

# 基于开源IP/SoC和开源EDA工具的芯片全链条设计—— 定制属于自己的RISC-V游戏掌机

**缪宇飏**

中国科学院计算技术研究所

2024年8月23日

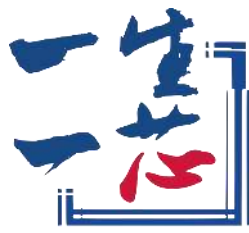


# 2030 “三步走” 规划

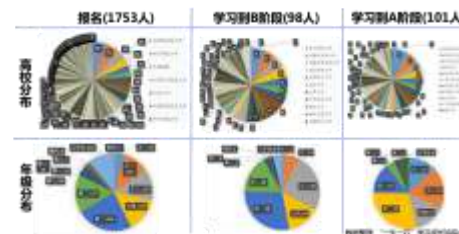
- **第一步：开源SoC——用3-5年为社区提供经过流片验证的高质量RISC-V开源核、开源SoC设计**
  - RISC-V处理器核IP、外围IP等
- **第二步：用开源工具链构建开源SoC——用5-7年逐步构建一套基于开源EDA工具链、开源IP、开源工艺库的开源SoC芯片设计流程**
  - 将商业版工具、IP逐渐替换为开源版
  - 实现本科生用全开源工具开发开源芯片，带着自己芯片毕业
- **第三步：用开源工具链自动化构建开源硬件——用10-15年开发更智能、更自动化的开源工具，提高设计验证效率**
  - 形成开源芯片设计生态，降低芯片开发门槛

# 我们当前在开展的项目

## 芯片 人才培养



- 培养学生“**全链条芯片设计**”和“**软硬件协同设计**”，独立解决问题
- 已开展六期，累计报名人数超6000人，覆盖超350所高校



## 开源 EDA工具链

iEDA

- 面向国产28nm工艺
- 构建全链条芯片设计的开源EDA平台



## 开源IP



- 开发处理器核IP及一系列接口类IP, 如SDRAM, UART, VGA, 嵌入式GPU, I2C, I2S等

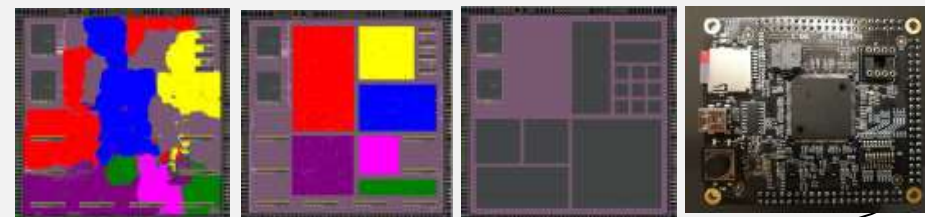


开源IP基础组件库

## SoC&后端流 片计划

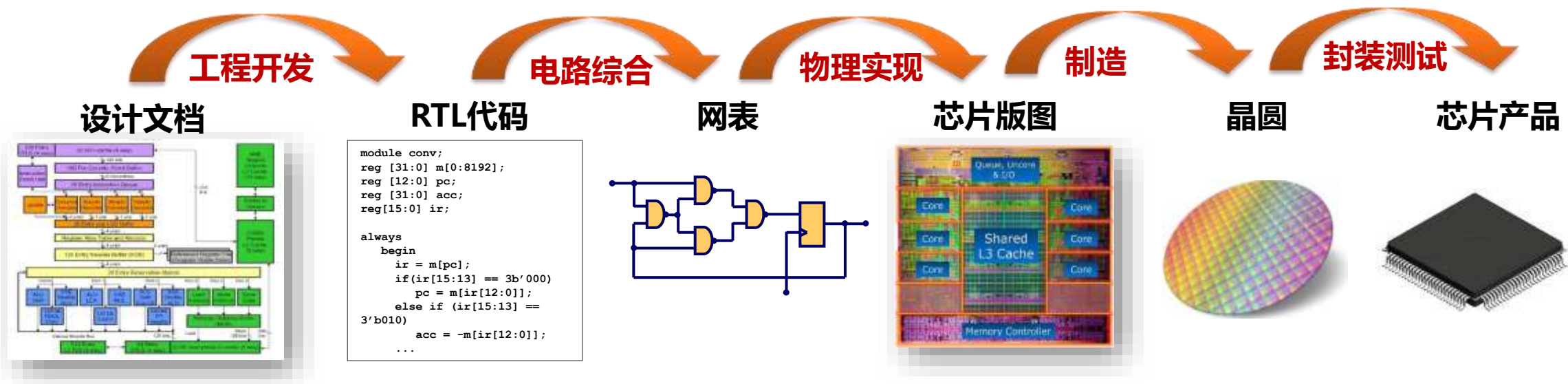
Design

- **2022年4个批次，合计10颗芯片**
- 支持多种粒度的芯片集成方法
- 支持从 spec定义到流片点亮
- 支持常用的开源IP



# 芯片全链条设计

- **工程开发**: 基于设计文档编写RTL代码
- **电路设计**: 将RTL代码经过逻辑设计和工艺映射成网表
- **版图设计**: 用EDA工具将设计好功能等网表物理化成GDS版图
- **芯片制造**: 将设计好的版图通过光刻制造封装形成芯片

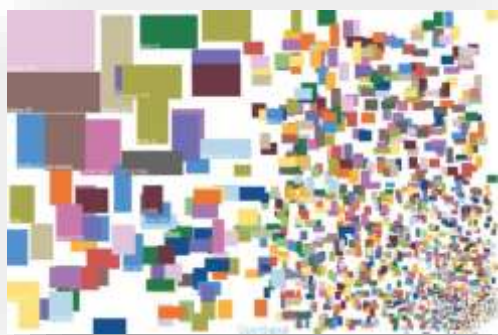


# 一、关于构建开源芯片全链条设计的思考

# 市场的多样化需求，牵引了开源芯片的新机遇

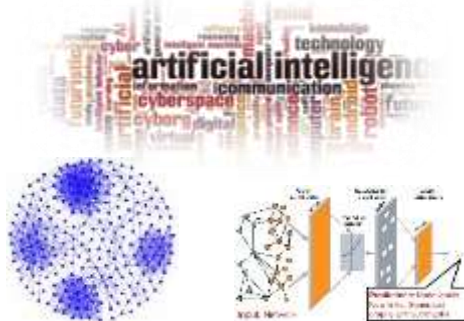
- **需求牵引**：AIoT、工业物联网、5G通信、无人驾驶等形成了多样化芯片设计需求
- **技术驱动**：人工智能和图计算等（芯片性能）、GPU和专用加速芯片（设计效率）

需求和指标更加严苛和多样化



多样化芯片设计需求

芯片性能更高、流程更加自动化



人工智能和图计算等

设计效率更高、研发周期更短

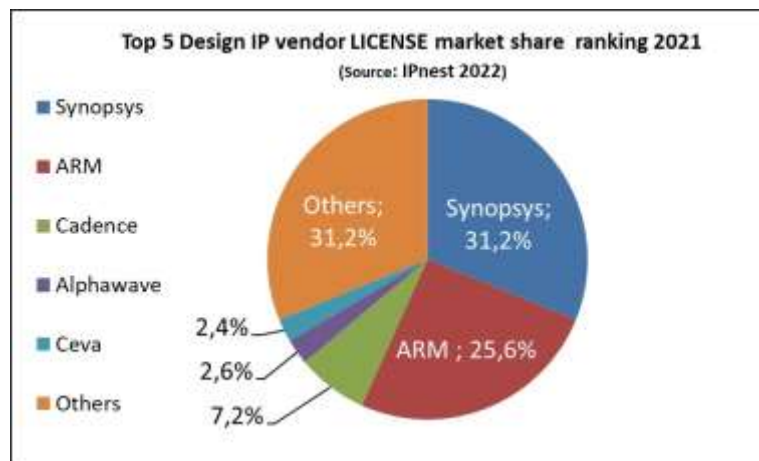


新型硬件和算力网络

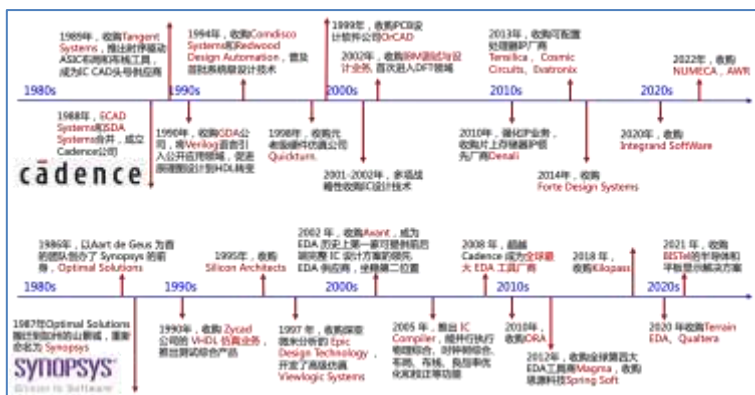


# 芯片全链条设计面临的困境

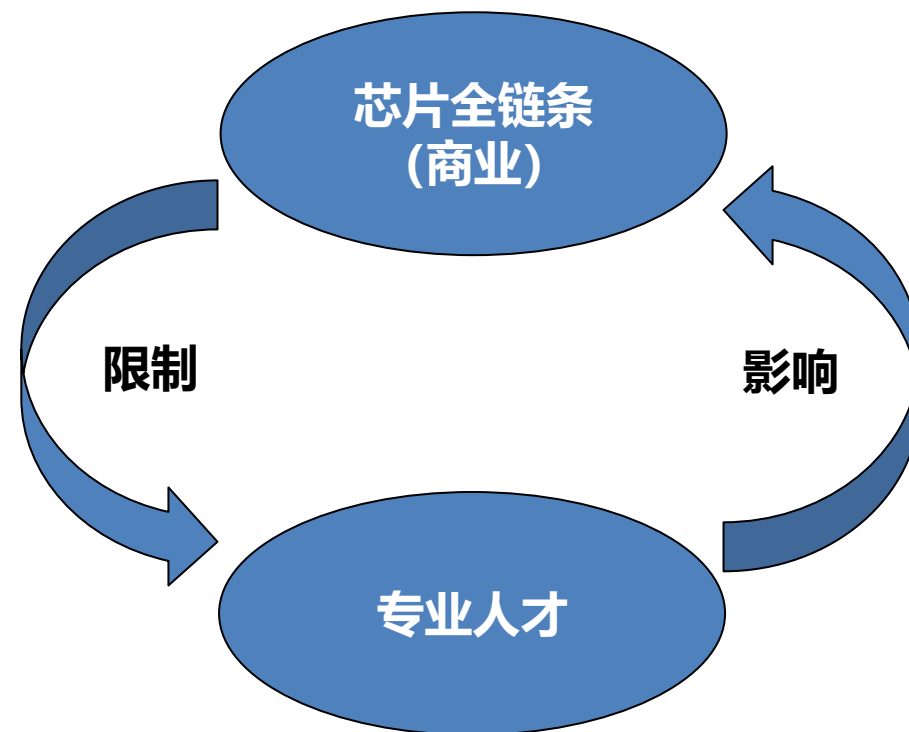
- IP/SoC和EDA技术被商业公司掌握，无法促进该领域的快速发展
- 技术跨度较大，缺少优秀的芯片前后端技术人才



中高端IP授权费价格高昂，SoC集成需要工程师具备一定的技术基础



国外EDA企业通过并购不断加强商业和技术垄断



技术壁垒会形成一个神奇的“闭环”

# 如何破解芯片全链条技术的困境？

- **问题定义：** 什么问题是有价值的和有意思的
  - 必要问题，痛点问题，以及国外有做但国内一直没做的问题
- **流程构建：** 如何设计一套可用于芯片全链条设计的流程
  - 参考国外经验，取其精华去其糟粕，做本土化设计和新技术试点
- **成果迭代：** 如何打开用户，让更多的人来用
  - 更多的测试用例，不同场景、规模和复杂度的芯片来打磨

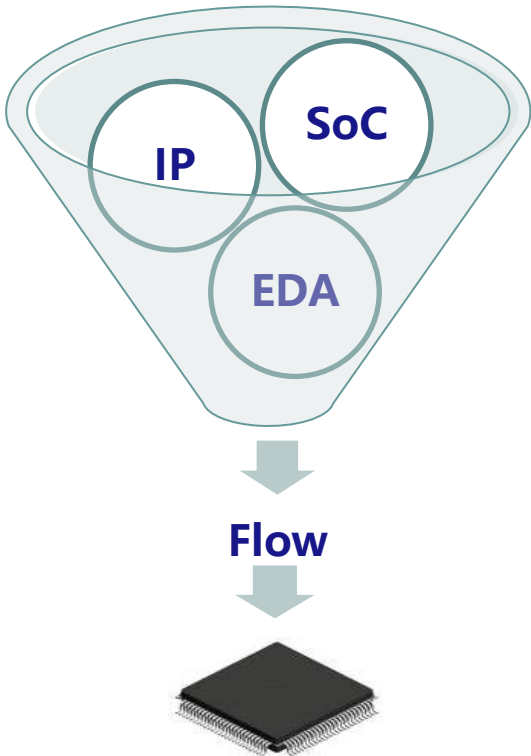


# 开源是支撑开放生态的重要保障

- 成功经验表明，**开源开放是构建繁荣的技术生态和产业生态的必要基础**
- 开源IP/SoC简化芯片前端设计，提高设计效率，大大降低Bug出现
- 开源EDA可以使部分算法设计工作前移到芯片端
- 开源IP/SoC、EDA为相关领域的科学研究和人才培养，提供了很好的思路

领域	代表系统	影响力体现
操作系统	Linux	重要的基础软件，支撑了整个开源生态体系
编译器	LLVM、GCC	
移动操作系统	Android	虽然现在已闭源，但当年以开源为起点， <b>统一了除iOS外的整个移动互联网生态</b>
人工智能	Caffe、TensorFlow Pytorch、PaddlePaddle	整个 AI 技术生态体系的基石
IP/SoC	----	一定基础（PULP Platform, lowRISC, LiteX）
EDA 工具	----	基础薄弱（Verilator, OpenRoad, OpenDACS）

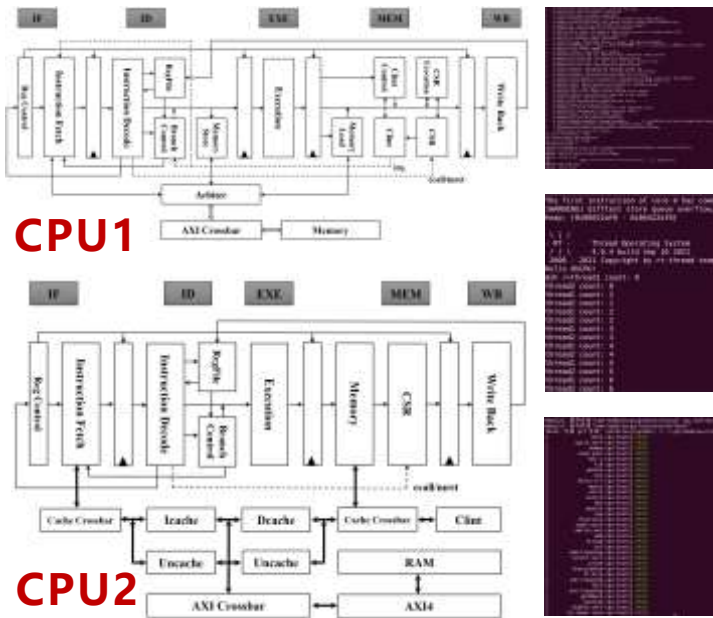
**共同特点：**皆为重要的基础性平台，**支撑各领域技术创新外，服务了巨大的产业经济**



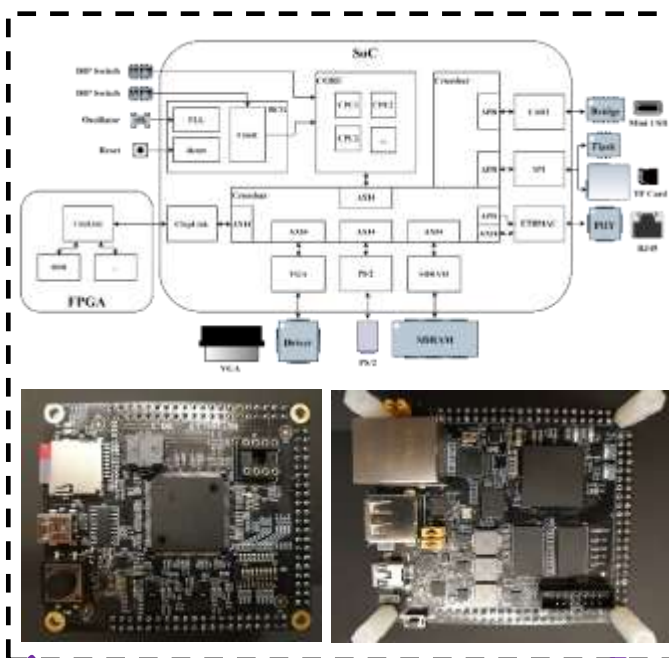
# 开源IP/SoC的价值

- 基于开源IP实现SoC的自动化敏捷集成
- 大大缩短处理器设计周期，提高芯片设计质量
- 可按需定制不同功能的芯片，具备较高的灵活度和可扩展性
- 为开源EDA工具提供数据输入，用于验证工具的功能和性能

