SoCGen: A Push Button Idea to GDS2 SoC Design Flow

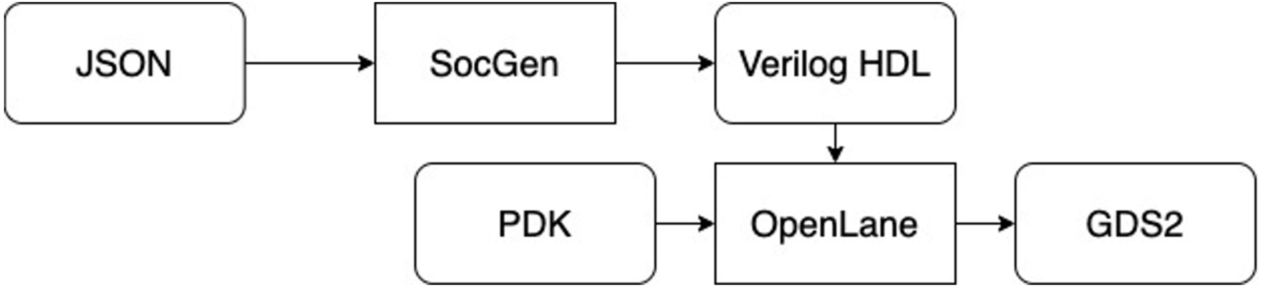
Habiba Gamal, Amr Gouhar, Mohamed Shalan

efabless.com

THE AMERICAN UNIVERSITY IN CAIRO

What is SoCGen?

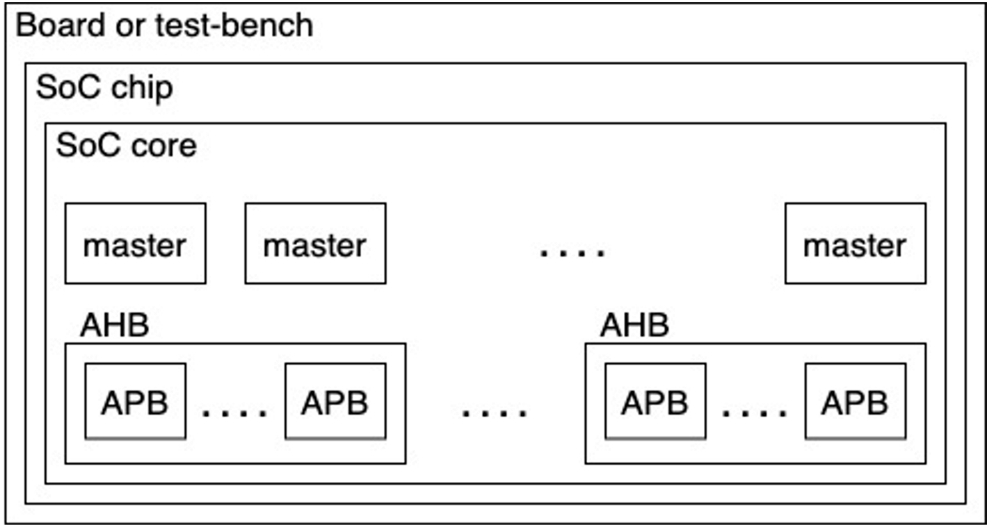
- System on Chip (SoC) design automation tool



