

官方网站 ysyx.org 微信公众号

Email: ysyx@bosc.ac.cn

“一生一芯”计划

——设计人生中的第一颗处理器芯片

余子濠

“一生一芯”项目组

2023.08



计算机系统与处理器
芯片课程虚拟教研室



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



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



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

一、“一生一芯”计划初衷

处理器芯片是最为复杂的一类芯片

- 芯片是指内含**集成电路**的硅片，是**半导体产业**的核心
- 芯片有**几十种**大门类，**上千种**小门类
- **处理器芯片**被公认为**芯片产业皇冠上的明珠**，设计复杂度高、难度大

各类芯片



处理器芯片



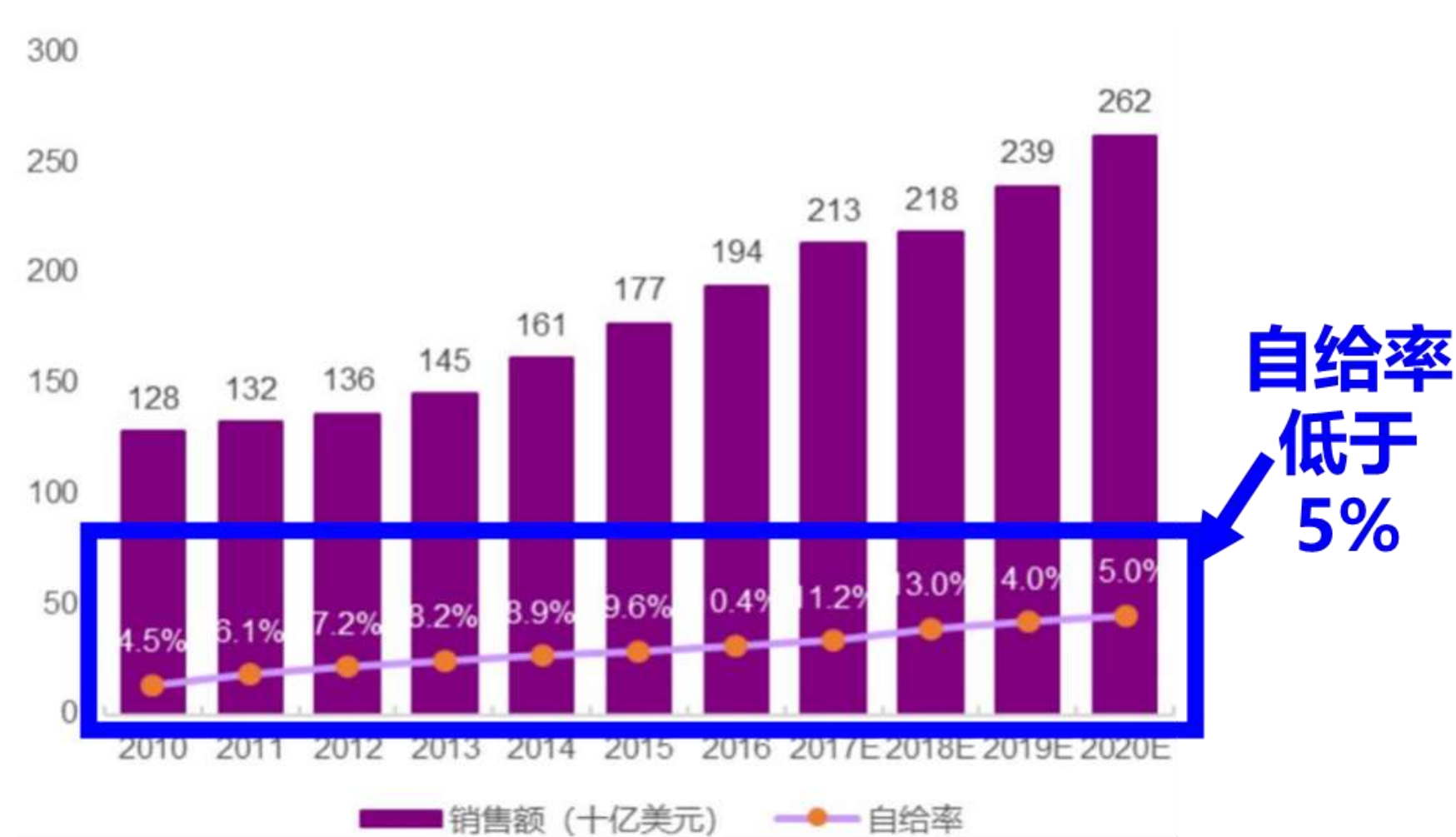
集成电路是中国第一大进口产品（2021年）

- 2021年进口**6355亿**个集成电路，进口额高达**4326亿美元**，远超过石油（**2573亿美元**）、铁矿（**1847亿美元**）
- 处理器与控制器芯片进口额超**2034亿美元**，占比高达**47%**

种类	进口金额 (亿美元)	占比
处理器及控制器	2034	47.0%
存储器	1220	28.2%
放大器	156	3.6%
其他	916	21.2%
总数	4326	100%

国产芯片任重道远

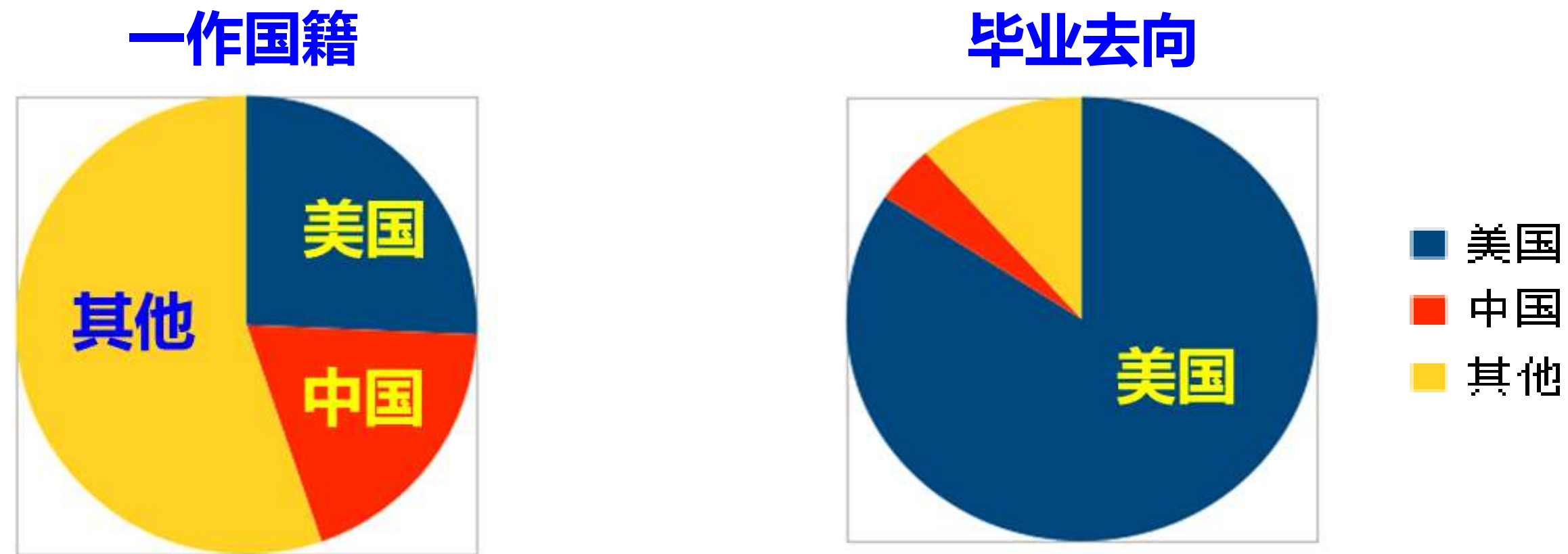
- 在很多领域，国产芯片市场占有率仍不足5%



系统	设备	核心集成电路	国产芯片占有率
计算机系统	服务器	MPU	0%
	个人电脑	MPU	0%
	工业应用	MCU	2%
通用电子系统	可编辑逻辑设备	FPGA/EPLD	0%
	数字信号处理设备	DSP	0%
通信装备	移动通信终端	Application Processor	18%
		Communication Processor	22%
		Embedded MPU	0%
		Embedded DSP	0%
	核心网络设备	NPU	15%
存储设备	半导体存储器	DRAM	0%
		NAND Flash	0%
		NOR Flash	5%
显示及视频系统	高清电视/智能电视	Image Processor	5%
		Display Driver	0%

我国优秀处理器芯片人才储备严重不足

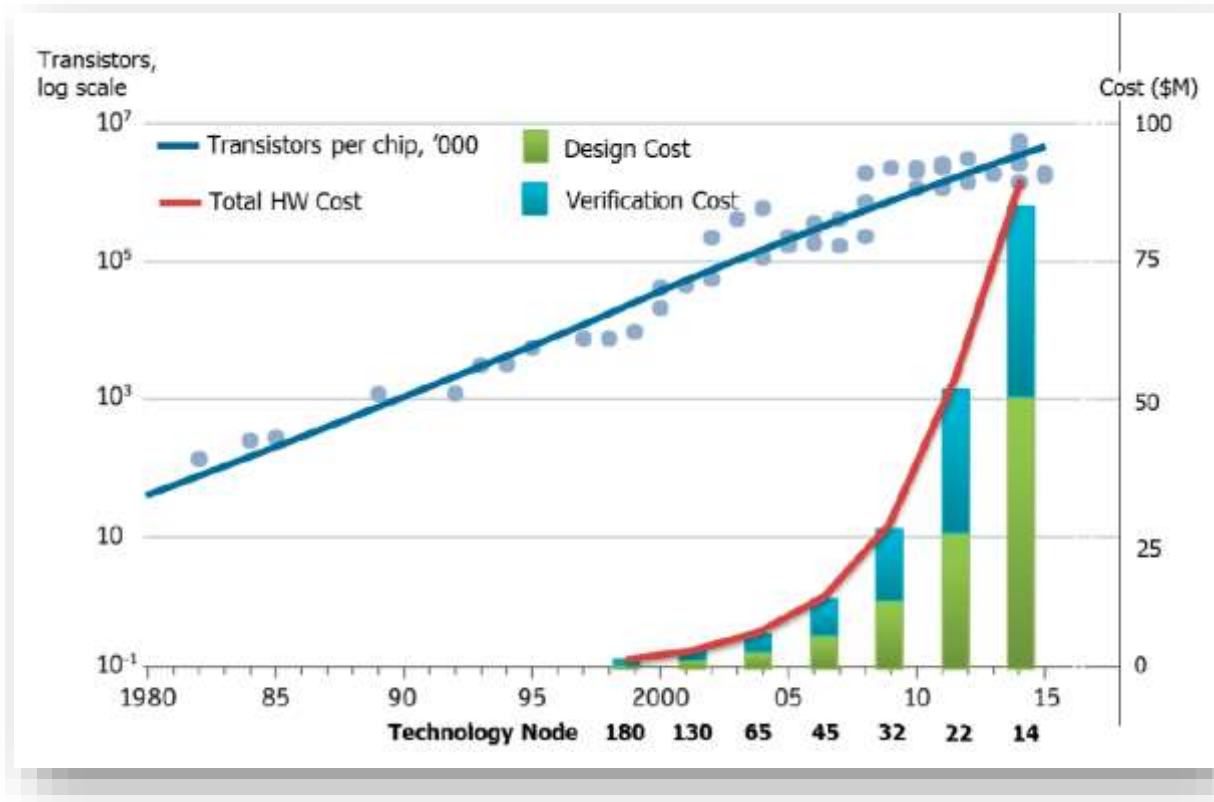
- **85% v.s. 4%**: 2008-2017十年体系结构国际顶级会议ISCA论文的第一作者**85%**在美国, 仅有**4%**在中国, 不足美国的**1/20**, 差距巨大
- 中国**加快**处理器芯片人才**培养规模与速度**, 迫在眉睫



