



# “一生一芯”计划

## 从零开始设计自己的RISC-V处理器芯片

余子濠

2025.07



中国科学院计算技术研究所

INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES



中国科学院大学

University of Chinese Academy of Sciences

# 一、背景

# “一生一芯”计划总体介绍

基于开源新赛道的一种贯通课程的实践型开放式大规模人才培养计划

## 面向所有 芯片设计爱好者

- 不限学校
- 不限专业
- 不限年级
- 不限基础

\*支持在校生免费流片



吸引  
提升  
培养

软硬件协同

前后端全链条



熟悉

深入

了解

CS

EE



打破教育资源  
不平衡的壁垒

突破传统课程的边界,  
融合EE和CS的全栈人才培养

培养后进入开源社区/  
企业, 攻关卡脖子领域

# “一生一芯”的组成

## 教学方案

计算机系统栈  
学习阶段  
验证  
处理器设计流程  
软硬件协同  
优化中的权衡

## 技术支撑

应用程序  
RTL仿真器  
EDA  
ISA模拟器  
SoC  
IP  
PCB  
DiffTest  
形式化验证  
运行时环境

## 组织运营

报名参与  
助教  
在线学习  
入学答辩  
调试考核  
周会  
学习记录

## 社区合作

CBAS认证机制  
高校合作路线  
开源项目  
海外学生  
企业实习  
多语言支持

## 二、**教学方案**

# 计算机系统能力培养的重点目标

- 理解计算机系统各抽象层**及其关联**
  - 是从事计算机系统设计、调试、优化等相关工作的基础
  - 何时能关联很大程度上决定学生何时掌握系统能力

- 已有教学方案

- 在课程间关联
  - 组成原理、操作系统、编译
- 在课程内关联
  - 计算机系统基础 -> 组成原理、操作系统、编译
  - 根据计算机系统教学改革的结果，效果优于上述方案

- 我们的教学方案——在每节课中关联（粒度更细）

计算机系统抽象层的转换

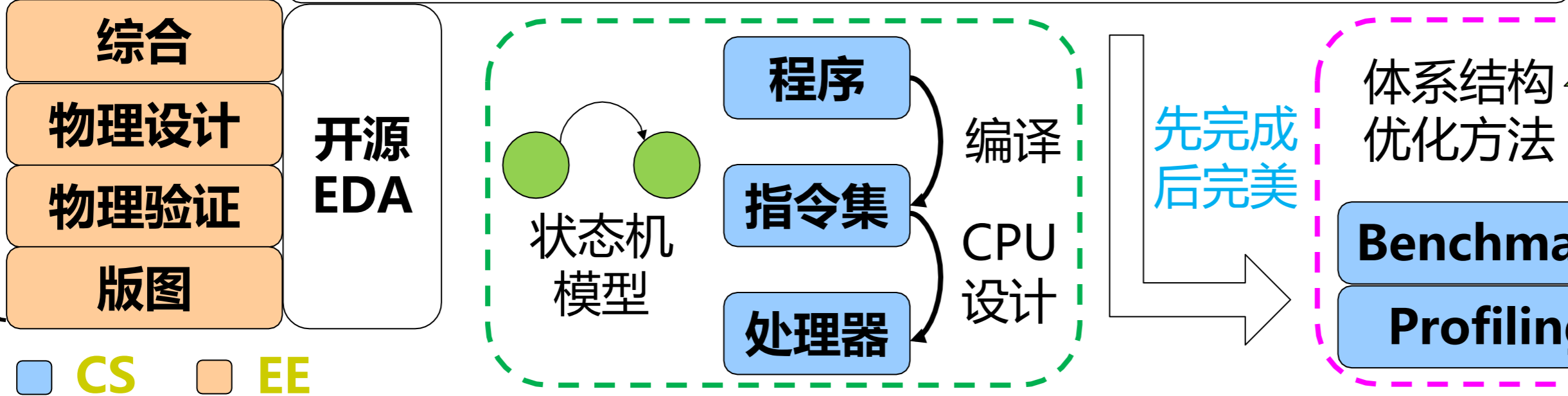


# 知识图谱

软硬件协同

前后端全链条

应用程序	从C语言到二进制、ELF文件和链接	游戏	OS应用
运行时环境	AM裸机运行时环境	IO库	C库
操作系统			自制OS
指令集	RISC-V指令集、程序的机器级表示	I/O指令	自陷指令
微结构	模拟器、单周期	总线、设备	异常机制
电路	SoC计算机系统: UART、SPI控制器、Flash颗粒.....		



**解决工程问题的基础设施:** lint工具、sanitizer、printf、trace、gdb、波形、profiler.....  
 使用正确的模式写出好代码 (不言自明、不言自证)、测试、断言

**独立解决问题的能力:**

- 科学提问
- 看手册、读代码、搜索互联网
- 学会使用/改进/制造工具

缓存 Cache设计

并行 指令流水线

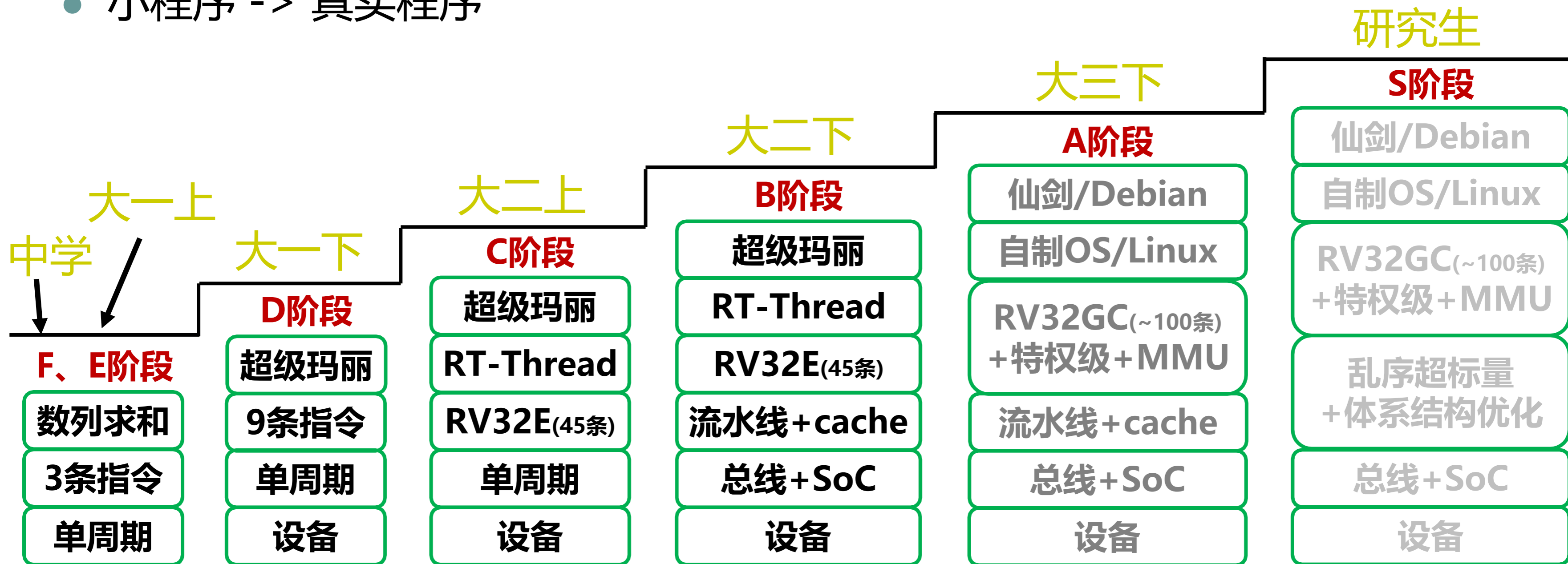
预测 分支预测

专用 乘/除法器

CS EE

# 阶段划分

- 高标准、低门槛的教学方案——覆盖从中学到研究生的知识
  - 小而简单的系统 -> 大而复杂的系统
  - 小程序 -> 真实程序



# 软硬件协同的学习安排

C = C language(软件)  
 R = RISC-V(ISA)  
 P = Processor(硬件)  
 T = Tools(基础设施)

RV32E  
 系统

简介

简单系统

阶段	序号	任务	讲义	课件	视频	C	R	P	T
预学习阶段	1	如何科学地提问	📄	📄	🎬				
	2	Linux系统安装和基本使用	📄	📄	🎬				✓
	3	计算机系统的状态机模型	-	📄	🎬	✓	✓	✓	
	4	复习C语言	📄	📄	🎬	✓			✓
	5	程序的执行和模拟器	-	📄	🎬	✓	✓		
	6	搭建verilator仿真环境	📄	-	-	✓		✓	
	7	数字电路基础实验	📄	📄	🎬			✓	
	8	完成PA1	📄	📄	🎬	✓			✓

■申请入学答辩

SoC & 优化

基础阶段	9	支持RV32IM的NEMU	📄	📄	🎬	✓	✓	✓	
	10	程序的机器级表示(上)	-	📄	🎬	✓	✓		
	11	程序的机器级表示(下)	-	📄	🎬	✓	✓		
	12	用RTL实现最简单的处理器	📄	-	-				✓
	13	AM运行时环境	📄	📄	🎬	✓			
	14	工具和基础设施	-	📄	🎬				✓
	15	支持RV32E的单周期NPC	📄	📄	🎬				✓
	16	ELF文件和链接	-	📄	🎬	✓	✓		
	17	设备和输入输出	📄	📄	🎬	✓	✓	✓	
	18	调试技巧	-	📄	🎬				✓
	19	异常处理和RT-Thread	📄	📄	🎬	✓	✓	✓	
	20	总线	📄						✓
	21	SoC计算机系统	📄						✓
	22	性能和简易缓存	📄						✓
	23	流水线	📄						✓

■达成B阶段流片指标

A阶段: M扩展, S模式 & U模式, A 扩展, xv6, Linux, 微结构优化

# 高标准：培养计算机软硬件协同系统能力(B阶段)

- 理解全系统架构
  - 程序，操作系统，ISA，处理器，电路
- 理解系统的工作原理
  - 游戏如何在系统中运行？

超级玛丽

RT-Thread

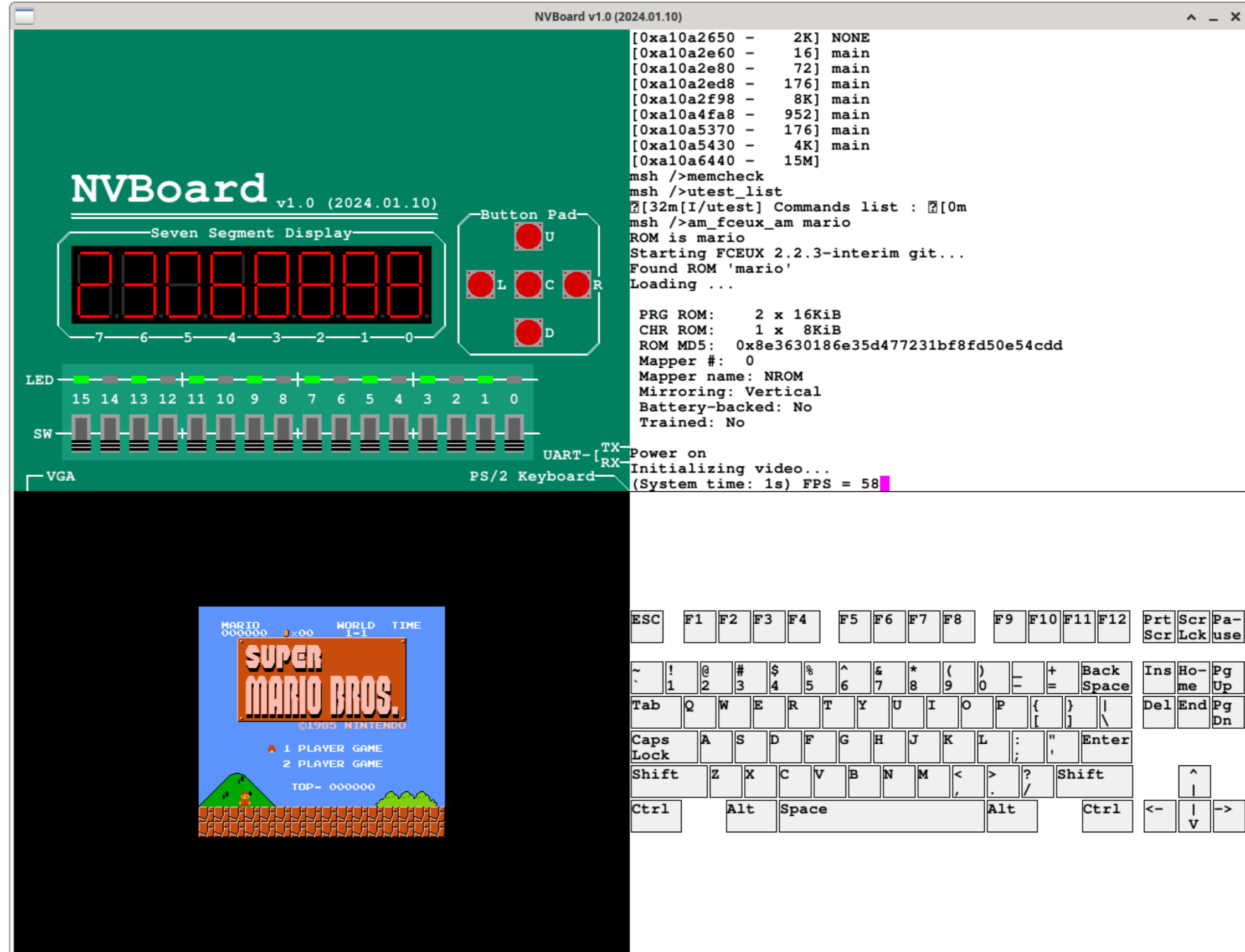
AM裸机运行时

RISC-V指令集

NPC处理器

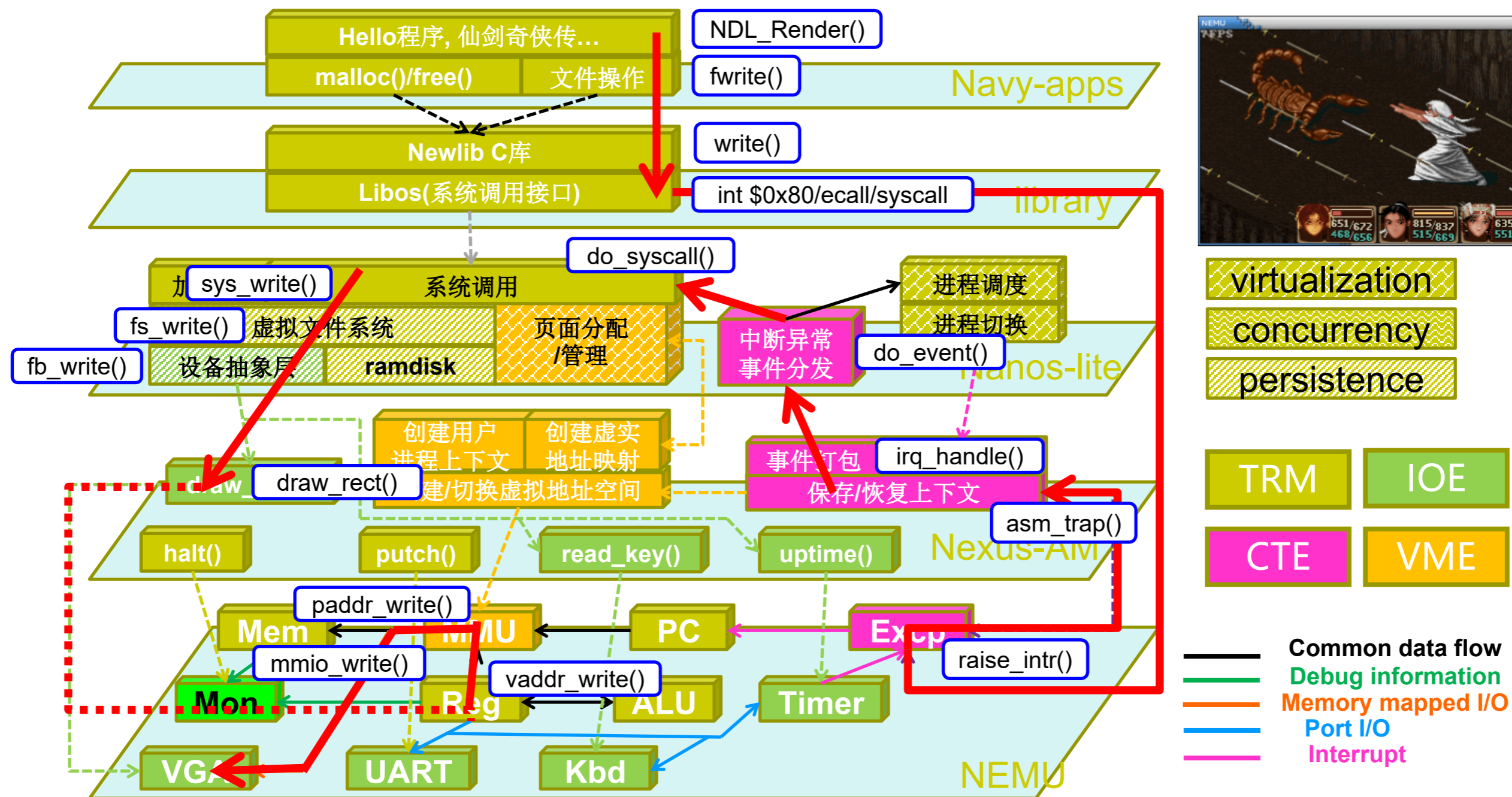
ysyxSoC

NVBoard



# 案例——游戏如何更新屏幕

- 学生亲手构建计算机系统
  - 理解 “程序如何在自己设计的处理器上运行”



# 案例——锻炼全栈调试能力

- 在课程中，解决问题需要**全系统栈调试能力**
  - 真实环境下，问题现象往往在应用层暴露，但原因在其他层次

印象最深刻的bug是npc启动rt-thread时的bug。这是一个软硬件协同的bug，分散地藏在了硬件代码和AM代码中。

起初编译运行，发现rt-thread启动后没有打印字符。观察了ftrace的输出，发现rt-thread似乎在空转。于是尝试运行yield-os，直接报出检查失败。查看此处检查，发现是检查切换后的上下文是否为空。遂决定先为npc添加etrace支持。（准备放在CSR模块中，写入mcause和mepc时记录）添加后运行rt-thread，结果之触发少数几次异常便又开始空转。

再次尝试运行yield-os，却不知道bug处在软件还是硬件中。决定先为npc添加新的diffest，使用diffest来排查硬件问题。开启diffest运行yield-os，未报出寄存器值不同。于是初步认为是软件的问题，明天再进一步探究。第二天，在yield-os中添加一系列打印代码，观察打印结果，发现PCB被不知道什么时候改掉了。于是启动npc监视点功能后，观测两个PCB的地址何时被写入。通过监视点发现，PCB的值确实被提前改写了，需要查看相关pc对应的指令。查看相关指令后发现，有一条sw指令将ra存入了PCB的位置。而且ra是0。且另一个PCB的首地址数据也被修改了。