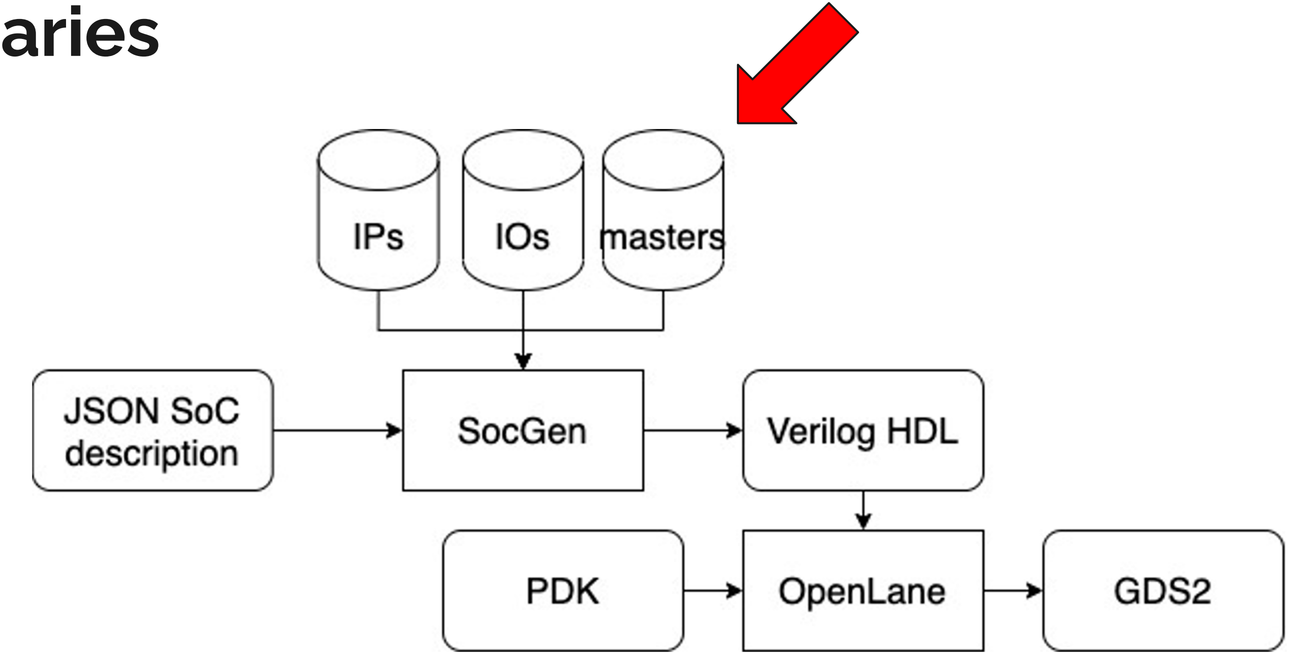
Motivation

- Facilitate SoC design
- Reduce time-to-fabrication
- Making use of the common features between different SoCs:
  - bus protocols,
  - Intellectual Properties (IPs)

Hierarchy

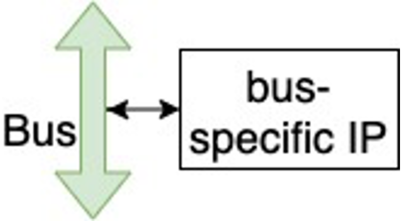

Board or test-bench
SoC chip
SoC core
master
master
....
master
AHB
AHB
APB
....
APB
....
APB
....
APB