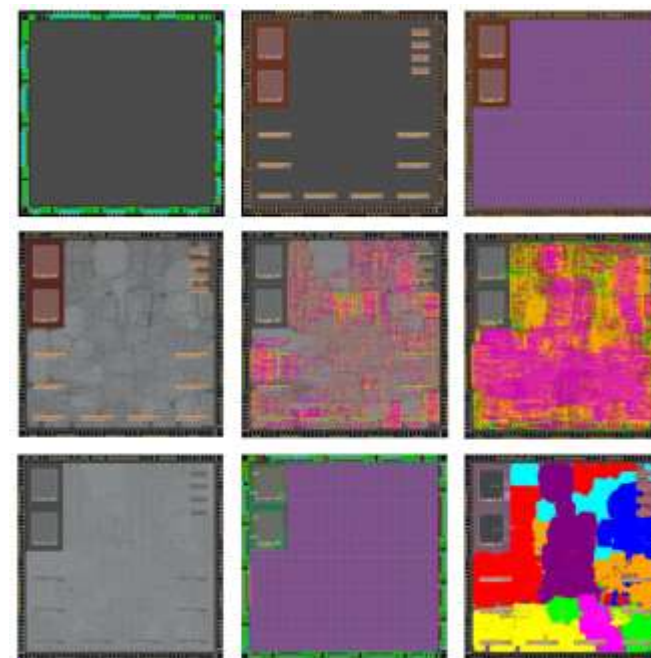
## 处理器核设计



## 集成开源IP形成SoC



## SoC物理实现



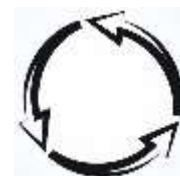
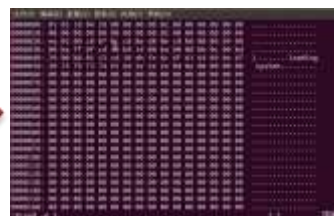
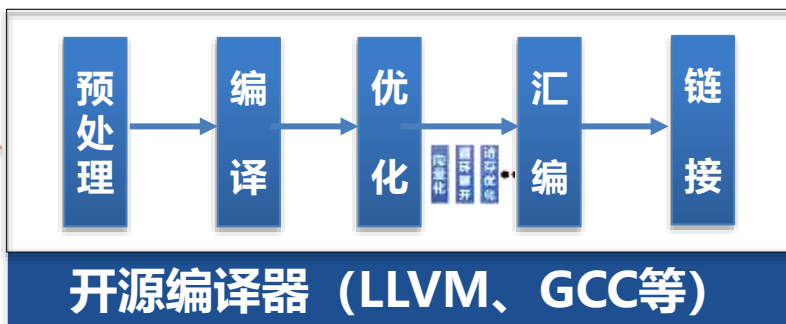
使用脚本实现IP的自动化集成和验证，缩短全链条设计周期

# 开源EDA的价值

- 助力EDA算法设计，如跨层优化、in-house EDA 设计、AI+EDA算法等等
- 为基础性和变革性的EDA方法（EDA云原生和异构加速等），提供基础平台支撑
- 开源在技术生态、人才流动等方面，可以跟商业EDA和芯片公司形成较好呼应
- 可以结合开源IP等，支撑开源芯片设计解决方案

## 软件设计流程

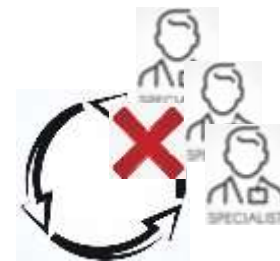
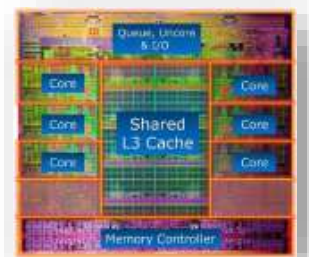
gcc -O3  
Llvm -O3



完备的自动化设计

## 芯片设计流程

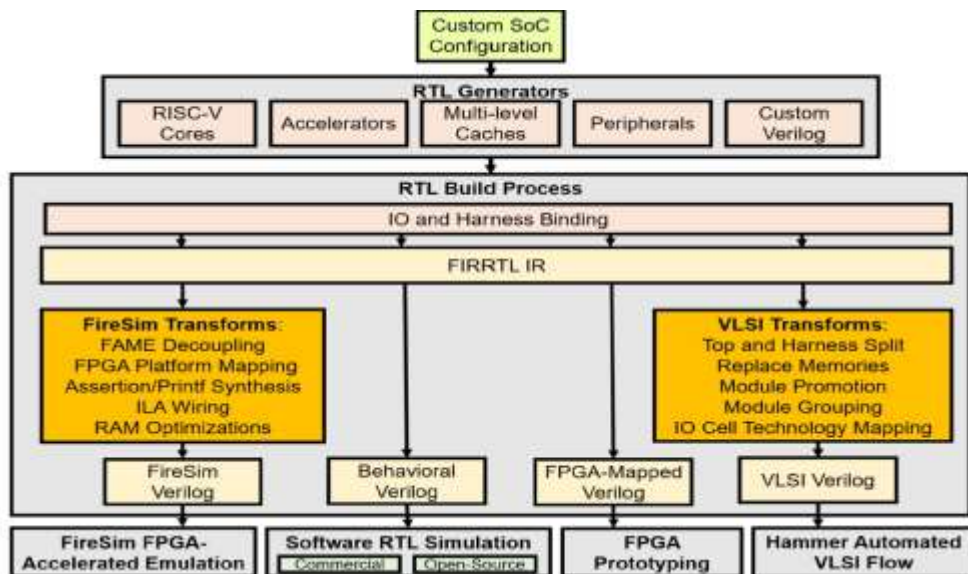
ieda -O3



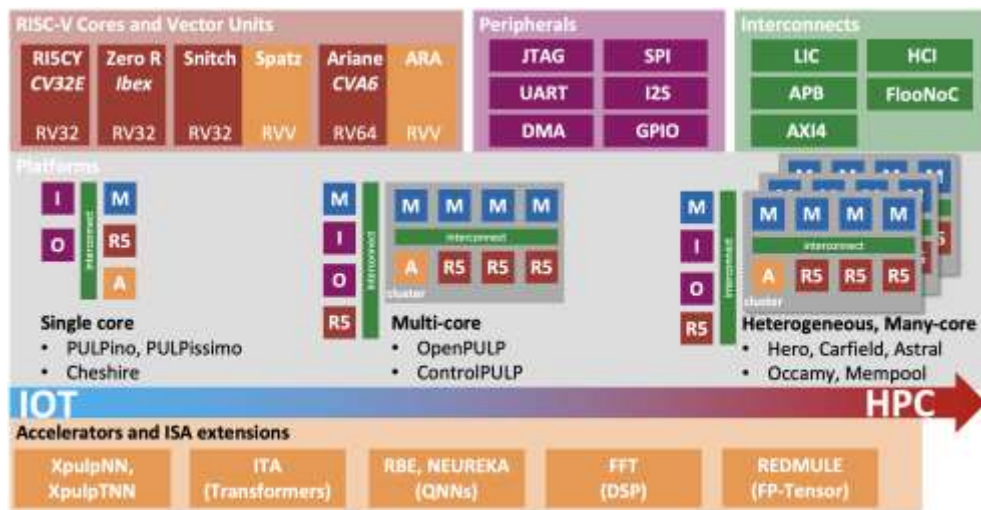
部分自动化  
尚需耗费大量人力，  
极度依赖专家经验



# 国际开源IP/SoC发展动态

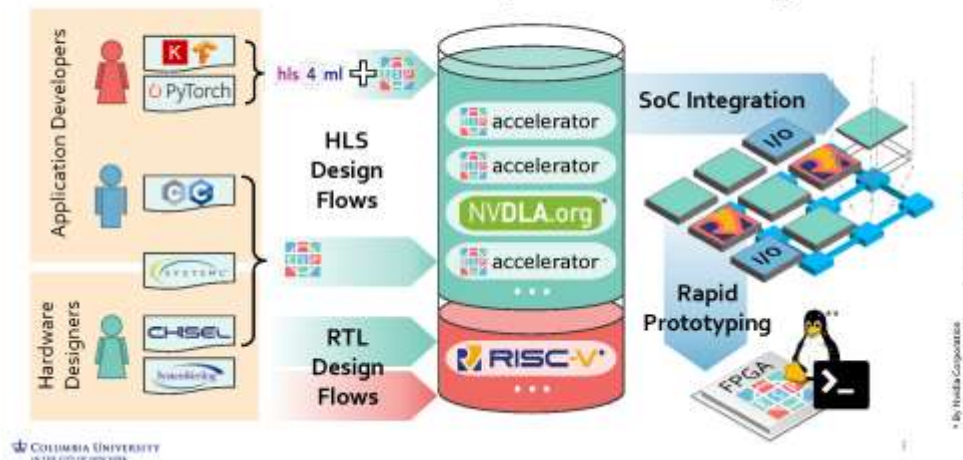


ChipYard

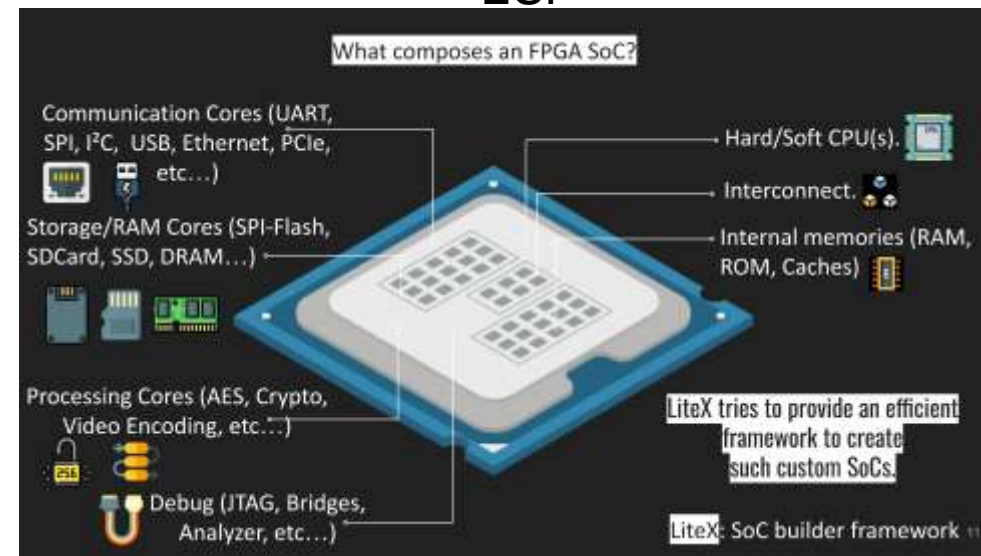


PULP Platform

## ESP Vision: Domain Experts Can Design SoCs



ESP



LiteX

# 国际开源EDA发展动态

项目	最近发版	Flow	点工具集成	PDK	资助方	国家	启动时间	流片	License
ORFS	2024-01-06	RTL-GDS	开源工具+自研工具	开源工艺、GF12、TSMC65	美国防高级研究计划局, Precision Innovations, OpenROAD Initiative	美国	2018	600余次 (包含 OpenLane)	BSD-3
OpenLane	2024-05-16	RTL-GDS	开源工具	开源工艺	Efabless	美国	2020	450余次	Apache-2.0
SiliconCompiler	2024-05-23	仿真验证 ASCI+FPGA RTL-GDS	开源工具+商业工具	开源工艺、GF22/12、Intel16	Zero ASIC	美国	2020	ZeroASIC商业项目多次流片	Apache-2.0
Hammer	2023-11-14	仿真验证 ASCI+FPGA RTL-GDS	开源工具+商业工具	开源工艺、商业28/22/16/14/12	UC伯克利Chipyard团队	美国	2017	7次	BSD-3
mflowgen	2024-02-24	RTL-GDS	开源工具+商业工具	开源工艺、TSMC40/28/16、IBM180	南加州大学/斯坦福团队	美国	2020	6次	BSD-3
Qflow	2024-05-05	RTL-GDS	开源工具	开源工艺	Open Circuit Design, Efabless	美国	2011	1次	GPL
DATC-RDF	2023-11-01	RTL-GDS	开源学术工具	开源工艺	美国防高级研究计划局, IEEE-CEDA, NSF	美国	2018	0	BSD-3
Alliance/Coriolis	2023-08-12	仿真验证 RTL-GDS	开源工具+自研工具	开源工艺	LIP6, 索邦大学, 法国科学院	法国	1990	1次	GPL-2.0
LibrEDA	活跃研发中	Netlist-GDS	开源工具+自研工具	开源工艺	NLnet Foundation	欧盟	2020	0	AGPL-3.0
VSDFLOW	2021停止更新	RTL-GDS	开源工具	开源工艺	VLSI System Design Corporation	印度	2017	0	Apache-2.0

# 国际开源EDA发展动态

- 2024年3月8日，500位欧洲的教授和科研学者联合签署声明，呼吁政府重视开源EDA对芯片教育和产业创新的巨大价值，建议欧洲相关部门加大对开源EDA的资助力度

## Importance of Open-Source EDA Tools for Academia

### Open Letter on European Strategic and Funding Directions

To Whom It May Concern

*March 8, 2024*

The recent semiconductor shortage and shifts in global political relations have changed the European roadmap on semiconductors and chip design significantly. A mix of incentives for new fabrication facilities for advanced technologies and the ambitious goals to (re)-build leading-edge chip design capabilities in Europe are key cornerstones of the European Chips Act. Under this impulse, various funding actions have been successfully launched, for instance in the area of the creation of IP based on the RISC-V instruction set.

Universities have to be an integral part of Europe's ambitions and are heavily involved in research activities on various levels. This is crucial for two reasons: First, they are incubators of innovative ideas. Second, and equally important, they are of key importance to educating future generations of chip designers and related jobs. The high demand for an increased workforce can only be satisfied with tight coordination and the best support of universities. In these efforts, we believe that open source is a key success factor. The availability of open-source RISC-V IPs developed in Europe has, for example, been of seminal importance for a significant ramp-up of research activities and the rapid build-up of a vibrant research community and innovation ecosystem around RISC-V hardware and software. This has led to new groups getting involved with computer architecture and an increase in innovative research that is currently funded to get commercialized at higher technology readiness levels.