ISCA十年论文第一作者统计情况(2008-2017)

处理器芯片设计人才培养面临的挑战

芯片设计门槛高



Source: Andreas Olofsson, *Intelligent Design of Electronic Assets (IDEA)*, 2017

传统专业/课程间关联弱



Image Source: <https://dzone.com/articles/how-does-the-asic-design-flow-cycle-works>

学习坡度高

应用程序

运行时环境

操作系统

指令集

微结构

电路

综合

物理设计

物理验证

版图

写RTL

学生
预期

实际是个
系统工程

尚缺少类似临床实习环节



“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



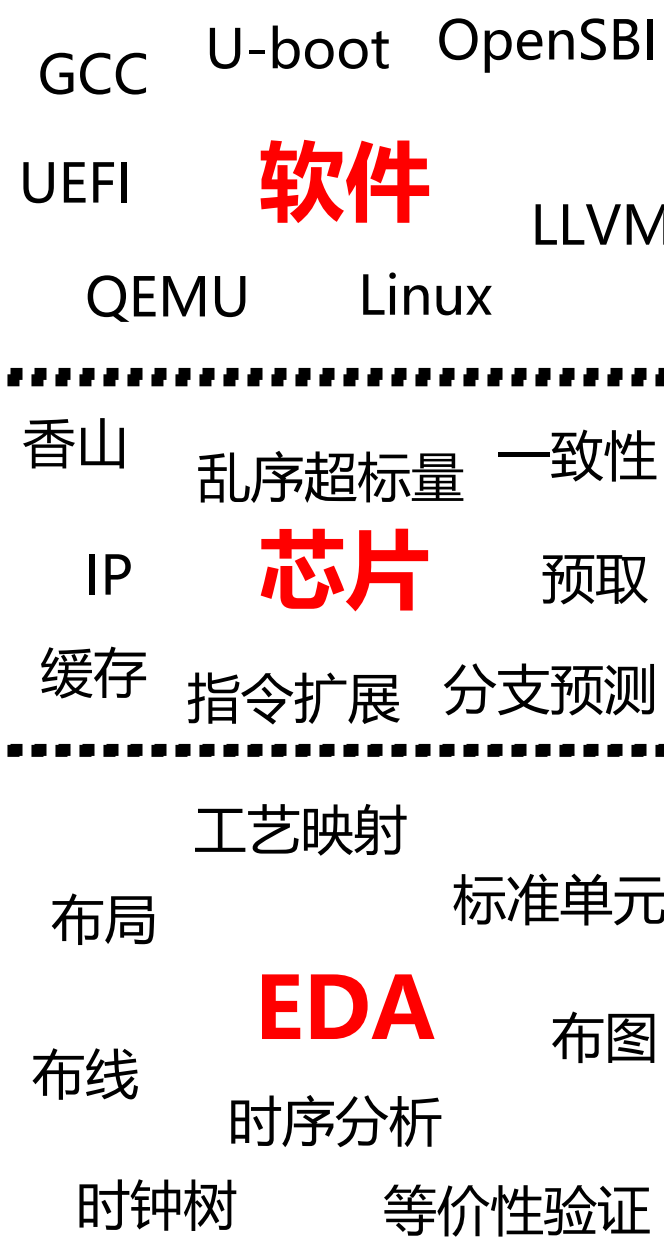
熟悉

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”计划执行情况

- 自2019年起已历六期，**循序渐进，逐步放大规模，保持教学质量**
- 数据统计时间: 2023年8月19日

← 感兴趣

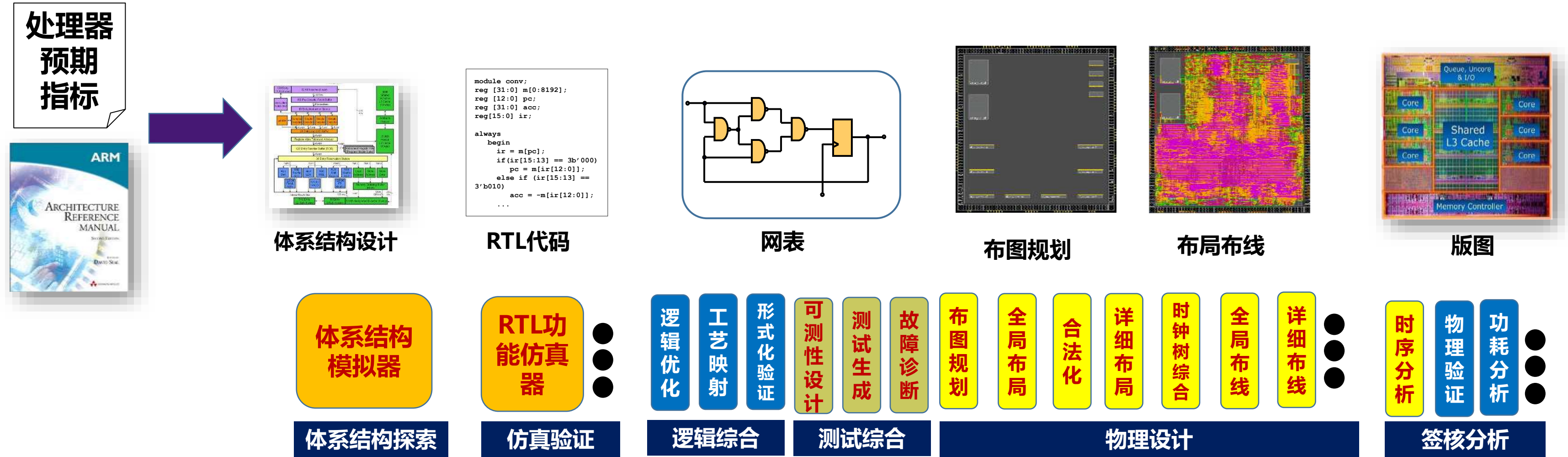
← 投入学习

期数	启动时间	报名结束时间	报名人数	覆盖高校	持续学习人数	达成流片指标人数
一	2019年8月	-	5	1	5	5
二	2020年8月	-	11	5	11	11
三	2021年7月	2021年9月	760	168	215	51
四	2022年2月	2022年8月	1753	328	215	18(学习指标提升)
五	2022年8月	2023年7月	1881	379	155	10(持续增加中)
六	2023年7月	未结束	821	190	37	未进行至流片阶段
七	预计2024年1月启动					

二、处理器芯片设计流程

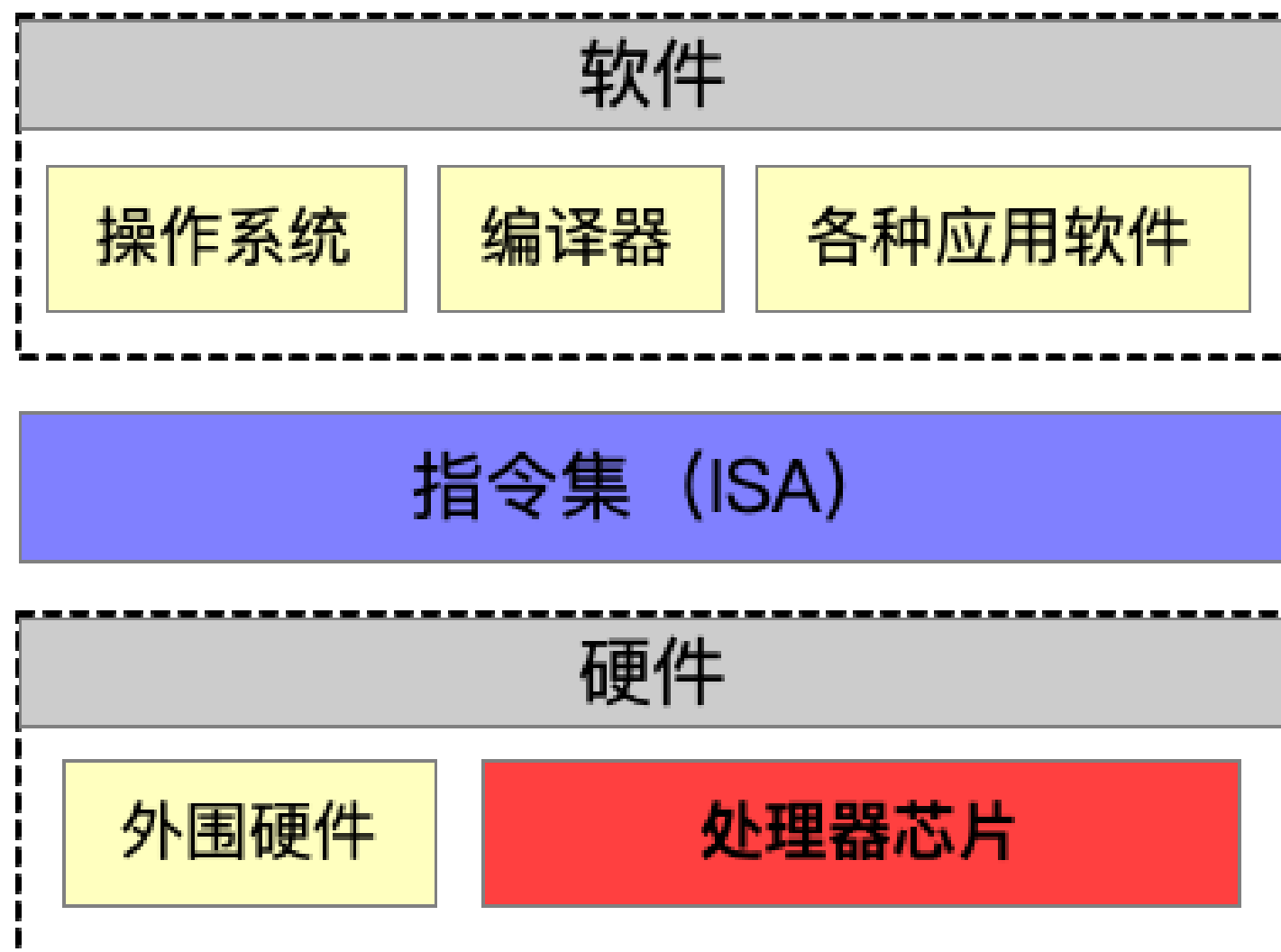
处理器芯片设计流程

● 从指令集到流片版图



指令集 (ISA) 是一种规范标准

- 指令集架构 (Instruction Set Architecture), 简称指令集
- 计算机系统中硬件与软件之间交互的规范标准
- RISC-V是一种指令集