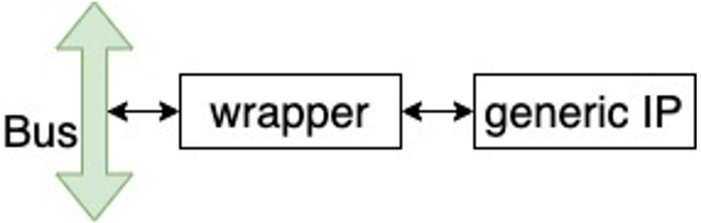
Libraries


IPs
IOs
masters
JSON SoC description
SocGen
Verilog HDL
PDK
OpenLane
GDS2

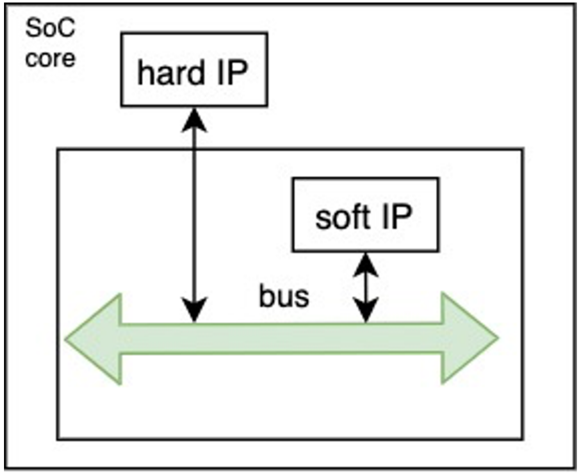
IPs library

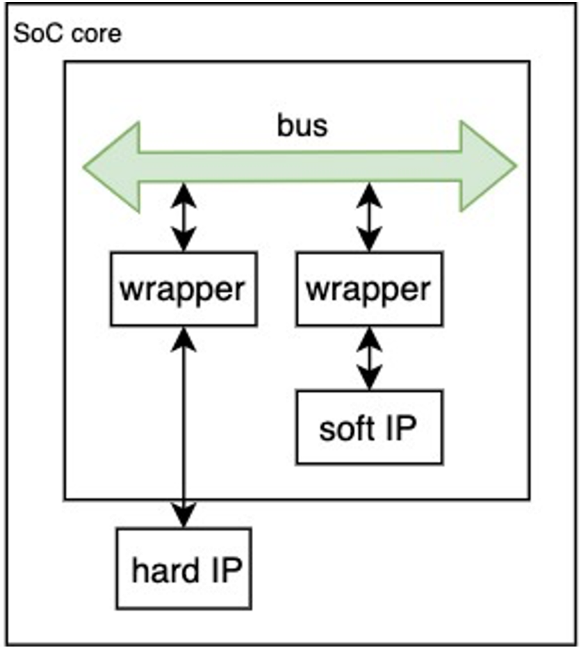
- JSON descriptions and verilog HDL for open-source verified IPs
- Examples:
    - UART, I2C master, SPI master,
    - timer, PWM, watchdog,
    - QSPI flash controller, SRAM controller
- Can be bus-specific or generic





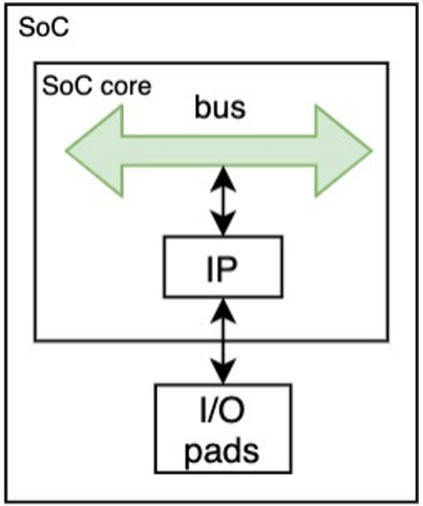
Hard IPs Vs. Soft IPs





IOs library

- JSON descriptions and Verilog HDL for IOs that are not technology specific
- Behavioral code used for simulation → not for hardening
- IOs included:
  - Digital input
  - Digital output
  - Digital input/output
  - Analog

