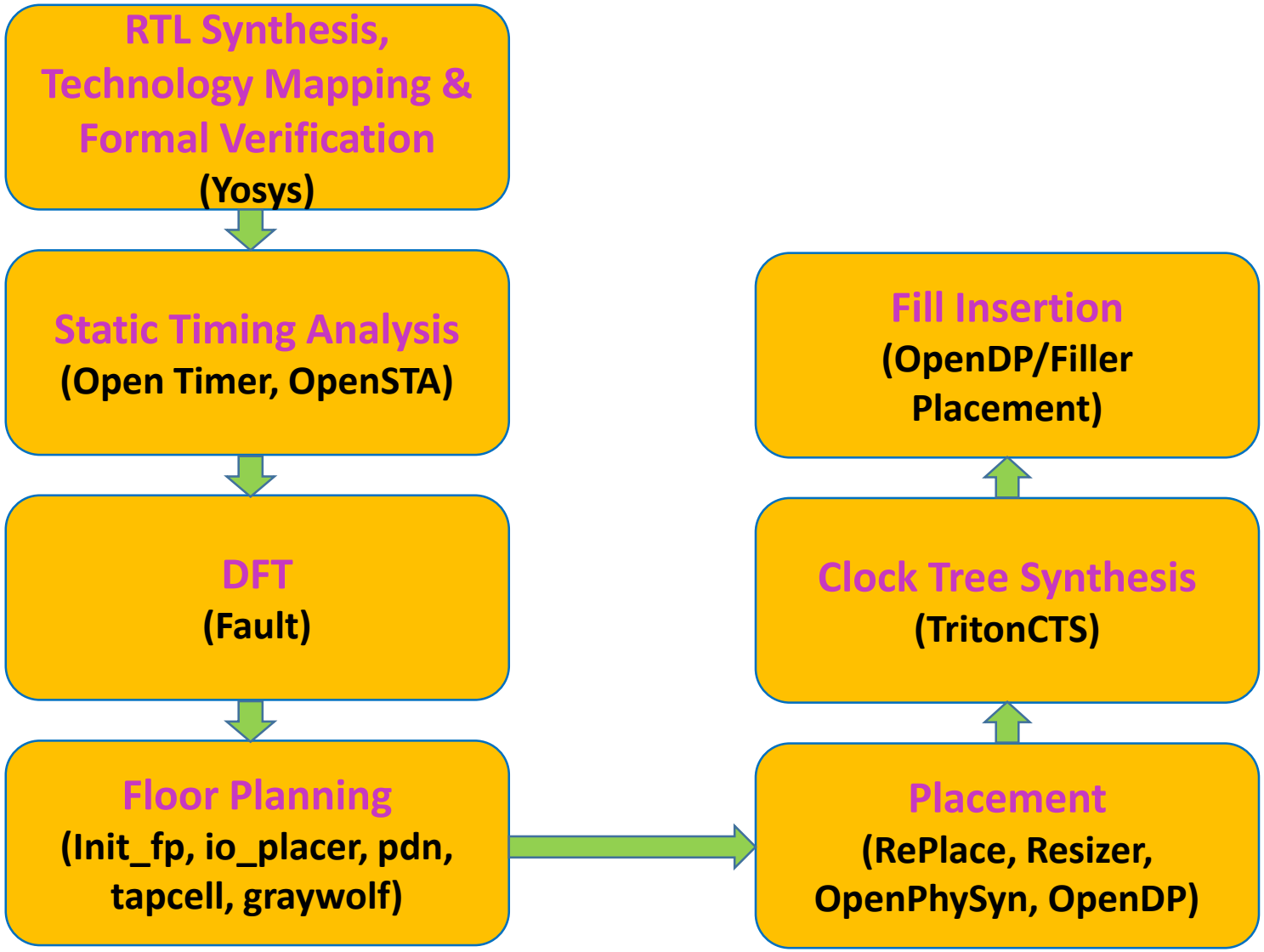
Design Stages and Tools



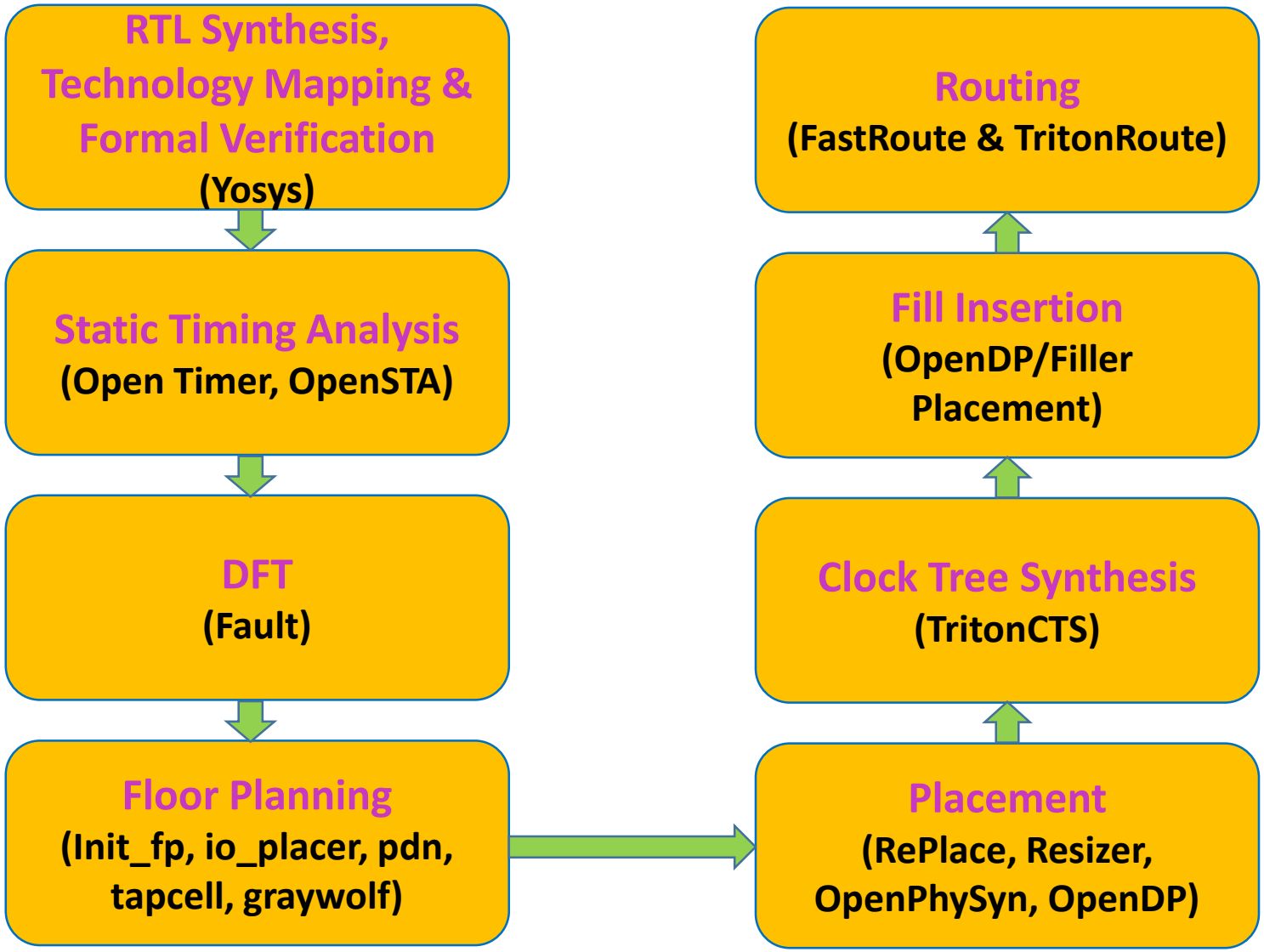
Design Stages and Tools



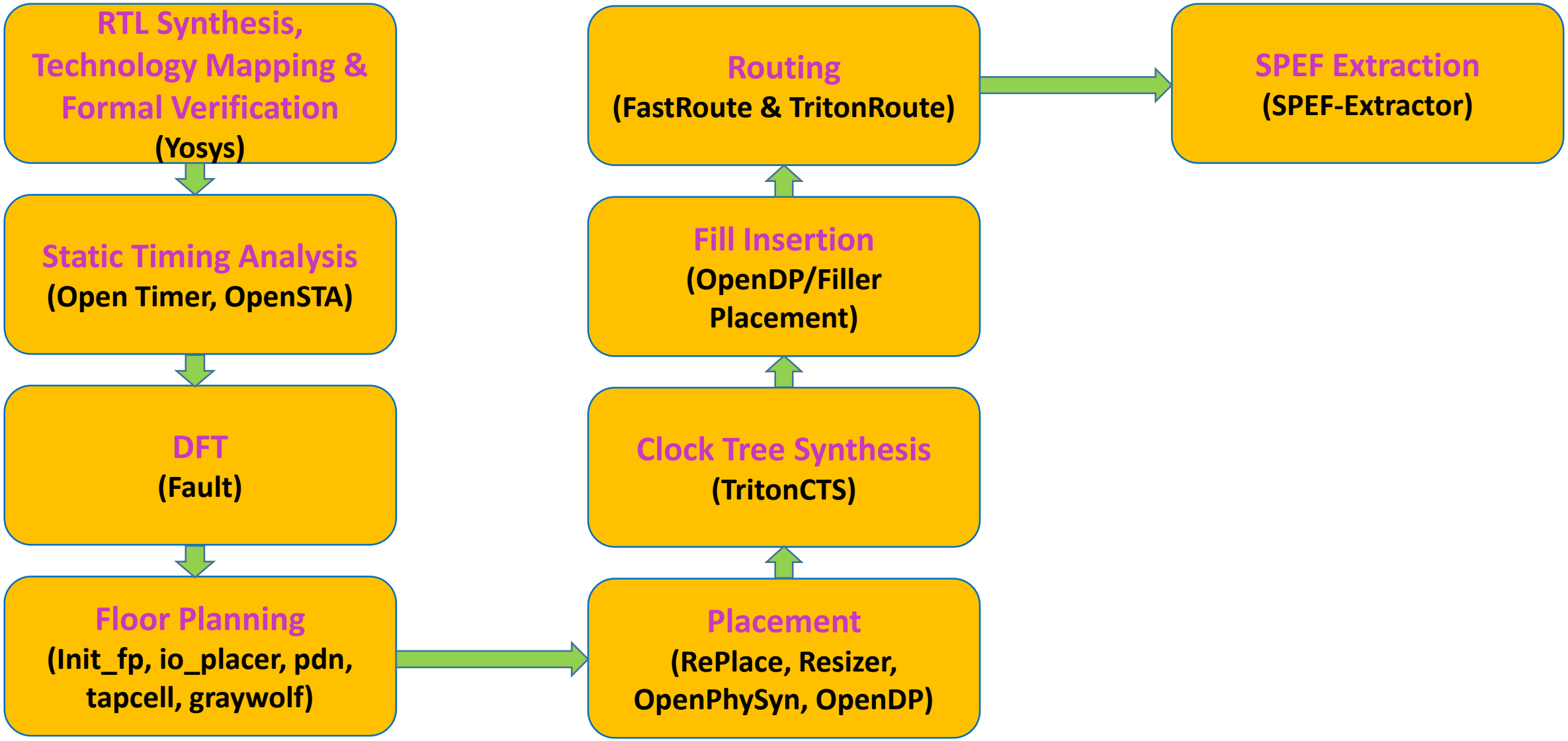
Design Stages and Tools