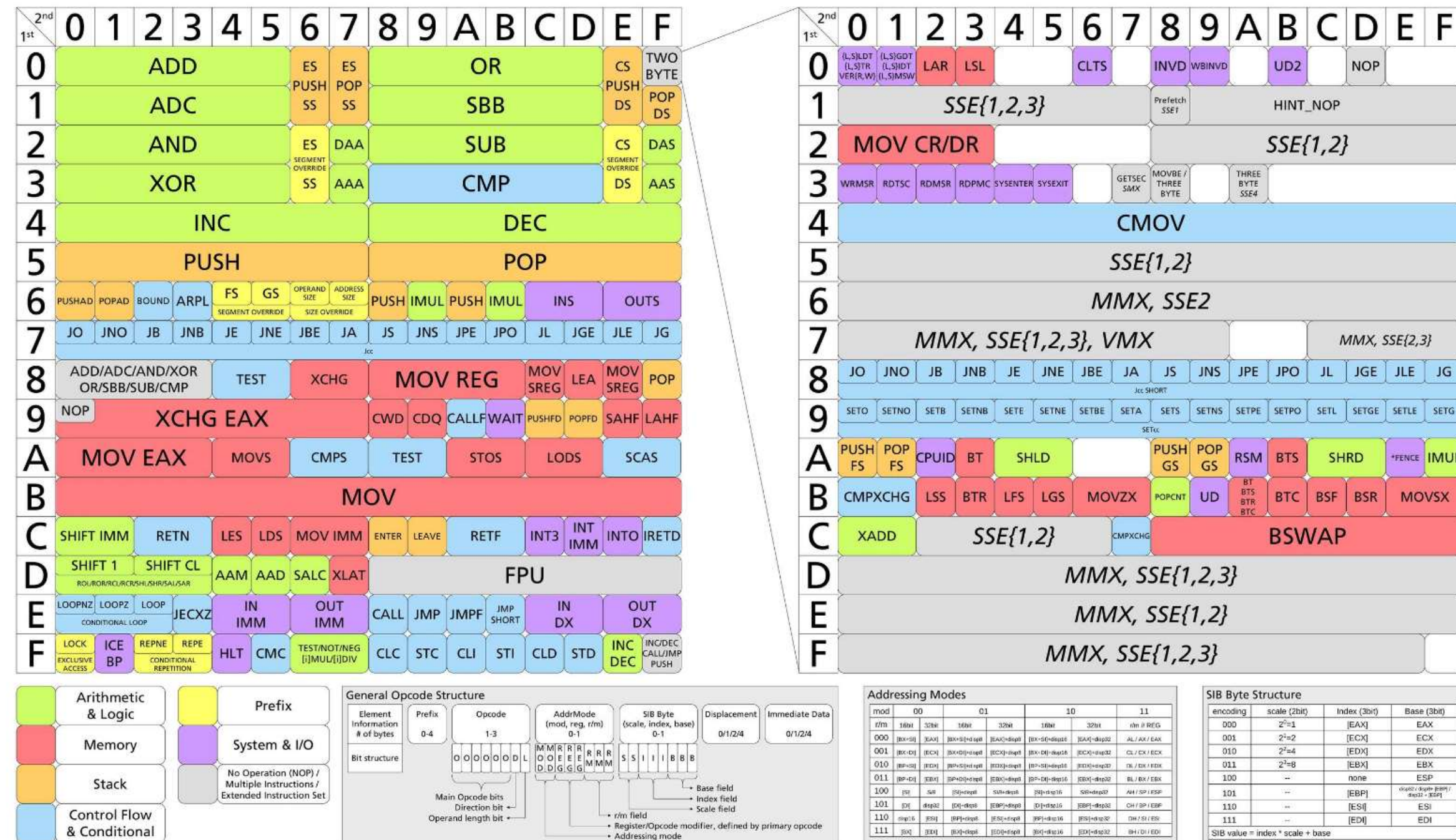
指令集可类比于螺母和螺钉的尺寸规范



指令集 → 微架构 → 版图 → 芯片产品

- 指令集是规范，定义指令的格式、功能

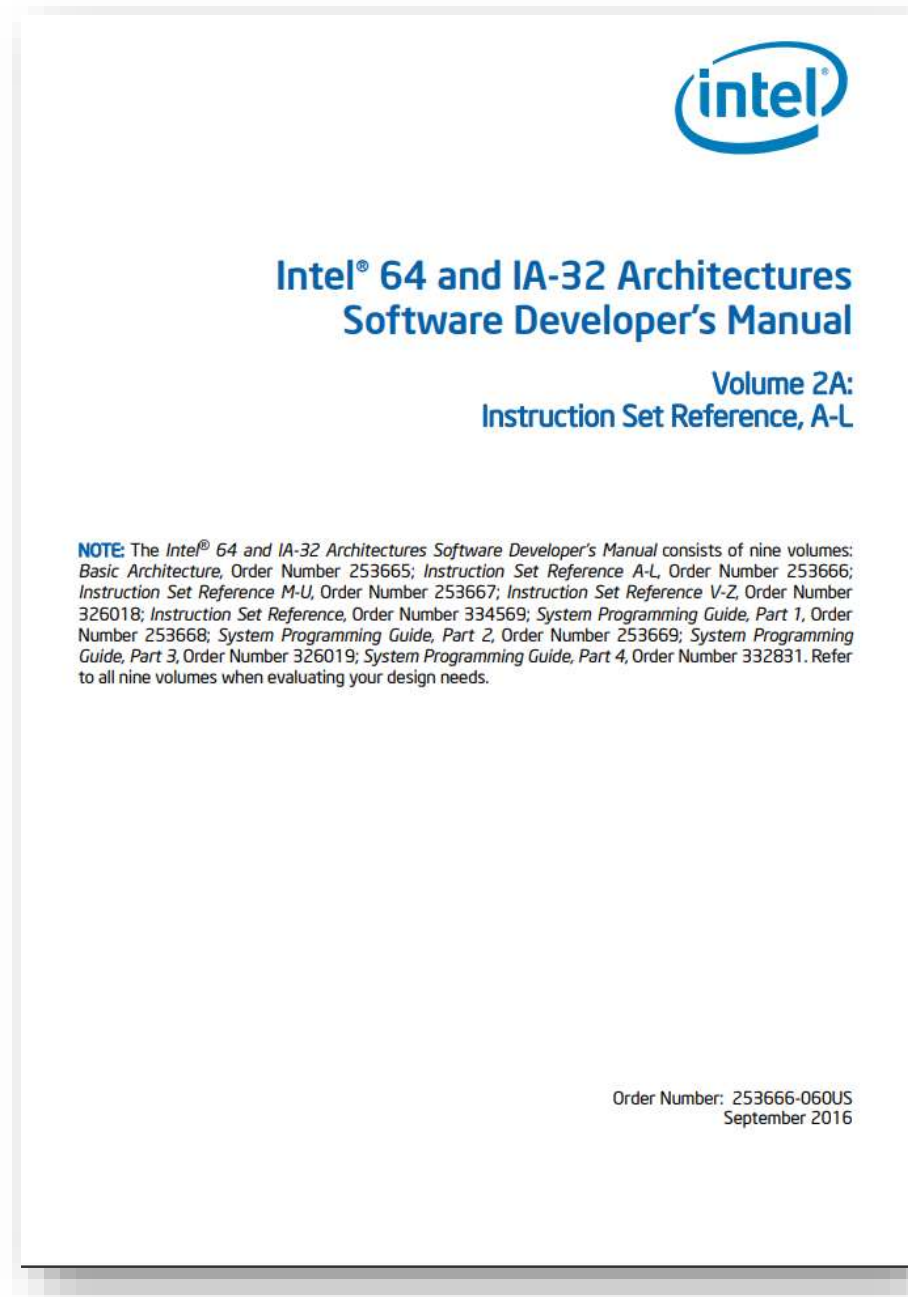
x86 Opcode Structure and Instruction Overview