A key next step is to turn innovation in computer architecture for the European market into chip designs that utilize the fabrication facilities and skilled workers that create those designs. In this trajectory, it is our strong belief that **open-source chip design tools (Electronic Design Automation -EDA- tools) are essential for the European renaissance in chip design education and innovation:**



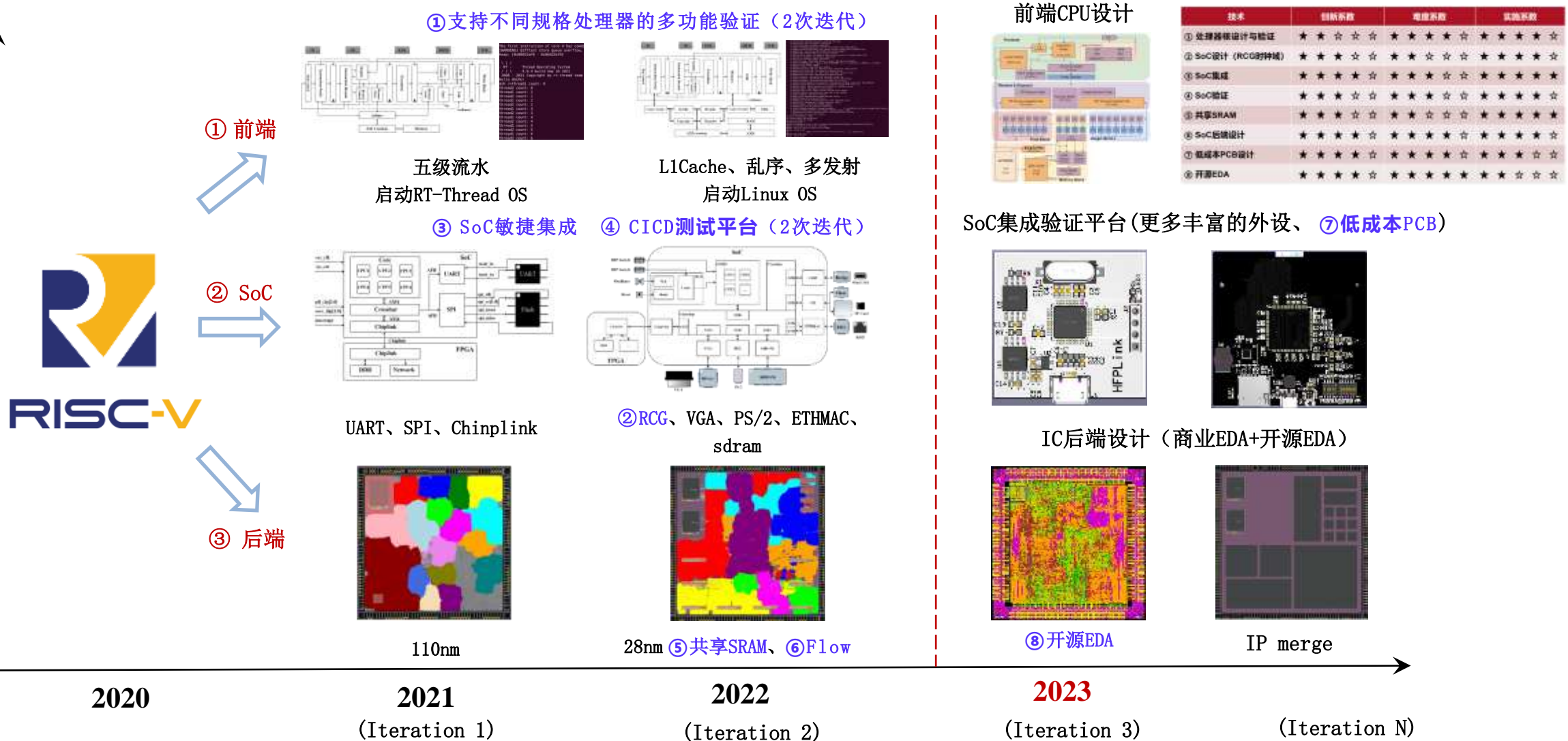
## 二、芯片全链条设计的 最新进展

# 技术路线

RISC-V

## SoC芯片敏捷设计和验证平台技术路线①②③及技术创新①②③④⑤⑥⑦⑧

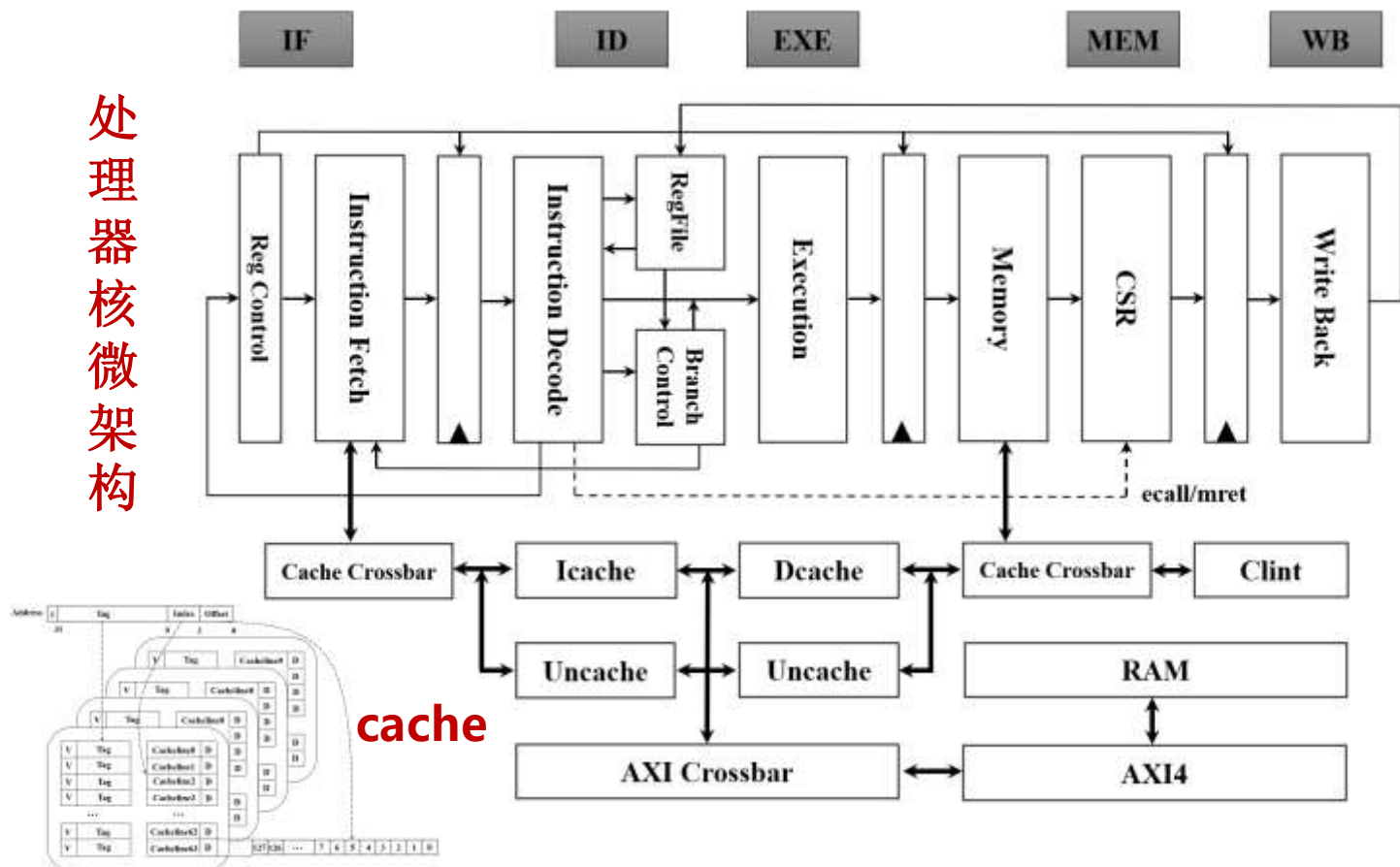
SoC芯片设计及平台技术路线



# 处理器核设计与功能验证 RISC-V

- 支持不同规格（功能、性能、面积）的处理器核接入SoC
- 统一规范顶层接口，通过AXI总线进行通信，包含Cache、可启动Linux OS(Chisel/Verilog)

处理器核微架构



```
...  
The first instruction of core 0 has committed. DiffTest  
[WARNING] diffTest store queue overflow, diffTest at  
heap: [0x00022af0 - 0x00022af0]  
...  
RT Thread Operating System  
4.0.4 build Sep 18 2021  
2006 - 2021 Copyright by rt-thread team  
Hello OSCP!  
ash />thread1 count: 0  
thread2 count: 0  
thread1 count: 1  
thread2 count: 1  
thread1 count: 2  
thread2 count: 2  
thread1 count: 3  
thread2 count: 3  
thread1 count: 4  
thread2 count: 4  
thread1 count: 5  
thread2 count: 5  
thread1 count: 6  
thread2 count: 6
```

启动 Linux OS

```
...  
The first instruction of core 0 has committed. DiffTest  
[WARNING] diffTest store queue overflow, diffTest at  
heap: [0x00022af0 - 0x00022af0]  
...  
RT Thread Operating System  
4.0.4 build Sep 18 2021  
2006 - 2021 Copyright by rt-thread team  
Hello OSCP!  
ash />thread1 count: 0  
thread2 count: 0  
thread1 count: 1  
thread2 count: 1  
thread1 count: 2  
thread2 count: 2  
thread1 count: 3  
thread2 count: 3  
thread1 count: 4  
thread2 count: 4  
thread1 count: 5  
thread2 count: 5  
thread1 count: 6  
thread2 count: 6
```

启动RT-Thread OS

```
...  
fatal: 不是 git 仓库: libraries/yysysoc/.../git/modul  
...  
min3-cpu-tests) PASS!  
quick-sort-cpu-tests) PASS!  
dummy-cpu-tests) PASS!  
leap-year-cpu-tests) PASS!  
mov-c-cpu-tests) PASS!  
fib-cpu-tests) PASS!  
wanshu-cpu-tests) PASS!  
div-cpu-tests) PASS!  
hello-rt-cpu-tests) PASS!  
mops-cpu-tests) PASS!  
shift-cpu-tests) PASS!  
switch-cpu-tests) PASS!  
max-cpu-tests) PASS!  
pascal-cpu-tests) PASS!  
sum-cpu-tests) PASS!  
recursion-cpu-tests) PASS!  
load-store-cpu-tests) PASS!  
matrix-mul-cpu-tests) PASS!  
add-cpu-tests) PASS!  
string-cpu-tests) PASS!  
bit-cpu-tests) PASS!  
add-longlong-cpu-tests) PASS!
```

运行字符版马里奥

```
...  
fatal: 不是 git 仓库: libraries/yysysoc/.../git/modul  
...  
min3-cpu-tests) PASS!  
quick-sort-cpu-tests) PASS!  
dummy-cpu-tests) PASS!  
leap-year-cpu-tests) PASS!  
mov-c-cpu-tests) PASS!  
fib-cpu-tests) PASS!  
wanshu-cpu-tests) PASS!  
div-cpu-tests) PASS!  
hello-rt-cpu-tests) PASS!  
mops-cpu-tests) PASS!  
shift-cpu-tests) PASS!  
switch-cpu-tests) PASS!  
max-cpu-tests) PASS!  
pascal-cpu-tests) PASS!  
sum-cpu-tests) PASS!  
recursion-cpu-tests) PASS!  
load-store-cpu-tests) PASS!  
matrix-mul-cpu-tests) PASS!  
add-cpu-tests) PASS!  
string-cpu-tests) PASS!  
bit-cpu-tests) PASS!  
add-longlong-cpu-tests) PASS!
```

RISC-V64I指令集测试

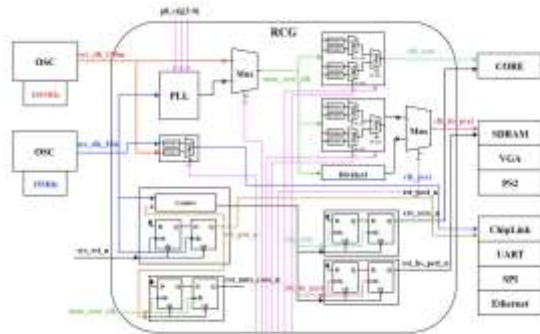
```
...  
fatal: 不是 git 仓库: libraries/yysysoc/.../git/modul  
...  
min3-cpu-tests) PASS!  
quick-sort-cpu-tests) PASS!  
dummy-cpu-tests) PASS!  
leap-year-cpu-tests) PASS!  
mov-c-cpu-tests) PASS!  
fib-cpu-tests) PASS!  
wanshu-cpu-tests) PASS!  
div-cpu-tests) PASS!  
hello-rt-cpu-tests) PASS!  
mops-cpu-tests) PASS!  
shift-cpu-tests) PASS!  
switch-cpu-tests) PASS!  
max-cpu-tests) PASS!  
pascal-cpu-tests) PASS!  
sum-cpu-tests) PASS!  
recursion-cpu-tests) PASS!  
load-store-cpu-tests) PASS!  
matrix-mul-cpu-tests) PASS!  
add-cpu-tests) PASS!  
string-cpu-tests) PASS!  
bit-cpu-tests) PASS!  
add-longlong-cpu-tests) PASS!
```

中断测试

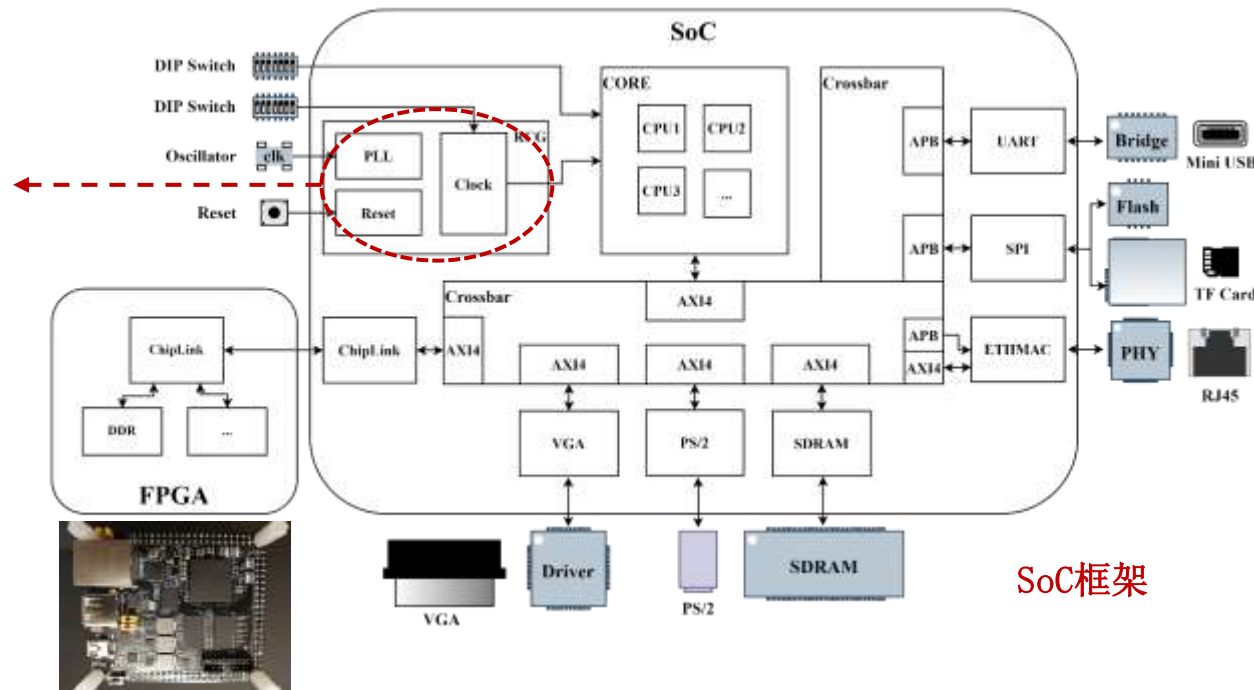
环境调用异常测试

# SoC设计 (提供丰富的开源IP) RISC-V

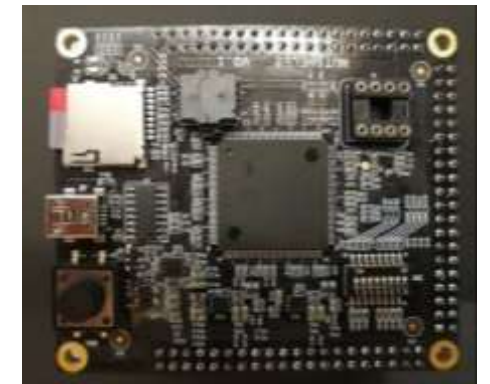
- 包括RISC-V处理器核、**PLL (锁相环)**、UART控制器 (通用异步收发传输器)、SPI控制器 (同步串行总线)、SDRAM (同步动态随机存储器)、VGA (视频图形阵列)、PS2(串行I/O接口)、ETHMAC (以太网媒体访问控制)、Chiplink (外围设备与访存系统的连接总线)
- 三个时钟域: 核时钟域、低速外设(ls\_peri)、高速外设(hs\_peri)
- 使用PLL将低频的时钟倍频, 通过拨码开关配置来选择CPU的工作频率