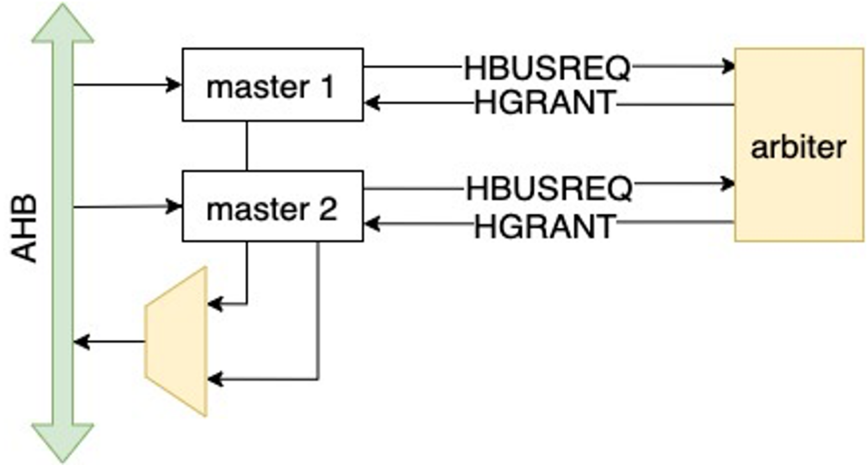


Masters library

- JSON descriptions for ARM Cortex M0 and ARM Cortex M3
- JSON description and verilog HDL for N5, open-source core
- Masters currently have to be bus-specific