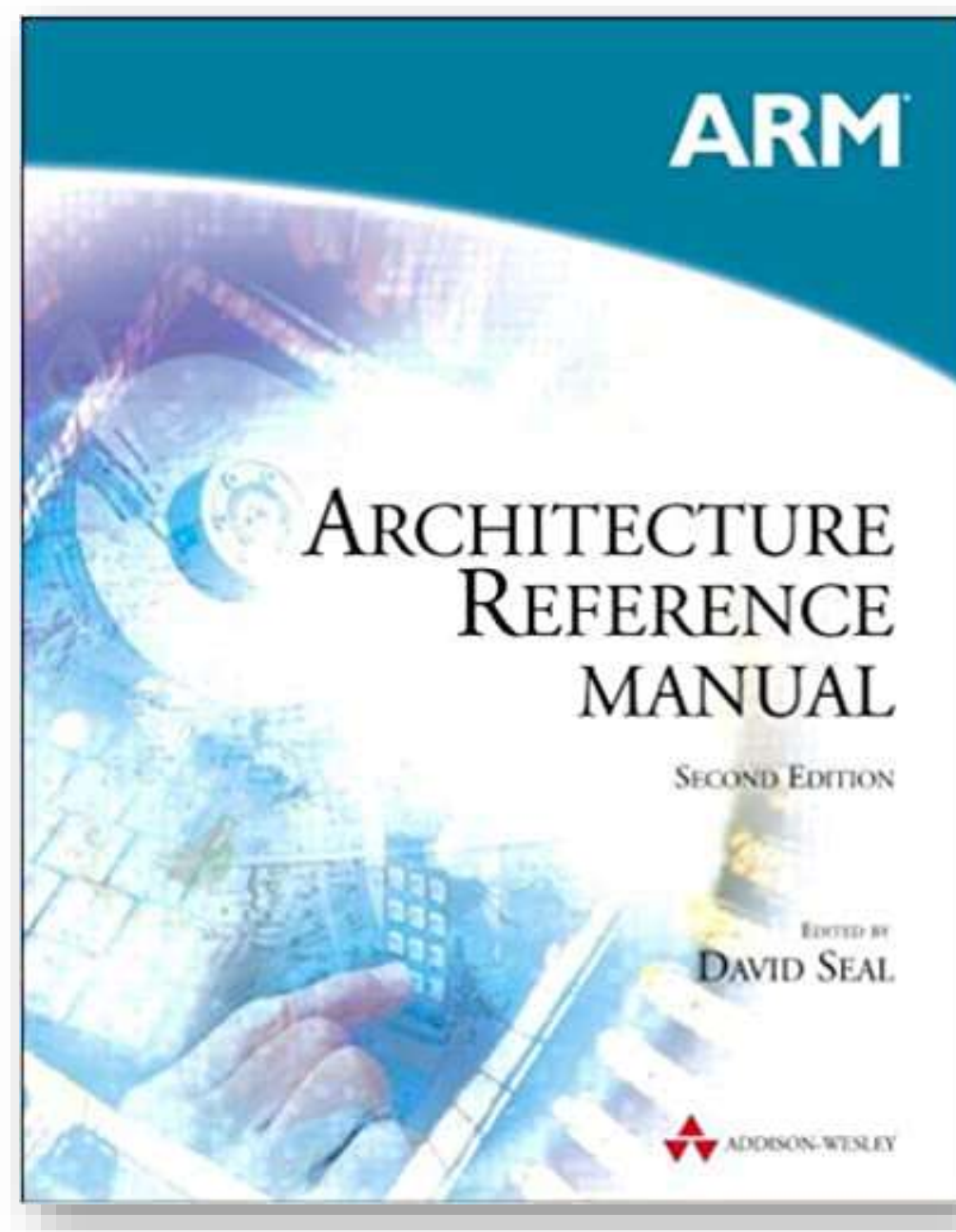
指令集 → 微架构 → 版图 → 芯片产品

- 指令集规范记录于**手册(Manual)**中

X86手册：约2200页



ARM手册：约2700页

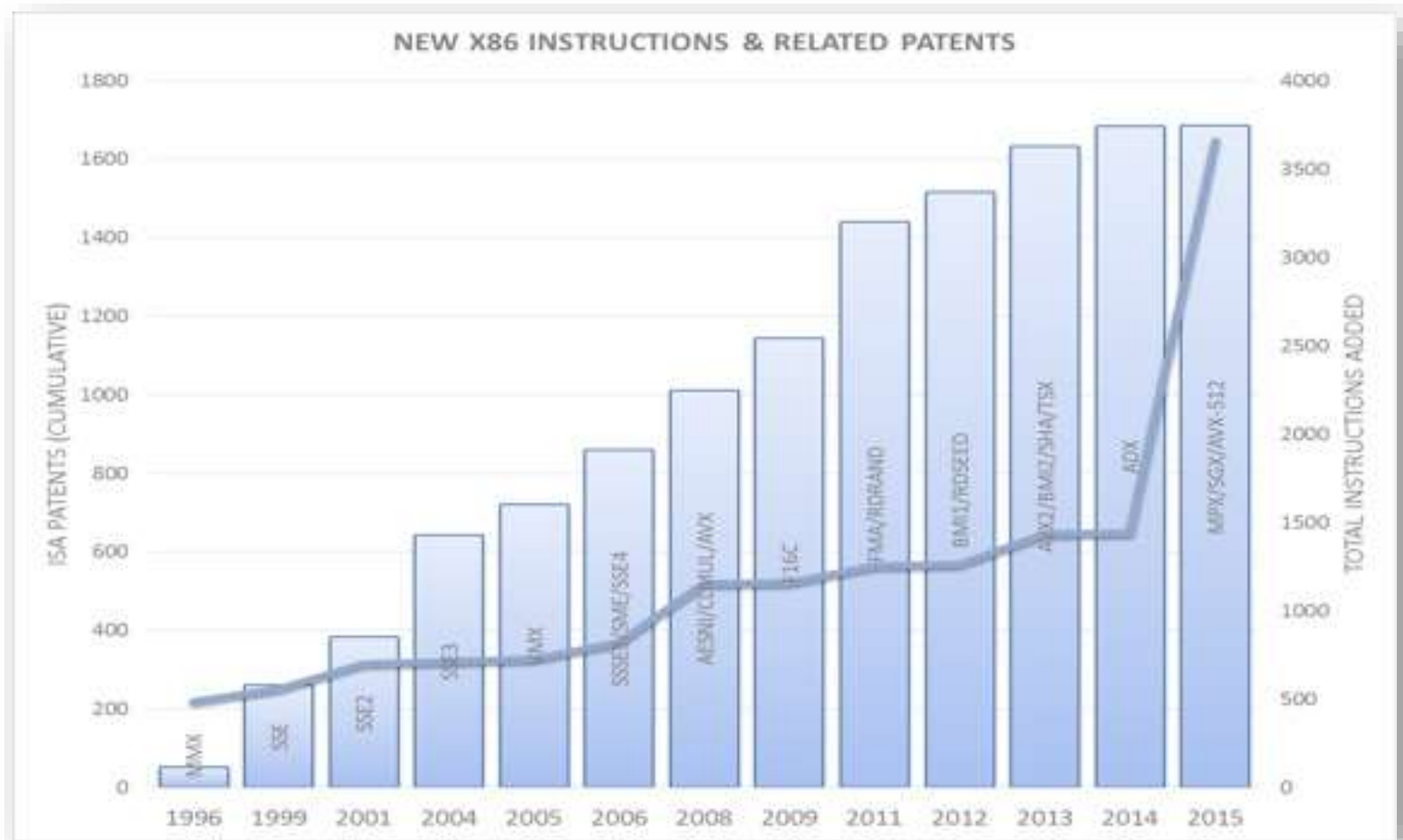


RISC-V手册：约200页



RISC-V v.s. X86/ARM: 指令更精巧

- X86从1978年的80条指令，增长到2015年的**3600条**
- RISC-V基础部分 (RV32I) 只有**47条**指令，并已冻结



RISC-V ①				RISC-V Reference Card ④			
Base Integer Instructions (32/64/128)				RV Privileged Instructions (32/64/128) ②			
Category	Name	Fmt	RV(32/64/128) Base	Category	Name	Fmt	RV(32/64/128) Priv
Loads	Load Byte	I	LB rd, rs1, imm	CSR Access	Atomic R/W	R	CSRWR rd, csr, rs1
	Load Halfword	I	LH rd, rs1, imm		Atomic Read & Set Bit	R	CSRRS rd, csr, rs1
	Load Word	I	LD rd, rs1, imm		Atomic Read & Clear Bit	R	CSRRC rd, csr, rs1
	Load Byte Unsigned	I	LBU rd, rs1, imm		Atomic R/W Imm	R	CSRWRI rd, csr, rs1
	Load Half Unsigned	I	LHU rd, rs1, imm		Atomic Read & Set Bit Imm	R	CSRRSI rd, csr, imm
Stores	Store Byte	S	SB rs1, rs2, imm	Change Level	Env. Call	R	SCALL
	Store Halfword	S	SH rs1, rs2, imm		Environment Breakpoint	R	EBREAK
	Store Word	S	SW rs1, rs2, imm		Environment Return	R	ERET
Shifts	Shift Left	R	SLL rd, rs1, rs2	Trap Redirect to Supervisor	Redirect Trap to Hypervisor	R	HTH
	Shift Left Immediate	I	SLLI rd, rs1, shamt		Hypervisor Trap to Supervisor	R	HTS
	Shift Right	R	SRL rd, rs1, rs2	Interrupt	Wait for Interrupt	R	MW
	Shift Right Immediate	I	SRLI rd, rs1, shamt		Supervisor FENCE	R	BFENCE, SF
	Shift Right Arithmetic	R	SRA rd, rs1, rs2				
Arithmetic	ADD	R	ADD rd, rs1, rs2	Optional Multiply-Divide Extension: RV32M			
	ADD Immediate	I	ADDI rd, rs1, imm	Multiply	MUL	R	MUL rd, rs1, rs2
	SUBTRACT	R	SUB rd, rs1, rs2		MULH	R	MULH rd, rs1, rs2
	Load Upper Imm to PC	U	LDUP rd, imm		MULHSB	R	MULHSB rd, rs1, rs2
					MULHSU	R	MULHSU rd, rs1, rs2
Logical	XOR	R	XOR rd, rs1, rs2	Divide	DIV	R	DIV rd, rs1, rs2
	XOR Immediate	I	XORI rd, rs1, imm		DIVU	R	DIVU rd, rs1, rs2
	OR	R	OR rd, rs1, rs2	Remainder	REM	R	REM rd, rs1, rs2
	OR Immediate	I	ORI rd, rs1, imm		REMU	R	REMU rd, rs1, rs2
	AND	R	AND rd, rs1, rs2	Optional Atomic Instruction Extension: RVA			
Compare	Set <	R	SLT rd, rs1, rs2	Load	Load Reserved	R	LR rd, rs1
	Set < Immediate	I	SLTI rd, rs1, imm		Store Store Conditional	R	SC rd, rs1, rs2
	Set < Unsigned	R	SLTU rd, rs1, rs2	Swap	SWAP	R	SWAP rd, rs1, rs2
	Set < Imm Unsigned	I	SLTIU rd, rs1, imm		ANDSWAP	R	ANDSWAP rd, rs1, rs2
				Add	ADD	R	ADD rd, rs1, rs2
Branches	Branch =	SB	BEQ rs1, rs2, imm		Logical	R	ANDXOR rd, rs1, rs2
	Branch <	SB	BNE rs1, rs2, imm		AND	R	AND rd, rs1, rs2
	Branch < Immediate	SB	BLT rs1, rs2, imm		ANDOR	R	ANDOR rd, rs1, rs2
	Branch < Unsigned	SB	BGE rs1, rs2, imm		Min/Max	R	AMIN rd, rs1, rs2
	Branch < Imm Unsigned	SB	BLTU rs1, rs2, imm		Maximum	R	AMAX rd, rs1, rs2
Jump & Link	Jump & Link Register	UJ	JAL rd, imm	Min/Max	Minimum Unsigned	R	AMINU rd, rs1, rs2
	Synch thread	I	FENCE		Maximum Unsigned	R	AMAXU rd, rs1, rs2
	System Break	I	SBREAK	16-bit (RVC) and 32-bit Instruction Formats			
	System CALL	I	SCALL	CI	16-bit	16-bit	16-bit
	Counters	Rd	RD		32-bit	32-bit	32-bit
System	Read CYCLE upper Half	Rd	RD		64-bit	64-bit	64-bit
	Read TIME upper Half	Rd	RD		128-bit	128-bit	128-bit
	Read INSTR RETired	Rd	RD		256-bit	256-bit	256-bit
	Read INSTR upper Half	Rd	RD		512-bit	512-bit	512-bit
					1024-bit	1024-bit	1024-bit

RISC-V v.s. X86/ARM：模块化

- 实现一个X86/ARM处理器，需实现**所有上千条**指令，复杂度极高

- RISC-V指令集采用**模块化**设计

- 必要的RV32I只有47条指令
- 其余指令可选扩展
- 可根据需求**自由组合，灵活适配**
 - 嵌入式(关注处理器成本) – RV32I
 - 嵌入式(关注存储器成本) – RV32IC
 - 教学 – RV32IMA
 - 桌面 – RV64GC
 - 高性能 – RV64GCV
- 支持自定义指令