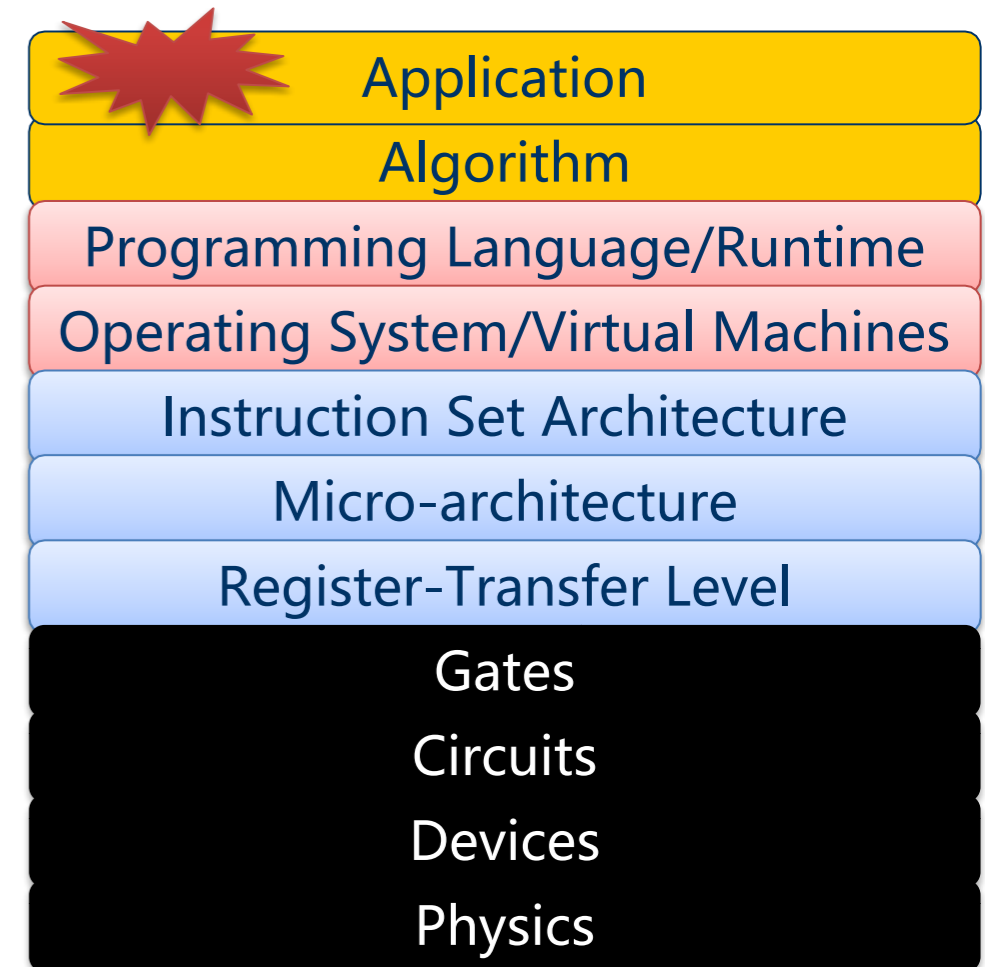
则那个0被写入的过程应该是：触发了第一次自陷，然后在第一次调用schedule函数后，从\_\_am\_irq\_handle返回。但是返回时恢复上下文，给ra加载了一个0进去。这个0在从\_\_am\_asm\_trap返回后，执行f函数，在保存ra寄存器时将0存到了该PCB的原本指向上下文指针所指的位置。

此处ra是我在创建上下文结构体时初始化的，初始化的值就是0。但是此时的sp为何会是个这个值？

旧值为0x80008a30，恢复上下文后加上CONTEXT\_SIZE（代码中为80）后变为0x80008a80。但是这比实际的上下文大小大了4。

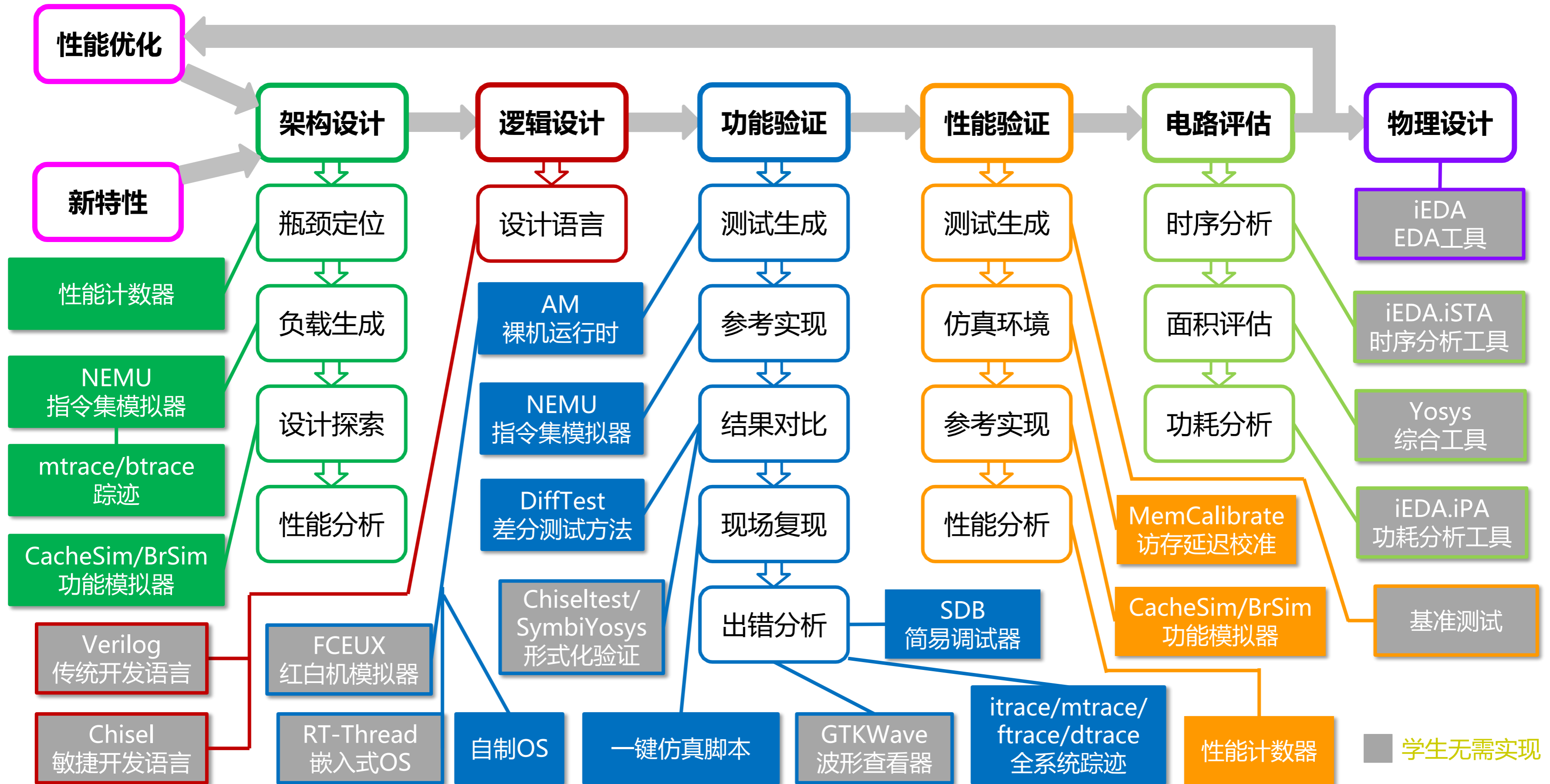
问题就出在这里！两个PCB在内存中是相邻摆放，偏移之后，值就会偏到另一个PCB中去。将上下文大小改为正确的值之后，yield-os成功运行。但是rt-thread还是无法运行。打开diffest之后报错。此时尝试剑走偏锋，将npc的镜像文件给riscv32e-nemu跑，结果跑通了，成功启动rt-thread。于是确定此时软件已经被修复，而硬件中应该有问题。这是第二天做的事情。

第三天去cpu中查看波形，发现一条lhu指令的地址计算出了问题。顺着信号查找，发现在译码时立即数种类识别出了问题。修改后重新编译运行，成功启动rt-thread。



学生的调试记录  
(一个代表性案例)

# 高标准：培养处理器芯片前后端全链条能力(B阶段)



# 不仅仅是RTL

## ● 先进的验证方法

- DiffTest – 在线指令行为测试
- 形式化验证 – 要么证明正确性, 要么给出反例
  - 带cache (DUT) v.s. 不带cache (REF)
  - 流水线 (DUT) v.s. 单周期 (REF)

## ● 性能评测和优化

- IPC – 用模拟器评估 (CacheSim, BrSim) + 通过性能计数器分析
- 频率 - Yosys (综合) + iEDA.iNO (网表优化) + iEDA.iSTA (时序分析)

## ● 功耗评估和优化 [WIP]

- iEDA.iPA (功耗分析)

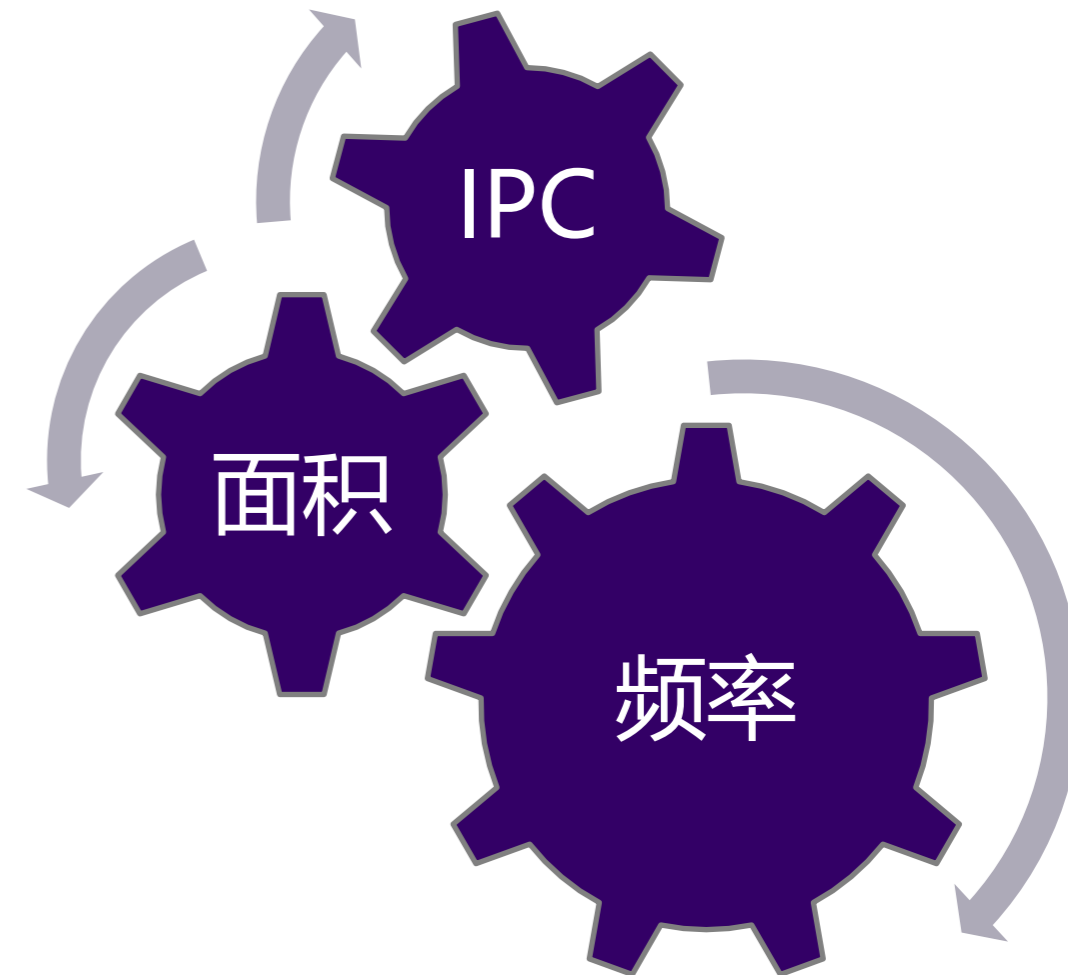
## ● 物理设计全流程

- iEDA (开源EDA工具)

# 案例——优化中的权衡

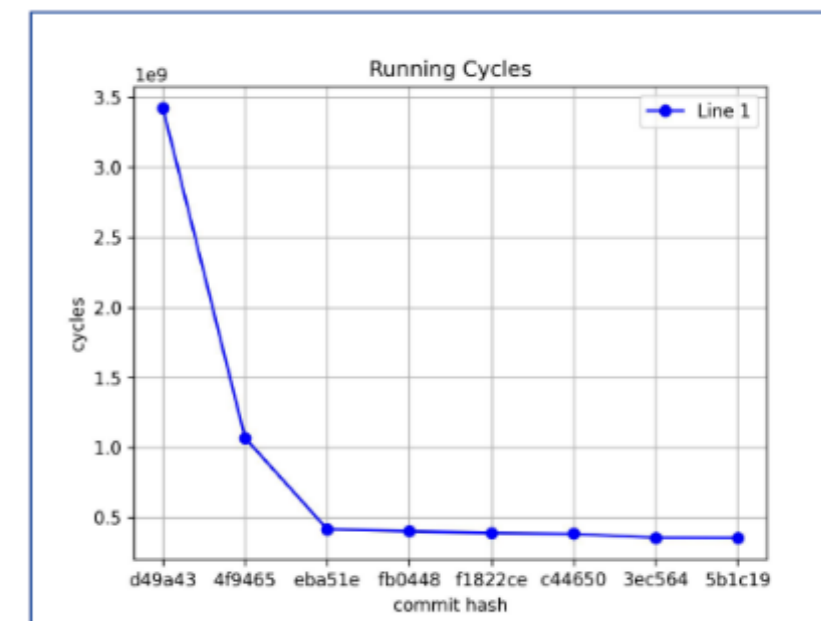
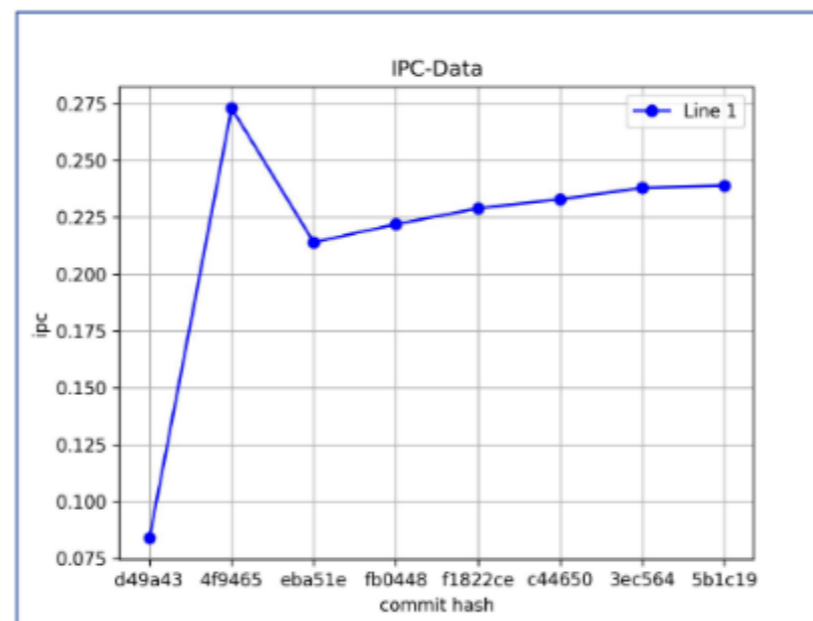
## ● 一生一芯中的开销和决策

IPC	综合频率	综合面积
-----	------	------



性能记录

	A	B	C	D	E	F	G	H	I
1	性能分析								
2	commit	bench	架构	说明	仿真周期数	仿真指令数	IPC	综合频率	综合面积
3	bf70feb4f77732ace4c720ee5a5f5a29fc8f5dae	dummy	Eryth-Multi		15316338	46509	0.00304	370.462MHz	26580.32
4	ad64212dbf61095221f92e2ad6500c518e70f57b		Eryth-Multi	STA时禁用commit模块				371.154MHz	26271.49
5	bb2c03f1d37014b86bb8b54b975686f9422d90db		Eryth-Qingyuan	单发射五级流水。静态预测。无CSR支持				399.381MHz	30805.19
6	489c97728a4526c24fb17fc78c8ab45239a0fbd0	dummy	Eryth-Qingyuan		18390887	39636	0.00216	397.640MHz	30796.42
7	19184c7ad51acad412c30d04546f435a2a1deaf9	dummy	Eryth-Qingyuan	BPU动态预测	16396636	46266	0.00282	330MHz	37961.39
8	6edea36d4112e7bb12f3c3eec62ef2ac86db3dd6	dummy	Eryth-Qingyuan	使用AXI的SDRAM	16389426	141429	0.00863	330.234MHz	37878.13
9	f33d84ae7fecf49c3005cb48f841abbbae338c7d	dhrystone	Eryth-Qingyuan	循环次数: 10000	141452948	10001872	0.07071	330.011MHz	39061.57
10	7a2afe007fa505b6f31f066cdd719a2bde4d0dca	dhrystone	Eryth-Qingyuan	循环次数: 10000	141713484	10001906	0.07058	330.011MHz	39061.57



学生作业——评估一项优化技术对频率、面积、IPC的影响

# 工程思维和方法的训练

- Linux命令行 更多内容: 第六期 “一生一芯” 第2课——[《工具是第一生产力——Linux入门教程》](#)
  - 常用命令, strace, PATH环境变量, shell, 命令别名, 正则表达式, 任务管理, 输入输出重定向, 管道, 脚本
  - 工具和C程序
- 编码风格 更多内容: 第六期 “一生一芯” 第5课——[《程序的执行和模拟器》](#)
  - 编写可读可维护的代码
    - 不言自明 - 仅看代码就能明白是做什么的(specification)
    - 不言自证 - 仅看代码就能验证实现是对的(verification)
  - 使用正确的编程模式写出好代码
    - 防御性编程/减少代码中隐含依赖/编写可复用代码/使用合适语言特性

# 工程思维和方法的训练

## ● 读代码的方法

更多内容: 第六期 “一生一芯” 第7课——[《NEMU代码导读》](#)

- 看源码(source): 可以得知每一处静态细节, 但较繁琐
- 看踪迹(trace): 容易了解运行动态行为, 但不全面
- 先从容易理解的方式下手(trace), 再结合程序的动态行为理解静态代码

## ● 工具和基础设施

更多内容: 第六期 “一生一芯” 第12课——[《工具和基础设施》](#)

- 代码开发——make, tmux, ccache, inotifywait, trace, gdb
- 处理器的差分测试方法 (DiffTest)

## ● 调试方法和技巧

更多内容: 第六期 “一生一芯” 第16课——[《调试技巧选讲》](#)

- 调试理论: fault -> error -> failure

# 开放理念：学习资源均开放

## 第五期"一生一芯"课程主页

- 课时: 每周六19:00~21:00
  - 直播链接 | 录播链接
- 答疑: 每周日19:00~20:00 (通过预习学习答疑后由助教通知)
- 报名流程 | 报名常见问题

## 课件和讲义

0. C = C语言(程序/模拟器/系统软件) | R = RISC-V指令集 | P = 处理器设计 | T = 工具

### 预学习阶段

- 一生一芯概述 | 如何科学地提问
- 工具是第一生产力——Linux入门教程 | Linux系统安装和基本使用
- 计算机系统的状态机模型 C R P | 复习C语言
- 从C语言到二进制的程序 C T
- 程序的执行和模拟器 C R P | 搭建verilator仿真环境 | 数字电路基础实验
- NEMU代码导读 C T | 完成PA1

### B阶段

- RISC-V指令集 C R P | 支持RV64IM的NEMU
- 程序的机器级表示 C R
- RISC-V单周期处理器设计 P | 用RTL实现最简单的处理器
- Abstract Machine裸机运行时环境 C | 运行时环境和基础设施
- ELF文件和链接 C R
- 工具和基础设施 T | 支持RV64IM的单周期NPC
- 设备和输入输出 C R P | 设备和输入输出
- 调试技巧进阶 T

### A阶段

- 异常处理 C R P | 简单的异常处理机制
- 计算机系统软件栈 C | 用户程序与系统调用 | 精彩纷呈的用户程序

### 计算机系统的状态机模型

余子濠

中国科学院 计算技术研究所

计算机系统与处理器 芯片课程虚拟教研室

- 课程主页
- 讲义42万字
- 课件1000+页, 10万字
- 教学视频时长40+小时

### Linux系统安装和基本使用

安装一个Linux操作系统

获取"一生一芯"框架代码

```
git clone -b ysyx2024 git@github.com:2050214/ysyx-workbench.git
```

! 安装系统是独立解决问题的最简单的训练

树立正确的价值观 接受最大程度的训练

### B站账号: 一生一芯-视频号

### 解密黑科技 - 现代方法

使用工具查看宏展开结果

回顾: 使用gcc的-E参数可以输出预处理结果

- 但直接编译会报错: 找不到头文件

解决方案: 在Makefile文件的编译规则中添加命令

```
# Compilation patterns
$(OBJ) DIR/%.o: %.c
@echo + CC $@
@mkdir -p $(dir $@)
$(CC) $(CFLAGS) -c -o $@ $<
$(CC) $(CFLAGS) -E -o $@.i $<
$(call call_fixdep, $@,om,d, $@)
```

展开的结果不好阅读

- 使用代码格式化工具

### 课件讲解+代码讲解+代码演示

# “一生一芯” 资源

- “一生一芯” 主页 - [ysyx.org](https://ysyx.org)
- 课程主页 - [ysyx.org/docs/](https://ysyx.org/docs/)
  - 包含报名链接, 常见问题, 课件, 讲义
  - 以及直播链接, 录播链接等

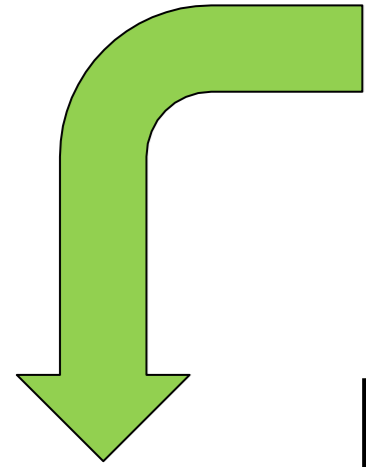


- B站账号 - 一生一芯-视频号
  - B站主页
    - <https://space.bilibili.com/2107852263>
  - 第五期录播集合
    - <https://space.bilibili.com/2107852263/channel/collectiondetail?sid=690279>
  - 第四期学生心得分享集合
    - <https://space.bilibili.com/2107852263/channel/collectiondetail?sid=1173655>

# 环境, 开发语言和工具依赖

- 软件配置 - Linux环境
  - 软件开发(模拟器, 系统软件, 应用程序)
    - C语言, 很少量的C++(类的基本使用) - gcc编译器
  - 硬件开发
    - Verilog或Chisel - 开源verilator仿真器 + 开源gtkwave波形查看器
    - 后端物理设计
      - 由工程师或线下实习生通过商业EDA完成
      - 正在进行: 将开源EDA工具融入教学流程
- 硬件配置 - 笔记本电脑即可, 无需FPGA
  - FPGA与ASIC流片原理不同, 使用ASIC EDA工具评估的结果更准确
  - FPGA主要用于仿真加速, 但学生的设计简单, FPGA流程时间反而长
    - 综合1小时, 运行1分钟 v.s. 仿真10分钟

# 低门槛：北京一零一中学初中生尝试学习“一生一芯”



F、E阶段

数列求和

3条指令

单周期

× 北京一零一中学初二项目制学习 ...