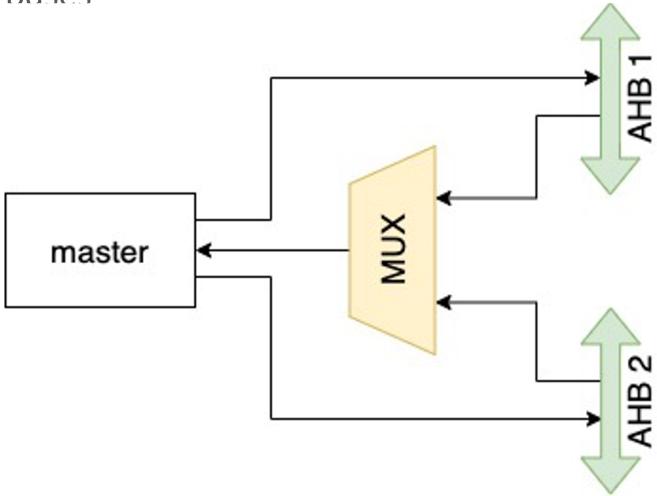
Supported Features

1) Multiple masters on the same bus



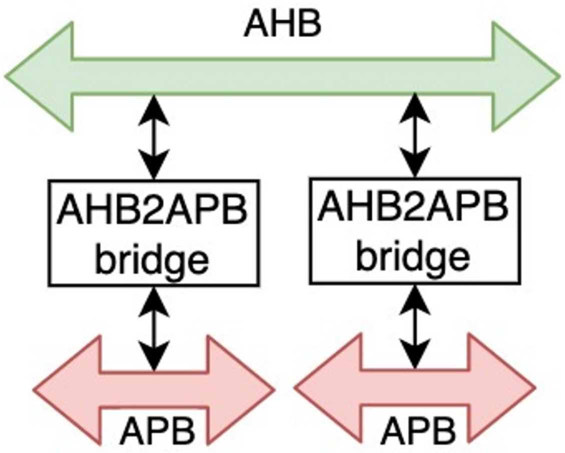
Supported Features

2) Same master on multiple buses



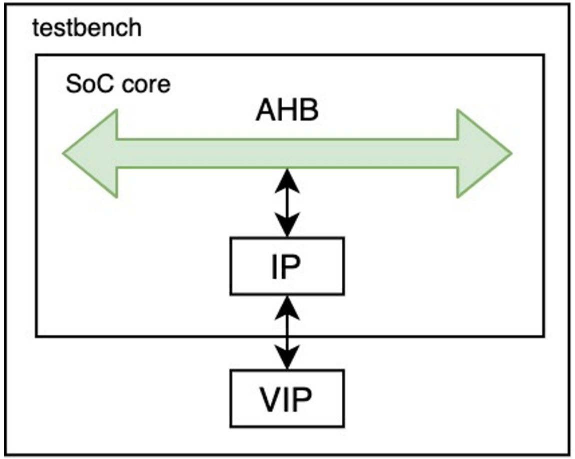
Supported Features

3) Multiple APBs on the same AHB



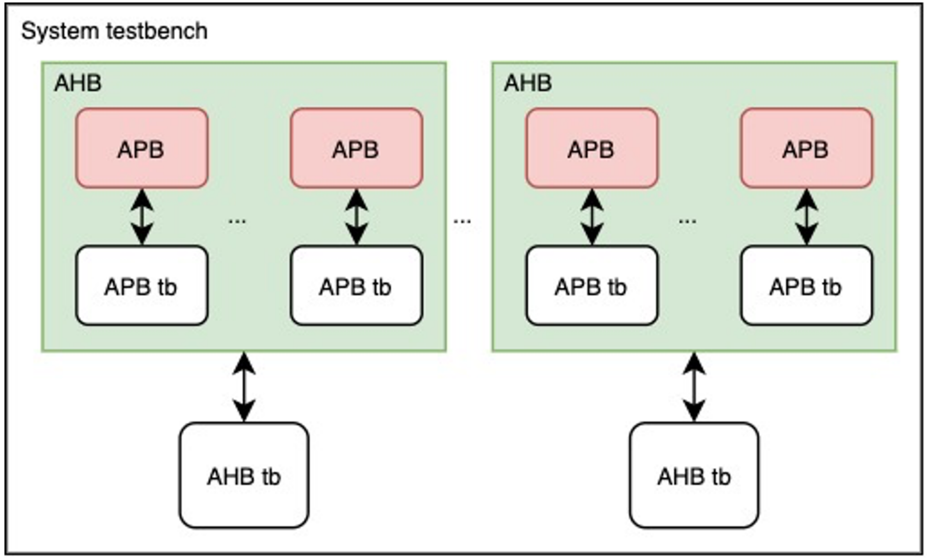
Supported Features

4) Verification IPs specified in IPs library for testing



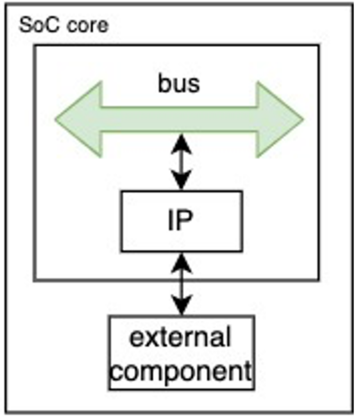
Testing

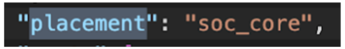
- Hierarchical testing
- Testing in the absence of real masters
- Configurable testbench:
  - Number of ticks
  - Location of hex file to load in flash
- Debug register for self-checking testbench



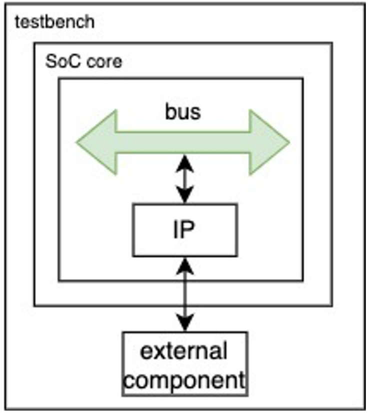
Configurable Options

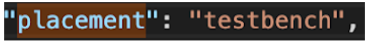
1) Placement level of components connected to external ports of IPs





```
"placement": "soc_core",
```

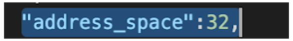




```
"placement": "testbench",
```

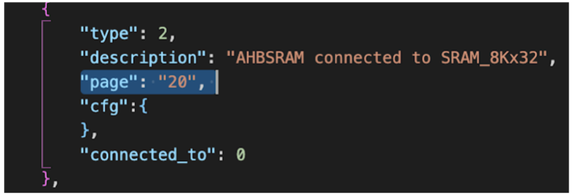
Configurable Options

2) Width of address line



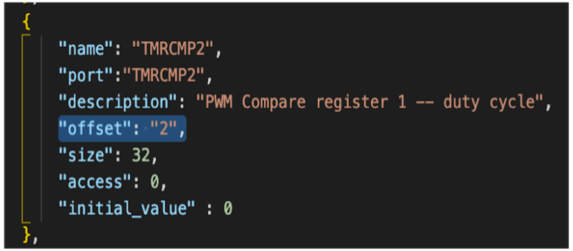
```
"address_space":32,
```

3) Base addresses of buses and components



```
{
    "type": 2,
    "description": "AHBSRAM connected to SRAM_8Kx32",
    "page": "20",
    "cfg":{
    },
    "connected_to": 0
},
```