Base Integer Instructions: RV32I and RV64I				Optional Multiply-Divide Instruction Extension: RV64M				Optional Vector Extension: RVV			
Category	Name	Fmt	RV32I Base	Category	Name	Fmt	RV64M (Multiply-Divide)	Category	Name	Fmt	RV32V/RV64V
Shifts	Shift Left Logical	R	SLIL rd,rs1,rs2	Multiply	Multiply High	R	MULH rd,rs1,rs2	SET Vector Len.	SETVL	R	SETVL rd,rs1
Shifts	Shift Left Logical Imm.	I	SLIL rd,rs1,shamt	Multiply High Sign/Uns.	MULHSU	R	MULHSU rd,rs1,rs2	Multiply High REHinder	VREH	R	VREH rd,rs1,rs2
Shifts	Shift Right Logical	R	SRL rd,rs1,rs2	Multiply High Uns.	MULHU	R	MULHU rd,rs1,rs2	Shift Left Log.	VSL	R	VSL rd,rs1,rs2
Shifts	Shift Right Logical Imm.	I	SRLI rd,rs1,shamt	Divide	Divide	R	DIV rd,rs1,rs2	Shift Right Log.	VSRL	R	VSRL rd,rs1,rs2
Shifts	Shift Right Arithmetic	R	SRA rd,rs1,rs2	Divide Unsigned	DIVU	R	DIVU rd,rs1,rs2	Shift R. Arith.	VSRA	R	VSRA rd,rs1,rs2
Shifts	Shift Right Arithmetic Imm.	I	SRAI rd,rs1,shamt	Remainder	Remainder	R	REM rd,rs1,rs2	Load Strided	VLD	I	VLD rd,rs1,imm
Arithmetic	ADD	R	ADD rd,rs1,rs2	Remainder Unsigned	REMU	R	REMU rd,rs1,rs2	Load Strided	VLDL	R	VLDL rd,rs1,rs2
Arithmetic	ADD Immediate	I	ADDI rd,rs1,imm					Load Indexed	VLDLX	R	VLDLX rd,rs1,rs2
Arithmetic	SUBtract	R	SUB rd,rs1,rs2					Store	VST	R	VST rd,rs1,imm
Arithmetic	SUBtract Immediate	I	SUBI rd,rs1,imm					Store Strided	VSTL	R	VSTL rd,rs1,rs2
Arithmetic	Load Upper Imm	I	LUI rd,imm					Store Indexed	VSTX	R	VSTX rd,rs1,rs2
Arithmetic	Add Upper Imm to PC	I	AUIPC rd,imm					AMO SWAP	AMOSWAP	R	AMOSWAP rd,rs1,rs2
Logical	XOR	R	XOR rd,rs1,rs2					AMO ADD	AMOADD	R	AMOADD rd,rs1,rs2
Logical	XOR Immediate	I	XORI rd,rs1,imm					AMO XOR	AMOXOR	R	AMOXOR rd,rs1,rs2
Logical	OR	R	OR rd,rs1,rs2					AMO AND	AMOAND	R	AMOAND rd,rs1,rs2
Logical	OR Immediate	I	ORI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Logical	AND	R	AND rd,rs1,rs2					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Logical	AND Immediate	I	ANDI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Compare	Set <	R	SLT rd,rs1,rs2					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Compare	Set < Immediate	I	SLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Compare	Set < Immediate	I	SLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Compare	Set < Immediate	I	SLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Branches	Branch <	R	BLT rd,rs1,rs2,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Branches	Branch < Immediate	I	BLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Branches	Branch < Immediate	I	BLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Branches	Branch < Immediate	I	BLTI rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Jump & Link	JAL	R	JAL rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Jump & Link	Jump & Link Register	I	JALR rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Synch	Synch thread	I	FENCE					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Synch	Synch Instr. & Data	I	FENCE.I					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Environment	CALL	I	ECALL					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Environment	BREAK	I	EBREAK					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read/Write	I	CSRRW rd,csr,rs1					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read & Set Bit	I	CSRRS rd,csr,rs1					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read & Clear Bit	I	CSRRC rd,csr,rs1					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read/Write Imm	I	CSRRWI rd,csr,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read & Set Bit Imm	I	CSRRSI rd,csr,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Control Status Register (CSR)	Read & Clear Bit Imm	I	CSRRCI rd,csr,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Loads	Load Byte	I	LB rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Loads	Load Halfword	I	LH rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Loads	Load Byte Unsigned	I	LBU rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Loads	Load Halfword Unsigned	I	LHU rd,rs1,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Stores	Store Byte	S	SB rs1,rs2,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Stores	Store Halfword	S	SH rs1,rs2,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2
Stores	Store Word	S	SW rs1,rs2,imm					AMO AND OR	AMOANDOR	R	AMOANDOR rd,rs1,rs2

RISC-V常用的可选扩展，可适配从嵌入式到高性能的各种场景

指令集 → 微架构 → 版图 → 芯片产品

- 微架构 (microarchitecture)设计，就是将**指令集**手册定义的功能**实例化**，然后通过工程开发，变成**源代码**

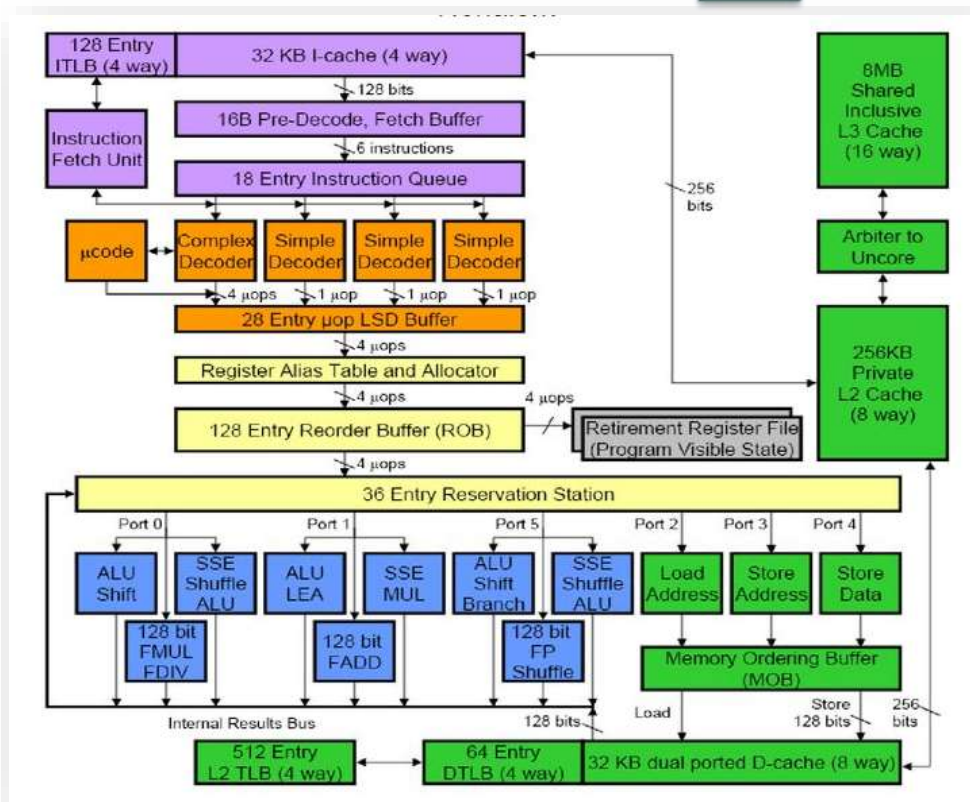
微架构设计

工程开发

RTL代码

EDA工具

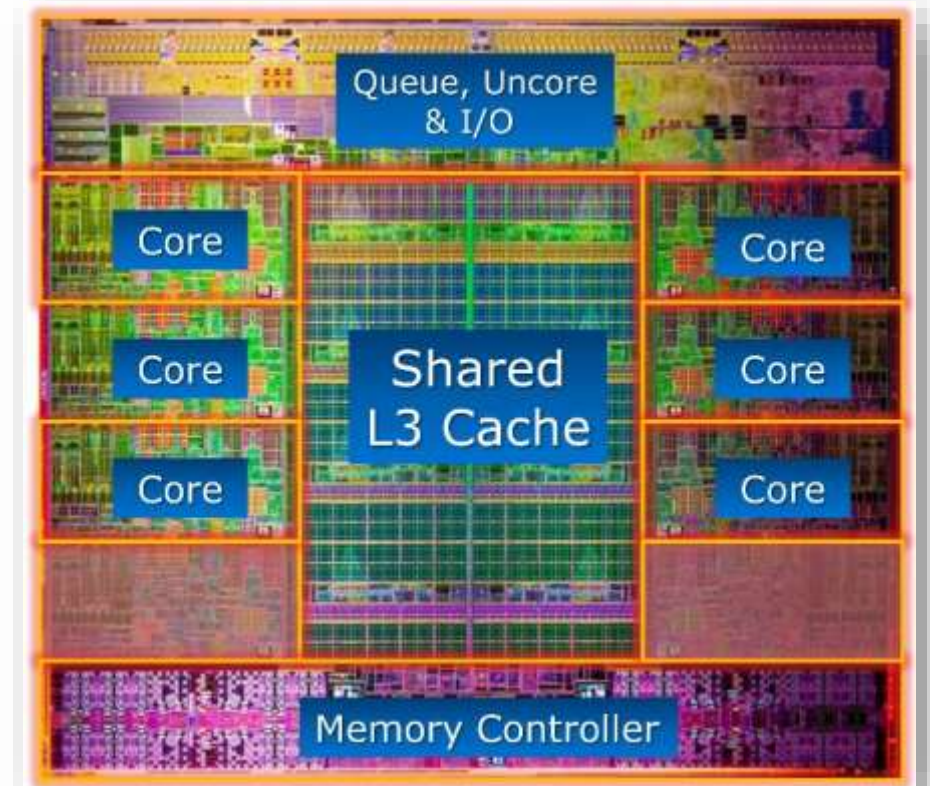
芯片版图



```
component DebugCoreTop is
port (
  -- Trigger and Data
  cu_Clk      : in    std_logic_vector(2 downto 0) := (others => '0');
  cu0_Trig    : in    t_trig_0 := (others => (others => '0'));
  cu1_Trig    : in    t_trig_1 := (others => (others => '0'));
  cu2_Trig    : in    t_trig_2 := (others => (others => '0'));
  cu0_Data    : in    t_data_0 := (others => (others => '0'));
  cu1_Data    : in    t_data_1 := (others => (others => '0'));
  cu2_Data    : in    t_data_2 := (others => (others => '0'));

  -- Downstream I2C
  SCL         : in    std_logic := '0';
  SDA         : inout std_logic := '0';

  -- Upstream
  gt_RefClk_p : in    std_logic := '0';
  gt_RefClk_n : in    std_logic := '0';
  gt_RX_p     : in    std_logic_vector(2 downto 0) := (others => '0');
  gt_RX_n     : in    std_logic_vector(2 downto 0) := (others => '0');
  gt_TX_p     : out   std_logic_vector(2 downto 0);
  gt_TX_n     : out   std_logic_vector(2 downto 0);
);
end component;
```