- 理解程序如何在处理器上运行的基本原理
- 掌握处理器芯片前端设计的步骤
- 了解芯片后端物理设计的流程

## 学习计划

- 覆盖知识: C语言程序设计, 系统编程, 指令集和汇编, 处理器组成与设计, 数字集成电路
- 学习计分为4个阶段, 学习内容陆续发布:
  - 预学习阶段 - 数字电路设计基础: 使用Logisim设计组合逻辑电路和时序逻辑电路
  - 基础(B)阶段 - 简单处理器设计: 使用Logisim设计9条指令的处理器, 运行超级玛丽
  - 进阶(A)阶段 - 处理器RTL开发: 使用硬件描述语言设计9条指令的处理器, 运行超级玛丽
  - 专家(S)阶段 - 可流片处理器设计: 设计9条指令, 支持总线的多周期处理器, 生成芯片版图
- 学习内容有一定挑战, 希望大家能享受处理器设计的乐趣, **不要把完成S阶段作为成功的唯一标准, 也不要卷掉其他同学作为学习目标**
- 可点击 [讲义](#) 一列中的图标跳转到相应页面
- 本页面不对外公开, 不知道URL则无法访问
- 后续学习内容将陆续发布

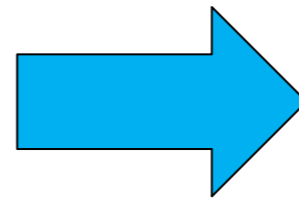
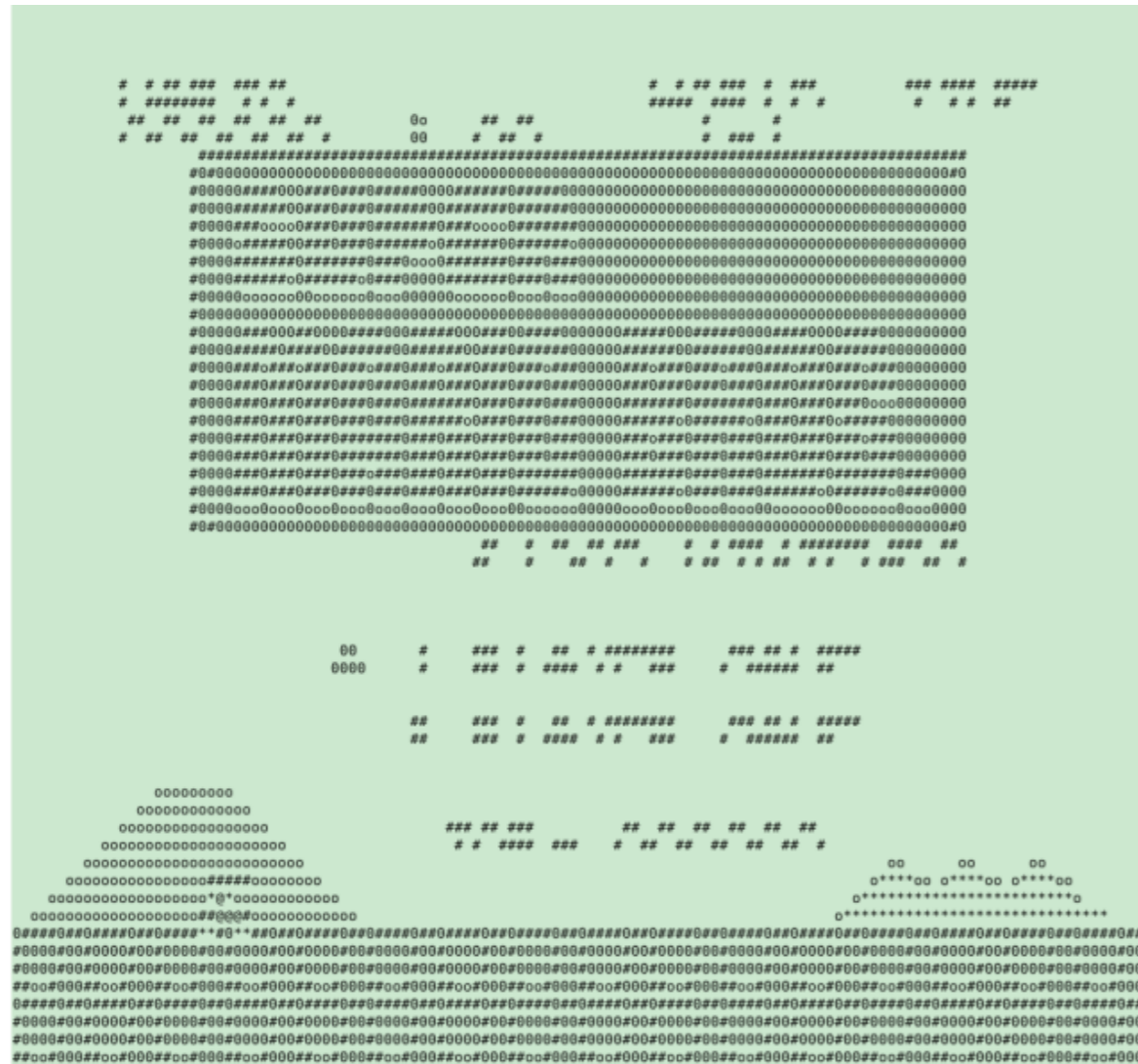
# 三、技术支持

# 应用程序, OS和运行时环境

- 应用程序
  - FCEUX (红白机模拟器), 基准测试, 其他游戏...
- OS
  - RT-Thread
  - **Nanos-Lite** (小型OS, 学生作业)
  - xv6, Linux内核, Debian (或其他发行版)
- 运行时环境
  - **Abstract Machine** (裸机运行时环境)
  - **Navy-apps** (Nanos-Lite上的运行时环境)
  - Linux运行时环境
- **绿色项目**属于南京大学的教学项目

# 在处理器上运行游戏

- 添加VGA控制器



D阶段: 运行超级玛丽 (1980s发行)

# 在处理器上运行游戏



A阶段: 运行仙剑奇侠传  
(1990s发行)



选做任务: 运行CLANNAD  
(2000s发行)

# SoC和设备控制器IP

- SoC - ysyxSoC
  - ysyx = 一生一芯的缩写
  - 基于RocketChip项目中的Diplomacy机制开发, 但未使用RocketChip中的任何RTL模块
  - 可生成Verilog代码
- 设备控制器IP
  - 由团队工程师开发或来自开源社区 - UART, SPI, I2C, PS2, VGA, PSRAM, SDRAM, ...
  - 学生作业 - CLINT, PLIC

# 仿真和验证

- RTL和网表仿真
  - Verilator – 仿真速度快, 但无法检测电路中的X态
    - 平时使用
  - Icarus Verilog – 能检测X态, 但仿真速度慢
    - 仅在提交代码前使用
  - Gtkwave – 波形查看器
- 验证
  - NEMU – 教学用的ISA模拟器, 学生作业
    - 由南京大学提出
  - DiffTest – 指令行为测试框架
    - 启发来源于软件工程领域的差分测试(Differential Testing)
  - chiseltest/SymbiYosys – 形式化验证

# 面向微结构设计的基准程序 (WIP)

- SPEC CPU是商业软件
  - 学生学习需要开源的基准程序

前端密集		计算密集		访存密集	
分支密集	C编译器, gobmk	整数密集	hmmmer, h264	流式访问	stream
ICache压力	RTL仿真器	浮点密集 (可选)	GEMM, Stencil	不规则访问	mcf, 图算法

- 覆盖处理器中的关键部件
  - 访存单元, 整数单元, 浮点单元, 分支预测器, cache等
- 可提供开源处理器的性能参考 (如香山)
  - 可进行性能打榜

# 板卡模拟

- NVBoard

- 虚拟板卡, 由南京大学提出

NVBoard v1.0 (2024.01.10)

Seven Segment Display

Button Pad

LED

SW

VGA

PS/2 Keyboard

```
[0xa10a2650 - 2K] NONE
[0xa10a2e60 - 16] main
[0xa10a2e80 - 72] main
[0xa10a2ed8 - 176] main
[0xa10a2f98 - 8K] main
[0xa10a4fa8 - 952] main
[0xa10a5370 - 176] main
[0xa10a5430 - 4K] main
[0xa10a6440 - 15M]
msh />memcheck
msh />utest_list
[32m[I/utest] Commands list : [0m
msh />am_fceux_am mario
ROM is mario
Starting FCEUX 2.2.3-interim git...
Found ROM 'mario'
Loading ...

PRG ROM: 2 x 16KiB
CHR ROM: 1 x 8KiB
ROM MD5: 0x8e3630186e35d477231bf8fd50e54cdd
Mapper #: 0
Mapper name: NROM
Mirroring: Vertical
Battery-backed: No
Trained: No

Power on
Initializing video...
(System time: 1s) FPS = 58
```

ESC F1 F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12 Prt Scr Pa-  
Scr Lck use

~ ! @ # \$ % ^ & \* ( ) \_ + Back Ins Ho- Pg  
1 2 3 4 5 6 7 8 9 0 = = Space me Up

Tab Q W E R T Y U I O P { } | Del End Pg  
[ ] \ / ] Dn

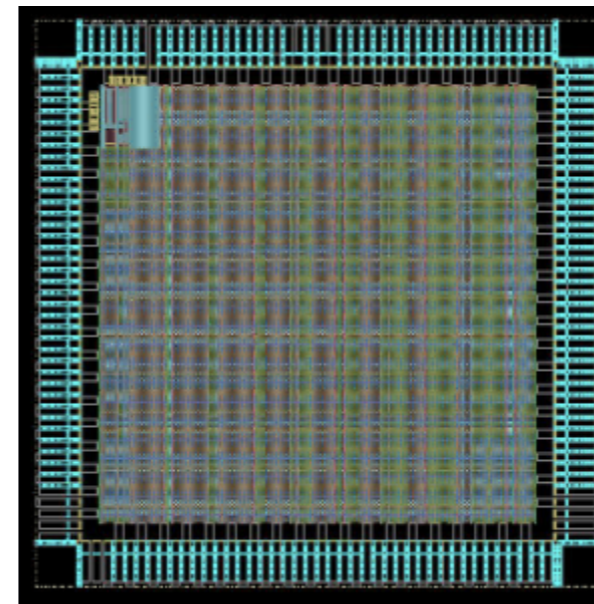
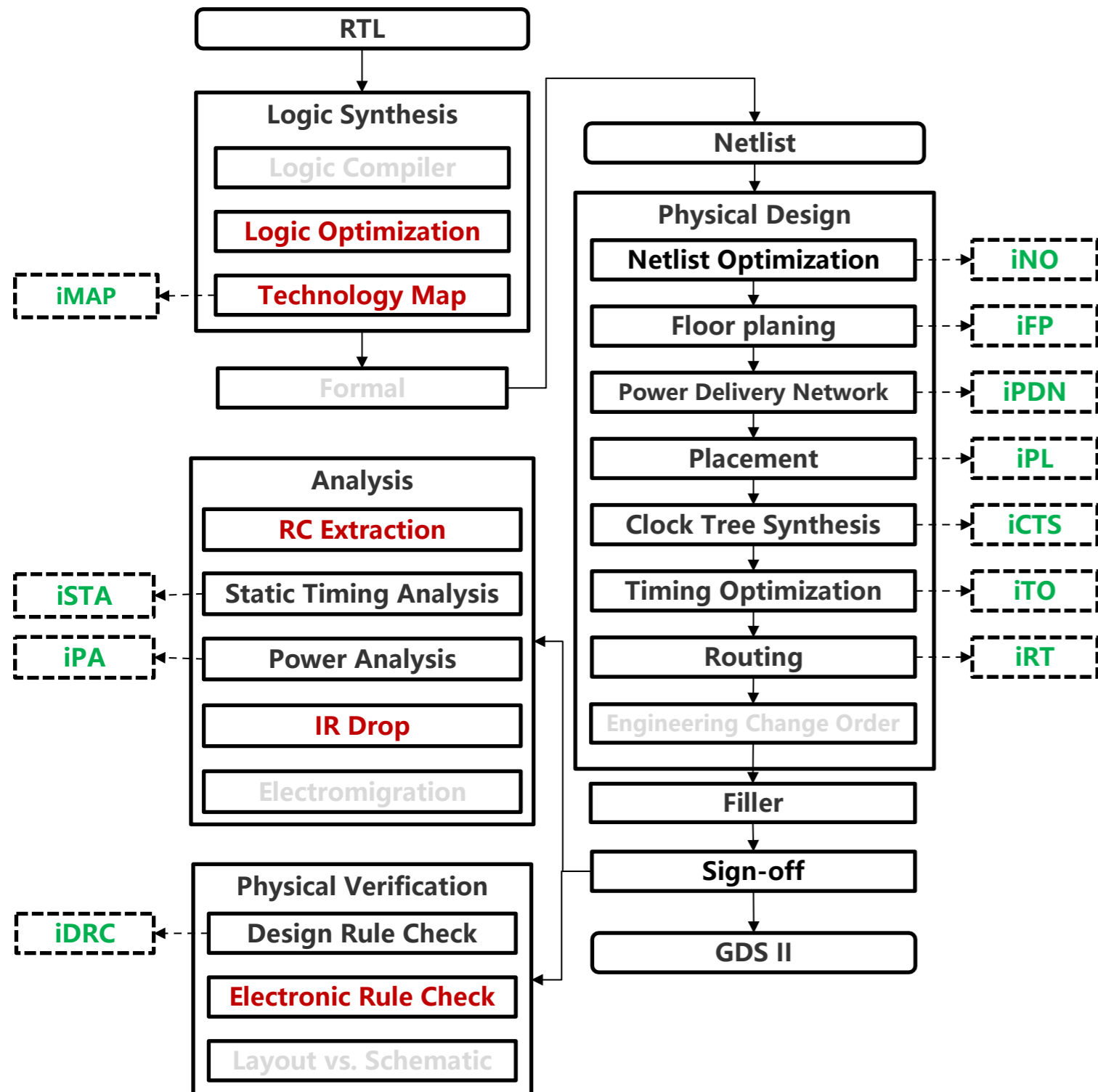
Caps A S D F G H J K L ; " Enter  
Lock ; ' Enter

Shift Z X C V B N M < > ? Shift ^  
/ . / Shift |

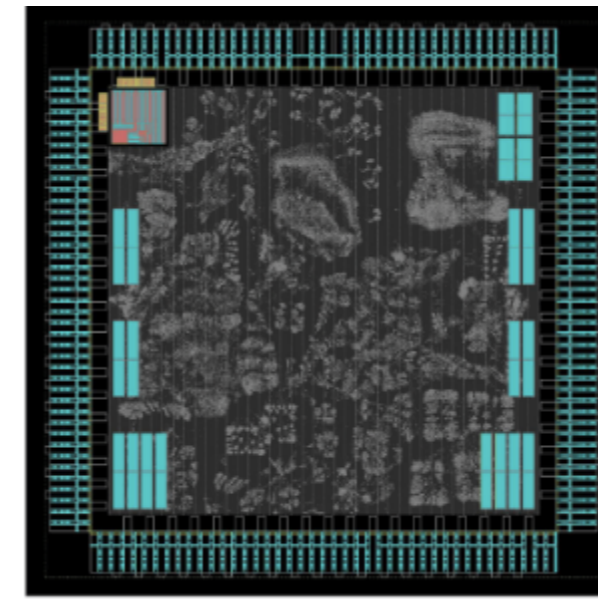
Ctrl Alt Space Alt Ctrl <- | ->  
V

# 综合和物理设计

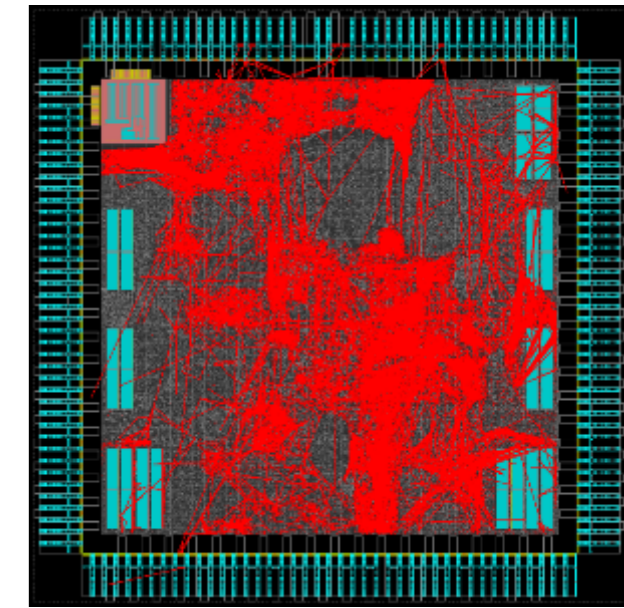
- RTL2GDS 流程 = Yosys, iEDA, Klayout, ...