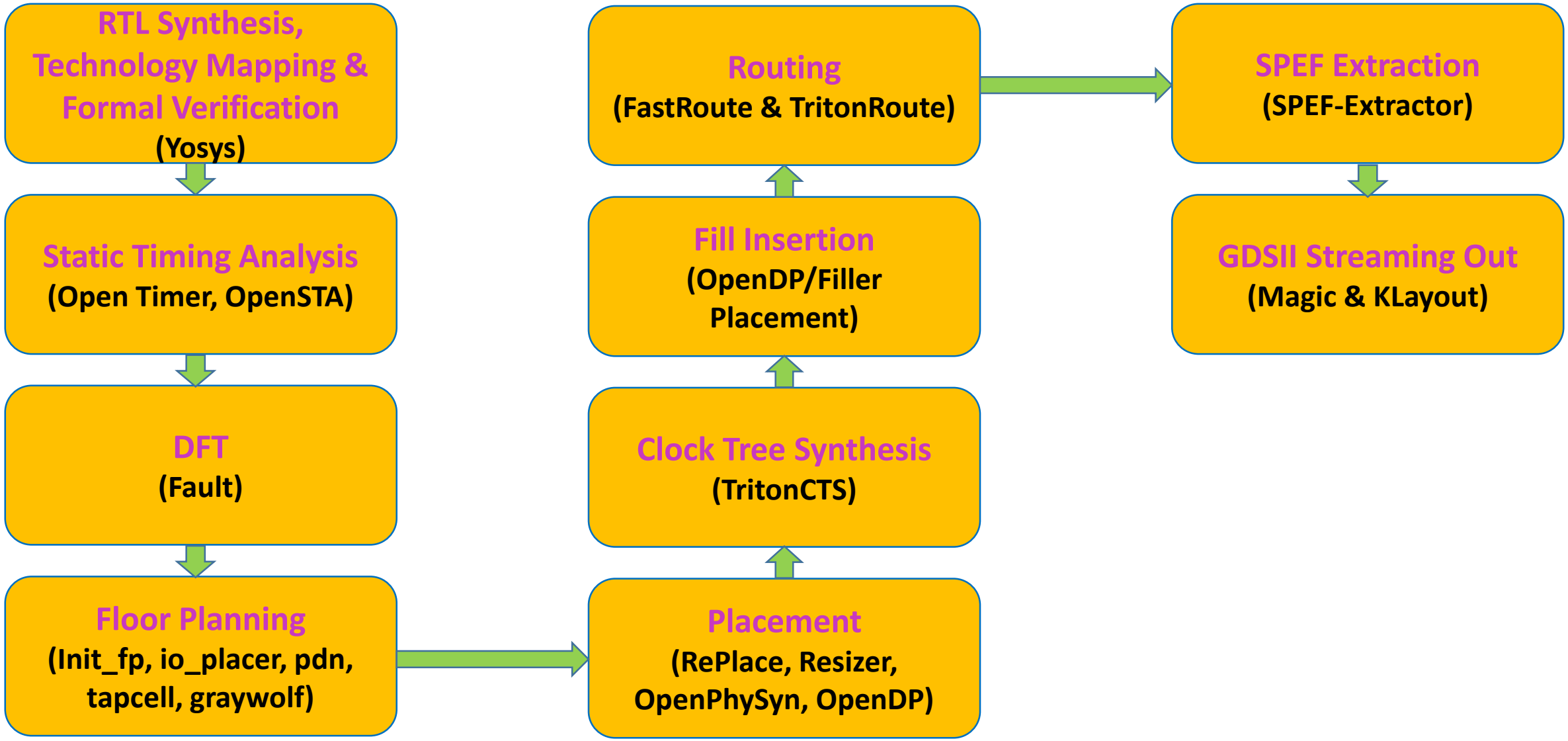
Design Stages and Tools



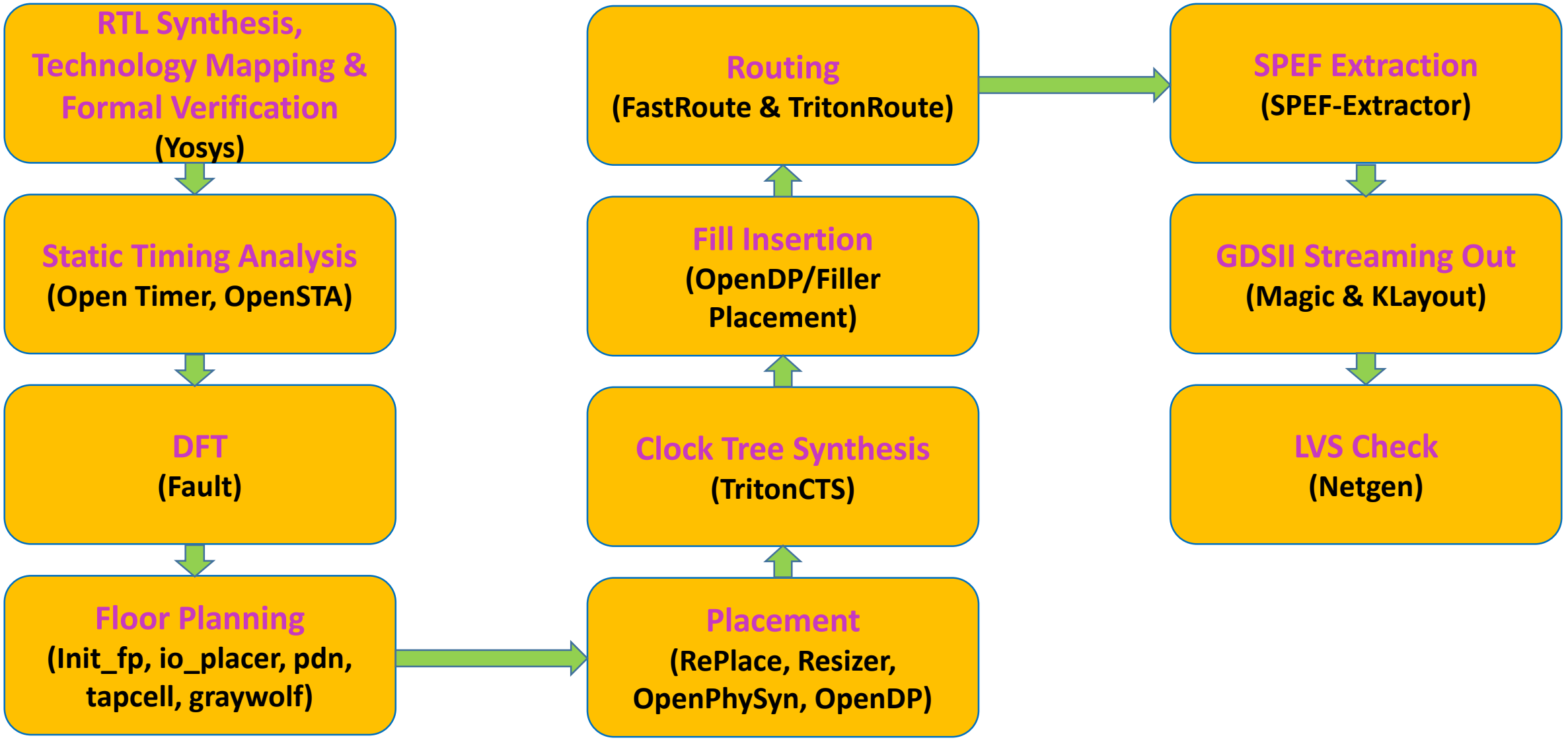
Design Stages and Tools



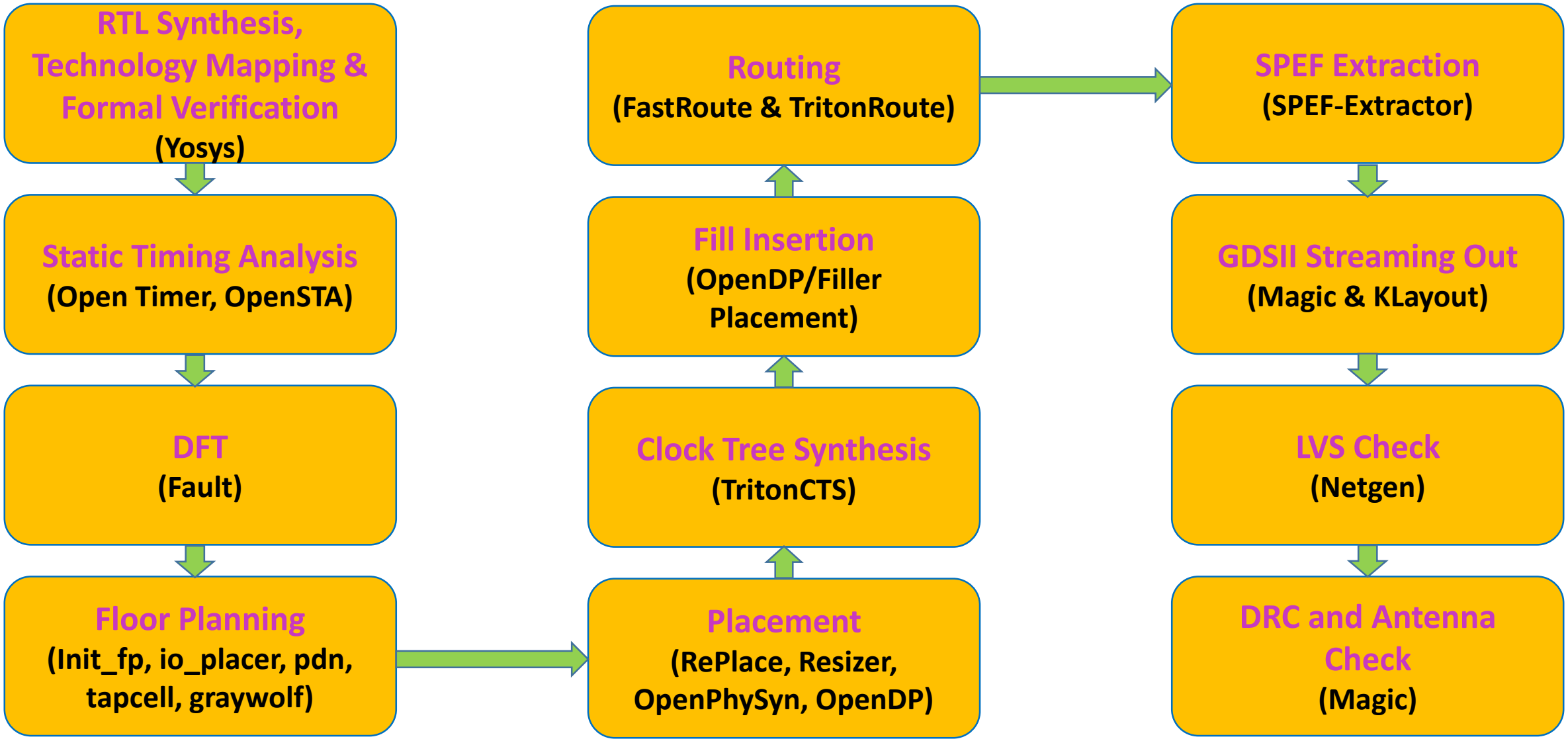
Design Stages and Tools



Design Stages and Tools



Design Stages and Tools



Important to Know

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements
  - **LVS** (Layout vs. Schematic)