晶振	core	ls_peri	hs_peri
25MHz	25MHz (1)	10MHz	25MHz
25MHz	25MHz (2)	10MHz	25MHz
25MHz	25MHz (3)	10MHz	25MHz
25MHz	25MHz (4)	10MHz	25MHz
25MHz	25MHz (5)	10MHz	25MHz
25MHz	25MHz (6)	10MHz	25MHz
25MHz	25MHz (7)	10MHz	25MHz
25MHz	25MHz (8)	10MHz	25MHz
25MHz	25MHz (9)	10MHz	25MHz
25MHz	25MHz (10)	10MHz	25MHz
25MHz	25MHz (11)	10MHz	25MHz
25MHz	25MHz (12)	10MHz	25MHz
25MHz	25MHz (13)	10MHz	25MHz
25MHz	25MHz (14)	10MHz	25MHz
25MHz	25MHz (15)	10MHz	25MHz
25MHz	25MHz (16)	10MHz	25MHz
25MHz	25MHz (17)	10MHz	25MHz
25MHz	25MHz (18)	10MHz	25MHz
25MHz	25MHz (19)	10MHz	25MHz
25MHz	25MHz (20)	10MHz	25MHz
25MHz	25MHz (21)	10MHz	25MHz
25MHz	25MHz (22)	10MHz	25MHz
25MHz	25MHz (23)	10MHz	25MHz
25MHz	25MHz (24)	10MHz	25MHz
25MHz	25MHz (25)	10MHz	25MHz
25MHz	25MHz (26)	10MHz	25MHz
25MHz	25MHz (27)	10MHz	25MHz
25MHz	25MHz (28)	10MHz	25MHz
25MHz	25MHz (29)	10MHz	25MHz
25MHz	25MHz (30)	10MHz	25MHz
25MHz	25MHz (31)	10MHz	25MHz
25MHz	25MHz (32)	10MHz	25MHz
25MHz	25MHz (33)	10MHz	25MHz
25MHz	25MHz (34)	10MHz	25MHz
25MHz	25MHz (35)	10MHz	25MHz
25MHz	25MHz (36)	10MHz	25MHz
25MHz	25MHz (37)	10MHz	25MHz
25MHz	25MHz (38)	10MHz	25MHz
25MHz	25MHz (39)	10MHz	25MHz
25MHz	25MHz (40)	10MHz	25MHz
25MHz	25MHz (41)	10MHz	25MHz
25MHz	25MHz (42)	10MHz	25MHz
25MHz	25MHz (43)	10MHz	25MHz
25MHz	25MHz (44)	10MHz	25MHz
25MHz	25MHz (45)	10MHz	25MHz
25MHz	25MHz (46)	10MHz	25MHz
25MHz	25MHz (47)	10MHz	25MHz
25MHz	25MHz (48)	10MHz	25MHz
25MHz	25MHz (49)	10MHz	25MHz
25MHz	25MHz (50)	10MHz	25MHz
25MHz	25MHz (51)	10MHz	25MHz
25MHz	25MHz (52)	10MHz	25MHz
25MHz	25MHz (53)	10MHz	25MHz
25MHz	25MHz (54)	10MHz	25MHz
25MHz	25MHz (55)	10MHz	25MHz
25MHz	25MHz (56)	10MHz	25MHz
25MHz	25MHz (57)	10MHz	25MHz
25MHz	25MHz (58)	10MHz	25MHz
25MHz	25MHz (59)	10MHz	25MHz
25MHz	25MHz (60)	10MHz	25MHz
25MHz	25MHz (61)	10MHz	25MHz
25MHz	25MHz (62)	10MHz	25MHz
25MHz	25MHz (63)	10MHz	25MHz
25MHz	25MHz (64)	10MHz	25MHz
25MHz	25MHz (65)	10MHz	25MHz
25MHz	25MHz (66)	10MHz	25MHz
25MHz	25MHz (67)	10MHz	25MHz
25MHz	25MHz (68)	10MHz	25MHz
25MHz	25MHz (69)	10MHz	25MHz
25MHz	25MHz (70)	10MHz	25MHz
25MHz	25MHz (71)	10MHz	25MHz
25MHz	25MHz (72)	10MHz	25MHz
25MHz	25MHz (73)	10MHz	25MHz
25MHz	25MHz (74)	10MHz	25MHz
25MHz	25MHz (75)	10MHz	25MHz
25MHz	25MHz (76)	10MHz	25MHz
25MHz	25MHz (77)	10MHz	25MHz
25MHz	25MHz (78)	10MHz	25MHz
25MHz	25MHz (79)	10MHz	25MHz
25MHz	25MHz (80)	10MHz	25MHz
25MHz	25MHz (81)	10MHz	25MHz
25MHz	25MHz (82)	10MHz	25MHz
25MHz	25MHz (83)	10MHz	25MHz
25MHz	25MHz (84)	10MHz	25MHz
25MHz	25MHz (85)	10MHz	25MHz
25MHz	25MHz (86)	10MHz	25MHz
25MHz	25MHz (87)	10MHz	25MHz
25MHz	25MHz (88)	10MHz	25MHz
25MHz	25MHz (89)	10MHz	25MHz
25MHz	25MHz (90)	10MHz	25MHz
25MHz	25MHz (91)	10MHz	25MHz
25MHz	25MHz (92)	10MHz	25MHz
25MHz	25MHz (93)	10MHz	25MHz
25MHz	25MHz (94)	10MHz	25MHz
25MHz	25MHz (95)	10MHz	25MHz
25MHz	25MHz (96)	10MHz	25MHz
25MHz	25MHz (97)	10MHz	25MHz
25MHz	25MHz (98)	10MHz	25MHz
25MHz	25MHz (99)	10MHz	25MHz
25MHz	25MHz (100)	10MHz	25MHz



外设地址映射

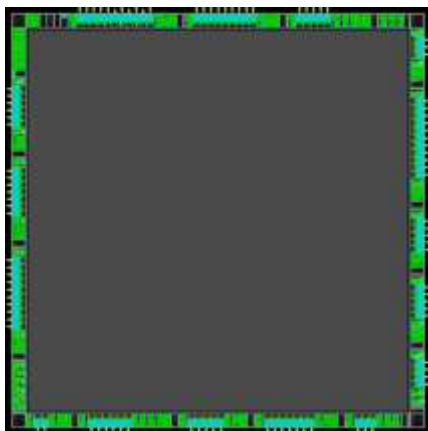


SoC板卡

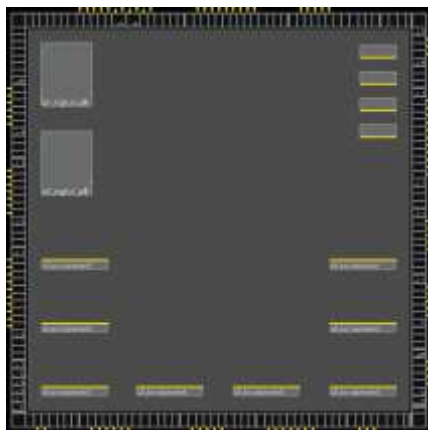
SoC框架



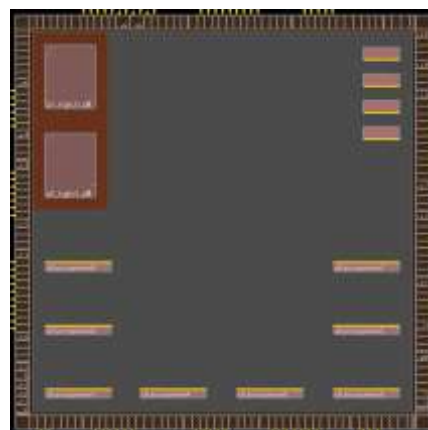
- 执行一条命令即可实现从初始化到布局布线的整个过程
- `gmake -f ../pr.makefile initialize floorplan preplace place CTS route`



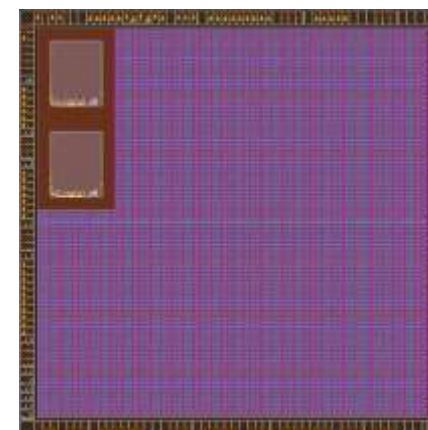
IO



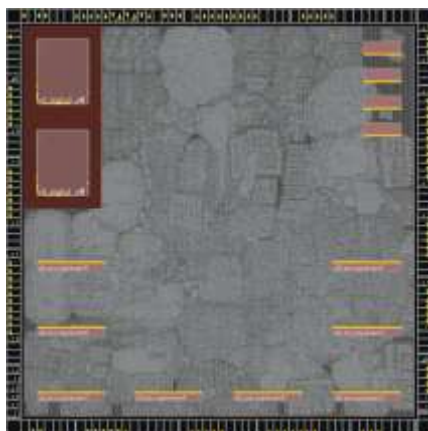
floorplan



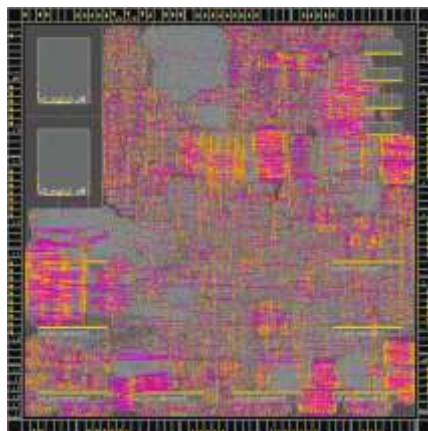
blockage



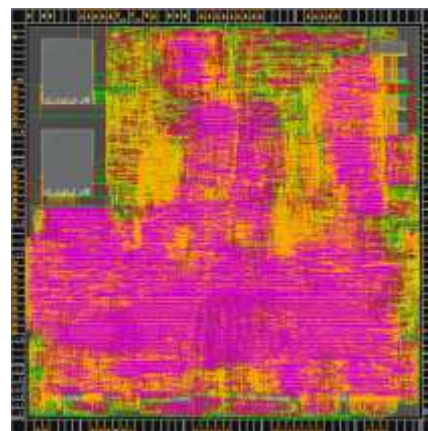
powerplan



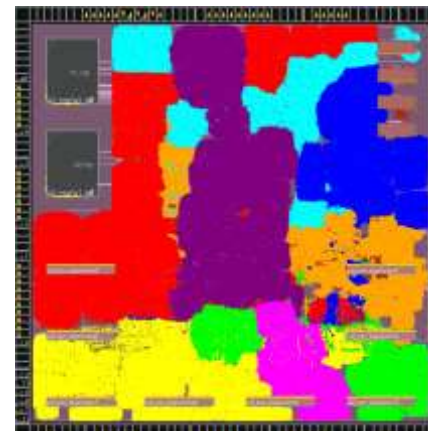
place



CTS



route



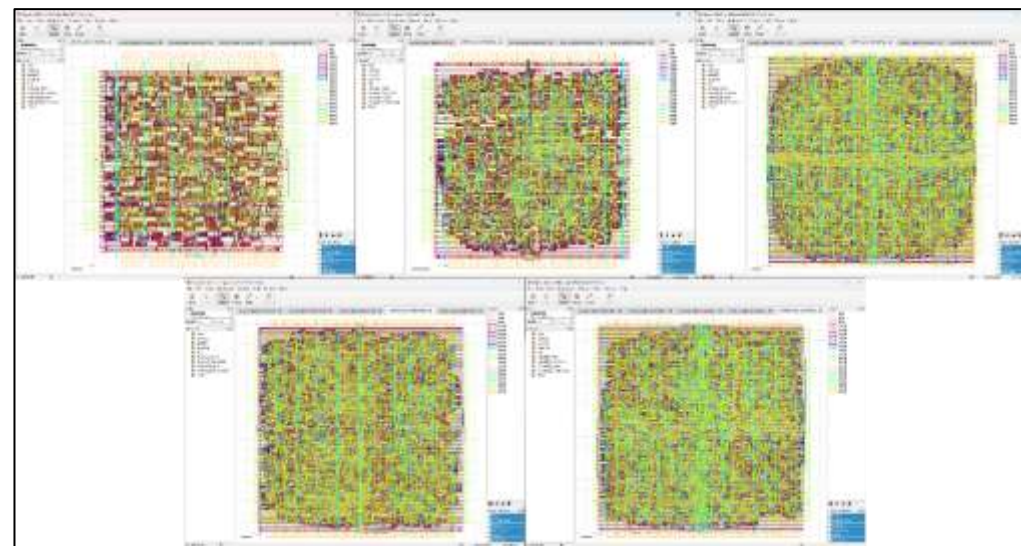
layout

# ① 开源EDA工具链进展

- **开源EDA工具链：**支持开源EDA工具，国产EDA工具和国际主流EDA工具
  - 现已支持110nm、28nm、40nm等主流工艺（smic/hl），支持接入国际主流EDA工具
  - 在开源SoC芯片上进行大量测试，CI/CD 自动化测试流程
  - 在 AtomGit和Gitee上开源，并对外宣传推广

Harry add iEDA scripts 7668c9d 1个月前 17 次提交		
bin	update iEDA binary and scripts	1个月前
picorv32a	update scripts	8个月前
result	delete example results (.def, .v)	3个月前
sky130	change env variables name	1个月前
tools	add iEDA scripts	1个月前
.gitignore	change env variables name	1个月前
README.md	init repo	9个月前
rtl2gds.sh	update iEDA binary and scripts	1个月前

开源EDA工具链代码（即将在AtomGit开源）

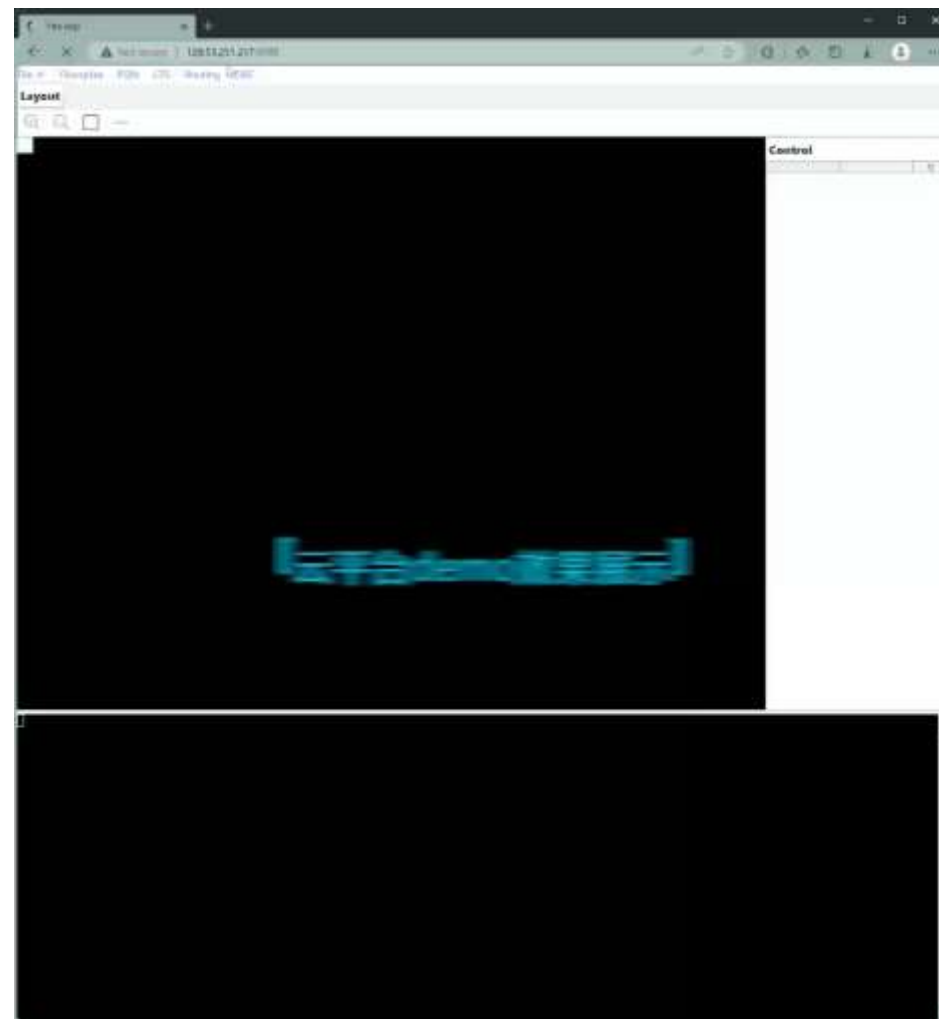
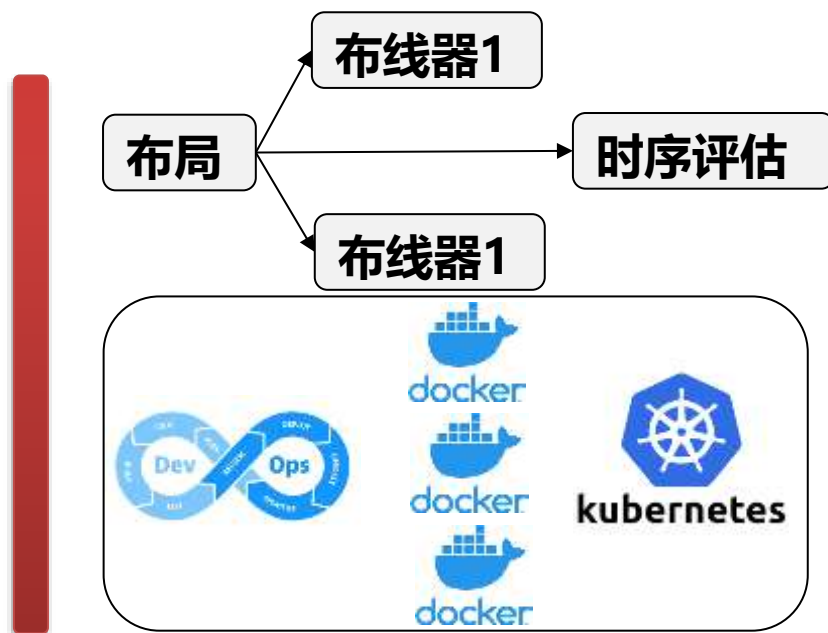
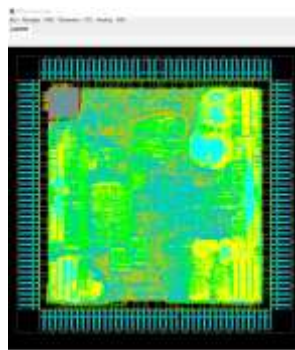