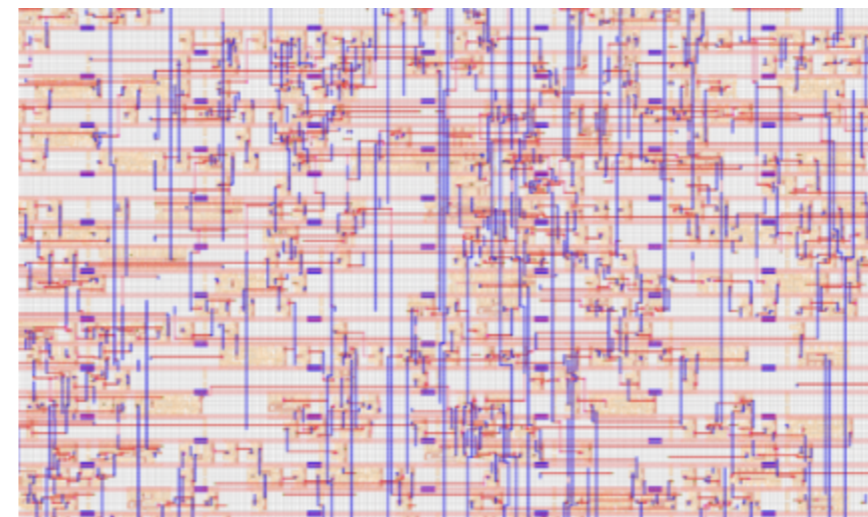
Layout Planning



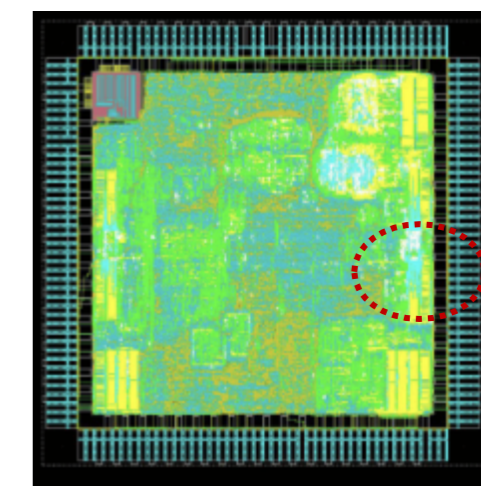
Placement



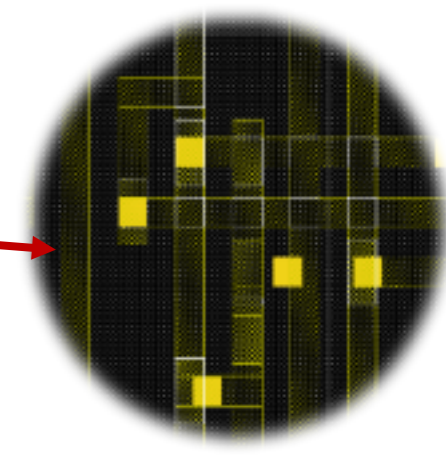
Clock Tree Synthesis (CTS)



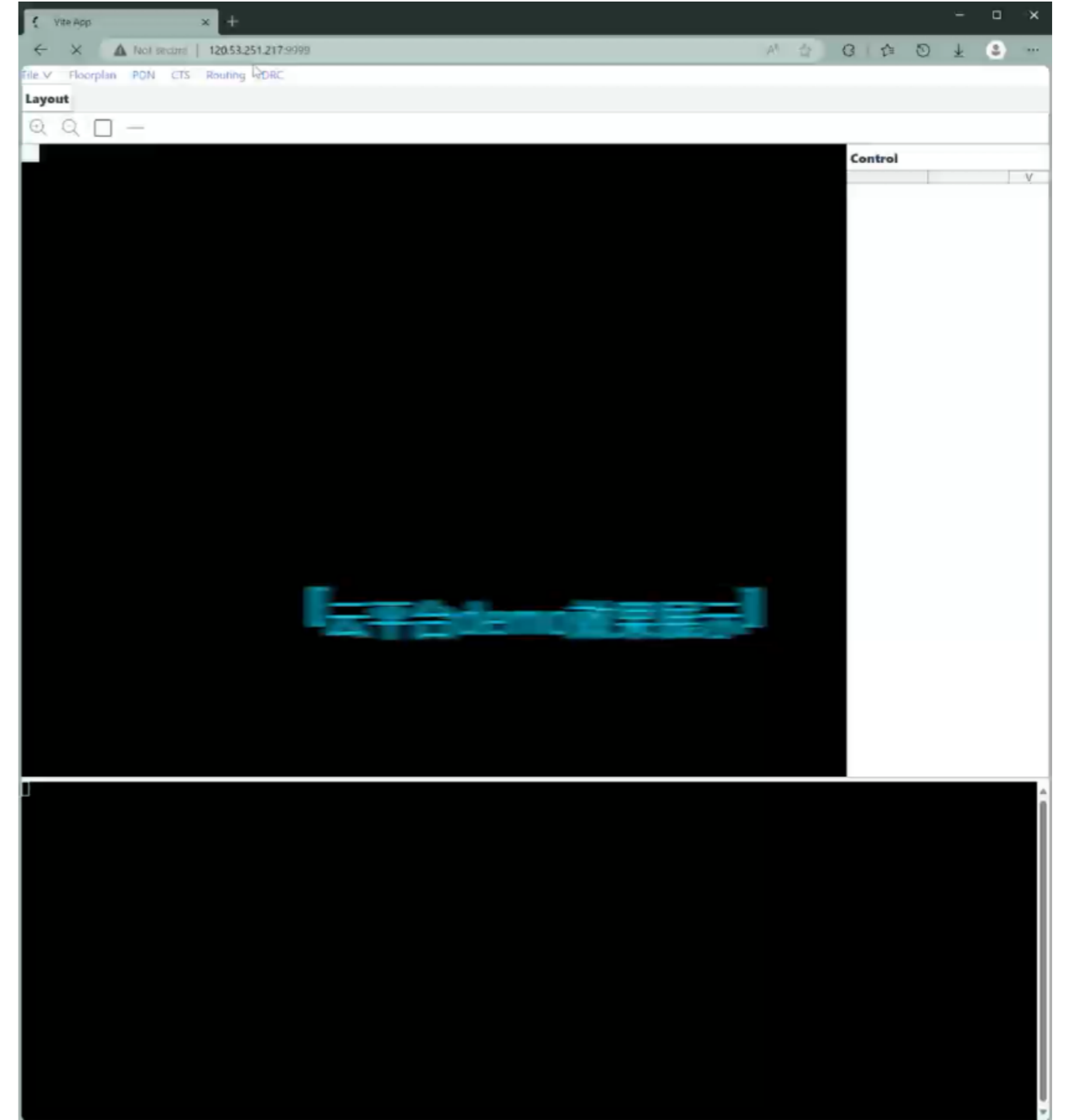
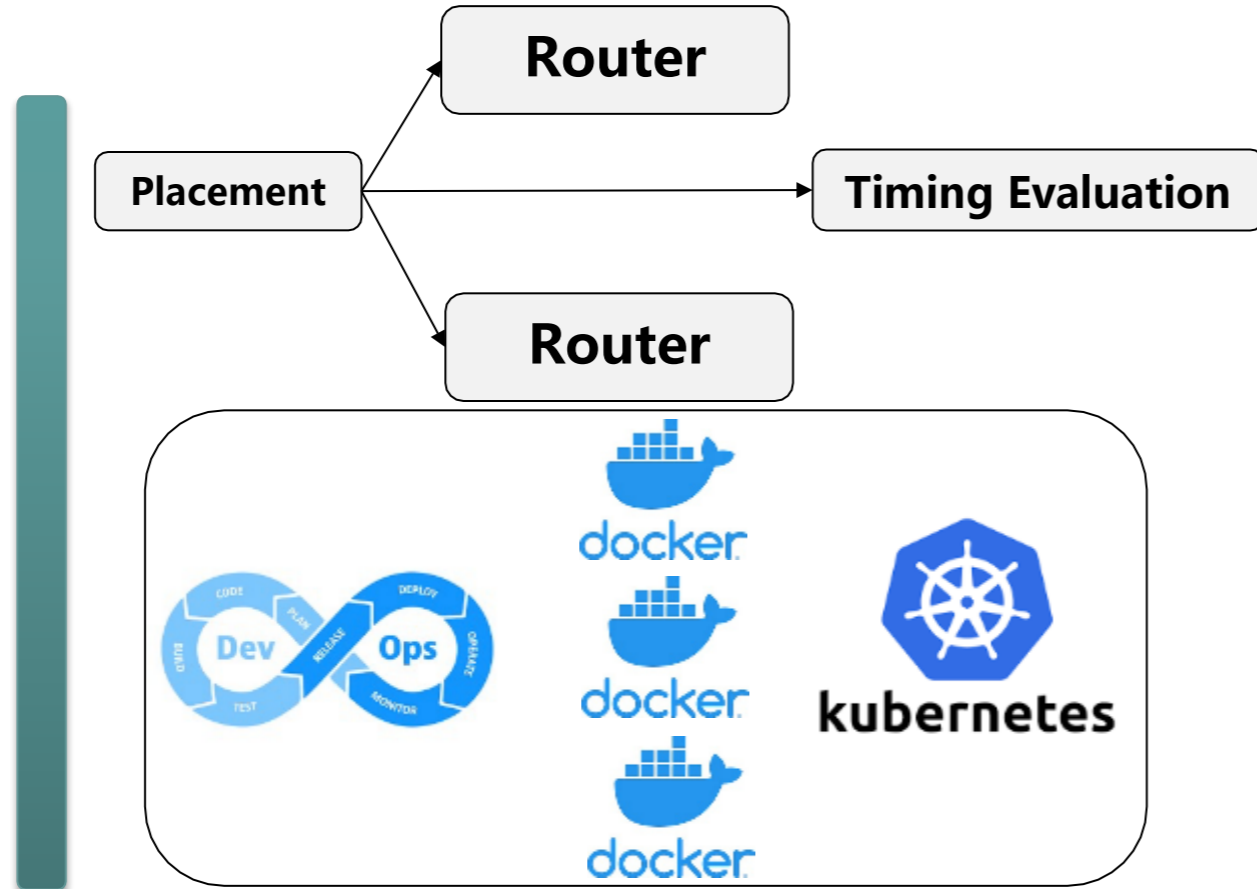
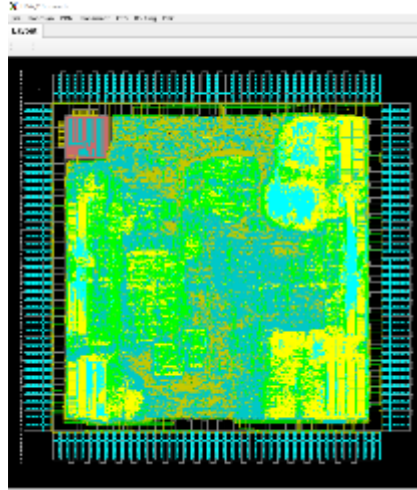
Routing



Layout and Design Rule Check (DRC)

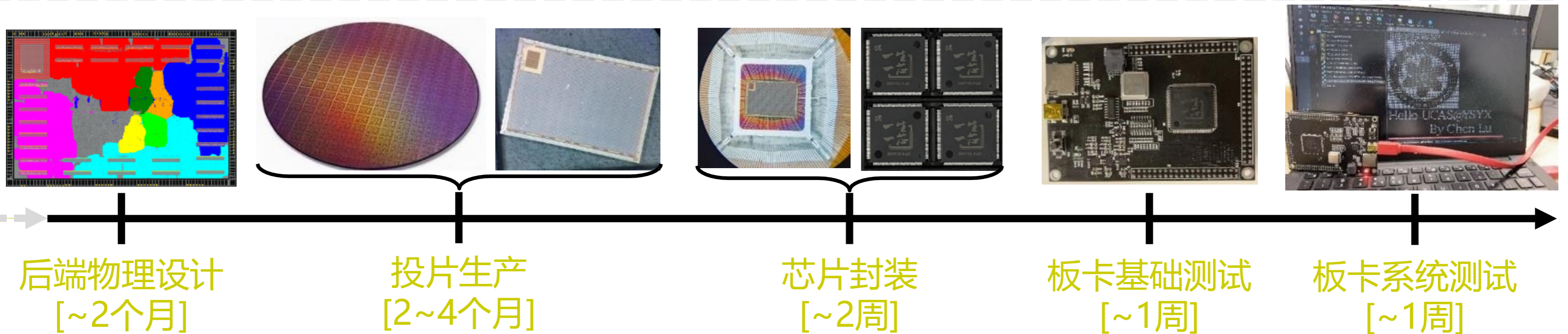
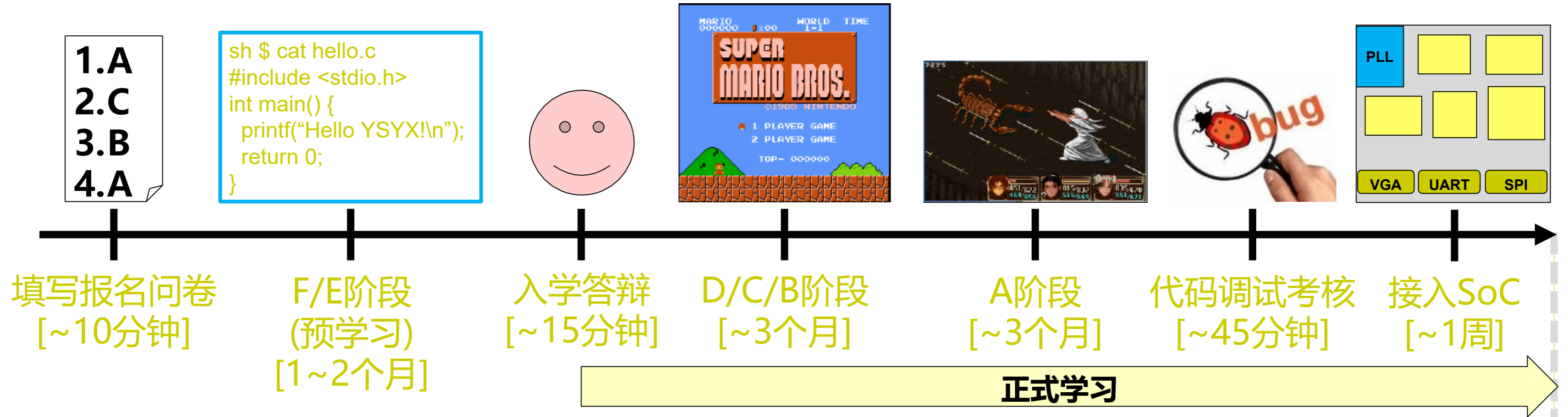


# 基于开源EDA工具的芯片设计云平台

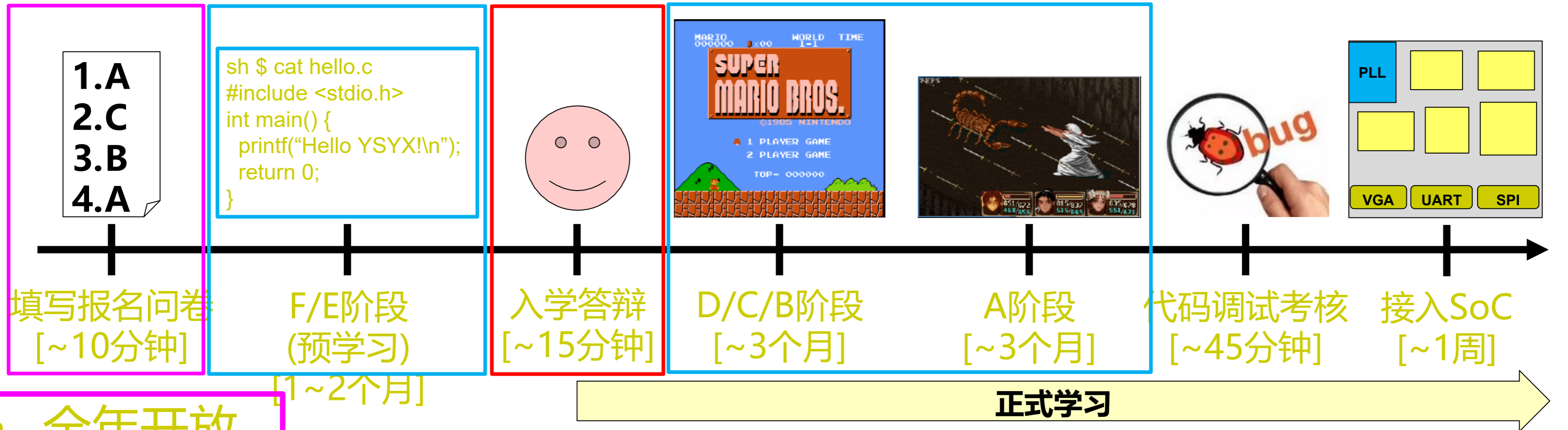


# 四、组织运营

# 学习路线图



# 学生的学习过程



- 全年开放
- 无门槛

- 助教提问, 检查学生学习是否扎实, 若否, 则给出学习建议
- 若答辩不通过, 准备后可再次申请

- 在线学习, 阅读讲义并完成编程练习
- 分微信讨论组, 每组20~30学生+2助教
- 开发任务独立完成, 鼓励组内讨论分享学习方法, 但不鼓励分享代码
- 做学习记录, 参加由助教组织的周会

# 学习无截止日期

- 完成学习的时间因人而异 (从数周到1~2年都有)
  - 学生来自不同高校, 专业, 年级, 基础也各不相同
  - 学得有快有慢/学得有早有晚/有的忙有的闲
- 若学生零基础, 至少需要500小时以上
  - 约等于修读计算机系系统方向所有必修课并完成所有大作业的时间
  - “一生一芯” 的工程量和复杂度也高于一般课程的大作业
- 学生自行安排其学习时间
  - 学完可随时申请入学答辩/考核

# 助教团队

- 从优秀學生中遴選
  - 启发學生思考, 而不是直接提供答案
  - 每次思考对学生来说都是锻炼的机会

## 助教列表 (2023.06)

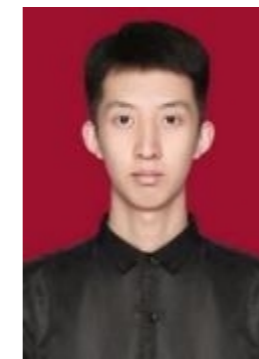
	Name	University	Grade
1	苗金标	中国科学技术大学	硕士二年级
2	段震伟	中国科学技术大学	硕士三年级
3	刘汉章	太原理工大学	本科三年级
4	曹勋	中国科学技术大学	硕士二年级
5	杨海帆	浙江工商大学	本科四年级
6	曹世洋	中国科学技术大学	硕士二年级
7	倪仁涛	东北大学	硕士一年级
8	魏人	兰州大学	本科四年级
9	吴佳宾	青岛大学	本科四年级
10	陈璐	中国科学院大学	直博一年级
11	栗金伦	太原理工大学	本科四年级