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements
  - **LVS** (Layout vs. Schematic), **ERC** (Electric Rule Check),

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements
  - **LVS** (Layout vs. Schematic), **ERC** (Electric Rule Check), **Xor Check**

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements
  - **LVS** (Layout vs. Schematic), **ERC** (Electric Rule Check), **Xor Check** and **Antenna check**

Important to Know

- **RTL** (Register Transfer Level) – Flow of data b/w hardware registers
- **STA** (Static Timing Analysis) – Validating the timing performance
- **Tap cells** (well tap and decap) – To prevent the latch-up issue
- **CTS** (Clock Tree Synthesis) – Distributing the clock equally
- **LEC** (Logic Equivalent Check) – To ensure the logical equivalency
- **LEF** (Library Exchange Format) – Abstract view of cells
- **DEF** (Design Exchange Format) – To get the complete information about the cell
- **SPEF** (Standard Parasitic Exchange Format) – Represent parasitic data of wires
- **DRC** (Design Rule Check) – Ensures the design meets manufacturing requirements
  - **LVS** (Layout vs. Schematic), **ERC** (Electric Rule Check), **Xor Check** and **Antenna check**
- **GDS** (Graphic Data Stream) – The final output product of the IC design

OpenLane

- An automated RTL to GDSII flow
- All ASIC implementation
- Supports sky130 PDK, gf180mcu PDK, other PDK's
- Open source utilities
- Single configuration file
- Two main use cases: Hardening a macro and Integrating a SoC
- Open Lane does everything from logic synthesis, placement, routing, optimizations and evaluations
- One full stack flow to develop chips to be fabricated
- Successfully tape out a family of RISC-V based SoC's called striVe.

Hardening Macros

- Base Requirements

S.No	Key	JSON	TCL
1	DESIGN_NAME	“DESIGN_NAME”: sample	set ::env(DESIGN_NAME) {sample}
2	VEERILOG_FILES	“VEERILOG_FILES ”: “dir::src/*.v”,	set ::env(VERILOG_FILES) [glob $::env(DESIGN_DIR)/src/*.v]
3	CLOCK_PORT	“CLOCK_PORT”: null	set ::env(CLOCK_PORT) {clk}
4	DESIGN_IS_CORE	“DESIGN_IS_CORE”: false	set ::env(DESIGN_IS_CORE) {0}

Hardening the Core

- The chip core would usually have other macros inside it.
  - Synthesis
  - Static Timing Analysis
  - Floorplan
  - IO Placement
  - Placement
  - Clock Tree Synthesis
  - Power Grid/Power Distribution Network
  - Routing
  - GDS Streaming
  - Final Reports and Checks

S.No	Key
1	VERILOG_FILES
2	VERILOG_FILES_BLACKBOX
3	EXTRA_LEFS
4	EXTRA_LIBS
5	EXTRA_GDS_FILES
6	SYNTH_READ_BLACKBOX_LIB
7	MACRO_PLACEMENT_CFG

Link

Chip Level Integration

The methodology views the chip using the following hierarchy:

Chip Level Integration

The methodology views the chip using the following hierarchy:

- Chip Core
  - The hard macros
  - The rest of the design

Chip Level Integration

The methodology views the chip using the following hierarchy:

- Chip Core
  - The hard macros
  - The rest of the design
- Chip IO
  - IO Pads
  - Power Pads
  - Corner Pads

Chip Level Integration

The methodology views the chip using the following hierarchy:

- Chip Core
  - The hard macros
  - The rest of the design
- Chip IO
  - IO Pads
  - Power Pads
  - Corner Pads

The current methodology goes as follows:

- Hardening the macros
- Hardening the core
- Hardening the full chip
- Power and Signal routing (Macros and Core)

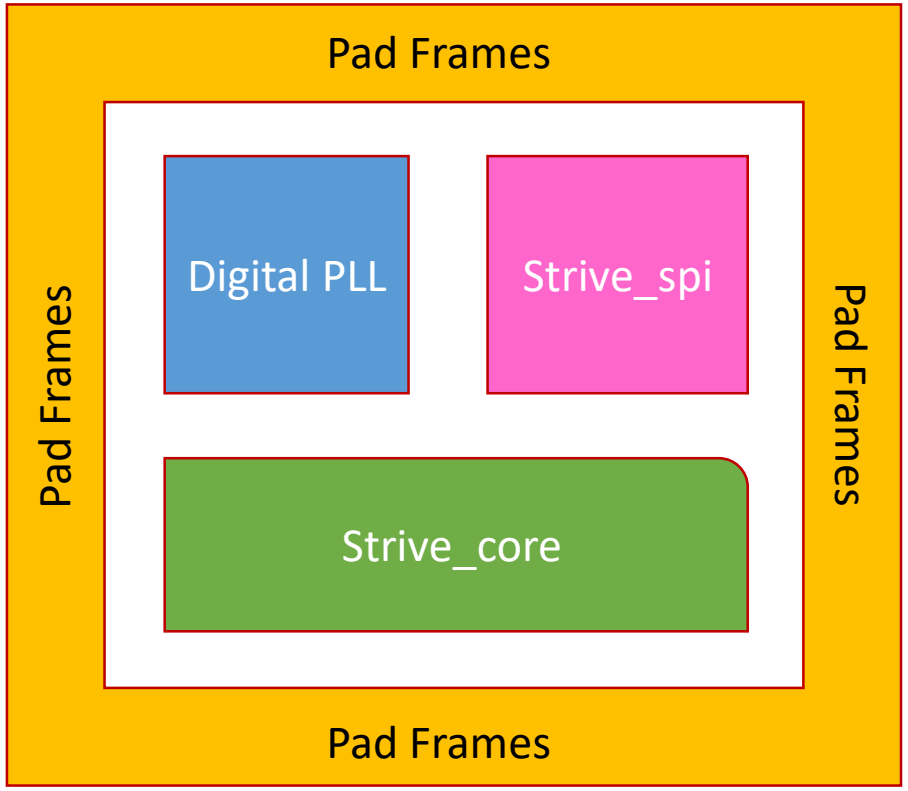
Chip Level Integration

The methodology views the chip using the following hierarchy:

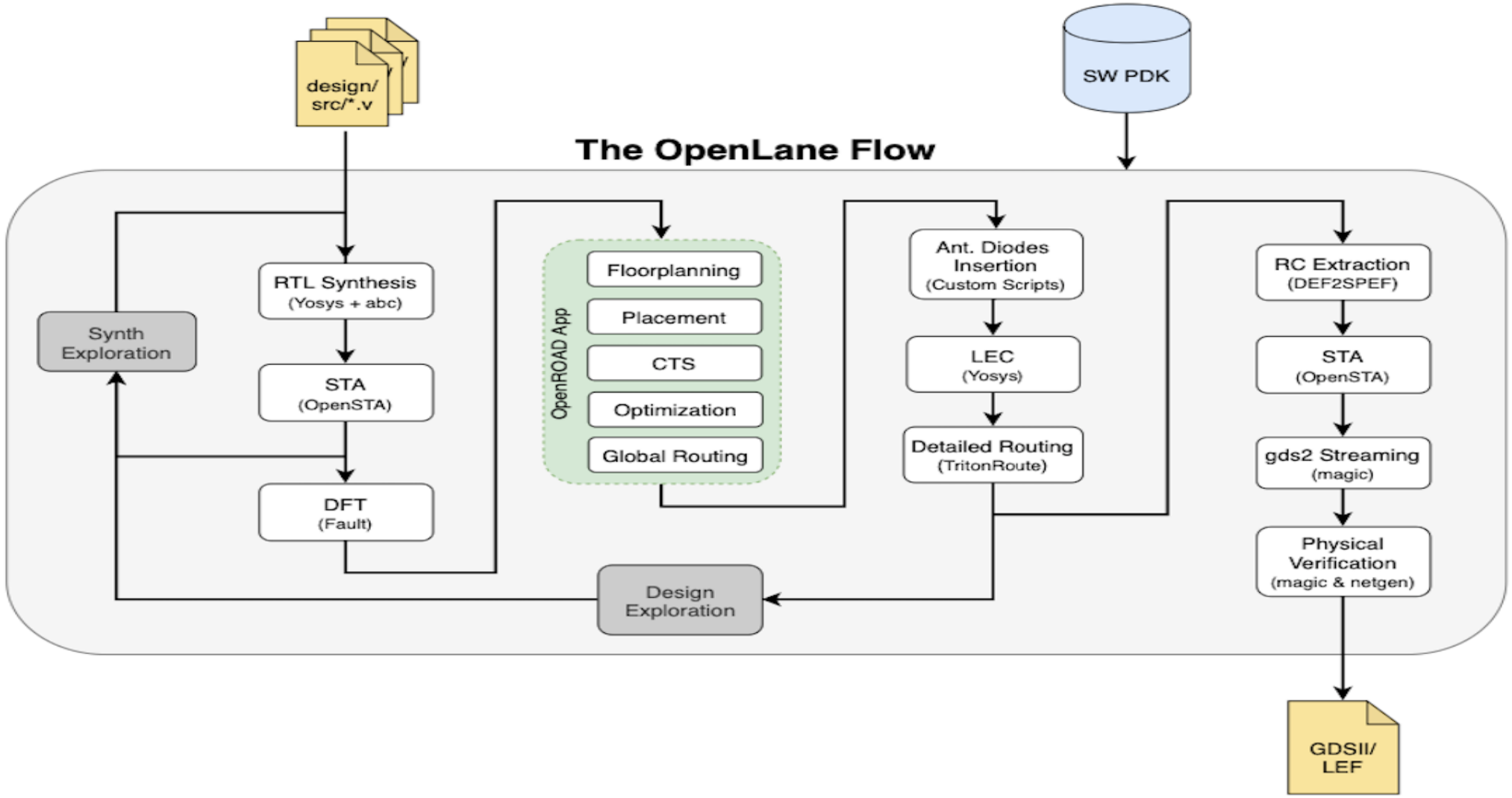
- Chip Core
  - The hard macros
  - The rest of the design
- Chip IO
  - IO Pads
  - Power Pads
  - Corner Pads

The current methodology goes as follows:

- Hardening the macros
- Hardening the core
- Hardening the full chip
- Power and Signal routing (Macros and Core)



OpenLane Architecture



Quick Start

Quick Start

Quick Start

Quick Start

Installing OpenLane

Prerequisites:

1. Linux is preferred. WSL to make installation easy.
2. Docker
3. Git
4. Python 3.6 or higher (venv and Pip)
5. GNU Make

Link

Screenshots of Execution flow

```
# user_project_wrapper
mkdir -p ./user_project_wrapper/runs/23_06_22_04_04
rm -rf ./user_project_wrapper/runs/user_project_wrapper
ln -s $(realpath ./user_project_wrapper/runs/23_06_22_04_04) ./user_project_wrapper/run
s/user_project_wrapper
docker run -it -v $(realpath /home/user/caravel1/..):$(realpath /home/user/caravel1/..)
 -v /home/user/caravel1/depen/pdks:/pdk -v /home/user/caravel1/caravel:/home/user/carav
el1/caravel -v /home/user/caravel1/depen/openlane_src:/openlane -v /home/user/caravel1/
mgmt_core_wrapper:/home/user/caravel1/mgmt_core_wrapper -e PDK_ROOT=/pdk -e PDK=sky130A
 -e MISMATCHES_OK=1 -e CARAVEL_ROOT=/home/user/caravel1/caravel -e OPENLANE_RUN_TAG=23_
06_22_04_04 -e MCW_ROOT=/home/user/caravel1/mgmt_core_wrapper -u 1000:1000 \
        efabless/openlane:2023.02.23 sh -c "flow.tcl -design $(realpath ./user_project_
wrapper) -save_path $(realpath ..) -save -tag 23_06_22_04_04 -overwrite -ignore_mismatc
hes"
OpenLane a35b64aa200c91e9eb7dde56db787d6b4c0ea12a
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 '../home/user/caravel1/openlane/user_project_wrapper/con
fig.tcl'...
[INFO]: PDK Root: /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: /home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_
04_04
[INFO]: Preparing LEF files for the nom corner...
[INFO]: Preparing LEF files for the min corner...
[INFO]: Preparing LEF files for the max corner...
```

Contd.,

```
[STEP 1]
[INFO]: Running Synthesis (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/synthesis/1-synthesis.log)...
[STEP 2]
[INFO]: Running Single-Corner Static Timing Analysis (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/synthesis/2-sta.log)...
[INFO]: Creating a netlist with power/ground pins.
[STEP 3]
[INFO]: Running Initial Floorplanning (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/floorplan/3-initial_fp.log)...
[INFO]: Floorplanned with width 2908.58 and height 3497.92.
[STEP 4]
[INFO]: Running IO Placement (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/floorplan/4-place_io.log)...
[STEP 5]
[INFO]: Performing Manual Macro Placement (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/placement/5-macro_placement.log)...
[STEP 6]
[INFO]: Running Tap/Decap Insertion (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/floorplan/6-tap.log)...
[INFO]: Power planning with power {vccd1 vdda1 vdda2 vccd2} and ground {vssd1 vssa1 vssa2 vssd2}...
[STEP 7]
[INFO]: Generating PDN (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/floorplan/7-pdn.log)...
[STEP 8]
[INFO]: Running Global Placement (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/placement/8-global.log)...
```