五颗芯片的设计版图



# 基于开源EDA工具链的芯片设计平台

- **云原生：** 重构EDA软件的设计方法学，细粒度切分EDA软件到工具、子工具、模块、子模块等级别，定义为微服务，部署到开放式云平台



## ② 开源芯片数据集

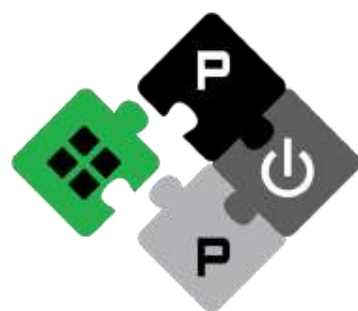
- 开源芯片 RTL 数据集（已包含四十余款开源核和一生一芯流片核）
- 构建面向智能EDA模型训练的EDA标签化数据集



香山



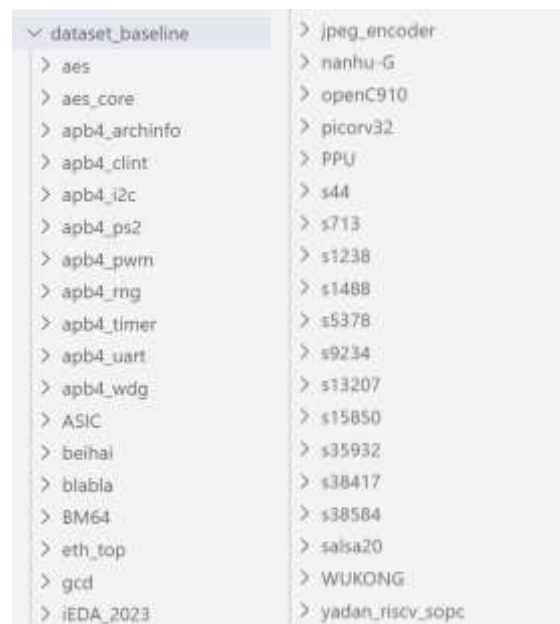
一生一芯



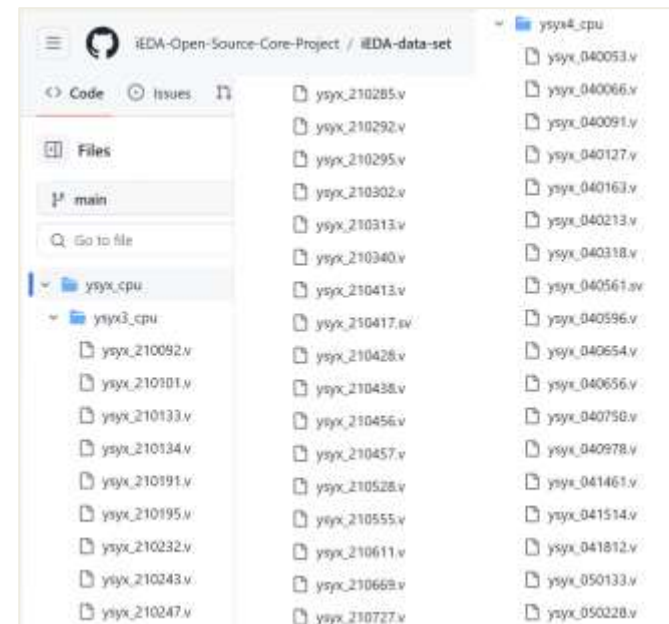
PULP



Xuantie 玄铁



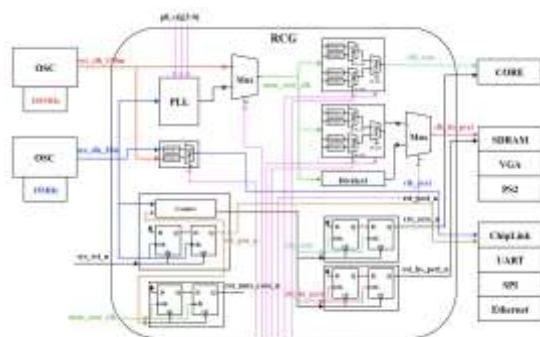
开源核RTL数据集



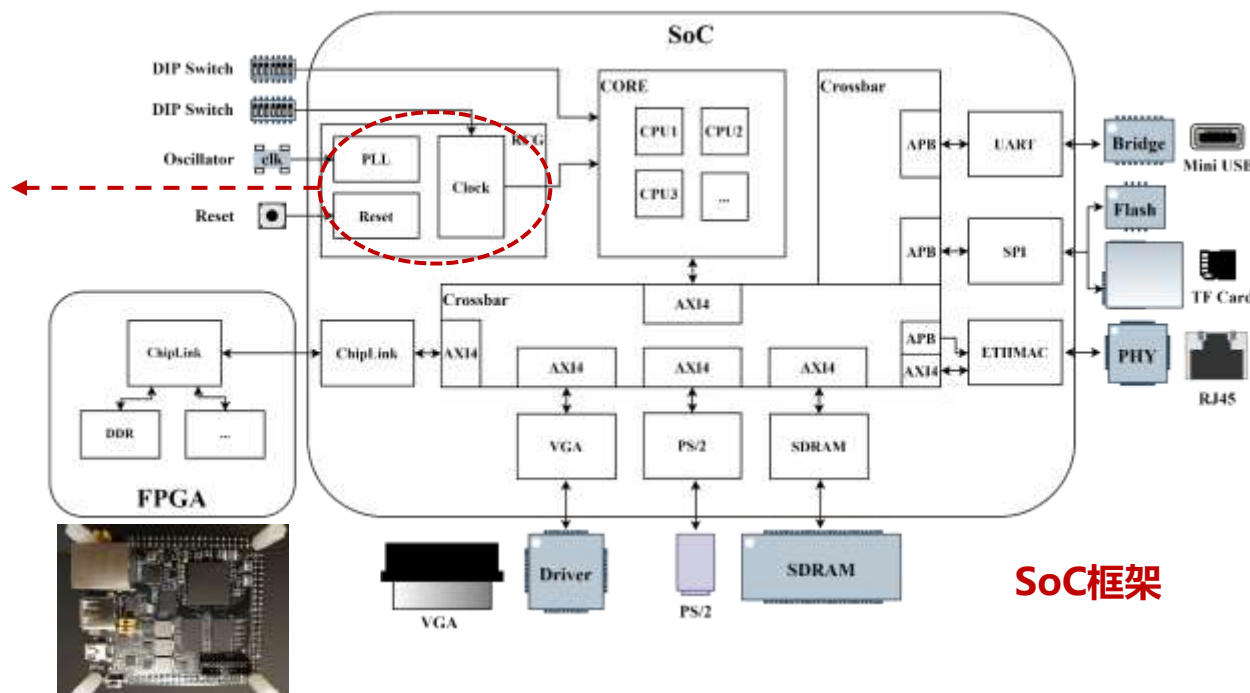
“一生一芯” RTL数据集

### ③ 设计面向市场销售的芯片产品

- 开源开发板：基于开源芯片（开源EDA和开源IP）的 RISC-V 开发板
  - 支持源代码级别复现和自定义修改，可面向市场销售
- 可量产芯片：联合芯片厂商，提供后端设计服务（基于混合EDA方案）



晶振	clk	clk_div	PLL输出	core	io_peri	io_peri
25MHz	F3091	6x22	100MHz	250MHz (16)	100MHz	250MHz
25MHz	F3091	6x21	100MHz	50MHz (12)	100MHz	250MHz
25MHz	F3091	6x20	100MHz	100MHz	100MHz	250MHz
25MHz	F3101	6x25	100MHz	100MHz (2)	100MHz (2)	250MHz
25MHz	F3091	6x28	100MHz	100MHz	100MHz (2)	250MHz
25MHz	F3101	6x22	100MHz	100MHz	100MHz (3)	250MHz
25MHz	F3101	6x20	100MHz	100MHz	100MHz (4)	250MHz
100MHz	F3090	6x40	无	100MHz	100MHz	250MHz (2)
100MHz	F3090	6x41	无	500MHz (12)	100MHz	250MHz (2)
100MHz	F3090	6x42	无	250MHz (16)	100MHz	250MHz



SoC框架

外设地址映射



SoC板卡

# ④ 服务高校及科研院所

- 服务社区、高校和企业的定制芯片和验证IP等需求

## 服务国内高校和研究所定制芯片

**浙江大学**  
ZHEJIANG UNIVERSITY

**深圳技术大学**  
SHENZHEN TECHNOLOGY UNIVERSITY

**深圳理工大学**  
SHENZHEN UNIVERSITY OF  
ADVANCED TECHNOLOGY

**北京邮电大学**  
Beijing University of Posts and Telecommunications

**南京大学**  
NANJING UNIVERSITY

**山东科技大学**  
SHANDONG UNIVERSITY OF SCIENCE AND TECHNOLOGY


**中国空间技术研究院**  
China Academy of Space Technology

**中国科学院上海微系统与信息技术研究所**  
Shanghai Institute of Microsystem and Information Technology Chinese Academy of Sciences

## 服务企业定制芯片和验证IP

**TERAMIC**  
太芮科技

## 服务国外高校和社区芯片验证

**NAZARBAYEV UNIVERSITY**

纳扎尔巴耶夫大学 (哈萨克斯坦)

## 各合作项目进展时间表

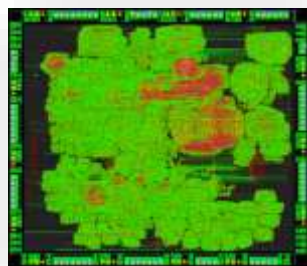
序号	客户名称	项目名称	项目简介	项目状态	2024-03-12	2024-03-27	2024-04-10	2024-04-24	2024-05-08
1	某高校	定制芯片	为该校研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
2	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
3	某研究所	定制芯片	为某研究所研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
4	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
5	某高校	定制芯片	为该校研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
6	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
7	某研究所	定制芯片	为某研究所研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
8	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
9	某高校	定制芯片	为该校研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
10	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
11	某研究所	定制芯片	为某研究所研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
12	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
13	某高校	定制芯片	为该校研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
14	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
15	某研究所	定制芯片	为某研究所研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
16	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
17	某高校	定制芯片	为该校研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
18	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					
19	某研究所	定制芯片	为某研究所研发一款定制芯片，用于其科研项目。项目进展顺利，预计下月完成。	进行中					
20	某企业	验证IP	为企业提供验证IP服务，用于其新产品开发。项目进展顺利，预计下月完成。	进行中					



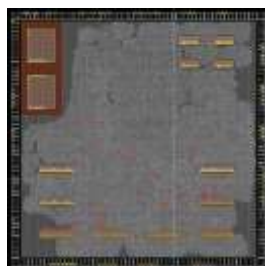
# 联动社区、高校和企业

- 支持开源社区、高校验证IP，提供后端设计服务和流片渠道等
- **免费**：开源社区的高质量攀升中的开源 IP 和开源 EDA 工具
- **付费（公益）**：高校科研和中小企业验证 IP 的需求

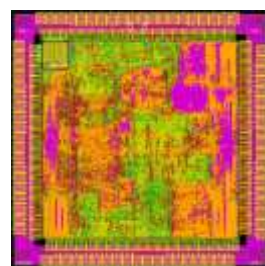
## 科学研究



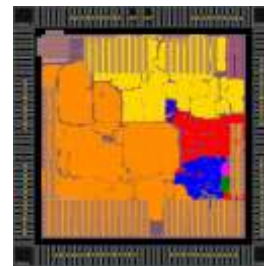
开源EDA 110nm验证芯片



开源EDA 110nm验证芯片



开源EDA 28nm验证芯片

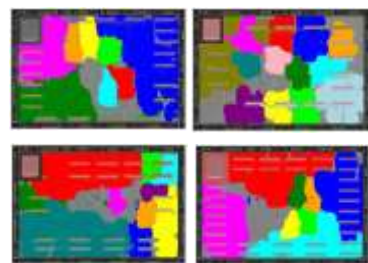


28nm类脑计算架构 &  
基于RISC-V的SSD主控芯片

本年度正在接洽：

- 模拟IP，如 28nm PLL等
- 模拟IP，如110nm PLL等
- 其他（如NoC、处理器安全机制等）

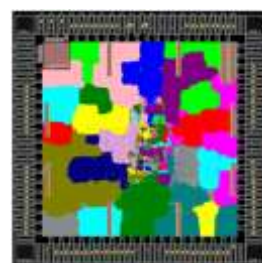
## 高校教学



一生一芯三期第一批39  
个处理器核



一生一芯三期第二批9  
个处理器核

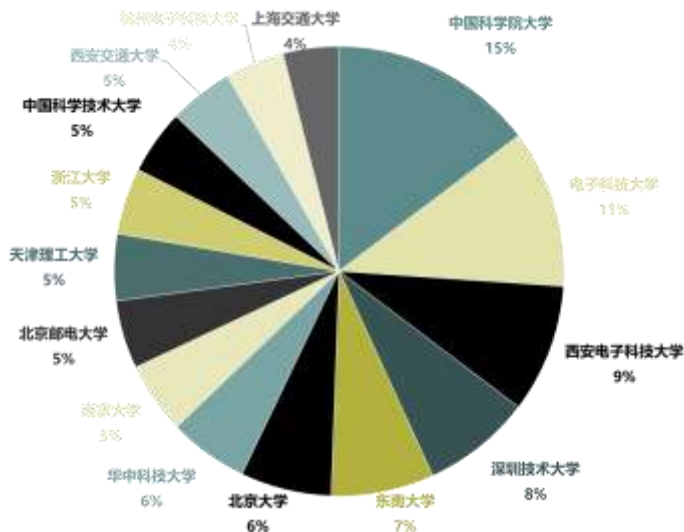
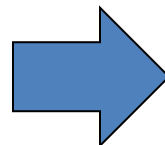
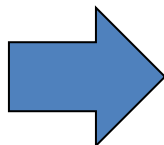
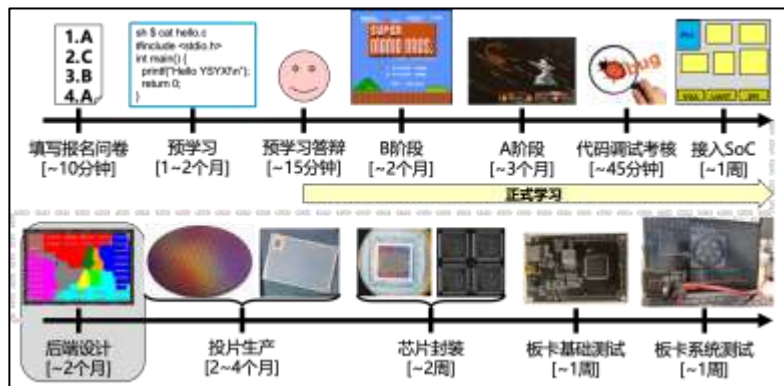