苗金标



段震伟



刘汉章



曹勋



杨海帆



曹世洋



倪仁涛



魏人



陈璐



栗金伦



吴佳宾

# 学生的学习记录

学号	姓名	学校	专业	年级	进度记录链接	"提问的智慧"读后感	实验报告	是否两周未更新进度记录	3.2	3.27	4.1	4.10	4.24	5.1	5.0.5	1.55	2.22	5.29	6.5	3.19	7.3	7.10	7.17	7.24	8.7	8.14		
ysyx_22040020	毛英楠	天津大学	计算机科学与技术	大四	<a href="#">ysyx_220020-毛英楠-天津大学-学习记录</a>	<a href="#">ysyx_220020-毛英楠-天津大学-读后感</a>		两周未更新	3.20			2022.4.11	S-关键: PA1	2022.2022	npc-c	2022.2022	2022.7.10	清假							2022.8.7	清假		
ysyx_22040047	吴浩宇	东北大学	计算机科学与技术	大三	<a href="#">ysyx_220047-吴浩宇-东北大学-学习记录</a>	<a href="#">ysyx_220047-吴浩宇-东北大学-读后感</a>			2022.3.20			2022.4.10	清假	PA2-3	npc读	2022.2022	2022.6.1	2022.2022	7.17	清假								
ysyx_22040091	冯浩原	中国科学院大学	计算机科学与技术	大三	<a href="#">ysyx_220091-冯浩原-中国科学院大学-学习记录</a>	<a href="#">ysyx_220091-冯浩原-中国科学院大学-读后感</a>	<a href="#">ysyx_220091-冯浩原-中国科学院大学-实验报告</a>			2022.3.20			2022.4.24	清	PA4.1													
ysyx_22040030	刘刚宇	东北大学秦皇岛分校	计算机类	大一	<a href="#">ysyx_220030-刘刚宇-东北大学秦皇岛分校-学习记录</a>	<a href="#">ysyx_220030-刘刚宇-东北大学秦皇岛分校-读后感</a>	<a href="#">ysyx_220030-刘刚宇-东北大学秦皇岛分校-实验报告</a>			2022.3.20	2022.3	2022.4.10	清假	PA2														
ysyx_22040063	包子旭	河南理工大学	计算机科学与技术	大二	<a href="#">ysyx_220063-包子旭-河南理工大学-学习记录</a>	<a href="#">ysyx_220063-包子旭-河南理工大学-读后感</a>			2022.3.20			2022.4.10	清假	PA2														
ysyx_22040031	苗恒	青岛科技大学	电子信息	研一	<a href="#">ysyx_220031-苗恒-青岛科技大学-学习记录</a>	<a href="#">ysyx_220031-苗恒-青岛科技大学-读后感</a>	<a href="#">ysyx_220031-苗恒-青岛科技大学-实验报告</a>		2022.3.20			2022.4.10	清假	PA2	PA2.1	PA2	2022.2022	2022.2022	2022.7.3	清	2022.7.17	清假	2022.8.7	清假	2022.8.14	清假		
ysyx_22040012	毕睿	海南大学	电子信息工程	大二	<a href="#">副本-ysyx_220012-毕睿-海南大学-学习记录</a>	<a href="#">ysyx_220012-毕睿-海南大学-读后感</a>	<a href="#">ysyx_220012-毕睿-海南大学-实验报告.pdf</a>		2022.3.20	2022.4.20	2022.4.20	2022.4.10	清假	PA2.1	PA2	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040099	蔡惠荣	太原理工大学	通信工程	大二	<a href="#">ysyx_220099-蔡惠荣-太原理工大学-学习记录</a>	<a href="#">ysyx_220099-蔡惠荣-太原理工大学-读后感</a>	<a href="#">ysyx_220099-蔡惠荣-太原理工大学-实验报告</a>		2022.3.20					PA1	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	npc告	准备接入diffset				2022.8.14	清假	
ysyx_22040025	邱尧尧	东北大学	物联网工程	大二	<a href="#">ysyx_220025-邱尧尧-东北大学-学习记录</a>	<a href="#">ysyx_220025-邱尧尧-东北大学-读后感</a>	<a href="#">ysyx_220025-邱尧尧-东北大学-实验报告</a>		2022.3.20					npc-1	进行中	PA2-2,	2022.npc-3	2022.6.19	清假	npc-加	npc-清	2022.8.7	清假					
ysyx_22040034	杜奕明	太原理工大学	软件工程	大四	<a href="#">ysyx_220034-杜奕明-太原理工大学-学习记录</a>	<a href="#">ysyx_220034-杜奕明-太原理工大学-读后感</a>	完成PA1, 无报告		2022.3.20					npc进行中	PA2-1, npc	2022.PA2-1	PA2-2	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假	
ysyx_22040094	黎梓浩	国防科技大学	软件工程	大三	<a href="#">ysyx_220094-黎梓浩-国防科技大学-学习记录</a>	<a href="#">ysyx_220094-黎梓浩-国防科技大学-读后感</a>	完成至PA1.1, 无报告		2022.3.20	2022.3	2022.3	2022.4.24	清	PA2	2022.5.2	npc-c	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假	
ysyx_22040046	王凯	中国科学技术大学	计算机科学与技术	大三	<a href="#">ysyx_220046-王凯-中国科学技术大学-学习记录</a>	<a href="#">ysyx_220046-王凯-中国科学技术大学-读后感</a>	<a href="#">ysyx_220046-王凯-中国科学技术大学-实验报告</a>		2022.3.20	2022.3	2022.3	2022.4.24	清	PA2	2022.5.2	npc-清	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040073	关嵩河	华南理工大学	集成电路工程	研一	<a href="#">ysyx_220073-关嵩河-华南理工大学-学习记录</a>	<a href="#">ysyx_220073-关嵩河-华南理工大学-读后感</a>	完成至PA1.1		2022.3.20		2022.4.20	2022.4.24	清	PA1	2022.5.2	npc-清	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040096	叶剑豪	杭州电子科技大学	软件工程	大三	<a href="#">ysyx_220096-叶剑豪-杭州电子科技大学-学习记录</a>	<a href="#">ysyx_220096-叶剑豪-杭州电子科技大学-读后感</a>	<a href="#">ysyx_220096-叶剑豪-杭州电子科技大学-实验报告</a>		2022.3.20			2022.4.10	清假	PA2.1	2022.sdb	watch	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040080	孙佳宇	电信科学技术研究院	电子科学与技术	研三	<a href="#">ysyx_220080-孙佳宇-电信科学技术研究院-学习记录</a>	<a href="#">ysyx_220080-孙佳宇-电信科学技术研究院-读后感</a>	<a href="#">ysyx_220080-孙佳宇-电信科学技术研究院-实验报告</a>		完成至PA1.1			2022.4.10	清假	PA2.2	NI	2022.5.29	2022.6.19	清假										
ysyx_22040040	祝静	天津理工大学	集成电路工程	研一	<a href="#">ysyx_220040-祝静-天津理工大学-学习记录</a>	<a href="#">ysyx_220040-祝静-天津理工大学-读后感</a>	<a href="#">ysyx_220040-祝静-天津理工大学-实验报告</a>		2022.3.20					PA2.1	完成	PA	2022.5.29	2022.7.3	清	流水钱	SoC中	2022.8.14	清假					
ysyx_22040070	杨利民	北京工商大学	软件工程	大三	<a href="#">ysyx_220070-杨利民-北京工商大学-学习记录</a>	<a href="#">ysyx_220070-杨利民-北京工商大学-读后感.docx</a>		等待更新中	3.20					npc-1	完成	PA	2022.5.29	2022.7.3	清	流水钱	SoC中	2022.8.14	清假					
ysyx_22040163	王晨宇	南通大学	计算机科学与技术	大二	<a href="#">ysyx_22040163-王晨宇-南通大学-学习记录</a>	<a href="#">ysyx_22040163-王晨宇-南通大学-读后感</a>	<a href="#">ysyx_22040163-王晨宇-南通大学-实验报告</a>		2022.3.20					PA3.3	PA3.3	2022.2022	2022.6.5	清假	npc-中	PA3完	2022.8.14	清假						
ysyx_22040068	王俊	海南大学	保研至微电子学, 研究方向:	大四	<a href="#">ysyx_220068-王俊-海南大学-学习记录</a>	<a href="#">ysyx_220068-王俊-海南大学-读后感.docx</a>	截至PA1.1 (问题有点多, 还没来得及汇总和撰写实验报告)		2022.3.20	2022.3.20	2022.3.20	2022.3.20	2022.3.20	PA1														
ysyx_22040188	徐铭伟	西安电子科技大学	信息与通信工程	研一	<a href="#">ysyx_22040188-徐铭伟-西安电子科技大学-学习记录</a>	<a href="#">ysyx_22040188-徐铭伟-西安电子科技大学-读后感</a>	完成至PA1.1		2022.3.20	2022.3.20	2022.3.20	2022.3.20	2022.3.20	PA1														
ysyx_22040214	朱吉宏	哈尔滨工业大学	电子信息	研一	<a href="#">ysyx_22040214-朱吉宏-哈尔滨工业大学-学习记录</a>	<a href="#">ysyx_22040214-朱吉宏-哈尔滨工业大学-读后感</a>	<a href="#">ysyx_22040214-朱吉宏-哈尔滨工业大学-实验报告</a>		2022.3.20					npc-1	完成	PA	2022.5.29	2022.7.3	清	流水钱	SoC中	2022.8.14	清假					
ysyx_22040539	周君宝	清华大学	电子科学与技术	大四	<a href="#">ysyx_22040539-周君宝-清华大学-学习记录</a>	<a href="#">ysyx_22040539-周君宝-清华大学-读后感</a>	<a href="#">ysyx_22040539-周君宝-清华大学-实验报告</a>		2022.3.20	2022.3.20	2022.3.20	2022.3.20	2022.3.20	PA1														
ysyx_22040017	解博元	东北大学秦皇岛分校	计算机科学与技术	大二	<a href="#">ysyx_220017-解博元-东北大学秦皇岛分校-学习记录</a>	<a href="#">ysyx_220017-解博元-东北大学秦皇岛分校-读后感</a>			2022.3.20	2022.3.20	2022.4.10	清假	PA1	2022.5.15	清假													
ysyx_22040067	李珍琪	国防科技大学	计算机科学与技术	大三	<a href="#">ysyx_220067-李珍琪-国防科技大学-学习记录</a>	<a href="#">ysyx_220067-李珍琪-国防科技大学-读后感</a>	<a href="#">ysyx_220067-李珍琪-国防科技大学-实验报告</a>		2022.3.20	2022.3.20	2022.4.24	清假																
ysyx_22040001	蒋雨龙	深圳大学	计算机科学与技术	大四	<a href="#">ysyx_220001-蒋雨龙-深圳大学-学习记录</a>	<a href="#">ysyx_220001-蒋雨龙-深圳大学-读后感</a>	<a href="#">ysyx_220001-蒋雨龙-深圳大学-实验报告</a>		2022.3.20	2022.3.20	2022.4.24	清假	PA2.1															
ysyx_22040178	王九龙	北京邮电大学	电子科学与技术	大三	<a href="#">ysyx_22040178-王九龙-北京邮电大学-学习记录</a>	<a href="#">ysyx_22040178-王九龙-北京邮电大学-读后感</a>	<a href="#">ysyx_22040178-王九龙-北京邮电大学-实验报告</a>		3.20	2022.3.27	清假																	
ysyx_22040374	崔家贺	南开大学	软件工程	大二	<a href="#">ysyx_22040374-崔家贺-南开大学-学习记录</a>	<a href="#">ysyx_22040374-崔家贺-南开大学-读后感</a>	<a href="#">ysyx_22040374-崔家贺-南开大学-实验报告</a>		2022.3.20	2022.3	2022.4.24	清假	PA2	2022.5.2	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假	
ysyx_22040154	徐步青	东南大学	电子科学与技术	大四	<a href="#">ysyx_22040154-徐步青-东南大学-学习记录</a>	<a href="#">ysyx_22040154-徐步青-东南大学-读后感</a>	<a href="#">ysyx_22040154-徐步青-东南大学-实验报告</a>		2022.3.20					PA2.3	声卡前	PA1	2022.5.2	2022.2022	2022.6.19	清假	npc-清	npc-清	2022.8.7	清假				
ysyx_22040193	苑文强	中国科学技术大学	电子信息	研二	<a href="#">ysyx_22040193-苑文强-中国科学技术大学-学习记录</a>	<a href="#">ysyx_22040193-苑文强-中国科学技术大学-读后感</a>	<a href="#">ysyx_22040193-苑文强-中国科学技术大学-实验报告</a>		2022.3.20					PA2	完成	PA1	2022.5.29	2022.7.3	清	2022.7.10	清假	2022.8.7	清假					
ysyx_22040142	刘森	上海海洋大学	计算机科学与技术	大三	<a href="#">ysyx_22040142-刘森-上海海洋大学-学习记录</a>	<a href="#">ysyx_22040142-刘森-上海海洋大学-读后感</a>	<a href="#">ysyx_22040142-刘森-上海海洋大学-实验报告</a>		2022.3.20					PA2	除PA2	除PA2	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	
ysyx_22040200	乌鑫龙	北京师范大学珠海分校	计算机科学与技术	大三	<a href="#">ysyx_22040200-乌鑫龙-北京师范大学珠海分校-学习记录</a>	<a href="#">ysyx_22040200-乌鑫龙-北京师范大学珠海分校-读后感</a>	<a href="#">ysyx_22040200-乌鑫龙-北京师范大学珠海分校-实验报告</a>		2022.3.20	2022.3.20	2022.3.27	清假	PA2.3	2022.npc	基础	2022.7.3	清假	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040414	林沛源	广东工业大学	集成电路设计与集成系统	大三	<a href="#">ysyx_22040414-林沛源-广东工业大学-学习记录</a>	<a href="#">ysyx_22040414-林沛源-广东工业大学-读后感</a>	<a href="#">ysyx_22040414-林沛源-广东工业大学-实验报告</a>		2022.3.20	2022.4.20	2022.4.24	清假	PA2.3	2022.npc	基础	2022.7.3	清假	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.2022	2022.8.14	清假
ysyx_22040413	暴劼	电子科技大学	电子信息工程	研二	<a href="#">ysyx_22040413-暴劼-电子科技大学-学习记录</a>	<a href="#">ysyx_22040413-暴劼-电子科技大学-读后感</a>	<a href="#">ysyx_22040413-暴劼-电子科技大学-实验报告</a>		2022.3.20	2022.4.20	2022.4.24	清假	PA2	2022.5.22	清假													
ysyx_22040595	郑佳纯	华南理工大学	电子科学与技术	大四	<a href="#">ysyx_22040595-郑佳纯-华南理工大学-学习记录</a>	<a href="#">ysyx_22040595-郑佳纯-华南理工大学-读后感</a>	<a href="#">ysyx_22040595-郑佳纯-华南理工大学-实验报告</a>		2022.3.20					木有权限														
ysyx_22040301	于皓哲	沈阳工业大学	电子科学与技术	研二	<a href="#">ysyx_22040301-于皓哲-沈阳工业大学-学习记录</a>	<a href="#">ysyx_22040301-于皓哲-沈阳工业大学-读后感</a>	<a href="#">ysyx_22040301-于皓哲-沈阳工业大学-实验报告</a>		2022.3.20	2022.4.20	2022.4.24	清假	PA1															
ysyx_22040562	李张勋	华中科技大学	集成电路设计与集成系统	大三	<a href="#">ysyx_22040562-李张勋-华中科技大学-学习记录</a>	<a href="#">ysyx_22040562-李张勋-华中科技大学-读后感</a>	<a href="#">ysyx_22040562-李张勋-华中科技大学-实验报告</a>		2022.3.20	2022.4.20	2022.4.24	清假	PA1															
ysyx_22040228	李国旗	齐鲁理工学院	电子信息工程	大二	<a href="#">ysyx_22040228-李国旗-齐鲁理工学院-学习记录</a>	<a href="#">ysyx_22040228-李国旗-齐鲁理工学院-读后感</a>	<a href="#">ysyx_22040228-李国旗-齐鲁理工学院-实验报告</a>		2022.3.20					npc-1	完成	PA1	2022.5.29	2022.7.3	清	流水钱	SoC中	2022.8.14	清假					
ysyx_22040221	管明星	山东科技大学	计算机技术	研一	<a href="#">ysyx_22040221-管明星-山东科技大学-学习记录</a>	<a href="#">ysyx_22</a>																						

# 学生的学习记录

花费时间

日期

任务

遇到的困难

如何解决

日期	计划任务	总时长	任务完成情况	卡了一段时间的bug
2月25日	安装Veril		安装完成	
2月26日	调试		调试完成	
2月27日	完成实验报		完成实验报	
3月1日	完成实验报		完成实验报	
3月3日	完成实验报		完成实验报	
3月6日	准备并		准备并	
3月7日	完成CSR指令		完成CSR指令	
3月8日	修正位		修正位	
3月9日	完成		完成	
3月10日	引入c		引入c	
3月11日	完		完	
3月12日	进一		进一	
3月14日	完成		完成	
3月15日	完成键		完成键	
3月16日	完成VGA及		完成VGA及	
3月17日	完成		完成	
3月18日	完成		完成	
3月19日	完成dtr		完成dtr	
3月20日	完成traco, 完		完成traco, 完	
3月21日	完成中		完成中	
3月22日	完成		完成	
3月22日	测试中断		测试中断	
3月24日	完成zemu与		完成zemu与	
3月25日	完成zemu与		完成zemu与	
3月26日	尝试引		尝试引	
3月27日	完成最基		完成最基	
2月28日	尝试完成MIPS的MEMU	5h	基础部分MIPS的DR19C-V差距不大, 实现起来的思路也大同小异	貌似MIPS的ISA没有更新, 直接make会报错, 查看了报错信息前几年遗留的bug
2月29日	完成MIPS的得分计算	5h	完成浮点寄存器, 可以将32位和64位得分数存到寄存器中	
3月30日	完成MIPS的得分		完成MIPS的得分	
3月31日	完成MIPS的得分		完成MIPS的得分	
4月1日	完成CSR指令		完成CSR指令	
4月2日	完成CSR指令		完成CSR指令	
4月3日	调试		调试	
4月4日	请明放假		请明放假	
4月5日	调试		调试	
4月6日	FPGA造型+起		FPGA造型+起	
4月7日	完成DDR3L内存		完成DDR3L内存	
4月8日	完成Flash绘		完成Flash绘	
4月9日	使用SPP和PMC作为外		使用SPP和PMC作为外	
4月10日	修正上电时序		修正上电时序	
4月11日	修正上电时序		修正上电时序	
4月12日	布线		布线	
4月13日	布线		布线	
4月14日	打板		打板	
4月15日	配置Verdi环境 & 学		配置Verdi环境 & 学	
4月16日	配置Virtuoso环境 & 开		配置Virtuoso环境 & 开	
4月17日	画CPU铜架结构		画CPU铜架结构	
4月18日	修正铜架结构		修正铜架结构	
4月19日	完成取指和R964I		完成取指和R964I	
4月20日	完成R964I		完成R964I	
4月21日	修改微架构		修改微架构	
4月22日	编写Testband		编写Testband	
4月23日	编写Testband		编写Testband	
4月24日	编写Testband		编写Testband	
4月25日	取指部分重写		取指部分重写	
4月26日	编写计分板		编写计分板	
4月27日	编写计分板		编写计分板	
4月28日	学习香山代码		学习香山代码	
4月29日	学习香山代码		学习香山代码	
4月30日				
5月1日	五一放假		五一放假	
5月2日				
5月3日				
5月4日				
5月5日	修Spina1HDL	2h	Spina1HDL在Pin	
5月6日	修Sp		修Sp	
5月7日	调试		调试	
5月8日	调试		调试	
5月9日	调试		调试	
5月10日	调试		调试	
5月11日	调试		调试	
5月12日	调试		调试	
5月13日	重写gol		重写gol	
5月14日	重写gol		重写gol	
5月15日	编写龙		编写龙	
5月16日	测试		测试	
5月17日	学习LoongArch《行		学习LoongArch《行	
5月18日	学习LoongArch《行		学习LoongArch《行	
5月19日	学习LoongArch《行		学习LoongArch《行	
5月20日	学习LoongArch《行		学习LoongArch《行	
5月21日	学习LoongArch《行		学习LoongArch《行	
5月22日	学习《重		学习《重	
5月23日	学习《重		学习《重	
5月24日	学习《重		学习《重	
5月25日	学习《重		学习《重	
5月26日	学习《重		学习《重	
5月27日	修复S		修复S	
5月28日	修复S		修复S	
5月29日	修复S		修复S	
5月30日	修复		修复	
5月31日	学习《		学习《	
6月1日	修复		修复	
6月2日	修复		修复	
6月3日	调试LA顶层		调试LA顶层	
6月4日	调试LA的AXI		调试LA的AXI	
6月5日	调试LA的AXI		调试LA的AXI	
6月6日	学习		学习	
6月7日	学习		学习	
6月8日	学习《超标		学习《超标	
6月9日	调试AXI总		调试AXI总	
6月10日	调试AXI总		调试AXI总	
6月11日	调试AXI总		调试AXI总	
6月12日	调试AXI总		调试AXI总	
6月13日	学习《超标		学习《超标	
6月14日	学习《超标		学习《超标	
6月15日	切换到Lattic		切换到Lattic	
6月16日	切换到Lattice芯片		切换到Lattice芯片	
6月17日	学习 yos		学习 yos	
6月18日	学习Reinde		学习Reinde	
6月19日	学习Reinde		学习Reinde	
6月20日	学习Reinde		学习Reinde	
6月21日	学习Reinde		学习Reinde	
6月22日	学习《超标		学习《超标	
6月23日	学习《超标		学习《超标	
6月24日	学习《超标		学习《超标	
6月25日	学习《超标		学习《超标	
6月26日	学习《超标		学习《超标	
6月27日	写DCache		写DCache	
6月28日	写DCache		写DCache	
6月29日	写DCache		写DCache	
6月30日	补MUL指令		补MUL指令	
7月1日	补DIV指令		补DIV指令	
7月2日	测试MUL和DIV		测试MUL和DIV	
7月3日	编译LA3		编译LA3	
7月4日	调试LA顶层		调试LA顶层	
7月5日	调试LA的AXI		调试LA的AXI	
7月6日	调试LA的AXI		调试LA的AXI	
7月7日	调试LA的AXI		调试LA的AXI	
7月8日	调试LA的AXI		调试LA的AXI	
7月9日	调试LA的AXI		调试LA的AXI	
7月10日	调试LA的p2mem	4h	p2mem	p2mem编写步骤挺多, 而且不知道是不是已经烧录进去了
7月18日	跑通difftest	6h	之前的设计时候没有考虑difftest, 导致模块之间只引出了总线, 调用了一下跳线特性引出了所需总线	之前的设计不太适合difftest仿真, difftest通过自定义总线进行数据传入和传出, 横跨了Petch和VB部分
7月19日	跑通difftest	4h	写了个makefile, 可以一次性导出verilog文件后再用verilator编译测试	Spina1HDL和difftest结合起来就很麻烦, 需要先导出才可以, 不可以直接调用内部verilator
7月20日	测试样例	5h	调试goldentrace, 跑完lab9	每一次都要重新生成trace文件, 导致重新跑一次需要很长时间, 调了一下makefile可以缓存一下
7月21日	测试样例	6h	官方的样例太大了, 就算是缓存了跑一次也需要五分钟以上, PA中的有些样例会小很多, 直接移植过来	由于只要测试指令是否实现正确, 指令完整测试可以拖到联调的时候, 龙芯提供的goldentrace环境, Ubuntu不行, 在CentOS下可以测试跑起来就比手调快多了, 但是查看波形很容易卡, 基本只能看上一小部分
7月22日	测试样例	4h	跑完ALU部分, 修正了一些有符号数处理的问题	貌似官方的程序会在这里编译不过, 有个类型写法有问题, 会引起vivado报错
7月23日	测试样例	4h	引入中断处理, 引出中断总线和csr总线	在modelsim仿真中乘法器是实时输出的, 但是似乎在verilator中会延迟时钟周期
7月24日	修复乘法器	6h	乘法器修改后在difftest中报错, 且全是零, 但是上板测试没有问题, 使用开器的乘法器又可以使用	Spina1HDL新版支持使用Xilinx仿真, 但是verilator翻了很多地方都没有支持IP核的仿真
7月25日	使用IP核模拟	6h	把乘法器和除法器换成了IP核, 上板测试没有问题, 但是跑difftest编译不过, 想了很多办法, 先用宏定义切换, 但是IP核是否能够准确使用不能保证改了一下全局时钟, 但是会导致外设时钟不对, 而外设时钟全部共享总体时钟, 全部把时钟迁移出来	一开始串口输出的全是乱码, 但是把波特率调高点居然就可以用了
7月26日	提升CPU频率	6h		
7月27日				
7月28日				
7月29日	电赛休息			
7月30日				
7月31日				
8月1日	分离APB总线时钟	4h	提升频率后发现串口波特率发生变化, 需要提升至原来两倍的波特率才可以使用, 但是UART16550使用的时钟和APB时钟是一样的, 需要修改总线时钟	一开始以为UART串口是内部分频至115200, 后来翻来翻去看看到分频器, 而且输出的时钟都是一样的, 最后才想到估计是p2mem和Linux软件定义了串口的波特率
8月2日	编译修改p2mem	6h	跨时钟域的总线不好处理, 尤其是总线上还有一堆外设串着, 打算直接调整p2mem的APB总线频率, 不过p2mem给的makefile非常的神奇, 看似编译了, 实际还是调了一个预编译的文件, 而且编译器内部和公开的也有区别, 最后半抄半改的好歹能编译出ROM文件了	貌似p2mem的APB总线频率定义有问题, 改了后和烧录区区别, 最后直接把波特率除以二编译
8月3日	修正initrd	6h	p2mem是能正常用了, 但是一运行Linux波特率又回去了, 但是这个波特率好像和p2mem关系不大, 更像是Linux硬件树里定义的	initrd是在p2mem的预编译文件中, 这一部分恰好没有源代码, 上板直接动flash但是p2mem好像是刷固件的, 没有带启动管理器
8月4日	编译上板测试coremark	5h	龙芯的coremark是直接烧录flash的模式, 并不是在Linux中跑coremark程序的模式, 但是给的配置文件都是硬编码, 只能从官方源代码一份, 然后交叉编译再上去跑	一开始跑不起来报非法指令, 还以为是CPU实现的有问题, 后来看了下difftest好像没有报错, 后来才发现这个是硬编码设定
8月5日	减少输出数量	6h	频率开高后时钟约束就跑不过, 看了下net延迟比logic延迟大太多了, 而且Layout图上很明显的跨区连线, 估计是输出太大, 限制输出数量后报错数量减少了	时钟约束受到扇出量影响力度相当大, 但是默认是按全局出做的, 尽管寄存器会减少很多网络延迟
8月6日	忽略指定path	6h	分析了一下剩下的超时path, 大部分都是经过mul, 看路径推测是由mul引起PC改变, 进而刷新流水线, 因为日常根本不会有这种情况, 直接忽略, 然后就没有时报错了	正常情况下ALU应该是不会影响PC值的, 但是在path中可以推测出来vivado认为ALU在一定条件下会影响PC值进而导致流水线冲刷 起延迟
8月7日	修正no drive on错误	4h	在Spina1HDL写CSR的时候, 只要操作CSR寄存器就会编译报错, 但是相同的逻辑在Chisel中是正常的, 而且生成的Verilog代码也没有问题	Chisel中的Bundle可以带方向, 但是Spina1HDL中的Bundle不能带方向, 手动在io中指定, 光方向就有一大段
8月8日	修正no drive on错误	3h	貌似只要操作Bundle中的任何一个信号, 都会导致Bundle里面所有的信号都报错	修改半天, 直到最后连最基础的寄存器赋值都报错, 不知道是哪儿的问题
8月9日	修正no drive on错误	3h	调试了半天才发现如果io的信号没有外部连接就会报错, 只有把全部信号全连上才行	增加一个模块就要修改IO引出配置, 但是引出配置要全部改才可以运行, 不如Chisel方便, 只需要改一部分就可以
8月10日	修正difftest接口信号名	6h	difftest运行起来报错, 检查了很久才发现是接口处的信号名全部加上了in_的前缀, 即使是blackbox引入的也会自动加上, 在源代码里一通删, 最后通过反射重命名信号名才可以	文档里面根本没提会自动修改生成的信号名, 而且也没提如何取消信号名前缀, 最后还得翻源代码
8月11日	新增中断信号	5h	直接把外部信号引入CSR模块, 一旦传入中断, 就直接设置寄存器	寄存器设置的时候会报Multiple Driver, 但是Spina1HDL没有Lookup, 手动展开成mux
8月12日	新增中断信号	6h	不能直接设置寄存器, 还必须先刷流水线, 这就导致了中断和其他模块连在一起, 只能全部加了中断和异常信号量	没想到中断的复杂程度会比CSR之类更高, 还要考虑中断优先级
8月13日	新增中断信号	5h	测试时候中断会引起当前寄存器内容丢失, 在中断中出不来	Spina1HDL的UInt类型自带信号线, 一旦使用就会在顶层生成一个信号量
8月14日				