Contd.,

```
[STEP 9]
[INFO]: Running Placement Resizer Design Optimizations (log: ../home/user/caravel1/open
lane/user_project_wrapper/runs/23_06_22_04_04/logs/placement/9-resizer.log)...
[STEP 10]
[INFO]: Running Detailed Placement (log: ../home/user/caravel1/openlane/user_project_wr
apper/runs/23_06_22_04_04/logs/placement/10-detailed.log)...
[STEP 11]
[INFO]: Running Clock Tree Synthesis (log: ../home/user/caravel1/openlane/user_project_
wrapper/runs/23_06_22_04_04/logs/cts/11-cts.log)...
[STEP 12]
[INFO]: Running Placement Resizer Timing Optimizations (log: ../home/user/caravel1/open
lane/user_project_wrapper/runs/23_06_22_04_04/logs/cts/12-resizer.log)...
[STEP 13]
[INFO]: Running Global Routing Resizer Timing Optimizations (log: ../home/user/caravel1
/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/routing/13-resizer.log)...
[STEP 14]
[INFO]: Running Diode Insertion (log: ../home/user/caravel1/openlane/user_project_wrapp
er/runs/23_06_22_04_04/logs/routing/14-obs.log)...
[STEP 15]
[INFO]: Running Detailed Placement (log: ../home/user/caravel1/openlane/user_project_wr
apper/runs/23_06_22_04_04/logs/routing/15-diode_legalization.log)...
[STEP 16]
[INFO]: Running Fill Insertion (log: ../home/user/caravel1/openlane/user_project_wrappe
r/runs/23_06_22_04_04/logs/routing/16-fill.log)...
[STEP 17]
[INFO]: Running Global Routing (log: ../home/user/caravel1/openlane/user_project_wrappe
r/runs/23_06_22_04_04/logs/routing/17-global.log)...
```

Contd.,

```
[STEP 18]
[INFO]: Writing Verilog (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/
23_06_22_04_04/logs/routing/17-global_write_netlist.log)...
[STEP 19]
[INFO]: Running Detailed Routing (log: ../home/user/caravel1/openlane/user_project_wrap
per/runs/23_06_22_04_04/logs/routing/19-detailed.log)...
[INFO]: No DRC violations after detailed routing.
[STEP 20]
[INFO]: Checking Wire Lengths (log: ../home/user/caravel1/openlane/user_project_wrapper
/runs/23_06_22_04_04/logs/routing/20-wire_lengths.log)...
[STEP 21]
[INFO]: Running SPEF Extraction at the min process corner (log: ../home/user/caravel1/o
penlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/21-parasitics_extraction.
min.log)...
[STEP 22]
[INFO]: Running Multi-Corner Static Timing Analysis at the min process corner (log: ../
home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/22-rc
x_mcsta.min.log)...
[STEP 23]
[INFO]: Running SPEF Extraction at the max process corner (log: ../home/user/caravel1/o
penlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/23-parasitics_extraction.
max.log)...
[STEP 24]
[INFO]: Running Multi-Corner Static Timing Analysis at the max process corner (log: ../
home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/24-rc
x_mcsta.max.log)...
```

Contd.,

```
[STEP 25]
[INFO]: Running SPEF Extraction at the nom process corner (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/25-parasitics_extraction.nom.log)...
[STEP 26]
[INFO]: Running Multi-Corner Static Timing Analysis at the nom process corner (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/26-rcx_mcsta.nom.log)...
[STEP 27]
[INFO]: Running Single-Corner Static Timing Analysis at the nom process corner (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/27-rcx_sta.log)...
[STEP 28]
[INFO]: Creating IR Drop Report (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/28-irdrop.log)...
[STEP 29]
[INFO]: Running Magic to generate various views...
[INFO]: Streaming out GDSII with Magic (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/29-gdsii.log)...
[INFO]: Generating MAGLEF views...
[STEP 30]
[INFO]: Streaming out GDSII with KLayout (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/30-gdsii-klayout.log)...
[STEP 31]
[INFO]: Running Magic Spice Export from LEF (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/31-spice.log)...
```

Contd.,

```
[STEP 29]
[INFO]: Running Magic to generate various views...
[INFO]: Streaming out GDSII with Magic (log: ../home/user/caravel1/openlane/user_projec
t_wrapper/runs/23_06_22_04_04/logs/signoff/29-gdsii.log)...
[INFO]: Generating MAGLEF views...
[STEP 30]
[INFO]: Streaming out GDSII with KLayout (log: ../home/user/caravel1/openlane/user_proj
ect_wrapper/runs/23_06_22_04_04/logs/signoff/30-gdsii-klayout.log)...
[STEP 31]
[INFO]: Running Magic Spice Export from LEF (log: ../home/user/caravel1/openlane/user_p
roject_wrapper/runs/23_06_22_04_04/logs/signoff/31-spice.log)...
[STEP 32]
[INFO]: Writing Powered Verilog (logs: ../home/user/caravel1/openlane/user_project_wrap
per/runs/23_06_22_04_04/logs/signoff/32-write_powered_def.log, ../home/user/caravel1/op
enlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/32-write_powered_verilog.l
og)...
[STEP 33]
[INFO]: Writing Verilog (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/
23_06_22_04_04/logs/signoff/32-write_powered_verilog.log)...
[STEP 34]
[INFO]: Running LVS (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_0
6_22_04_04/logs/signoff/34-lvs.lef.log)...
[STEP 35]
[INFO]: Running Magic DRC (log: ../home/user/caravel1/openlane/user_project_wrapper/run
s/23_06_22_04_04/logs/signoff/35-drc.log)...
[INFO]: Converting Magic DRC database to various tool-readable formats...
```

Contd.,

```
[STEP 36]
[INFO]: Running Magic DRC (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/36-drc.log)...
[INFO]: Converting Magic DRC database to various tool-readable formats...

[STEP 37]
[INFO]: Running OpenROAD Antenna Rule Checker (log: ../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/logs/signoff/37-antenna.log)...
[INFO]: Saving current set of views in '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/results/final'...
[INFO]: Saving current set of views in '../home/user/caravel1'...
[INFO]: Saving runtime environment...
[INFO]: Generating final set of reports...
[INFO]: Created manufacturability report at '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/reports/manufacturability.rpt'.
[INFO]: Created metrics report at '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/reports/metrics.csv'.
[WARNING]: There are max slew violations in the design at the typical corner. Please refer to '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/reports/signoff/27-rcx_sta.slew.rpt'.
[WARNING]: There are max fanout violations in the design at the typical corner. Please refer to '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/reports/signoff/27-rcx_sta.slew.rpt'.
[WARNING]: There are max capacitance violations in the design at the typical corner. Please refer to '../home/user/caravel1/openlane/user_project_wrapper/runs/23_06_22_04_04/reports/signoff/27-rcx_sta.slew.rpt'.
[INFO]: There are no hold violations in the design at the typical corner.
[INFO]: There are no setup violations in the design at the typical corner.
[SUCCESS]: Flow complete.
```