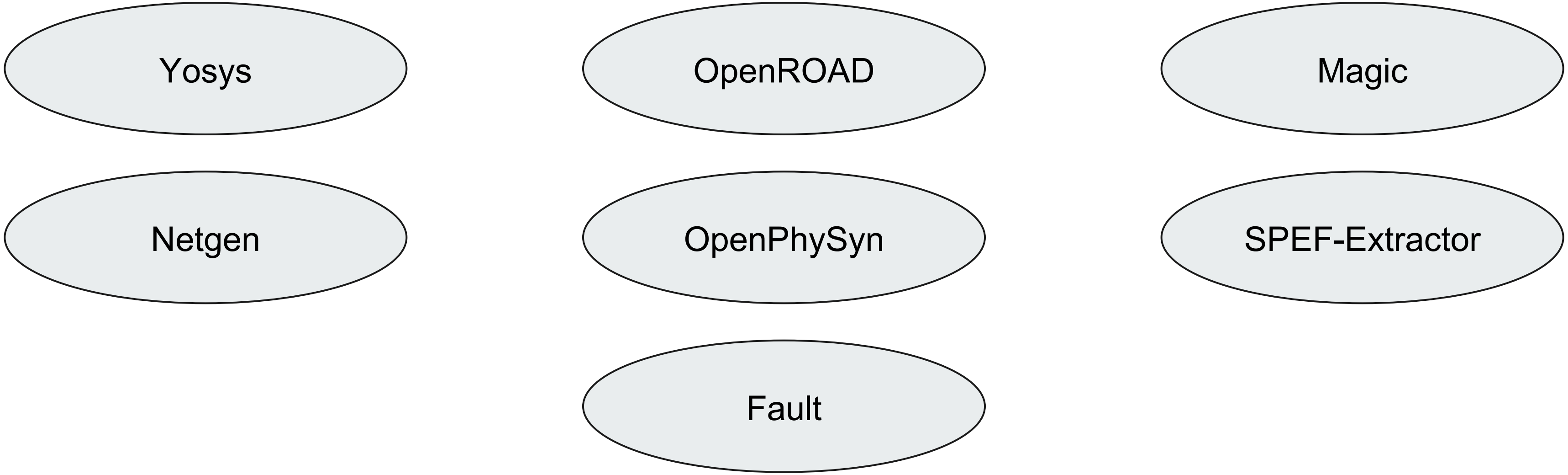
4) Offset addresses of registers within IPs



```
{
    "name": "TMRCMP2",
    "port":"TMRCMP2",
    "description": "PWM Compare register 1 -- duty cycle",
    "offset": "2",
    "size": 32,
    "access": 0,
    "initial_value" : 0
},
```

OpenLane

OpenLANE is an RTL to GDS-II automated open-source flow, based on:



Yosys

OpenROAD

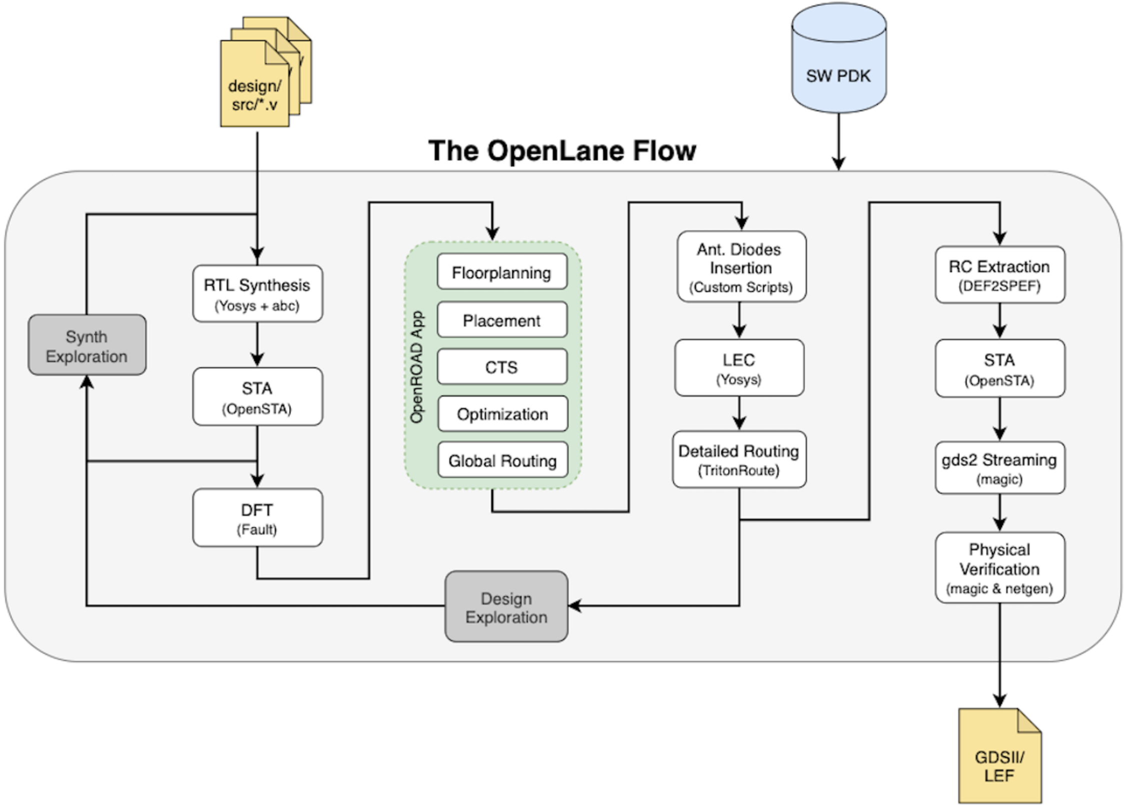
Magic

Netgen

OpenPhySyn

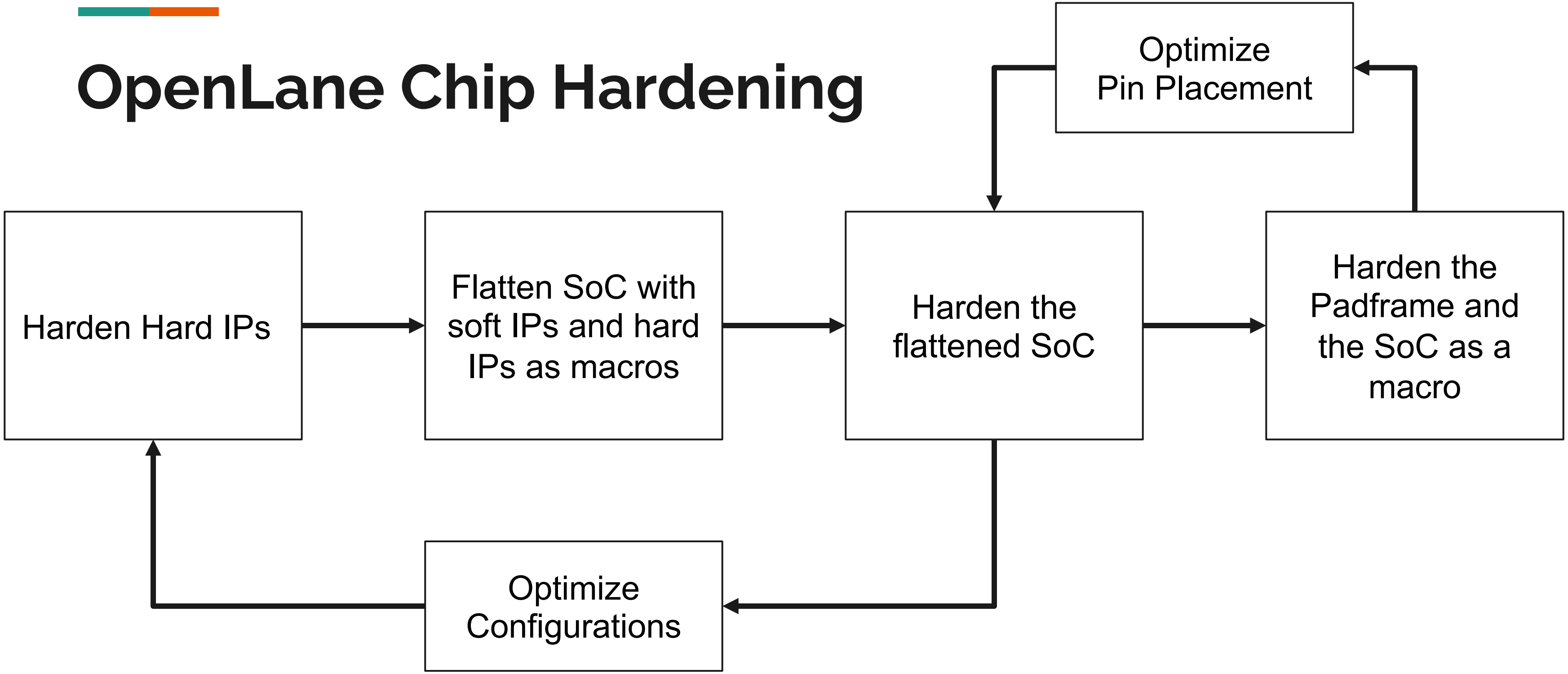
SPEF-Extractor

Fault



*Source: https://github.com/efabless/openlane/blob/master/doc/openlane.flow.1.png

OpenLane Chip Hardening



Optimize Pin Placement

Harden Hard IPs