# 累计报名人数超过1万

	开始时间	结束时间	报名人数	覆盖高校	通过入学答辩
第1期	2019年8月	2020年7月	5	1	5
第2期	2020年8月	2021年6月	11	5	11
第3期	2021年7月	2022年1月	760	168	215
第4期	2022年2月	2022年7月	1753	328	215
第5期	2022年8月	2023年6月	1881	379	155
第6期	2023年7月	2024年6月	3331	553	369
24级	2024年7月	2024年12月	2740	500	148
25级	2025年1月	-	2429	536	182
总计			12550	1031	1300

# 在线调试考核

- 申请流片时, 助教在学生提交的代码中**随机注入bug**
  - 覆盖硬件, 软件, 构建系统和仿真环境
- 学生需要通过在线会议**在1小时内完成调试**
  - 学生需要共享屏幕
- 根据调试过程, 助教可以评估学生
  - 对工程细节的了解是否足够深入
  - 能否从软硬件协同的角度分析问题
  - 是否知道在什么时候该用什么工具
  - 能否独立解决新问题

# 第三期 “一生一芯” 流片

## ● 流片情况:

- 2021年12月底班车: **39个处理器核** (包括助教5个、测试核2个)
- 2022年2月底班车: **9个处理器核**

## ● 答辩情况

- **C9 (10) , 985 (12) , 211 (9) , 普通高校 (11)**
- 电子/微电子/集成电路: **25人**; 计软: **13人**; 其他: **4人**
- **大一 (2) , 大二 (3) , 大三 (11) , 大四 (3)**
- **研一 (8) , 研二 (11) , 研三 (1) , 博一/二/三 (4)**

## ● 指标情况

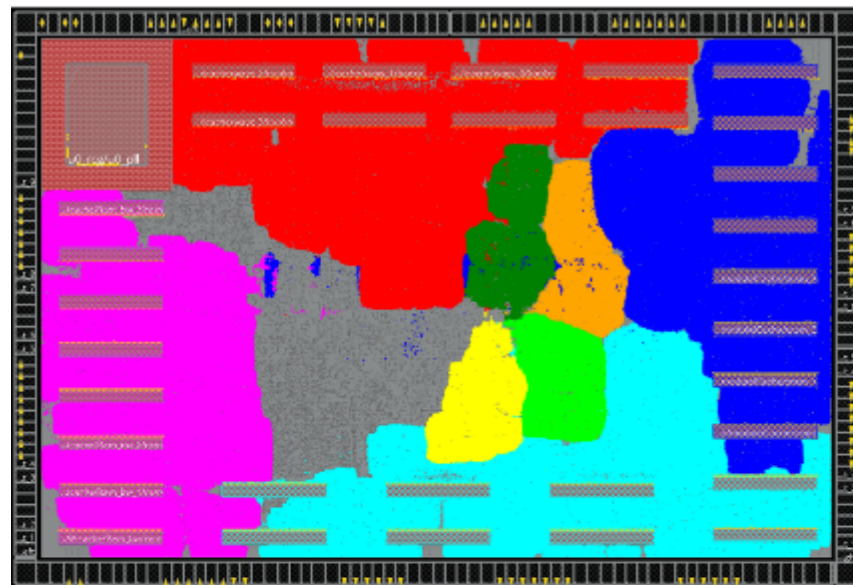
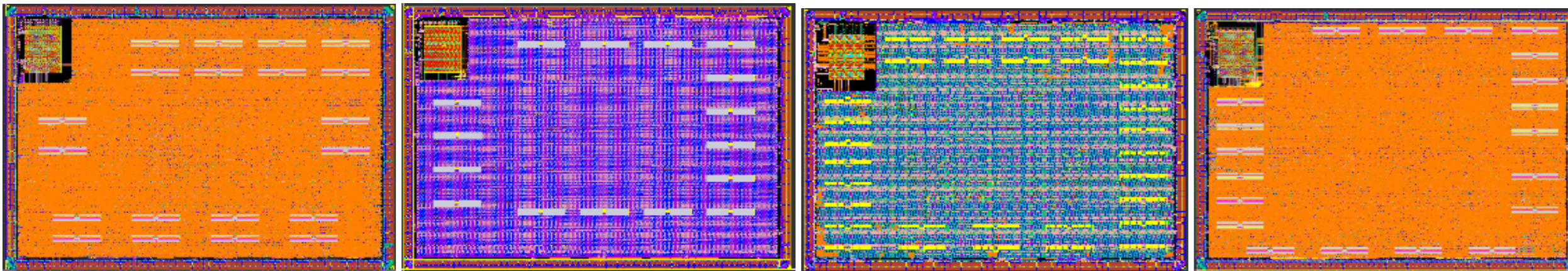
- 分支预测: 12个;
- 乱序执行: 3个;
- Chisel:16个; Verilog/sv: 31个
- Cache: 12个

学号	学校	年级	
ysyx_210596	中山大学	研二	微电子
ysyx_210760	扬州大学	大三	计算机科学与技术
ysyx_210407	长春理工大学	大四	电子科学与技术
ysyx_210366	广东工业大学	研二	控制科学与工程
ysyx_210703	上海大学	大三	计算机科学与技术
ysyx_210324	西北工业大学	研三	控制工程
ysyx_210747	北京航空航天大学	研二	电子信息工程
ysyx_210746	电子科技大学	研一	电子科学与技术
ysyx_210448	山东交通学院	大二	物联网工程

ysyx_210340	中国科学院大学	直博二年级	微电子学与固体电子学
ysyx_210092	西安电子科技大学	大三 (即将升)	计算机科学与技术专业嵌入
ysyx_210456	电子科技大学	研一	电子信息科学与技术
ysyx_210247	南京理工大学	研一	电子信息
ysyx_210243	华中科技大学	大三	电子信息与通信工程
ysyx_210544	南京航空航天大学	博士一年级	软件工程
ysyx_210232	青岛科技大学	大三	集成电路设计与集成系统
ysyx_210295	华东师范大学	研一	集成电路设计与集成系统
ysyx_210457	山东交通学院	大一	电子信息工程
ysyx_210458	太原理工大学	大二	水利
ysyx_210611	南京大学	大一	计算机科学与技术
ysyx_210285	南京大学	大二 (准大三)	计算机科学与技术
ysyx_210128	上海交通大学	大四	电子与计算机工程
ysyx_210727	华中科技大学	研二	计算机科学与技术
ysyx_210718	深圳大学	研二	电子信息
ysyx_210133	电子科技大学	研二	电子科学与工程学院

ysyx_210292	集美大学	大四	计算机科学与技术
ysyx_210191	南京理工大学	大三	计算机科学与技术
ysyx_210195	西安电子科技大学	硕士二年级	电子科学与技术
ysyx_210413	大连理工大学	研一	软件工程
ysyx_210428	沈阳工业大学	研二	电子科学与技术
ysyx_210313	电子科技大学	大三	微电子
ysyx_210302	复旦大学	研一	微电子学与固体电子学
ysyx_210184	清华大学	研二	集成电路工程
ysyx_210479	太原理工大学	大三	计算机科学与技术
ysyx_210013	西安交通大学	研二	微电子学与固体电子学
ysyx_210438	南京大学	直博二年级	电子信息技术
ysyx_210555	南京大学	研一	集成电路工程
ysyx_210528	中国农业大学	大三	电子信息工程
ysyx_210669	北京工业大学	研二	计算机技术
ysyx_210417	中国科学技术大学	研一	集成电路工程
ysyx_210134	浙江大学	大三	计算机科学与技术
ysyx_210152	重庆邮电大学	大三	电子信息工程

# 第三期 “一生一芯” 的成果



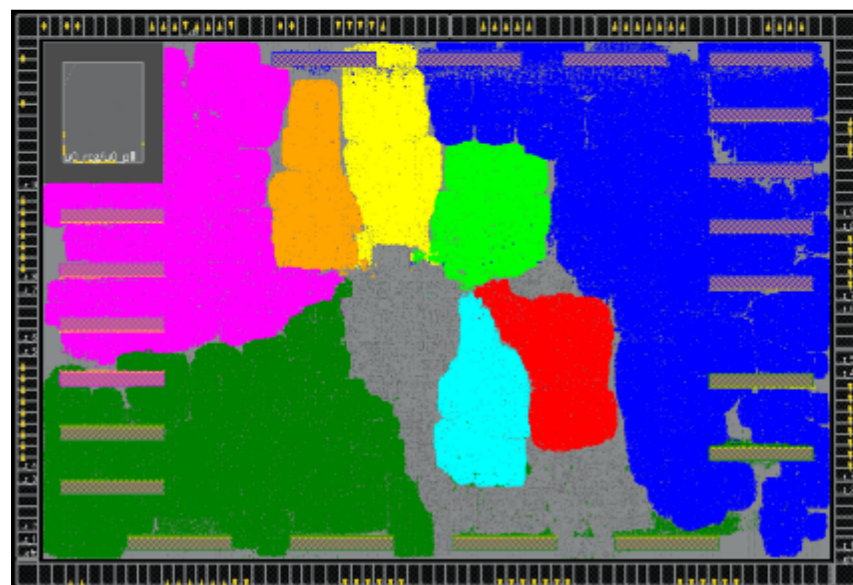
SoC\_1

- 210013
- 210153
- 210247
- 210285
- 210528
- 210539
- 210718
- 210727



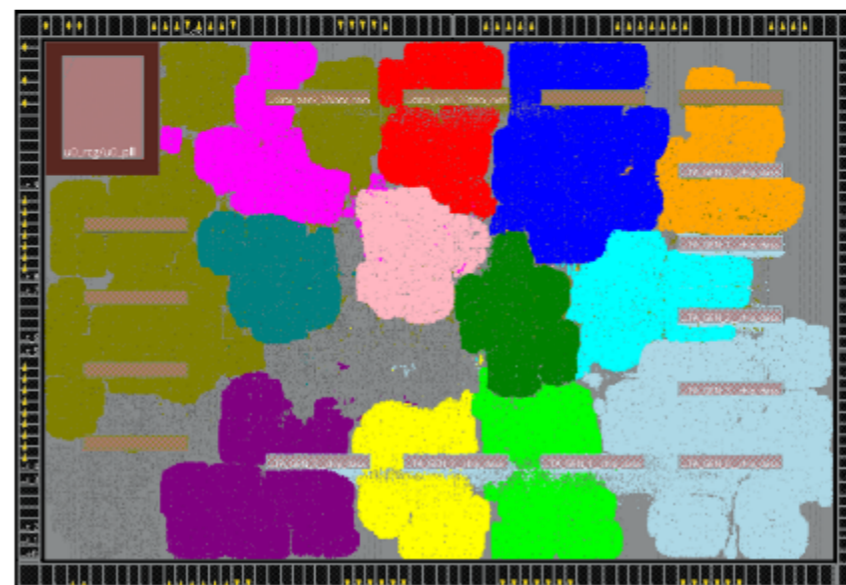
SoC\_2

- 210000
- 210101
- 210184
- 210456
- 210171
- 210232
- 210313
- 210340
- 219999
- 210413



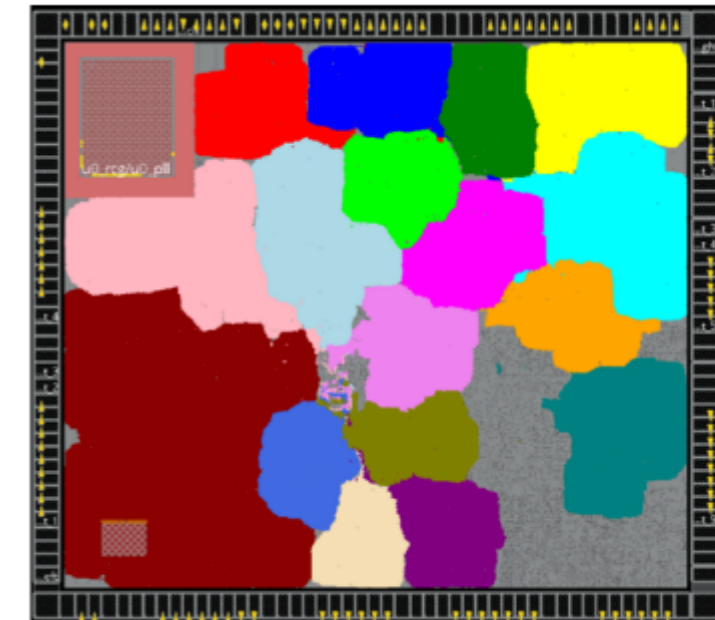
SoC\_3

- 210092
- 210128
- 210152
- 210295
- 210438
- 210479
- 210555
- 210669



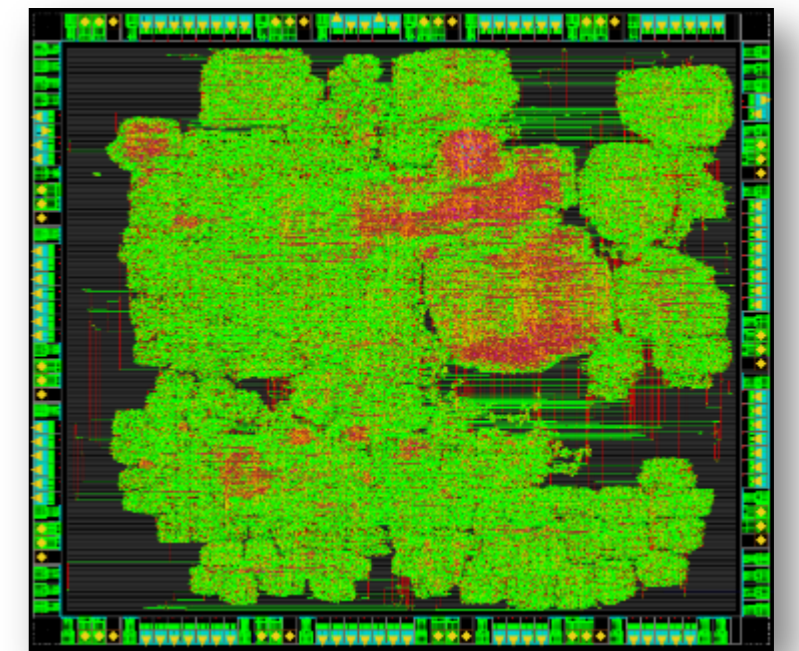
SoC\_4

- 210133
- 210134
- 210191
- 210195
- 210243
- 210292
- 210302
- 210417
- 210428
- 210457
- 210458
- 210544
- 210611