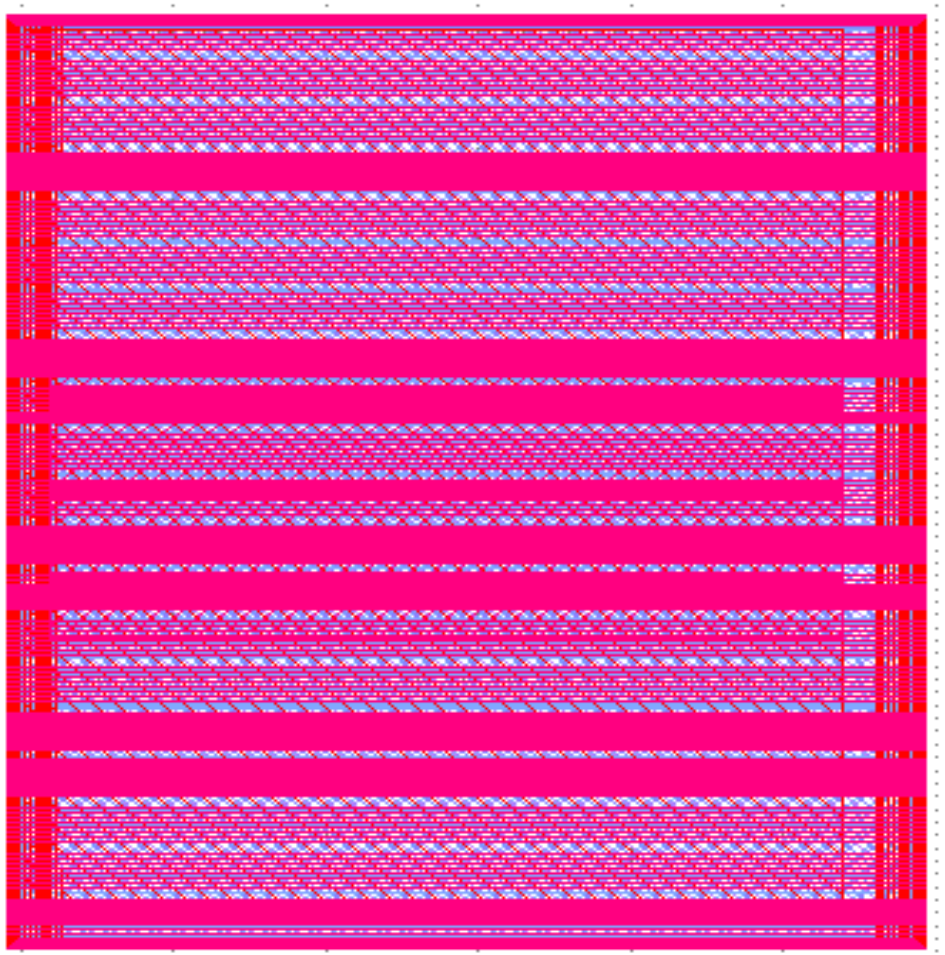
Design Stages and Tools of OpenLane

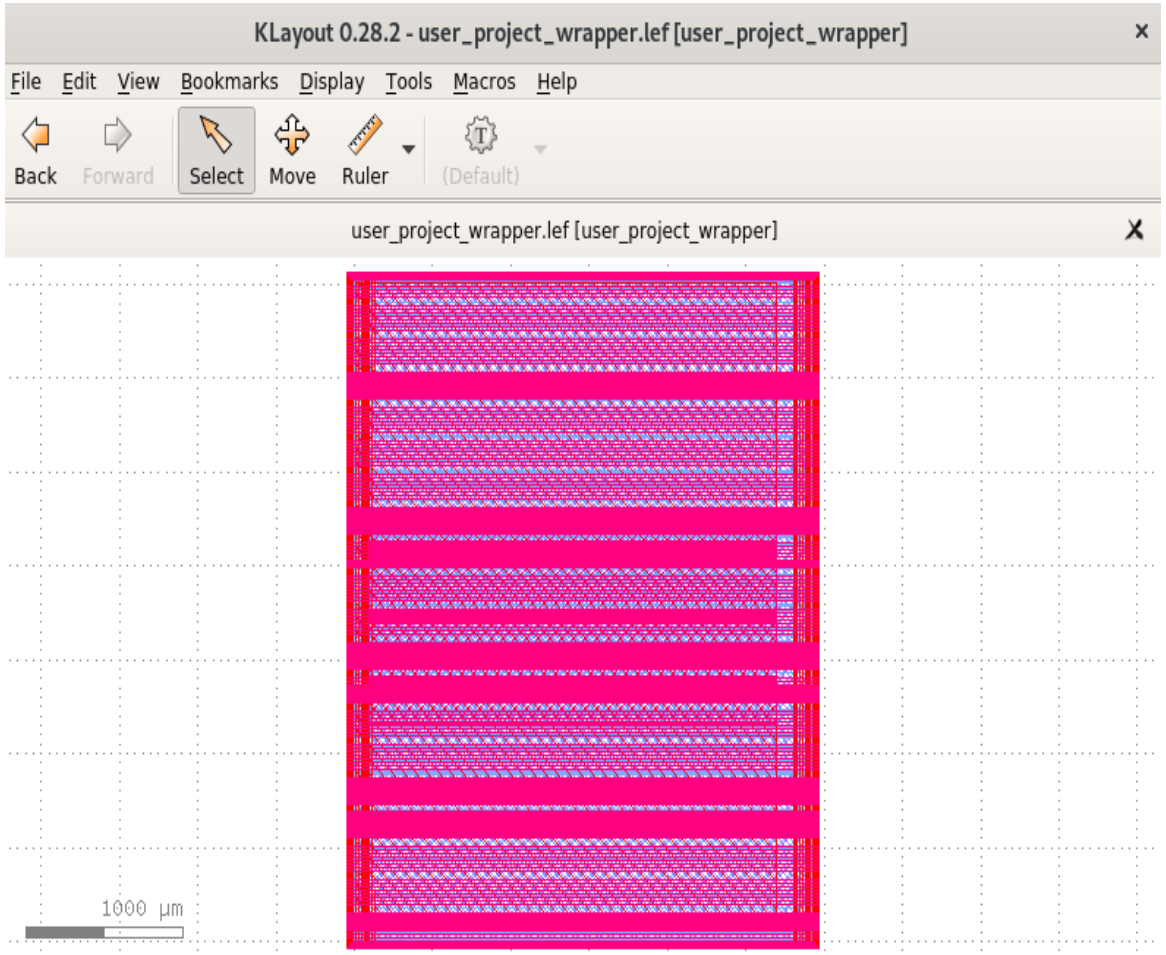
- Synthesis (yosys/abc, OpenSTA)
- Floor planning (init_fp, ioplacer, pdngen, tapcell)
- Placement (RePlace, Resizer, OpenDP)
- CTS (TritonCTS)
- Routing (FastRoute, TritonRoute, OpenRCX)
- Tape out (Magic, KLayout)
- Signoff (Magic, KLayout, Netgen, CVC)