RISC-V v.s. X86/ARM: 开放免费

指令集与微架构均有
三种**知识产权**模式

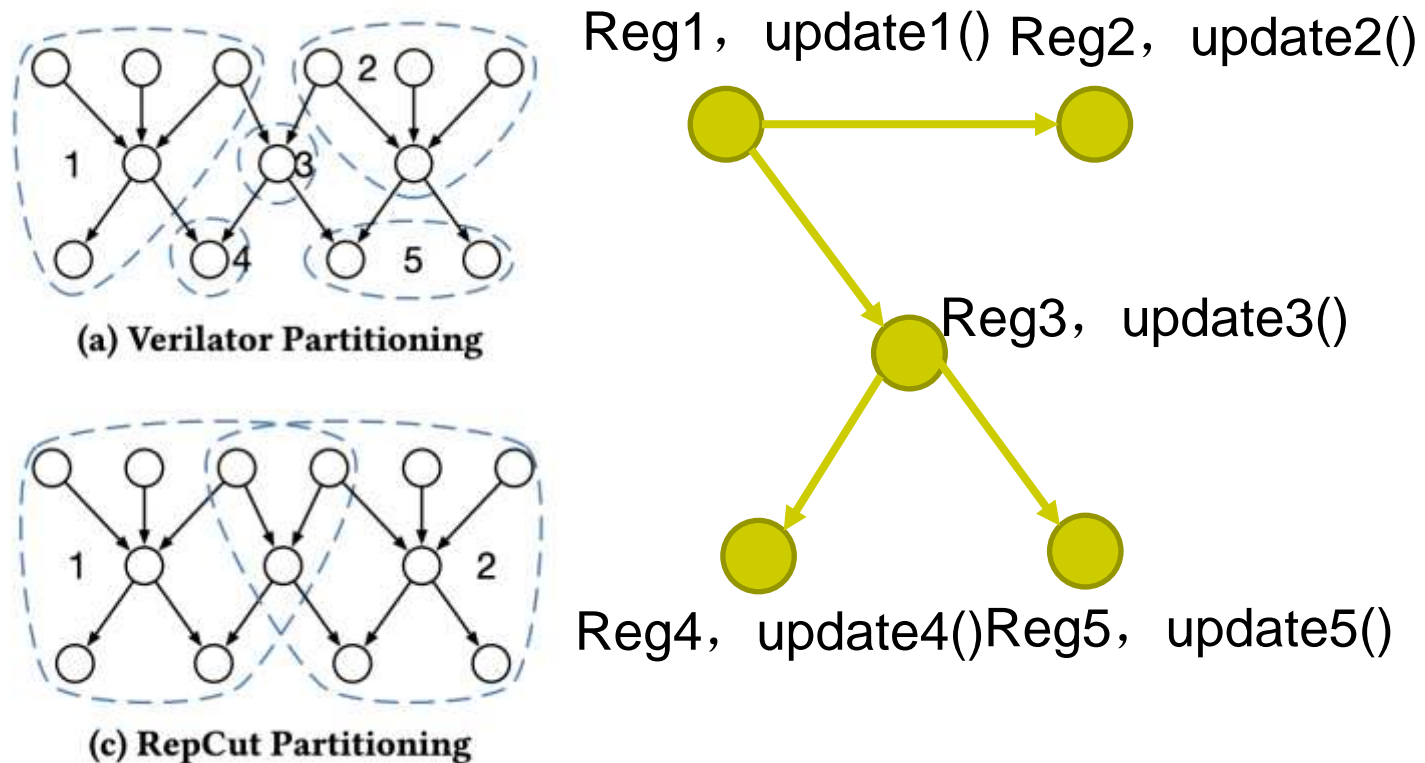
- 开放免费 (Open & Free)
- 可授权 (Licensable)
- 封闭 (Closed)

微架构设计 指令集	1 开放免费的设计	2 需授权的设计	3 封闭的设计	产品可选的设计 (对应各指令集)
开放免费的指令集 (RISC-V)	Berkeley的Rocket Chip/剑桥lowRISC/ 芯来科技蜂鸟E203	平头哥/SiFive/晶 心科技Andes的 RISC-V处理器核	Google和 NVIDIA的自研 RISC-V处理器	1 2 3
需授权的指令集 (ARM)		ARM的处理器设计, 如Cortex-A76等	基于ARM架构的 Apple处理器	2 3
封闭的指令集 (x86)			Intel和AMD的 处理器	3

指令集 → 微架构 → 版图 → 芯片产品

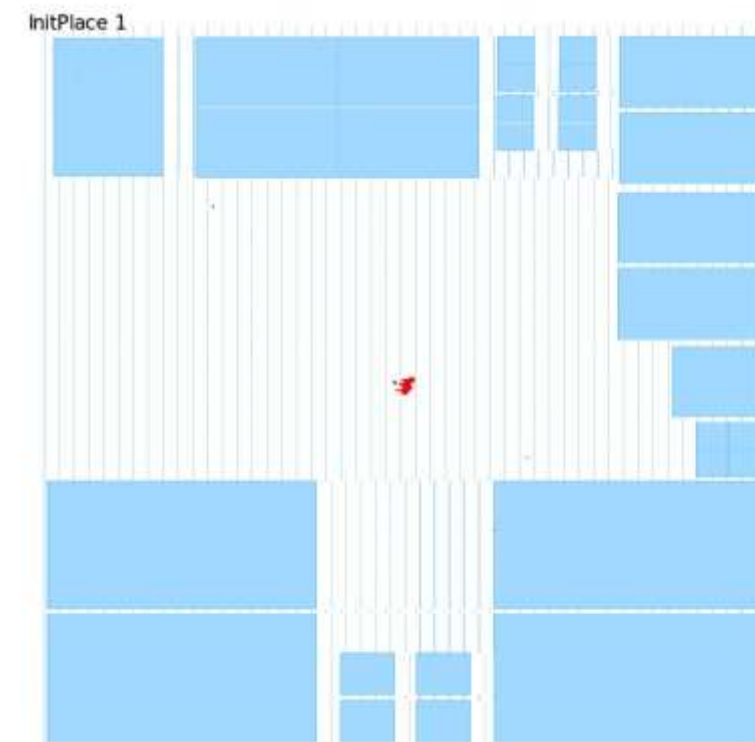
● 通过EDA工具将源代码生成版图

- **仿真验证**: 在一个超大的电路图 (Graph) 中, 进行值传播, 每个cycle 更新一次Reg值
- **综合**: 把一个亿门级电路, **转化为与非门等布尔逻辑表达**, 实现**门数最少或面积时序最优**等目标
- **布局**: 在**不到1cm²的平面空间**内, 摆放**数亿个长方形**单元 (需同时考虑其拓扑连接关系等)
- **时钟树综合**: 构建一颗时钟树, 从单一时钟源出发, 使其**到达数亿单元的时延等长**
- **布线**: 在多层绕线空间内, 解决**数亿单元的拓扑连线**问题, 并实现**线长最短**
- **时序分析**: 把数亿条连线关系, **建图并找出时延最长的路径**

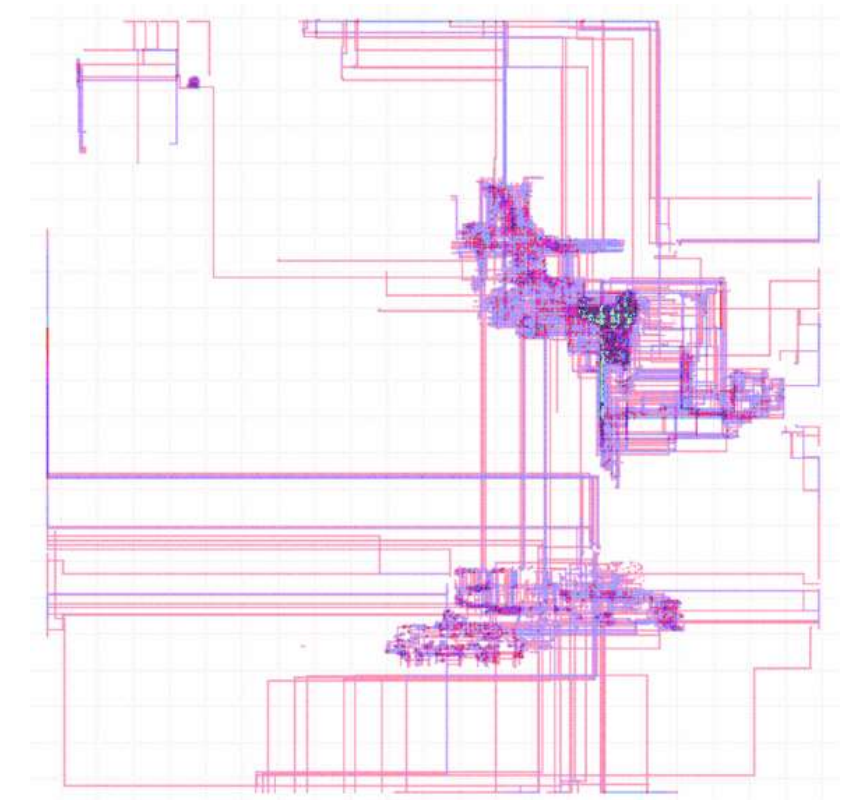


仿真验证中的图计算 (同时压迫 icache和dcache, stall很多)

RepCut: Superlinear Parallel RTL Simulation with Replication-Aided Partitioning (ASPLOS 2023)



“果壳-1” 芯片的Replace布局
By 陈仕健

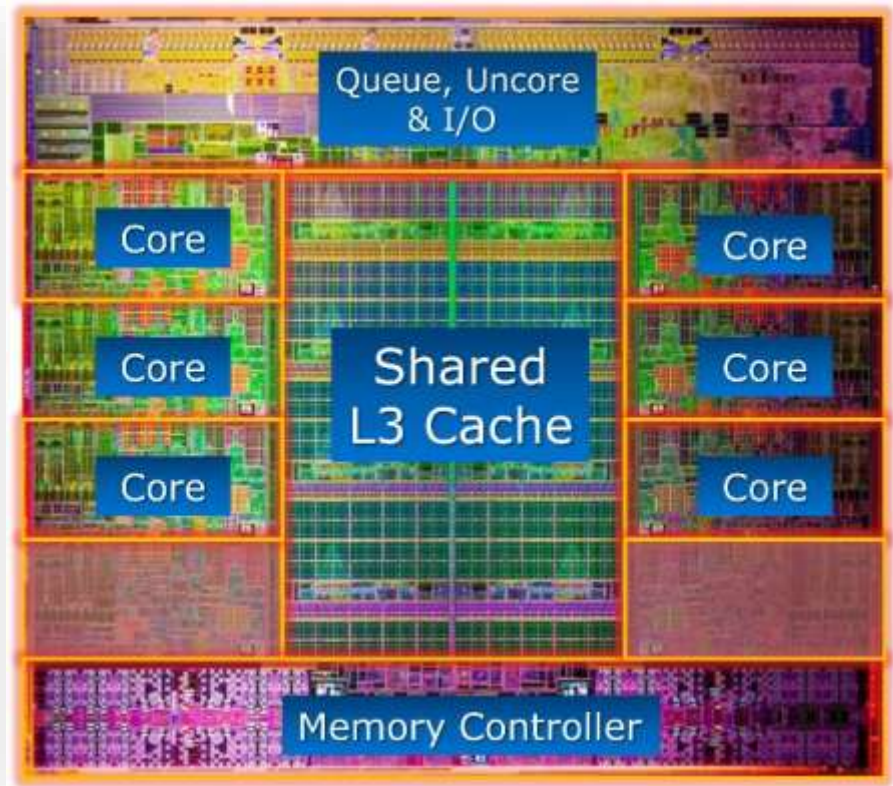


“果壳-1” 芯片的iGR布线
By 曾智圣

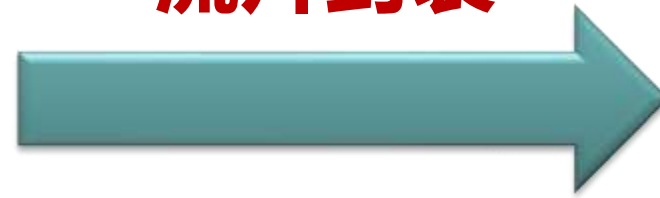
指令集 → 微架构 → 版图 → 芯片产品

- 将版图提交给台积电/中芯国际等企业**流片**，然后由长电等企业完成**封装**，获得芯片

芯片版图



流片封装



芯片产品



三、本科生帶着自己的芯片毕业

首期 “一生一芯” 计划
(2019年8月~12月)