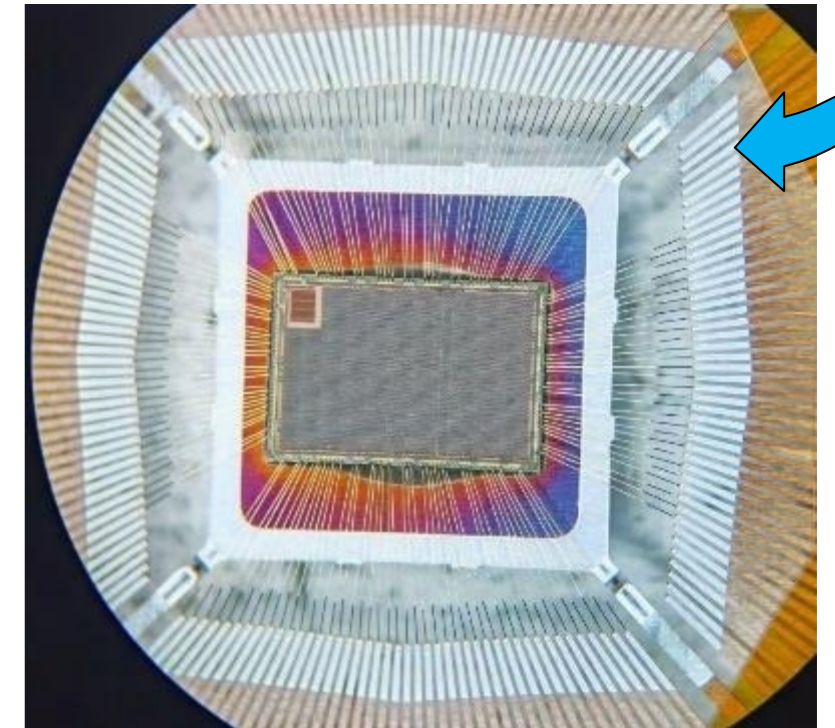
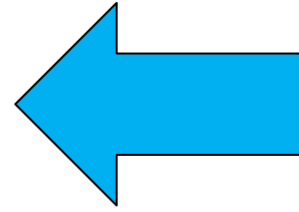
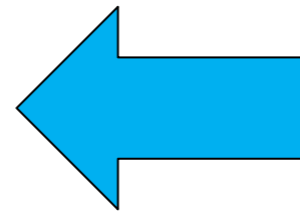
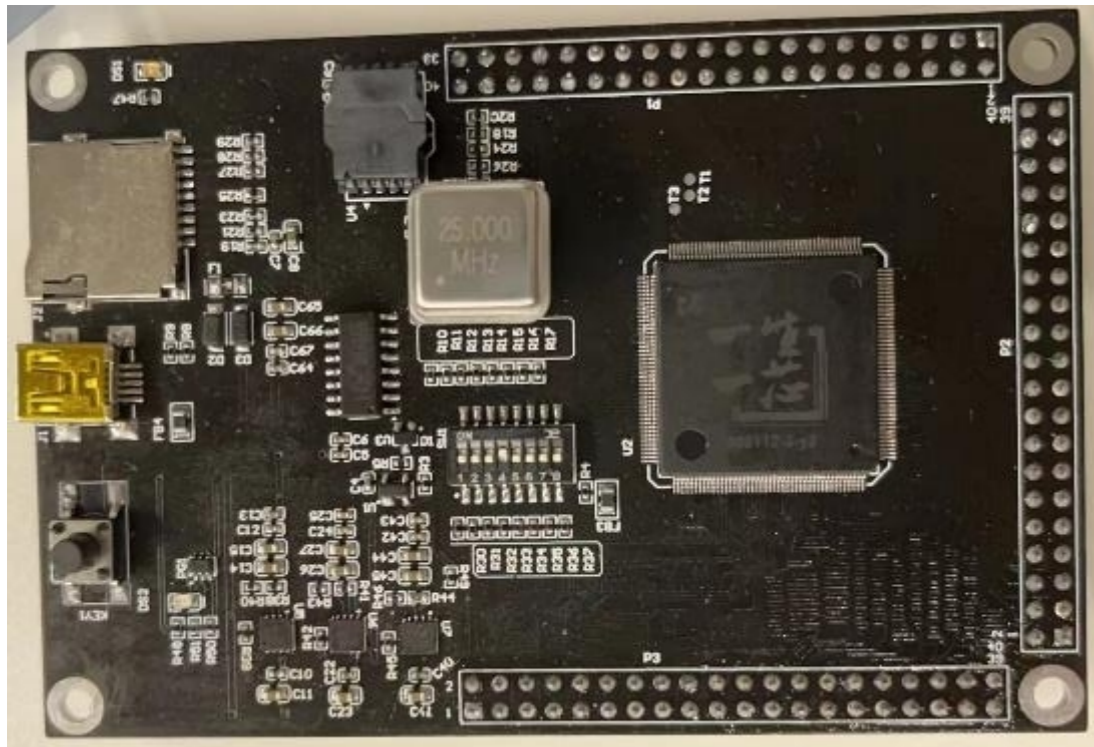
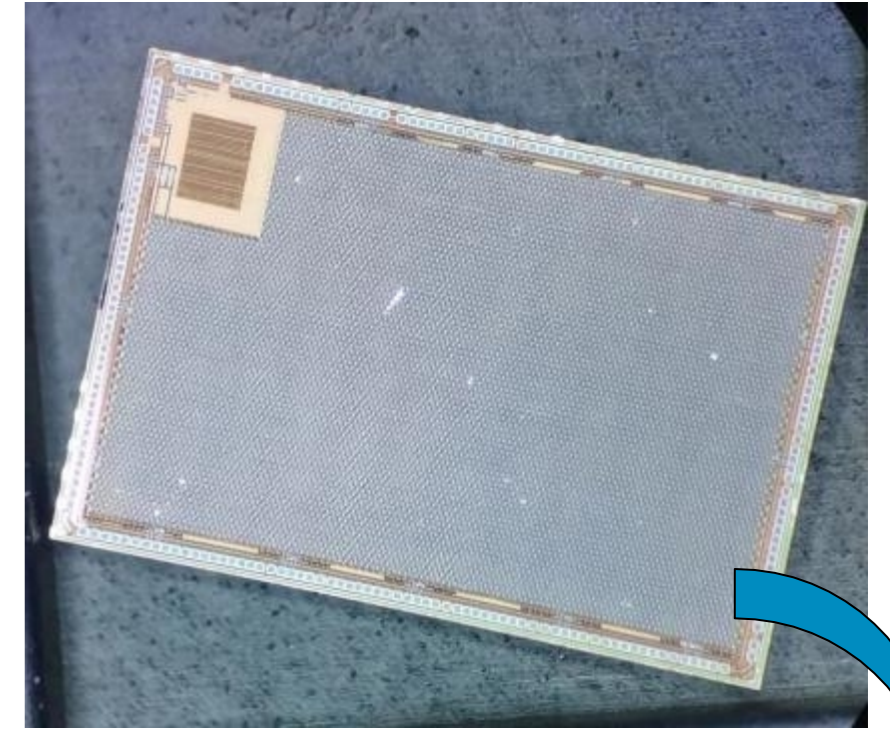
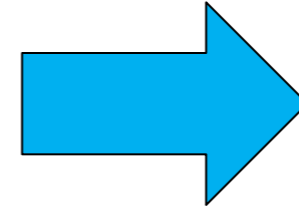
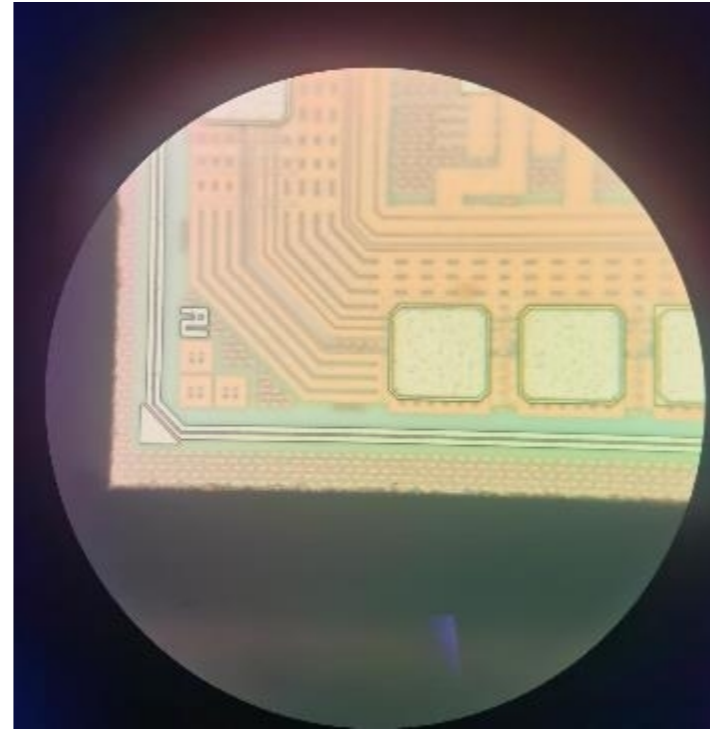
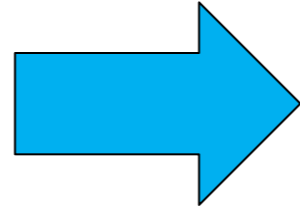
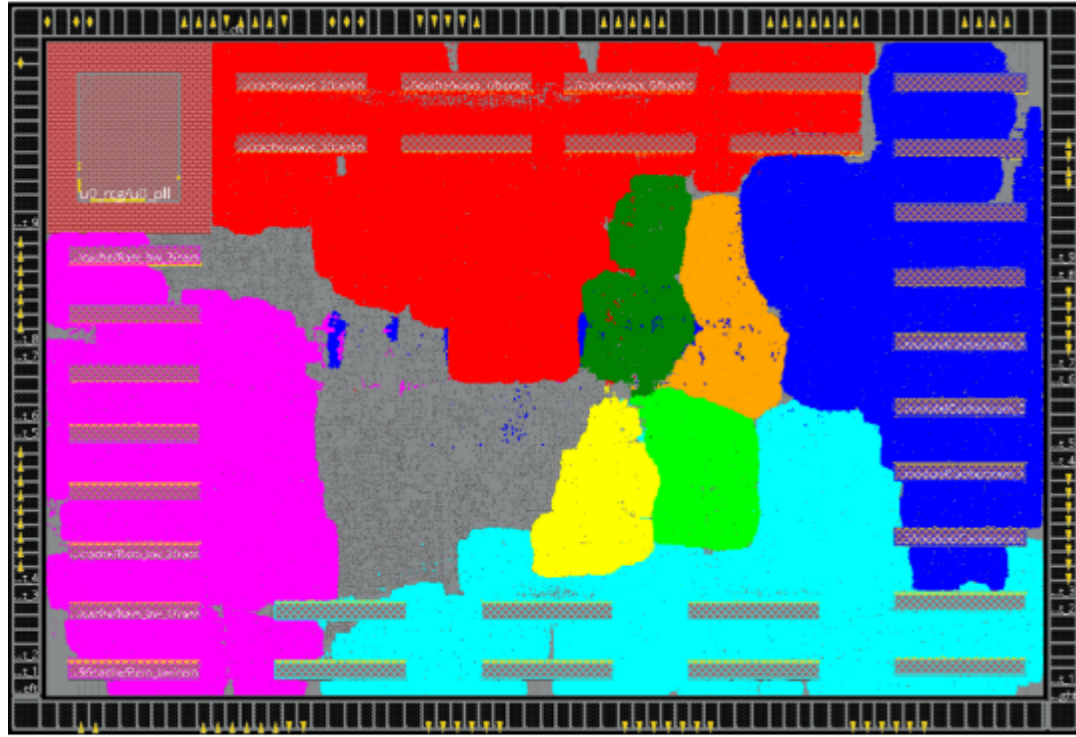
SoC\_5

- 210745
- 210407
- 210596
- 210703
- 210366
- 210448
- 210324
- 210760
- 210746
- chenxi
- chenguokai
- hbh
- lx
- yanyue
- yyh
- zjv
- zzy



用开源EDA工具设计并投片的  
110nm (一生一芯) 处理器核

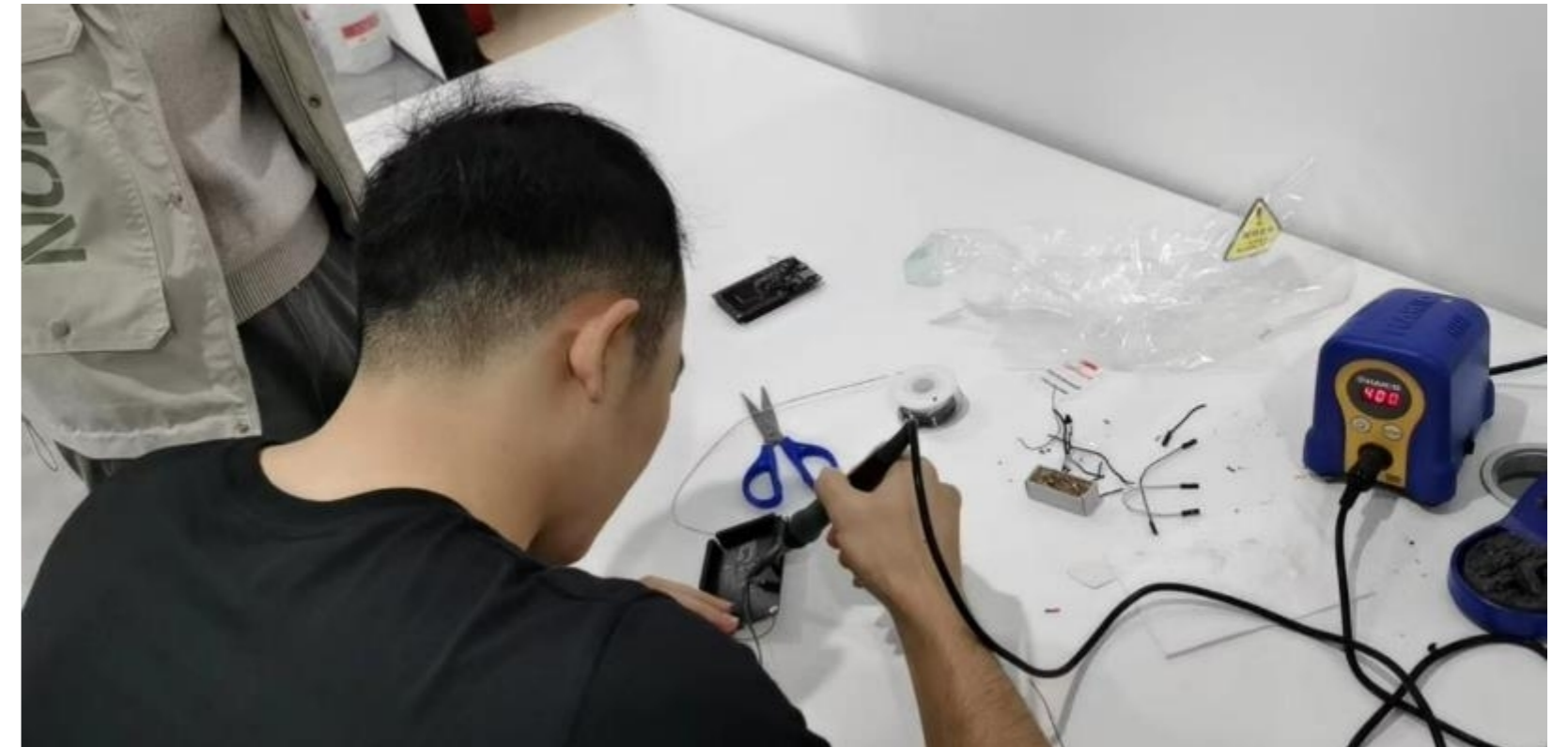
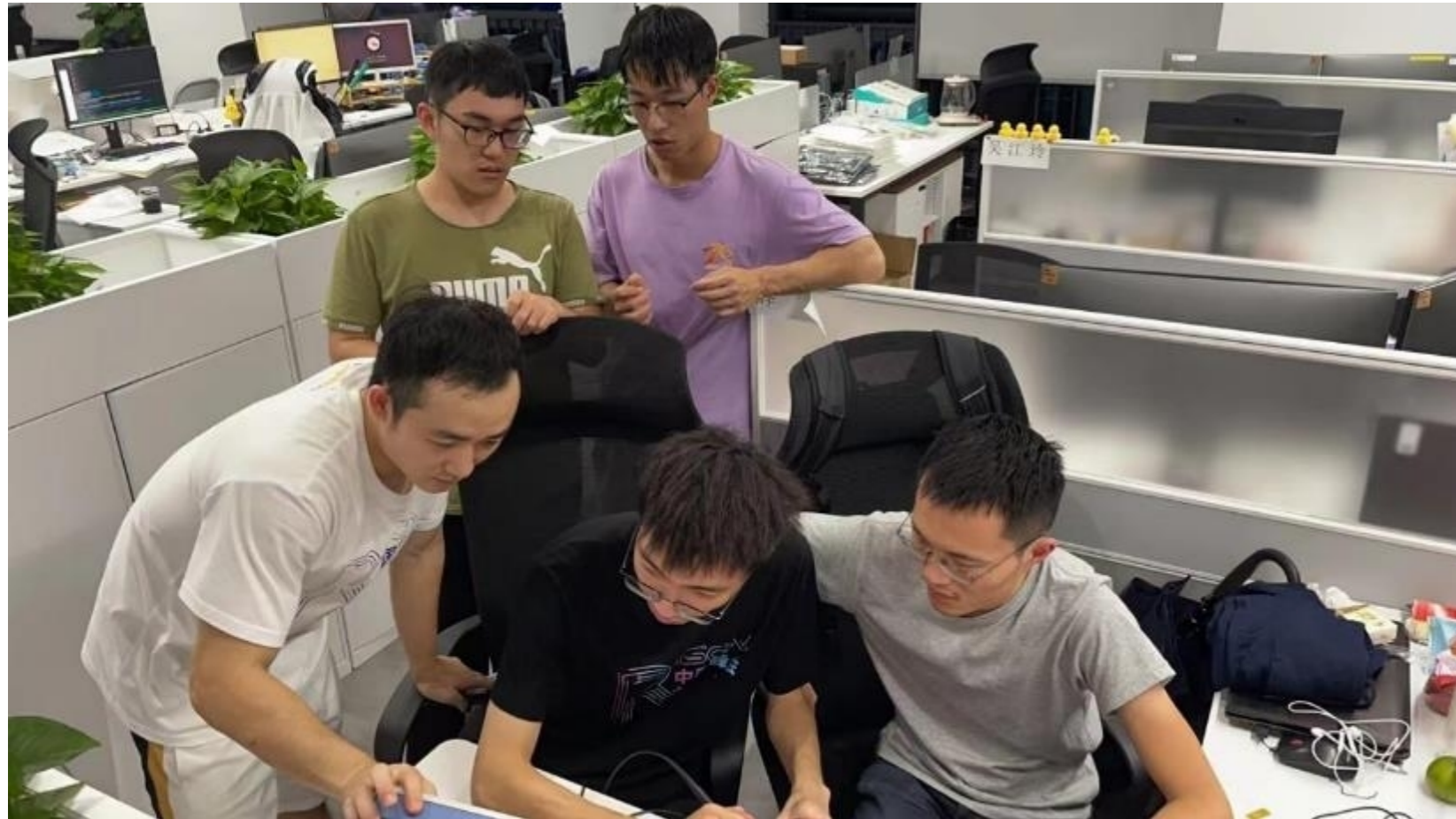
# 处理器芯片和板卡展示



# 第三期板卡元件测试

## 测试团队

- 黄健明(海南大学)
- 卢非凡(西安财经大学)
- 马壮(中国科学技术大学)
- 缪宇驰(鹏城实验室)
- 许立达(中科院微电子所)



# 第三期板卡软件测试

	hello world	memtest
flash	<pre>[10:59:26.279]收←◆Hello World! [11:04:04.430]收←◆Hello World! [11:04:06.275]收←◆Hello World!</pre>	<pre>[14:51:55.890]收←◆start test... mem tests prepared mem tests passed!!</pre>
mem	<pre>[14:19:33.762]收←◆Loading program of size: 208 bytes, expect 128 '# Loading.... ##### Load finished Exec app... Hello World!</pre>	<pre>[15:06:04.482]收←◆Loading program of size: 3840 bytes, expect 128 '# Loading.... ##### [15:06:04.745]收←◆##### Load finished Exec app... start test... mem tests prepared mem tests passed!!</pre>