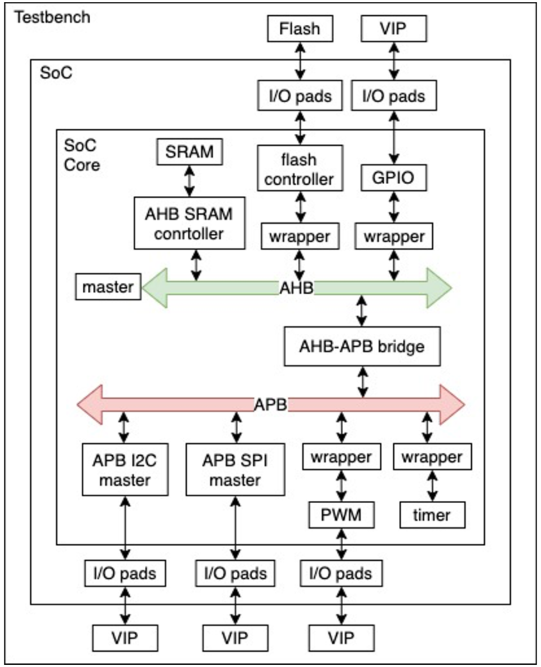
Flatten SoC with soft IPs and hard IPs as macros

Harden the flattened SoC

Harden the Padframe and the SoC as a macro

Optimize Configurations

Generated System



Work In Progress

- Support more bus types

- Add early stage estimators for area, power and clock frequency

Thank You!

scan
me