一生一芯四期第一批13  
个处理器核



## ⑥ 教学：应用到“一生一芯”流程

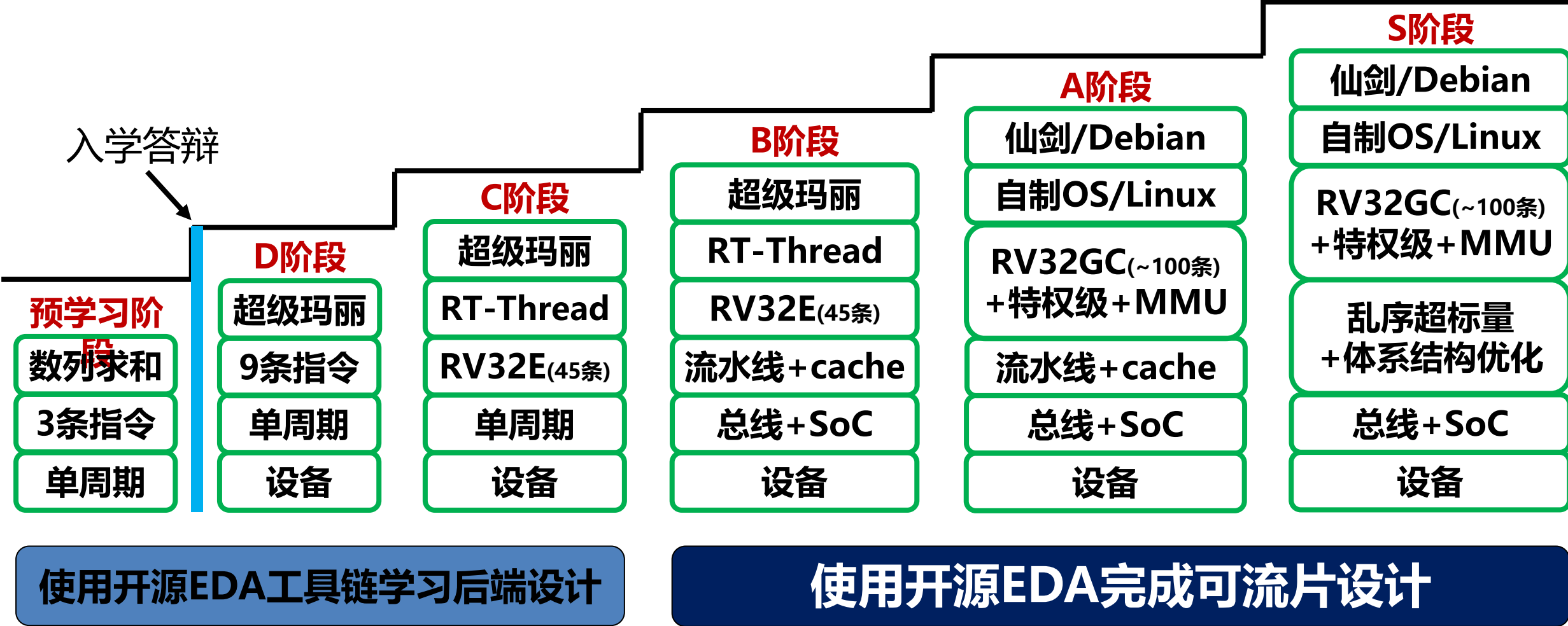
- 基于**开源模式**、**贯通课程**的**实践型**大规模芯片设计人才培养计划
- 助力人才培养、技术创新等，快速扩大用户规模，进入**研用结合的健康迭代流程**
- 开展到第六期，累积参与人数超**7000人**，覆盖**680所高校**
  - 坚持**开放性和公益性**，不限高校、专业和年级，**支持国内在校生免费流片**
  - 探索**低成本流片**方案，把每名同学的流片成本**降低到1万元**





# 教学：应用到 “一生一芯” 流程

■ 划分更多子阶段的 “一生一芯” 教学流程，扩展芯片人才培养规模



## ⑦ 愿景：使用开源EDA和开源IP设计芯片

- 使用开源EDA、开源IP和开源系统软件，设计芯片并流片验证，构建可以使用的原型系统



**2024年：**  
设计一个掌上游戏机

可面向市场销售的开源芯片开发板



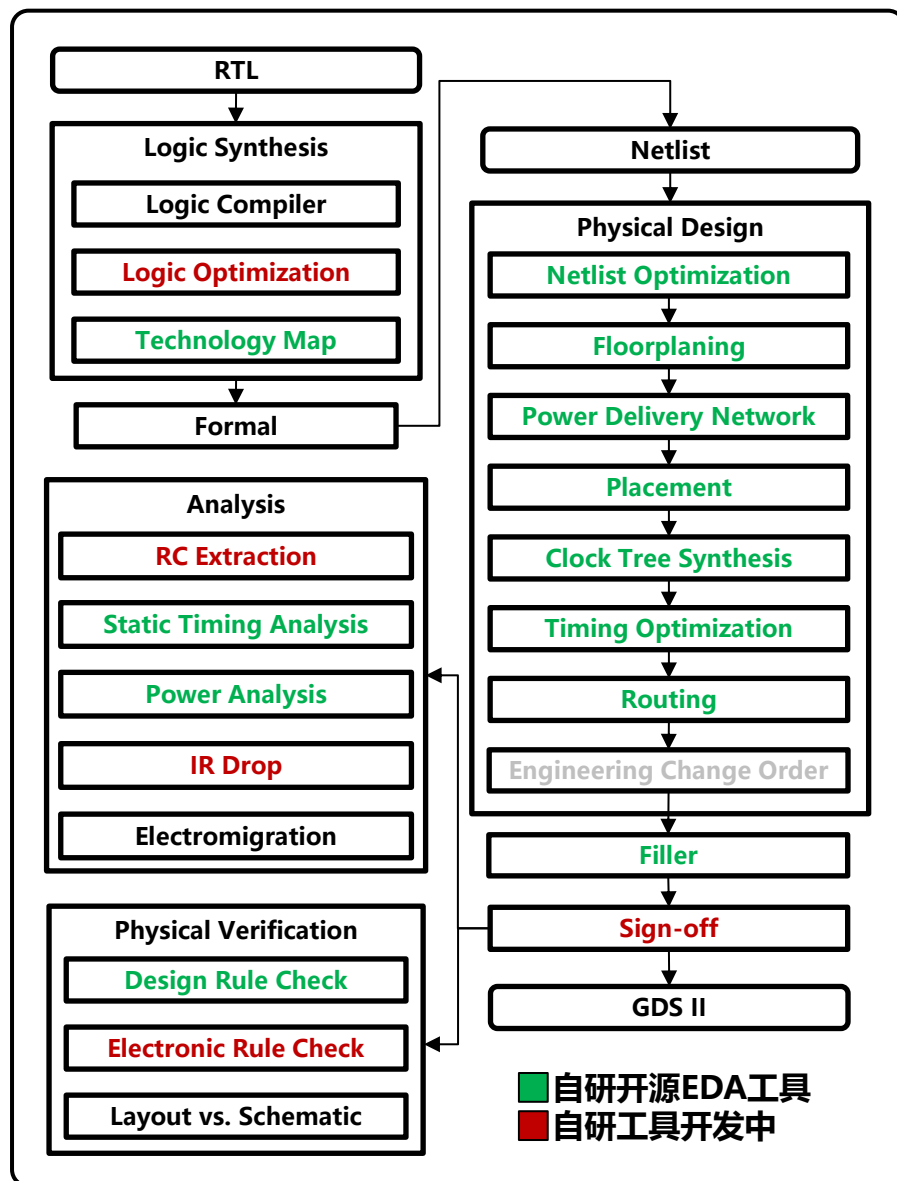
**2026~2028年：**  
设计一个平板，运行Android，  
支持联网游戏和视频等（使用香山和嵌入式GPU等）



**2028~2030年**  
设计一个笔记本，支持  
Ubuntu 和桌面办公等（使用  
香山和GPU等）

# 三、定制RISC-V游戏掌机 项目

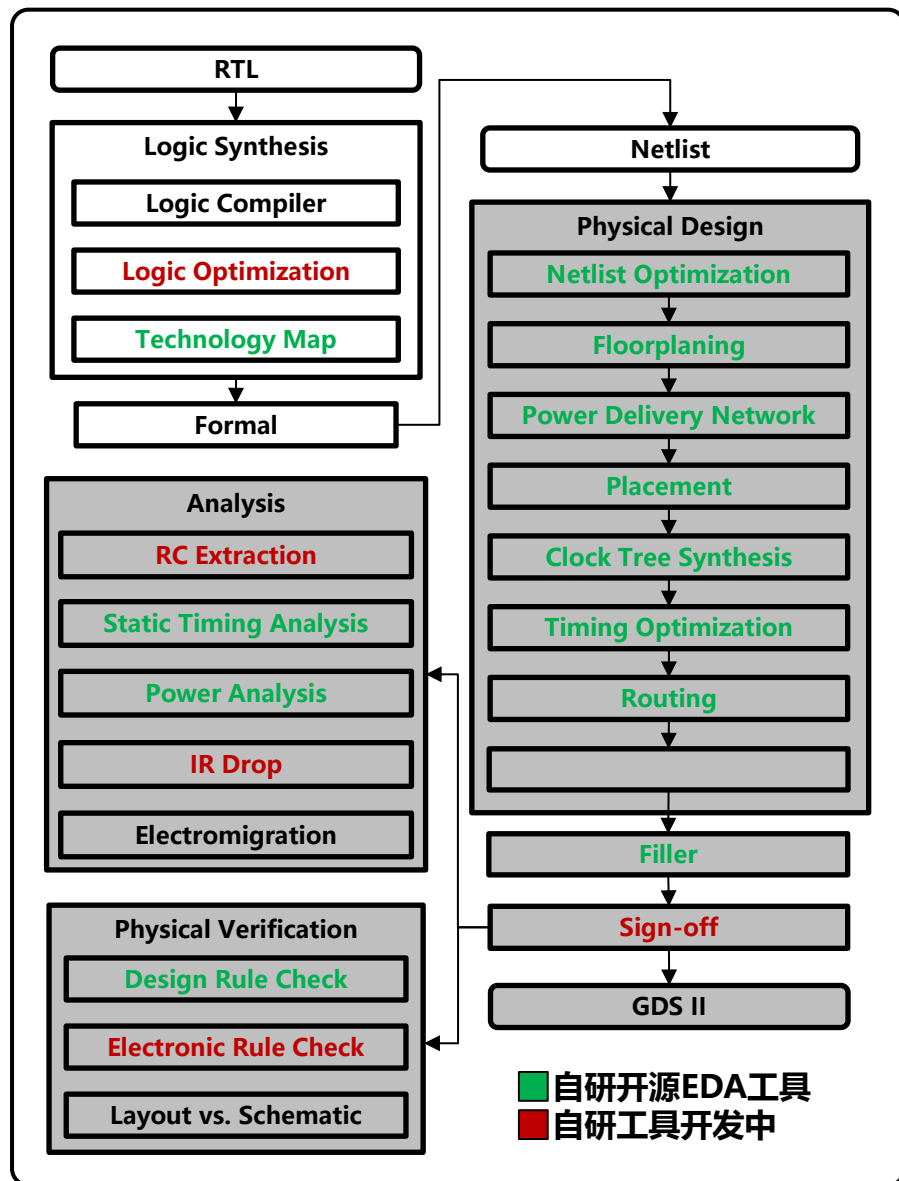
# 开源后端设计方案



## 使用开源EDA工具完成开源掌机SoC的后端设计

- 使用若干开源EDA工具：iEDA、sv2v、yosys、abc
- 覆盖逻辑综合、逻辑映射、布图规划、电源规划、网表优化、布局、时钟树、布线、静态时序分析、功耗分析等后端设计全流程
- 十余款自研工具（iEDA工具链，完成三次流片验证）、其余多款工具在研中
- 支持开源130nm和商业28nm工艺下的后端设计

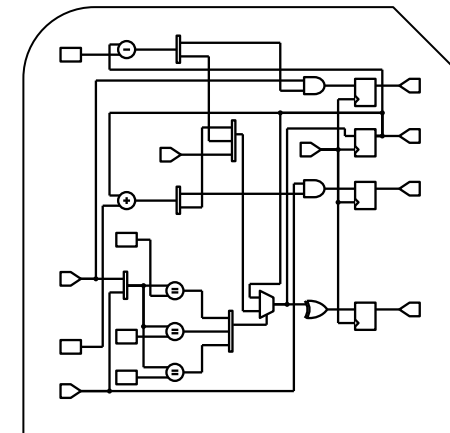
# 开源后端设计方案：逻辑综合



```
module conv;
reg [31:0] m[0:8192];
reg [12:0] pc;
reg [31:0] acc;
reg[15:0] ir;

always
begin
ir = m[pc];
if(ir[15:13] == 3b' 000)
pc = m[ir[12:0]];
else if (ir[15:13] ==
3' b010)
acc = -m[ir[12:0]];
...
```

SoC.sv (RTL)



SoC.v (Netlist)

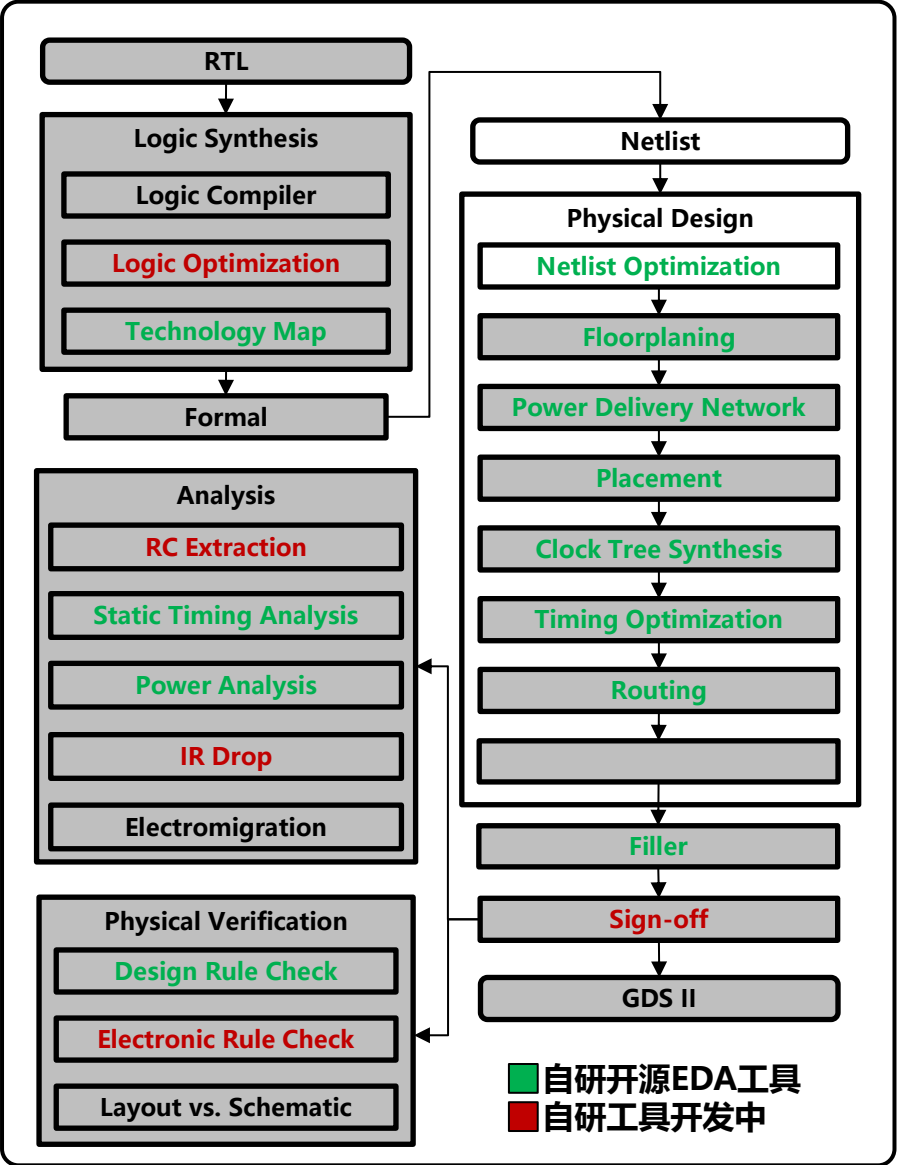
## 使用 Yosys 完成逻辑综合和逻辑映射 (Yosys-ABC)

- 目标：将寄存器级的芯片描述转化为门电路的网表级描述
- 输入：RTL代码 (SystemVerilog/Verilog)
- 输出：Netlist门级网表 (Verilog)
- 可选：sv2v工具进行SystemVerilog-Verilog转换

## 相关自研项目：

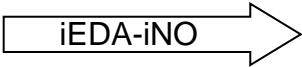
- iMAP 完成了基本工艺映射算法以及逻辑优化算子
- 物理综合 LogicFactory 工具正在开发中

# 开源后端设计方案：网表优化



```
===== Program start time 2023-01-28 19:55:47=====
[Result: ] Find 3011 Net with fanout violation.
[Result: ] Insert 5140 Buffers.
===== Program end time 2023-01-28 20:25:50=====
```

(示) 关键参数	数据	分析
Instance个数	302644	比Floorplan阶段多插入 <b>5140</b> 个buffers, 单元面积增加约 0.008cm^2
Net个数	311869	比Floorplan阶段多 <b>5140</b> 条net
Pin分布	pin (>= 32) = 7	大于 32 个pin的net由 <b>2893</b> 优化为 <b>7</b> 条