首期“一生一芯”计划

——让学生带着自己设计的处理器芯片毕业

- 2019年8月底，在国内**首次以流片为目标**，由**5位**2016级计算机学院本科生主导完成一款64位**RISC-V处理器SoC芯片**设计，于12月19日**流片**
- 芯片成功**运行Linux**操作系统与**国科大教学操作系统UCAS-Core**



金越



王华强



王凯帆

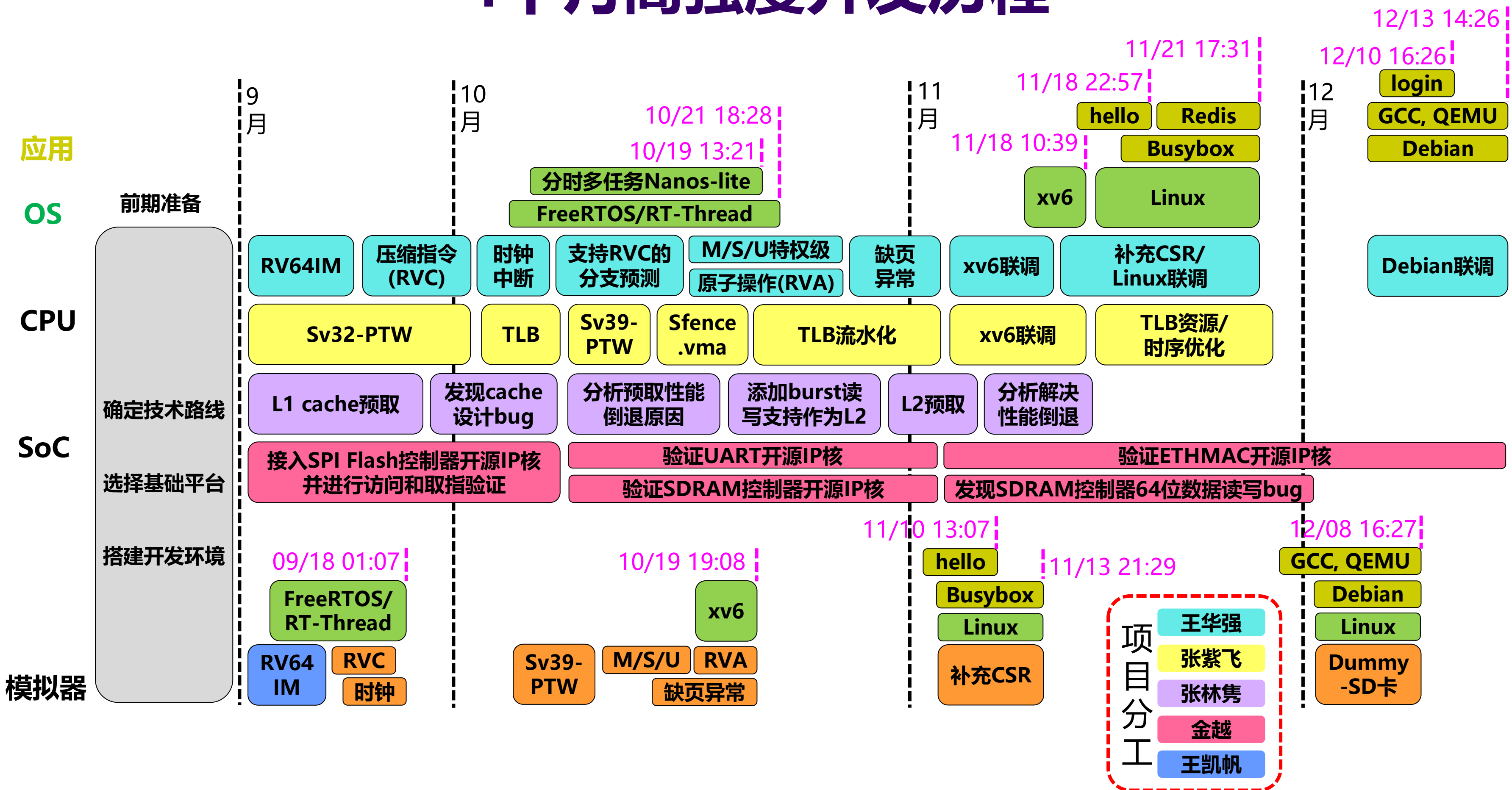


张林隽



张紫飞

4个月高强度开发历程

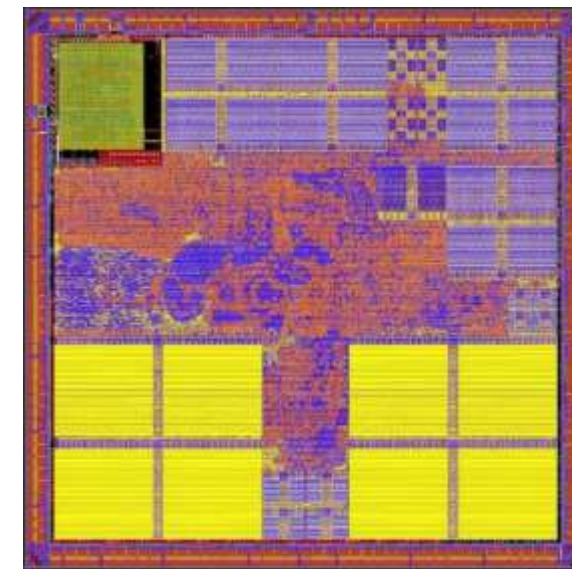


成果：果壳 (NutShell)

一款功能完整的RISC-V处理器

- 单发射9级流水顺序核
- RV64IMAC指令集，支持M/S/U特权级
- 两位饱和计数器分支预测，512项BTB，16项RAS
- 支持Sv39分页机制，支持硬件TLB填充
- 32K指令/数据L1 cache
- 支持L1 指令/数据cache读一致性
- 128K L2 cache，支持next line预取
- 使用Chisel语言开发
- 支持SDRAM、SPI flash、UART等外设
- 支持启动Linux 4.18.0内核
- 支持运行Busybox套件
- 在仿真/FPGA环境下支持启动Debian 11

基于国产110nm工艺完成流片



- 110nm工艺
- 10mm²
- 200mw@350MHz Typical
- TQFP100封装

学生感悟

依赖指导 => 主动探索

与之前实验最大的不同.....就是**没有先行者一步一步的详细指导**，而是要**自己寻找方法，独立实现**，然后进行验证甚至推倒重来。

使用者 => 创造者

胡伟武老师曾经说过，我们计算机系的同学应该学会怎么造计算机而不是怎么用计算机。我以前对这句话并不太有感触，相反曾经质疑国科大计算机系的课程设置这么多硬件的内容是否合理。但**真正参与到项目中才发现在大学里所学的知识 and 技能是真的有用**。

大部分知识在体系结构课程中...**工作原理也很简单**，只有短短的几行，但是**真正在代码中实现却比自己所想象的要困难得多**。

更自信、更有耐心

和4个月之前的自己相比.....**最重要的就是这种观念上的转变**。遇到bug不再在一个地方上死磕，而是从心理上告诉自己bug都是人写出来的，**只要有耐心，只要挖得足够深就一定能找到问题所在**。

成就感

真正参与到项目中才知道课程作业就像直接给人采摘的果园一样，但项目却是**给一片荒地和几颗果树苗，从开垦种植和施肥都要自己动手**，并且还不知道这样能不能结出果实。不知为何，总觉得**从0开始种出的果实要更甜一些**。

提升专业知识，锤炼心理素质

首期 “一生一芯” 计划取得成功

- 五位同学实现 “带着自己设计的处理器芯片毕业” 这一目标



“果壳”设计被RISC-V Global Forum接收

NutShell: A Linux-Compatible RISC-V Processor Designed by Undergraduates

Your submission was accepted for
RISC-V Global Forum 2020

The RISC-V Event Team

★ 详情

Dear Huaqiang,

It is our pleasure to inform you that we
have accepted your submission.

***Nutshell: A Linux-Compatible RISC-V
Processor Designed by Undergraduates,***

as a lightning talk, and would like to
welcome you as a speaker to the [RISC-V
Global Forum 2020](#), happening virtually
Thursday, September 3. Sessions will
overlap during US, Europe, and Asia
working hours. We ask that you please
review the information below and on the
[Speaker Guide](#) and complete all the
required items to confirm your speaking
engagement.