Results Directory Structure

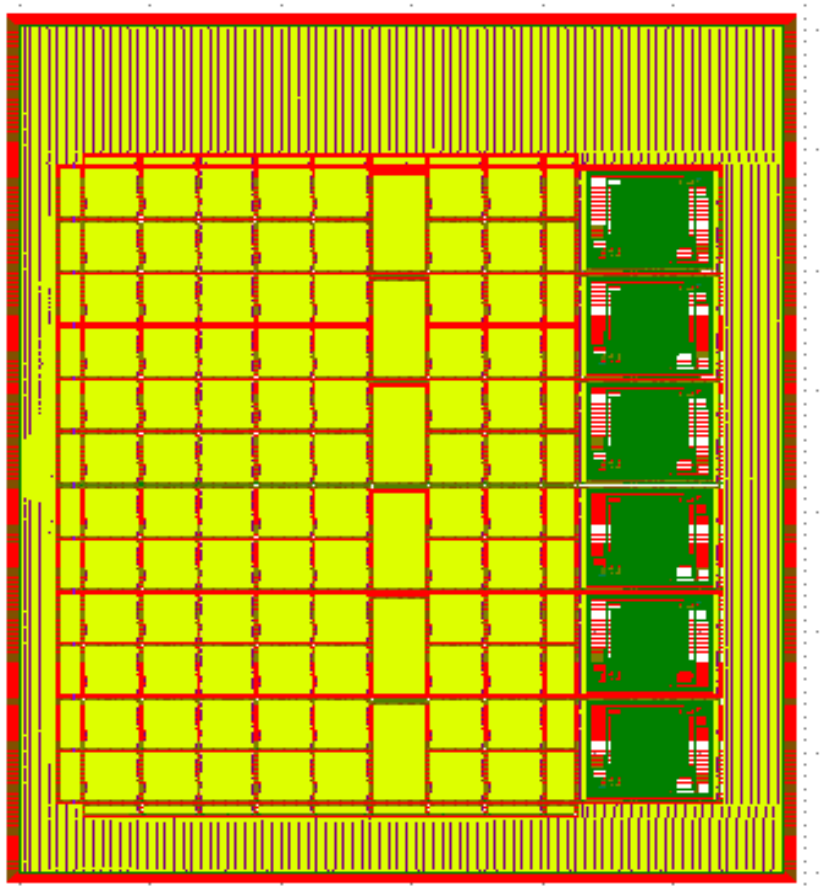
```
<design_name>
├── config.json/config.tcl
├── runs
│   ├── <tag>
│   │   ├── config.tcl
│   │   ├── {logs, reports, tmp}
│   │   │   ├── cts
│   │   │   ├── signoff
│   │   │   ├── floorplan
│   │   │   ├── placement
│   │   │   ├── routing
│   │   │   └── synthesis
│   │   ├── results
│   │   │   ├── final
│   │   │   ├── cts
│   │   │   ├── signoff
│   │   │   ├── floorplan
│   │   │   ├── placement
│   │   │   ├── routing
│   │   │   └── synthesis
```

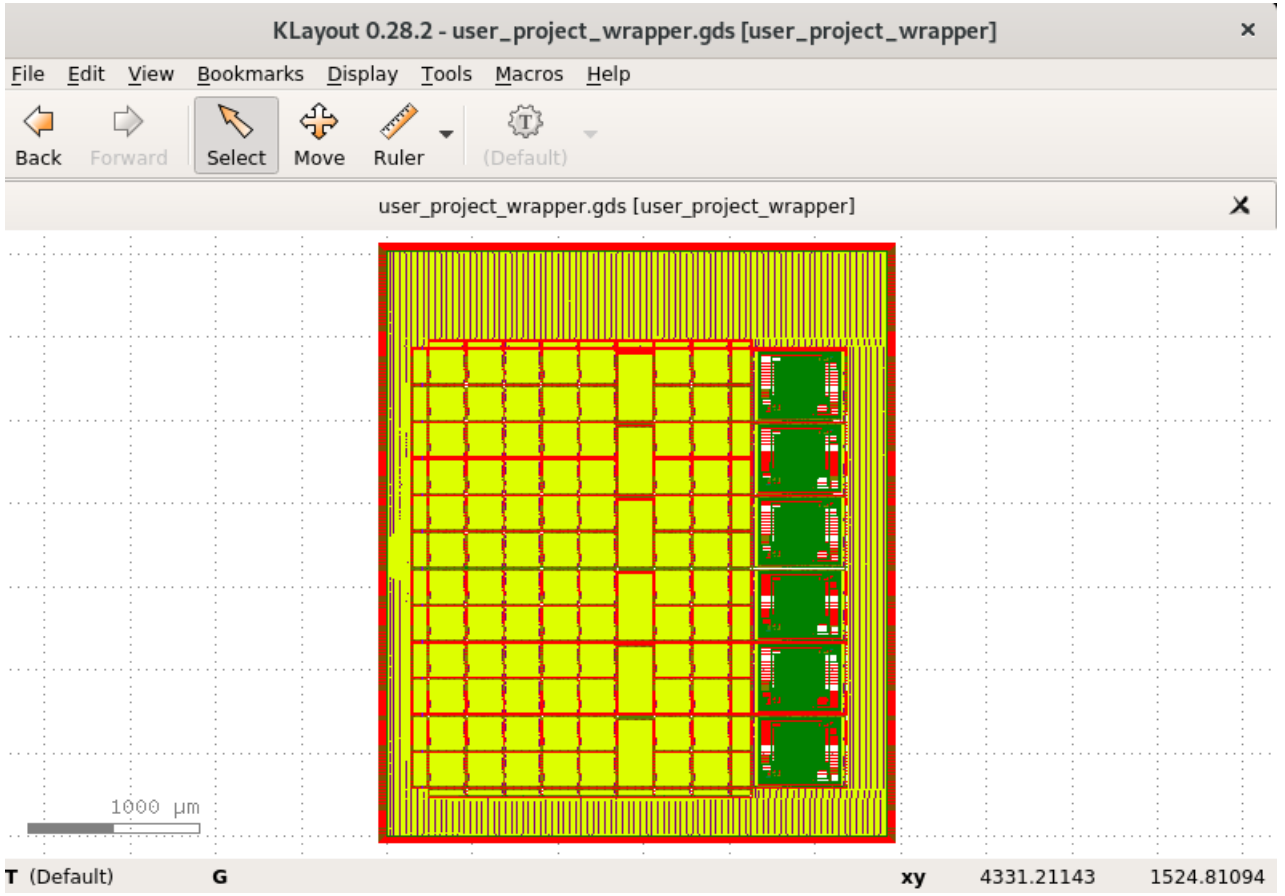
KLayout view of Open eFPGA (.gds)





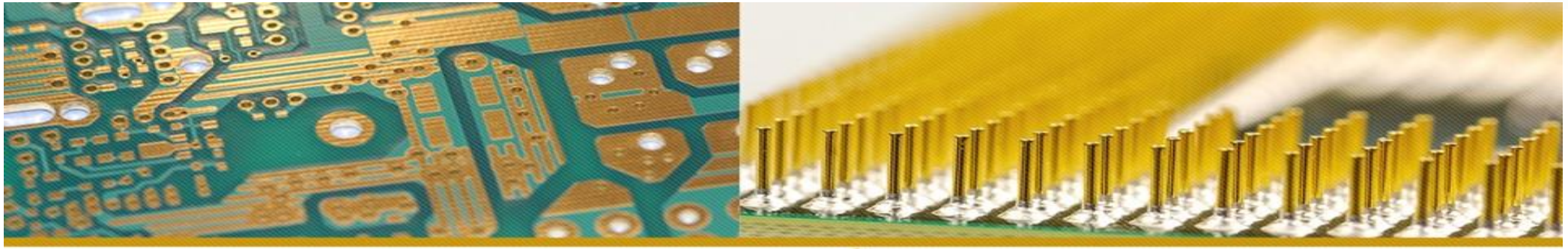
KLayout view of Open eFPGA (.lef)





References

- https://www.google.de/
- https://github.com/FPGA-research-Manchester/FABulous
- https:// github.com/The-OpenROAD-project/OpenLane
- https:// github.com/nguyendao-uom/open_eFPGA
- https:// github.com/efabless/caravel_user_project
- https://opelane.readthedocs.io/en/latest/getting_started/installation/index.html
- https://youtu.be/d0hPdkYg5QI
- https://www.youtube.com/live/2xF-beNHDTU?feature=share
- open Lane paper (Mohamed Shalan, Tim Edwards)



THANK YOU