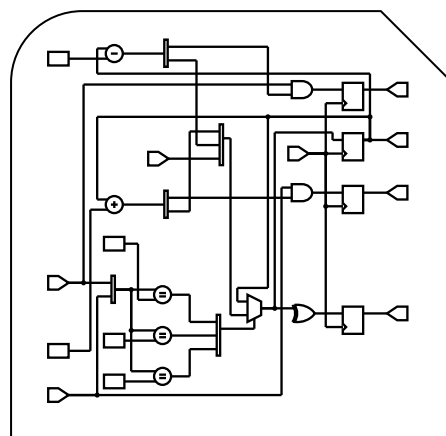
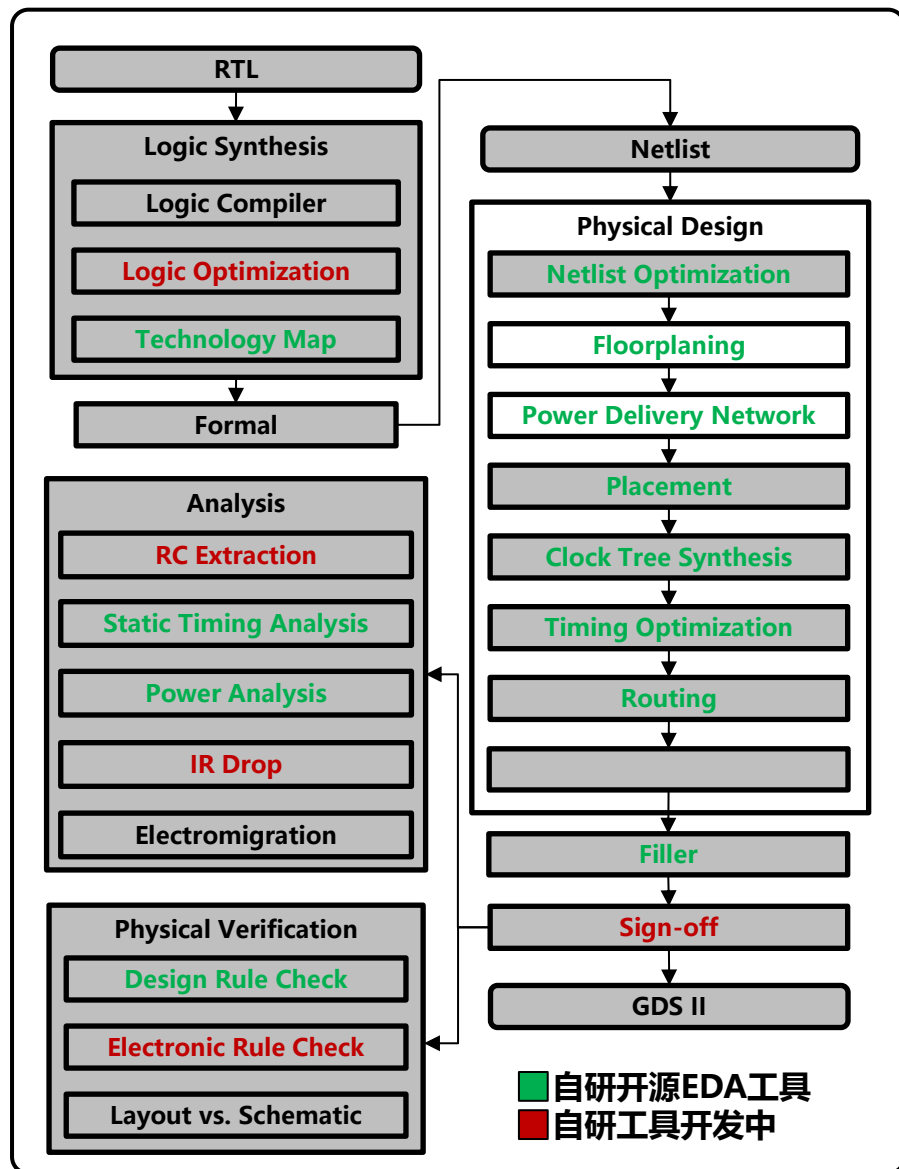
SoC.v (Netlist)  SoC\_fix\_fanout.v (Netlist)

## 使用iEDA完成网表优化 (iEDA-iNO)

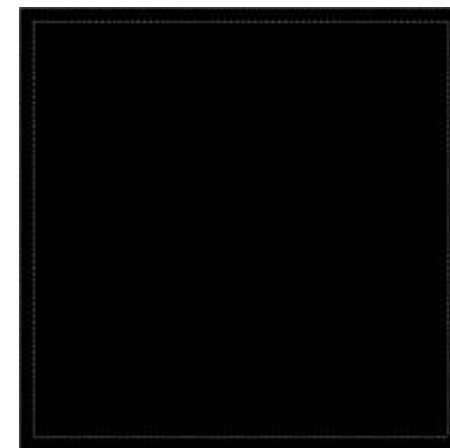
- 目标：确保每个逻辑门满足**最大扇出约束**
- 输入：Netlist门级网表 (Verilog)
- 输出：满足扇出约束的网表 (Verilog)
- 流程：构建和筛选线网、更换单元、插buffer、重构线网、迭代检查、保存输出



# 开源后端设计方案：布图规划



SoC\_fix\_fanout.v (Netlist)



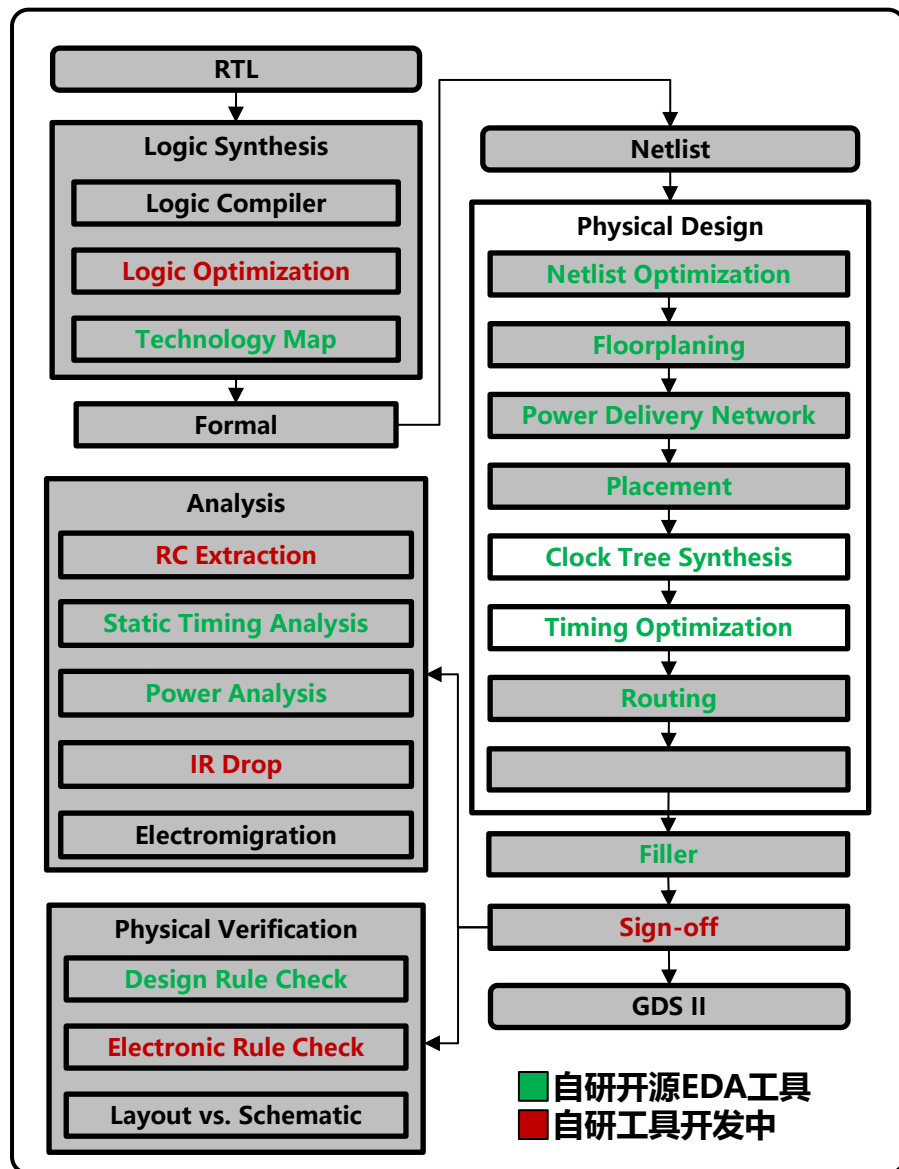
SoC\_iFP.def (Design Exchange Format)

## 使用iEDA完成布图规划 (iFP) 和电源规划 (iPDN)

- 目标：确认IO/供电/Marco宏单元等元素的位置
- 输入：Netlist门级网表 (Verilog)
- 输出：iFP.def (Design Exchange Format)
- 流程：版图初始化、IO单元规划、宏单元规划、物理单元放置、电源规划、保存输出def

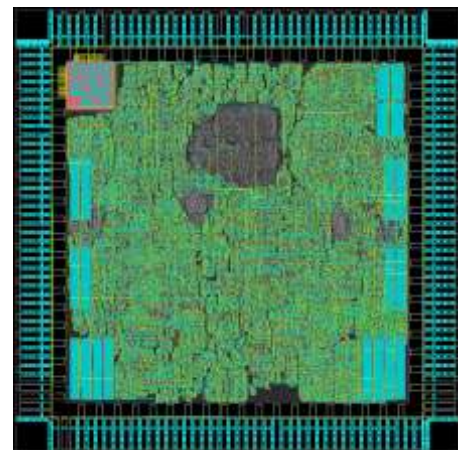


# 开源后端设计方案：时钟树综合



SoC\_iPL.def

iEDA-iCTS

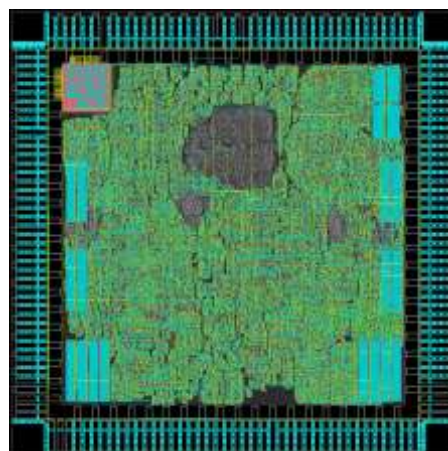
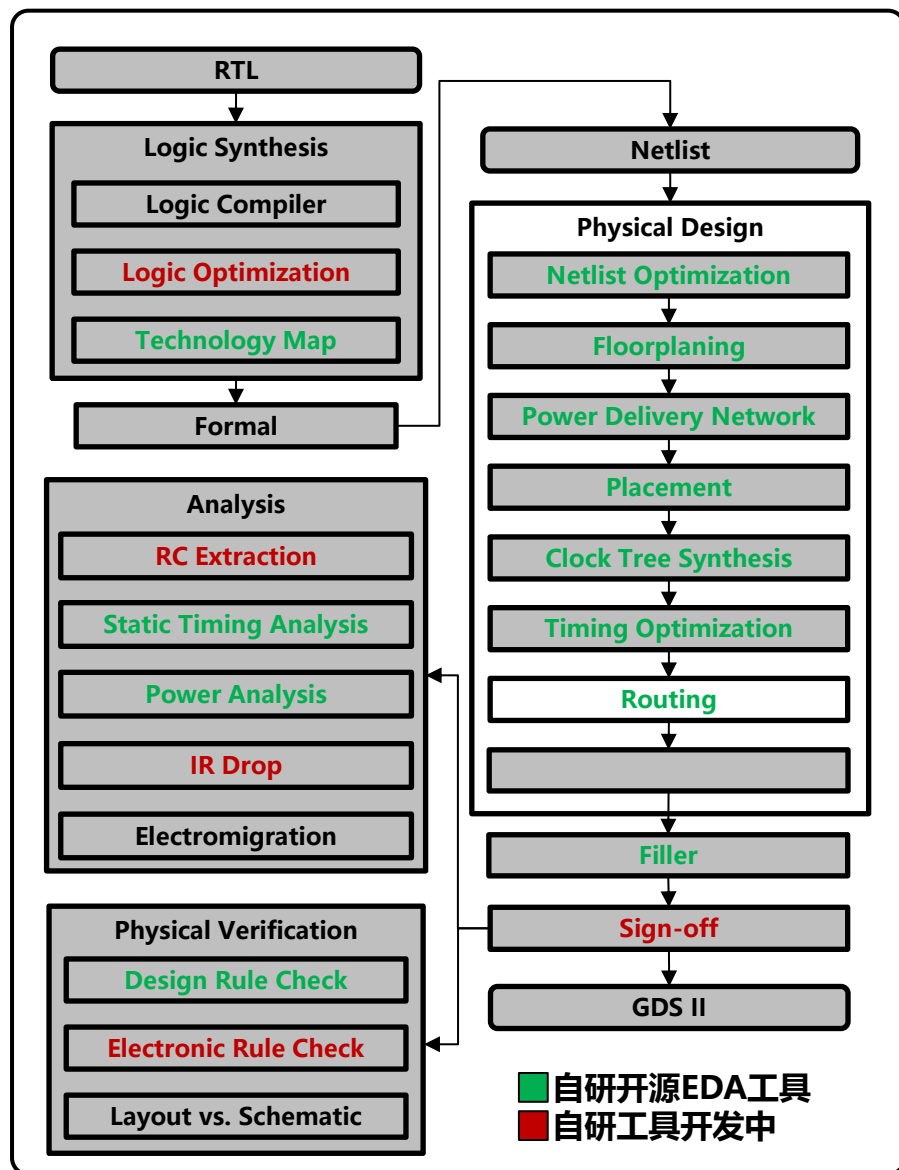


SoC\_iCTS.def

## 使用iEDA完成时钟树综合 (iEDA-iCTS)

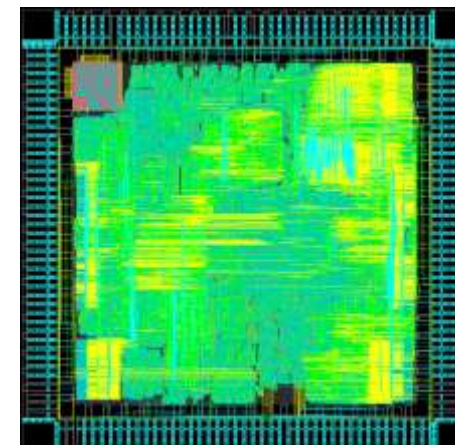
- 目标：生成时钟树，满足最大扇出/时序要求等约束
- 输入：SoC\_iPL.def (Design Exchange Format)
- 输出：SoC\_iCTS.def (Design Exchange Format)
- 流程：时钟分布、分布分解(聚类)、约束建模、线网生成(优化)、时序检查

# 开源后端设计方案：布线



SoC\_iCTS.def

iEDA-iRT

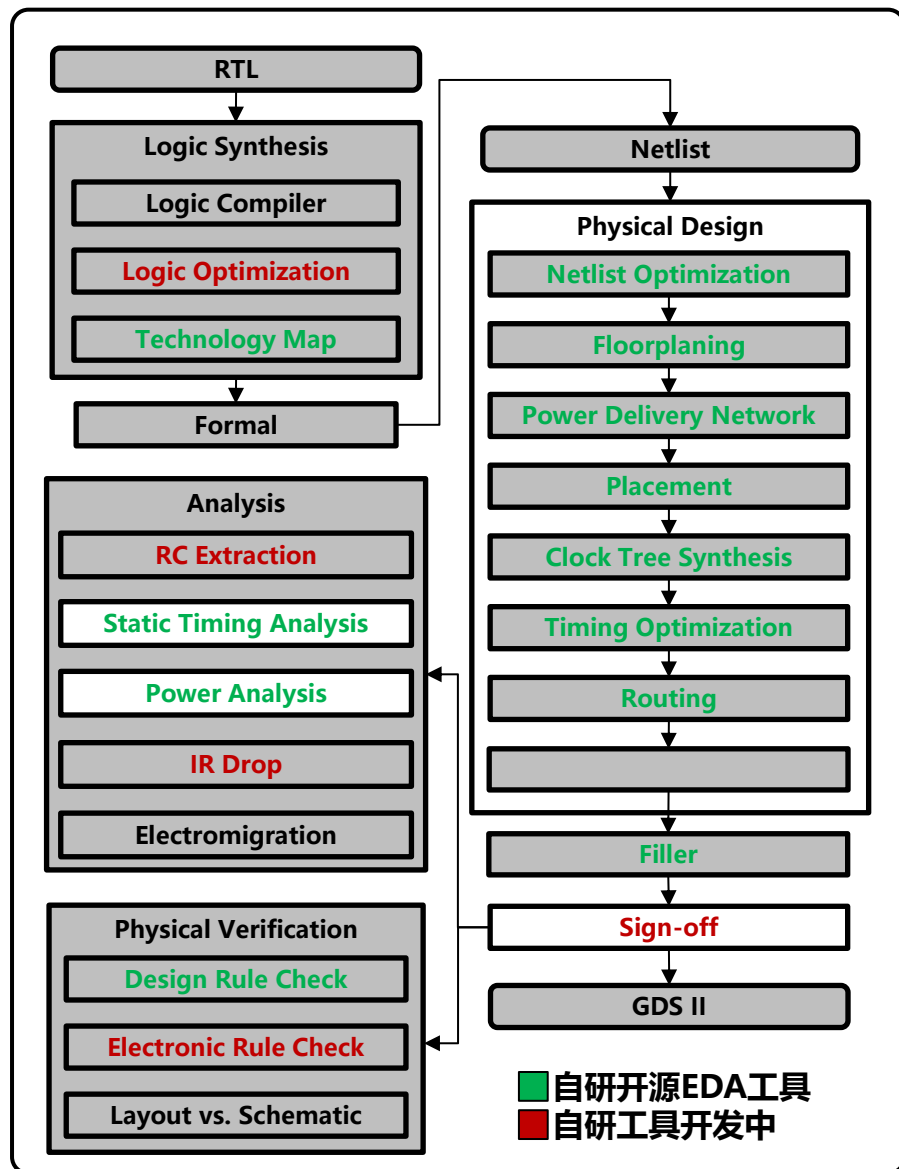


SoC\_iRT.def

## 使用iEDA完成全局/详细布线 (iEDA-iRT)

- 目标：生成线网，满足线长/时序/拥塞/DRC等约束
- 输入：SoC\_iCTS.def (Design Exchange Format)
- 输出：SoC\_iRT.def (Design Exchange Format)
- 流程：pin点分配、资源分配、全局布线、轨道分配、详细布线、布线优化