## 内存测试

```
[10:31:49.358]收←◆[mem data] cnt: 65929216(3ee0000), addr: 0x9f900000
[10:31:49.416]收←◆[mem data] cnt: 65994752(3ef0000), addr: 0x9f980000
[10:31:49.476]收←◆[mem data] cnt: 66060288(3f00000), addr: 0x9fa00000
[10:31:49.535]收←◆[mem data] cnt: 66125824(3f10000), addr: 0x9fa80000
[10:31:49.594]收←◆[mem data] cnt: 66191360(3f20000), addr: 0x9fb00000
[10:31:49.653]收←◆[mem data] cnt: 66256896(3f30000), addr: 0x9fb80000
[10:31:49.712]收←◆[mem data] cnt: 66322432(3f40000), addr: 0x9fc00000
[10:31:49.772]收←◆[mem data] cnt: 66387968(3f50000), addr: 0x9fc80000
[10:31:49.830]收←◆[mem data] cnt: 66453504(3f60000), addr: 0x9fd00000
[10:31:49.889]收←◆[mem data] cnt: 66519040(3f70000), addr: 0x9fd80000
[10:31:49.949]收←◆[mem data] cnt: 66584576(3f80000), addr: 0x9fe00000
[10:31:50.007]收←◆[mem data] cnt: 66650112(3f90000), addr: 0x9fe80000
[10:31:50.067]收←◆[mem data] cnt: 66715648(3fa0000), addr: 0x9ff00000
[10:31:50.126]收←◆[mem data] cnt: 66781184(3fb0000), addr: 0x9ff80000
[10:31:50.185]收←◆mem tests passed!!
[10:32:25.838]收←◆\0\0
```

## RT-Thread 测试

```
Load finished
Exec app...
heap: [0x80022590 - 0x86422590]

\ | /
- RT - Thread Operating System
/ | \ 4.0.4 build Nov 29 2022
2006 - 2021 Copyright by rt-thread team
Hello RISC-V!
thread1 count: 0
thread2 count: 0
msh />
[10:41:52.743]收←◆thread1 count: 1
thread2 count: 1

[10:41:53.385]收←◆thread1 count: 2
thread2 count: 2
```





# 流片成果展示(2)

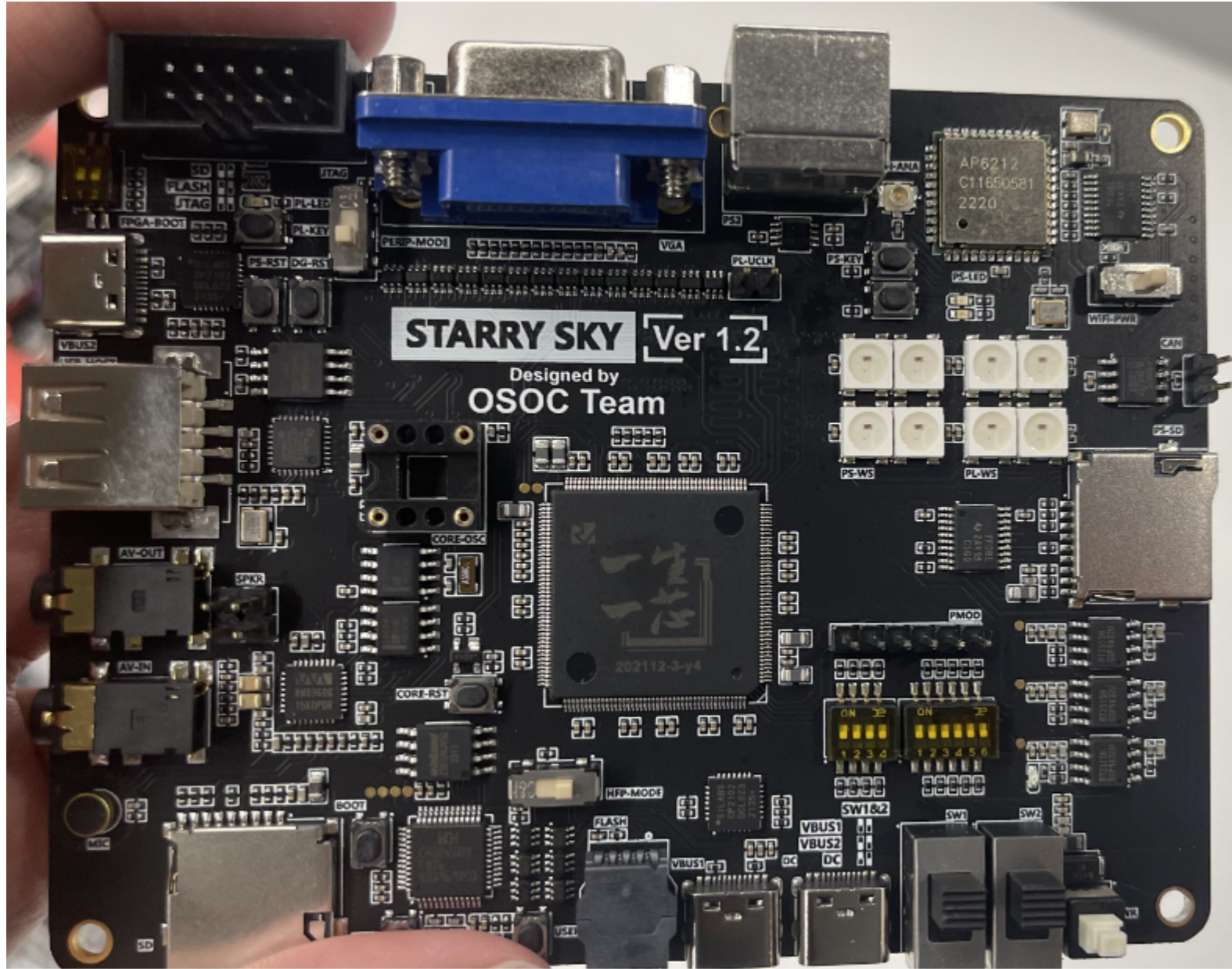
在Linux上运行

字符版2048游戏

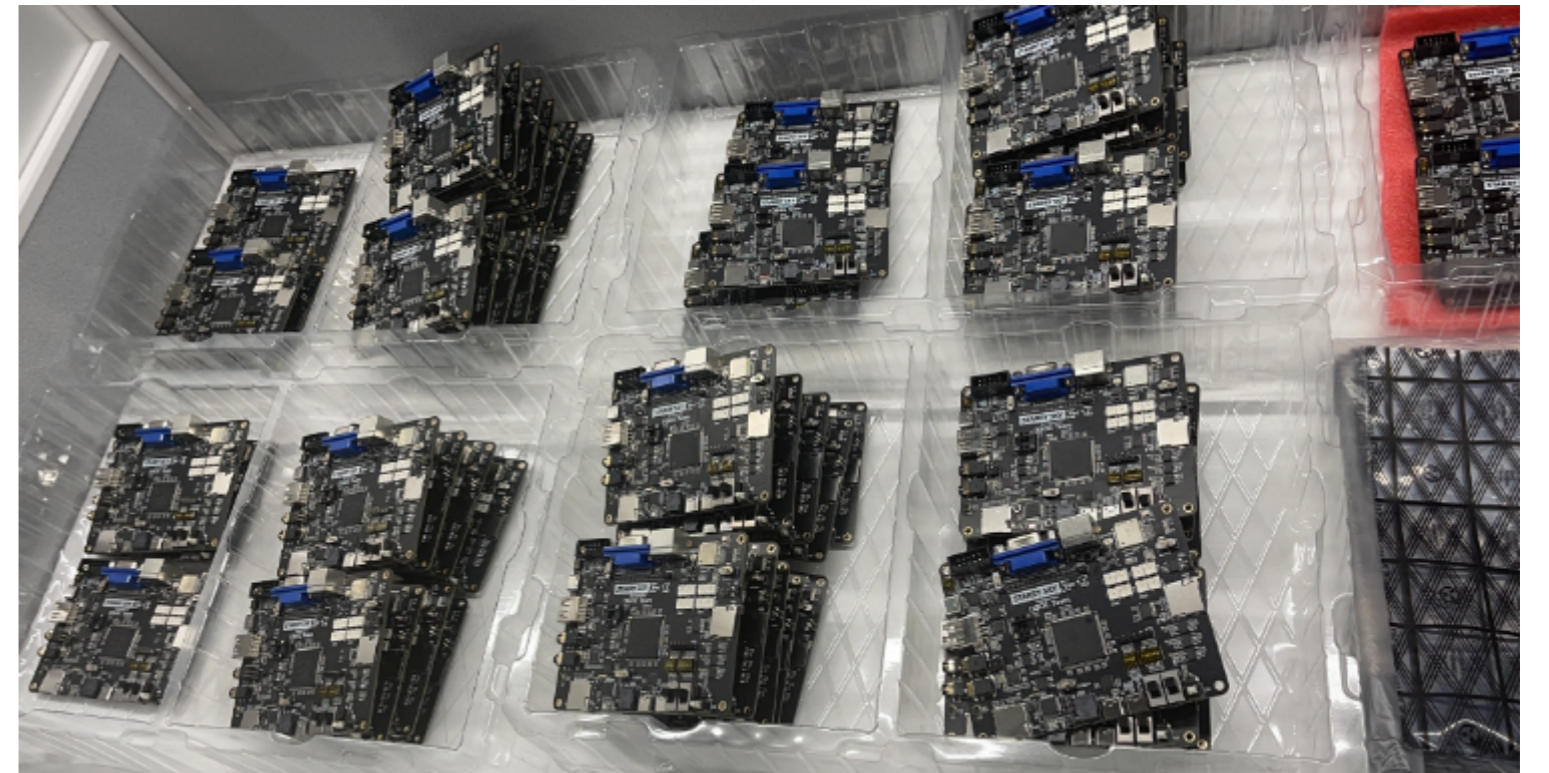


<https://www.bilibili.com/video/BV1CL411X7wV>

# 板卡设计和发放

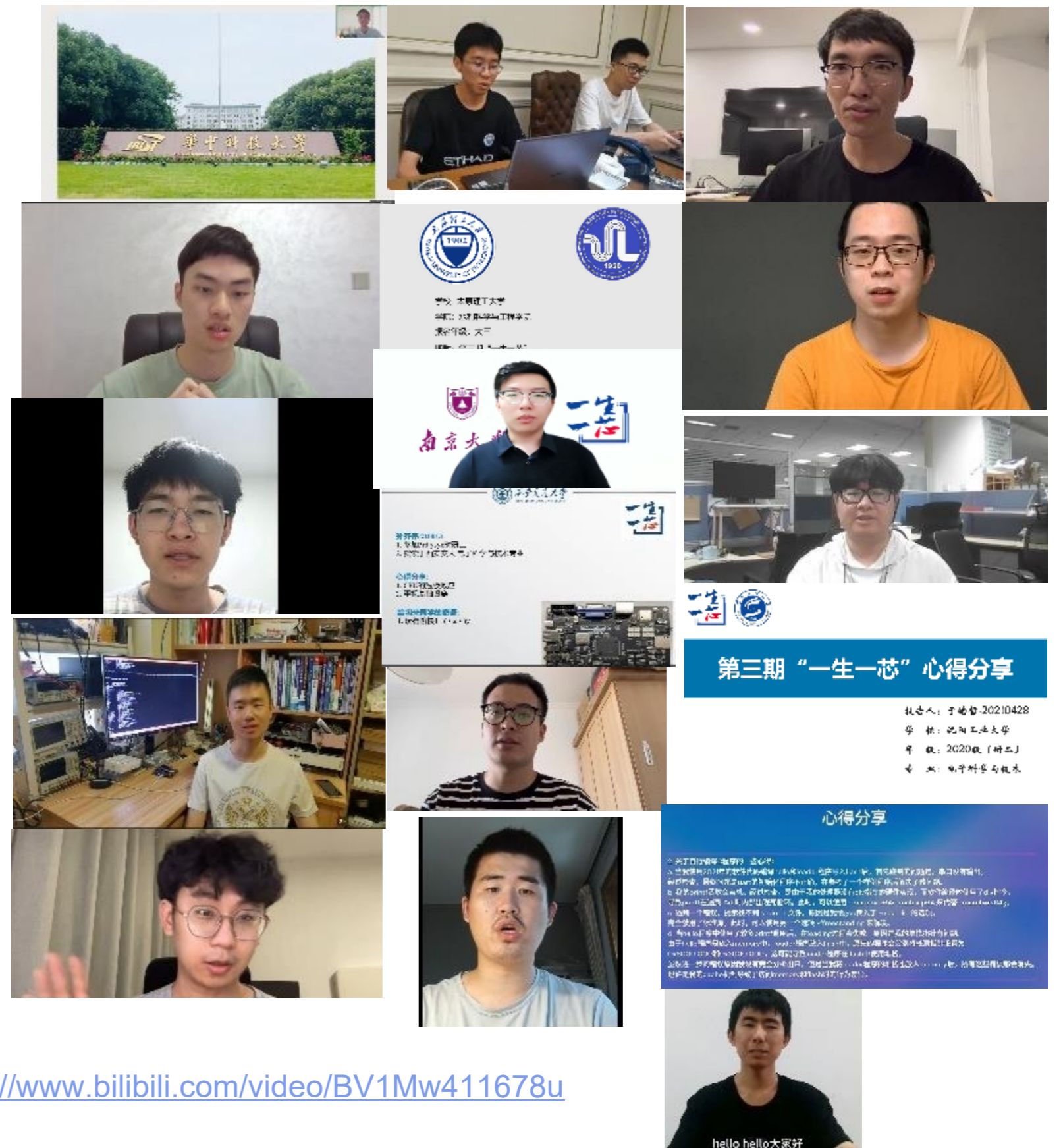
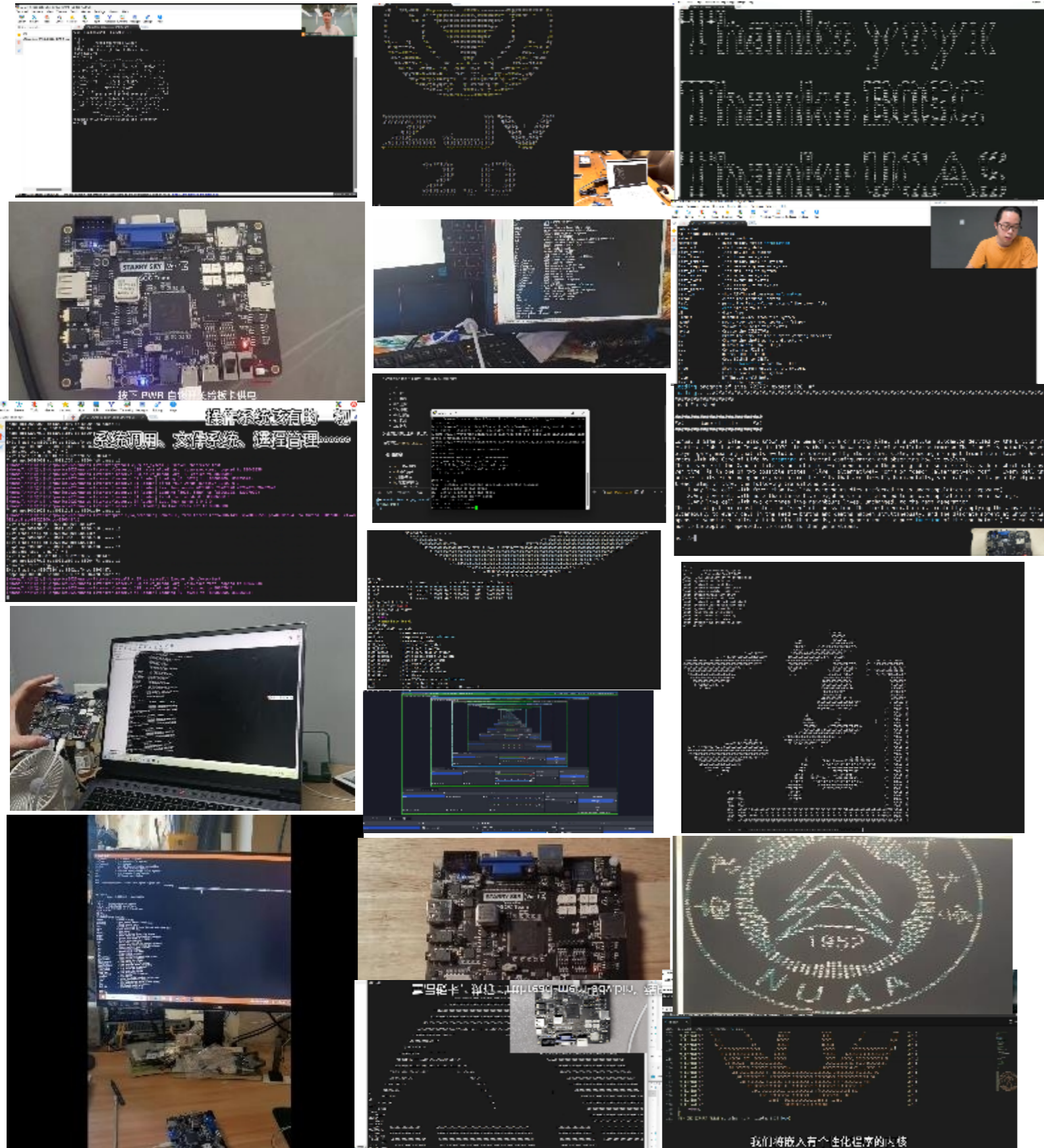


搭载“一生一芯”芯片的“星空”板卡



准备寄出的物料和板卡

# 学生录制的板卡点亮视频和心得分享视频



<https://www.bilibili.com/video/BV1Mw411678u>

# “一生一芯” 高校合作

## 初步合作阶段

## 加强合作阶段

## 深入合作阶段

**方案A：**  
定期出具学习情况报告  
**适用场景：**  
少量学生初步开始学习

**方案B：**  
定期出具报告+教学交流  
**适用场景：**  
有意向展开” 教学合作  
**优势：**  
“一生一芯” 就教学合作提出针对性方案

**方案C：**  
定期出具报告+配备助教+教学交流  
**适用场景：**  
适用于有一定学员规模，成立学习群组的情况  
**优势：**  
专配助教关注群内学员情况并答疑指导

**方案D：**达成“一生一芯” 学分认证  
**适用场景：**高校院系官方认证“一生一芯” 学习内容并赋予学分或者保研加分  
**优势：**将学生的课外实践与校内学分相结合，充分调动学员积极性

**方案E：**成立“一生一芯” 社团  
**适用场景：**高校院系成立正式社团  
**优势：**社团化管理和组织学生，“一生一芯” 项目组将给予一定的活动经费支持。

**方案F：**课程引入“一生一芯” 案例工具  
**适用场景：**高校院系课程可适应性调整  
**优势：**引入“一生一芯” 的案例和工具与课程相结合。

**方案G：**课程引入“一生一芯” 实验  
**适用场景：**高校院系课程可适应性调整  
**优势：**引入“一生一芯” 的实验增强课堂的实践性，培养学生动手能力

**方案H：**开设“一生一芯” 实践课程  
**适用场景：**高校院系纳入课程体系，开设“一生一芯” 实践课程  
**优势：**全面培养学生前后端全链条的知识体系和动手实践能力

**方案I：**开设“一生一芯” 训练营  
**适用场景：**高校院系提供场地和资源支持，“一生一芯” 负责教学组织  
**优势：**集中学生线下授课和学习，学习效率高，周期短，效果显著

**方案J：**开展科研教学项目合作  
**适用场景：**  
高校院系与“一生一芯” 有共同的项目合作方向  
**优势：**双方共同投入资源完成技术攻关和研发工作，项目验证和迭代的经验用于教学方案改革

以上方案逐步深入，亦可同时进行，多维度展开深度合作

# “一生一芯” 火热进行中

## 随到随学，全年开放报名

### 扫码或登录 [ysyx.org](http://ysyx.org) 点击 “立即参与”



官方网站  
[ysyx.org](http://ysyx.org)



“一生一芯” 公众号

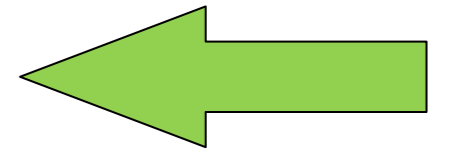


从零开始  
创造属于你的  
**RISC-V®** 处理器

项目介绍

PDF下载

报名参与



计算机系统与处理器  
芯片课程虚拟教研室



北京开源芯片研究院  
BEIJING INSTITUTE OF OPEN SOURCE CHIP



中国开放指令生态 (RISC-V) 联盟  
China RISC-V Alliance



上海处理器技术创新中心  
SHANGHAI INNOVATION CENTER FOR COMPUTING TECHNOLOGY