# 开源后端设计方案：静态时序分析/功耗分析



Point	Fanout	Capacitance	Resistance	Transition	Delta Delay	Derate	Incr	Path
u1_clk:XC (PD00EDG_V_G)		0.002	0.000	0.000		1.000	0.000	0.000r
sys_clk_100m (clock net)	1	0.002	0.000	0.000	0.000	1.000	0.000	0.000r
u1_clk_xc_donttouch:I (CK8D12BP40P140LVT)		0.002	0.000	0.000		0.885	0.011	0.011r
u1_clk_xc_donttouch:Z (CK8D12BP40P140LVT)	2	0.002	0.000	0.000	0.000	1.000	0.000	0.011r
sys_clk_100m_buf (clock net)		0.001	0.000	0.000		0.885	0.032	0.043r
u0_rcg/u1_lvt_ckmux2hdv4:I0 (CKMUX2D4BP40P140LVT)		0.006	0.000	0.018		1.000	0.000	0.043r
u0_rcg/u1_lvt_ckmux2hdv4:Z (CKMUX2D4BP40P140LVT)	5	0.006	0.000	0.018	0.000	0.885	0.021	0.064r
u0_rcg/mux_core_clk (clock net)		0.001	0.000	0.018		1.000	0.000	0.043r
u0_rcg/mux_core_clk_0_buf:I (CK8D4BP35P140)		0.000	0.000	0.015		0.885	0.021	0.064r
u0_rcg/mux_core_clk_0_buf:Z (CK8D4BP35P140)	17	0.000	0.000	0.015	0.000	1.000	0.000	0.064r
u0_rcg/mux_core_clk_0 (clock net)		0.000	0.000	0.015		0.885	0.021	0.064r
u0_rcg/mux_core_clk_div3/gt_enl_reg:CP (DFSND18WP40P140LVT)		0.000	0.000	0.015		1.000	0.000	0.064r
clock CLK_u1_clk_XC (rise edge)		0.000	0.000	0.015		0.885	0.021	0.064r
clock network delay (propagated)		0.000	0.000	0.015		0.820	0.045	0.109r
u0_rcg/mux_core_clk_div3/gt_enl_reg:CP (DFSND18WP40P140LVT)		0.001	0.000	0.009		1.000	0.000	0.109r
u0_rcg/mux_core_clk_div3/gt_enl_reg:Q (DFSND18WP40P140LVT)	1	0.001	0.000	0.009	0.000	1.000	0.000	0.109r
u0_rcg/mux_core_clk_div3/gt_enl (net)		0.001	0.000	0.009		1.000	0.000	0.109r
u0_rcg/mux_core_clk_div3/U_GT1:E (CKLNQD4BP40P140LVT)		0.001	0.000	0.009		1.000	0.000	0.109r
u1_clk:XC (PD00EDG_V_G)		0.002	0.000	0.000		1.000	0.000	0.000r
sys_clk_100m (clock net)	1	0.002	0.000	0.000	0.000	1.000	0.000	0.000r
u1_clk_xc_donttouch:I (CK8D12BP40P140LVT)		0.002	0.000	0.000		1.030	0.013	0.013r
u1_clk_xc_donttouch:Z (CK8D12BP40P140LVT)	2	0.002	0.000	0.000	0.000	1.030	0.038	0.051r
sys_clk_100m_buf (clock net)		0.001	0.000	0.000		1.000	0.000	0.013r
u0_rcg/u1_lvt_ckmux2hdv4:I0 (CKMUX2D4BP40P140LVT)		0.006	0.000	0.006		1.030	0.038	0.051r
u0_rcg/u1_lvt_ckmux2hdv4:Z (CKMUX2D4BP40P140LVT)	5	0.006	0.000	0.018	0.000	1.000	0.000	0.051r
u0_rcg/mux_core_clk (clock net)		0.001	0.000	0.018		1.030	0.025	0.076r
u0_rcg/mux_core_clk_0_buf:I (CK8D4BP35P140)		0.000	0.000	0.015		1.030	0.025	0.076r
u0_rcg/mux_core_clk_0_buf:Z (CK8D4BP35P140)	17	0.000	0.000	0.015	NA	1.000	0.000	0.076r
u0_rcg/mux_core_clk_0 (clock net)		0.001	0.000	0.000		0.000	0.000	0.076r
u0_rcg/mux_core_clk_div3/U_GT1:CP (CKLNQD4BP40P140LVT)		0.001	0.000	0.000		0.000	0.000	0.076r
clock CLK_u1_clk_XC (rise edge)		0.001	0.000	0.000		0.000	0.000	0.076r
clock network delay (propagated)		0.001	0.000	0.000		0.000	0.000	0.076r
u0_rcg/mux_core_clk_div3/U_GT1:CP (CKLNQD4BP40P140LVT)		0.001	0.000	0.000		0.000	0.000	0.076r
library hold time						-0.011	0.064	0.064r
clock reconvergence pessimism								0.309r
data require time								0.045r
data arrival time								0.045r
slack (NET)								0.045r

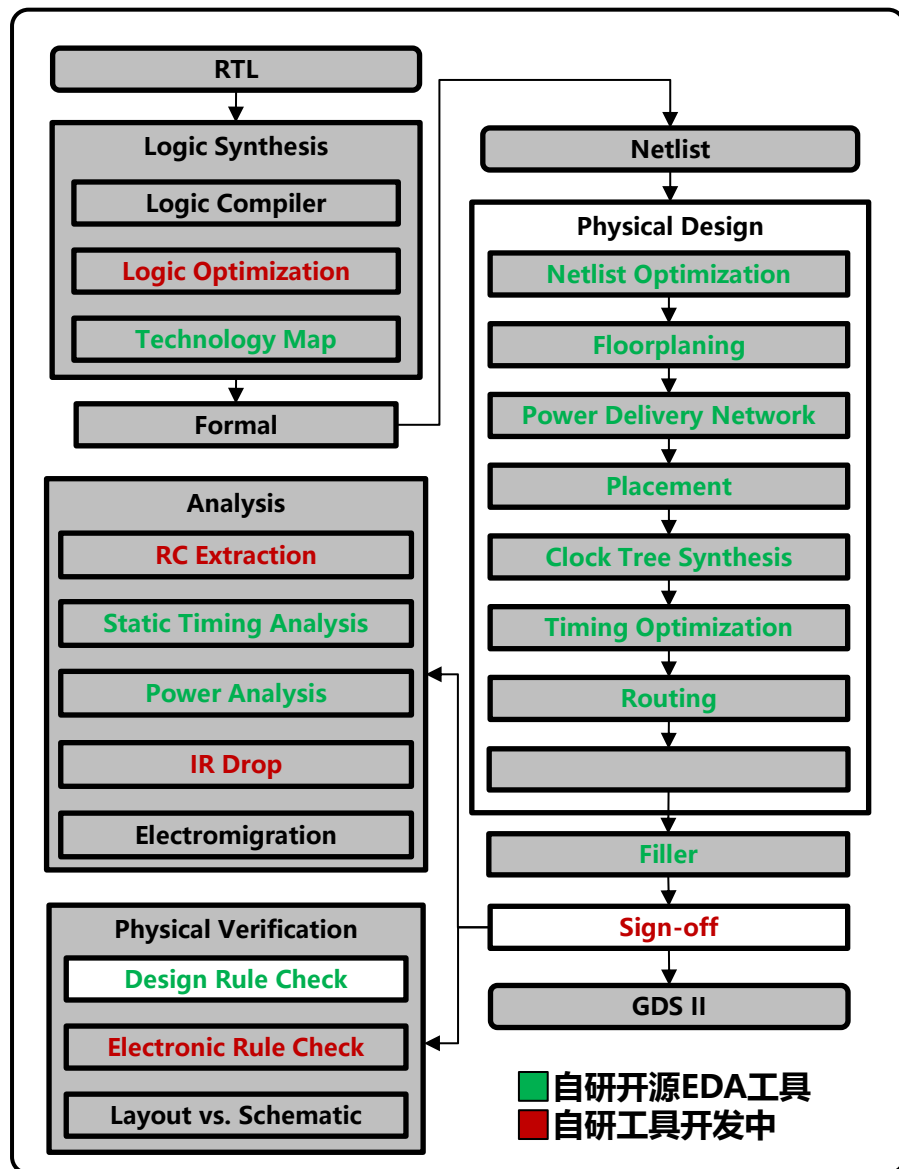
iEDA-iSTA 静态时序分析报告

Generate the report at 2023-05-06T09:54:06					
Report : Averaged Power					
Power Group	Internal Power	Switch Power	Leakage Power	Total Power	(%)
combinational	1.064e-07	5.063e-09	3.079e-08	1.422e-07	(27.595%)
sequential	2.862e-07	7.337e-09	7.963e-08	3.732e-07	(72.405%)
Net Switch Power == 1.240e-08 (2.406%)					
Cell Internal Power == 3.926e-07 (76.173%)					
Cell Leakage Power == 1.104e-07 (21.422%)					
Total Power == 5.154e-07					

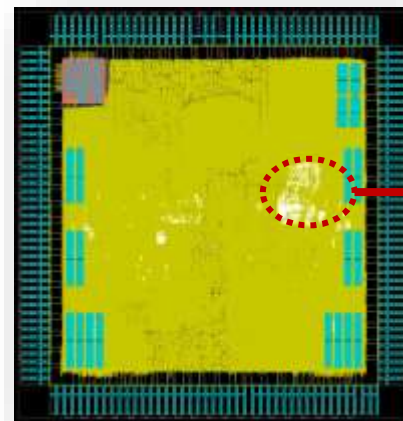
iEDA-iPW 功耗分析报告



# 开源后端设计方案：设计规则检查



Drc Summary	
DRC Type	Number
Cut Different Layer Spacing	433141
Cut EOL Spacing	197803
Cut Enclosure	152168
Cut EnclosureEdge	0
Cut Spacing	358281
Metal Corner Filling Spacing	10443
Metal EOL Spacing	869415
Metal JogToJog Spacing	0
Metal Notch Spacing	733497
Metal Parallel Run Length Spacing	864355
Metal Short	1745445
MinHole	1260
MinStep	670823
Minimal Area	1248072



iEDA-iDRC 检查报告/可视化

## 使用iEDA完成单元布局 (iEDA-iDRC)

- 目标：检查工艺要求的spacing/area/short等设计规则
- 输入：SoC\_iRT.def (Design Exchange Format)
- 输出：DRC报告

## 四、团队介绍

# CC (Chip Creative) 团队简介

- 构建开源的芯片设计解决方案，覆盖开源IP、开源EDA、人才培养、流片&定制
  - 来自多个单位的课题组联合团队，包含计算所、鹏城、开芯院等

## 开源IP：处理器核和基础IP



**处理器核**  
登山计划  
香山裁剪  
复刻雁栖湖



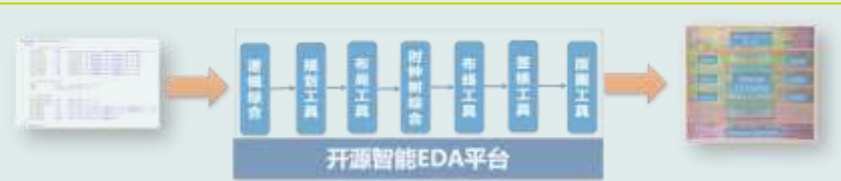
**接口IP**  
DDR控制器  
SDRAM、  
以太网等等

注：香山项目由香山团队主导研发和维护，CC团队深度参与

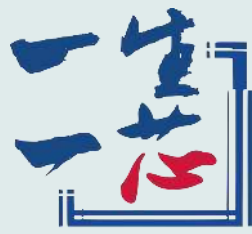
## 开源EDA：打通芯片从RTL到GSDII的设计全流程



仿真验证、逻辑综合、  
物理设计、签核分析  
流片三次  
支持110和28nm工艺

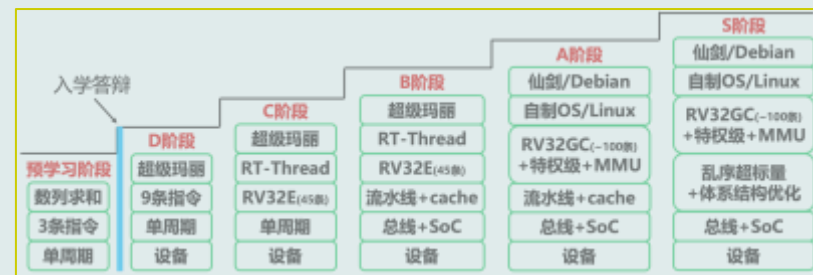


## 人才培养



累计报名**8300**余人，  
覆盖全球**680**余所高校

软硬件协同设计能力  
芯片全链条设计能力

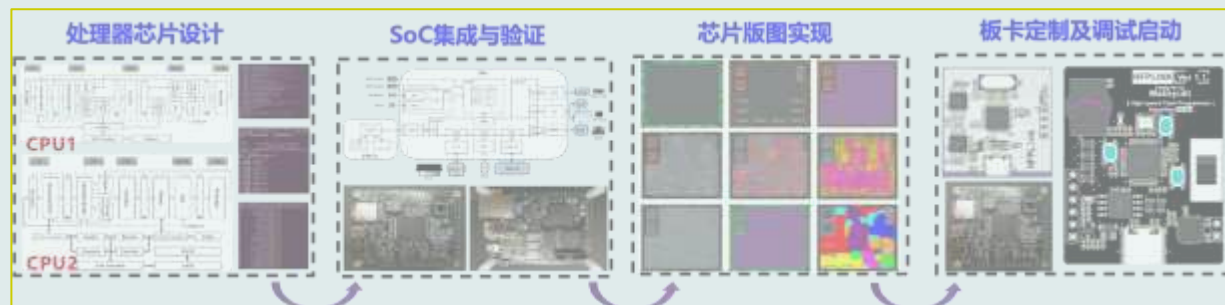


## 流片&定制

使用开源EDA和开源IP的解决方案 **开源掌机**

**SNOW (Shuttle NO Wait)**  
为高校和社区提供低成本流片班车和设计服务

支持近10个课题组/开源项目验证IP或定制芯片



# CC团队愿景：“开源”重构芯片设计

- CC团队秉承开源开放的理念，期望联合国内外认可开源芯片理念的团队，共同构建完全开源的芯片设计解决方案及其技术生态体系，用“开源”理念革新芯片设计方法学，实现“使用开源EDA工具，基于开源IP设计开源芯片”的终极愿景

- **第一步：开源SoC**——用3-5年为社区提供经过流片验证的高质量RISC-V开源核、开源SoC设计
  - RISC-V处理器核IP、外围IP等
- **第二步：用开源工具链构建开源SoC**——用5-7年逐步构建一套基于开源**EDA**工具链、开源**IP**、开源**工艺库**的开源SoC芯片设计流程
  - 将商业版工具、IP逐渐替换为开源版
  - **本科生使用开源工具开发开源芯片，带着自己芯片毕业**
- **第三步：用开源工具链自动化构建开源硬件**——用10-15年开发更智能、更自动化的开源工具，提高设计验证效率
  - 形成开源芯片设计生态，降低芯片开发门槛



使用开源**EDA**、开源**IP**和开源系统软件，设计芯片并流片验证，构建可以使用的原型系统

**谢谢  
请批评指正**