The event will take place on the virtual
event platform. MeetingPlay. The



RISC-V Global Forum

Barcelona

9:00 AM–3:00 AM

Shanghai

3:00 PM–9:00 AM

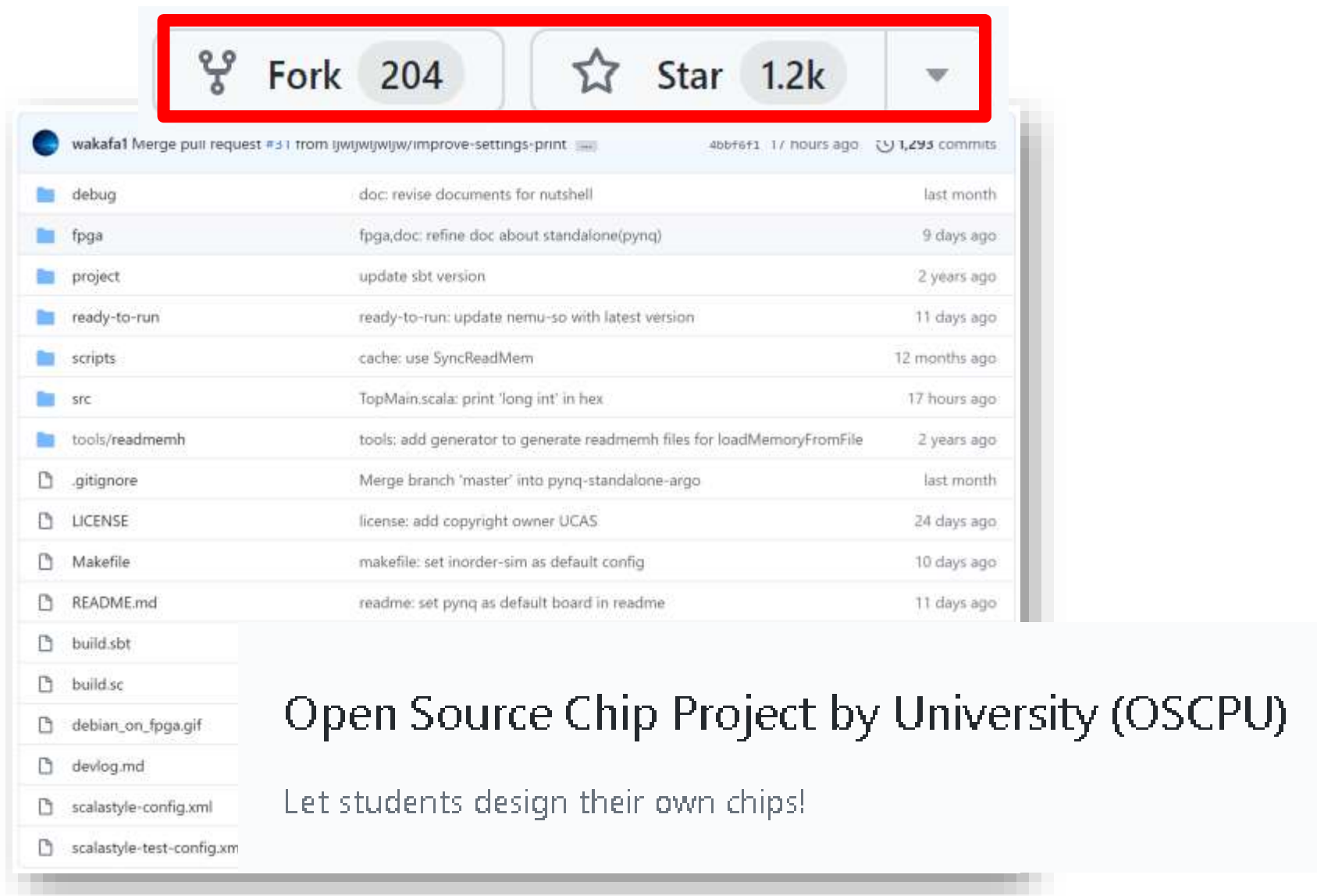
San Francisco

12:00 AM–6:00 PM

**全世界29个技术报告
唯一一个由本科生完成**

向国际社区开源，已有近百位使用者

- 果壳已在**GitHub**开源
 - 超过200个Fork



<https://github.com/OSCPU/NutShell>

- 典型案例: 中科院软件所已将**华为OpenEuler**操作系统移植到果壳上, 并建立果壳分支
 - 工业级操作系统成功应用在教学芯片上

```
[ 2.680000] clk: Not disabling unused clocks
[ 2.690000] Waiting for root device /dev/mmcblk0p2...
[ 2.760000] mmc0: new high speed SDHC card at address aaaa
[ 2.810000] mmcblk0: mmc0:aaaa S508G 7.40 GiB
[ 2.840000] mmcblk0: p1 p2
[ 2.980000] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data m
[ 2.990000] VFS: Mounted root (ext4 filesystem) on device 179:2.
[ 3.010000] devtmpfs: mounted
[ 3.020000] Freeing unused kernel memory: 176K
[ 3.030000] This architecture does not have kernel memory protection.
[ 4.300000] random: fast init done
[ 6.550000] systemd[1]: System time before build time, advancing clock.
[ 6.630000] systemd[1]: Failed to lookup module alias 'autofs4': Functi
[ 6.910000] systemd[1]: systemd v243-18 running in system mode. (+PAM +
+IDN2 -IDN +PCRE2 default-hierarchy=legacy)
[ 6.940000] systemd[1]: Detected architecture riscv64.

Welcome to openEuler 20.03 (LTS)!

[ 7.040000] systemd[1]: Set hostname to <openEuler-RISCV-rare>.
```

四、建设大规模人才培养流程

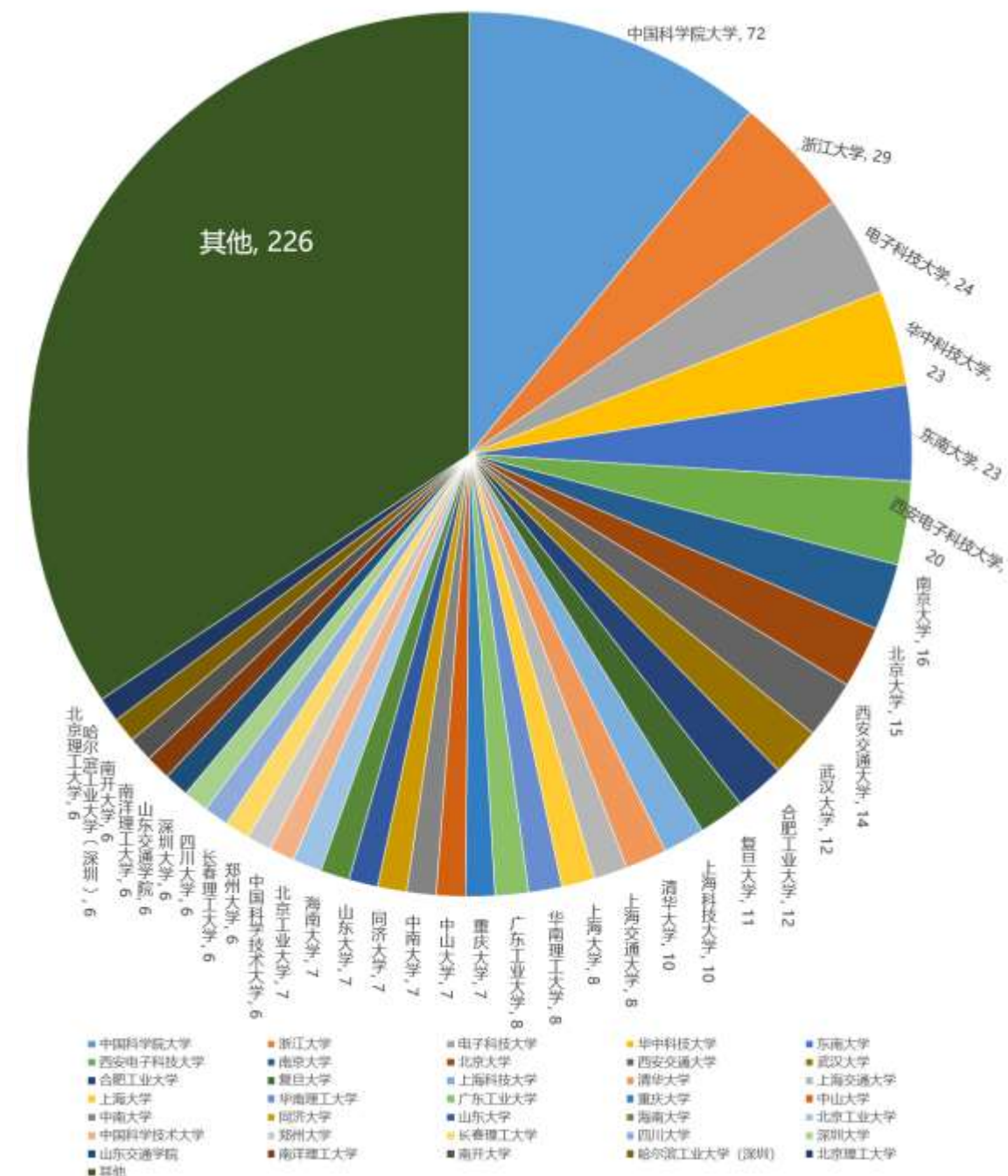
第三期 “一生一芯” 计划

(2021年7月~2021年12月)

第三期报名情况简析

● 报名总数：760人

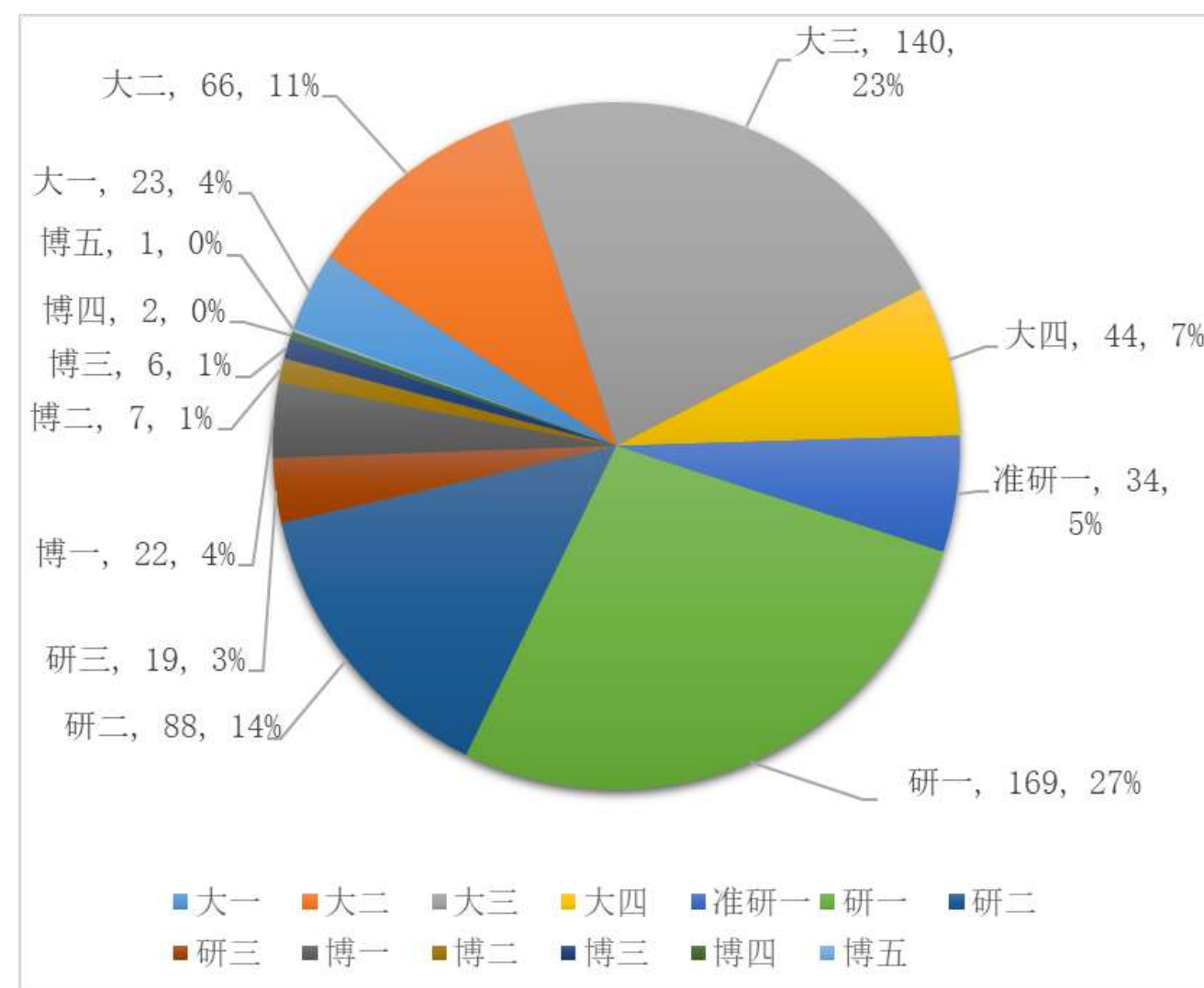
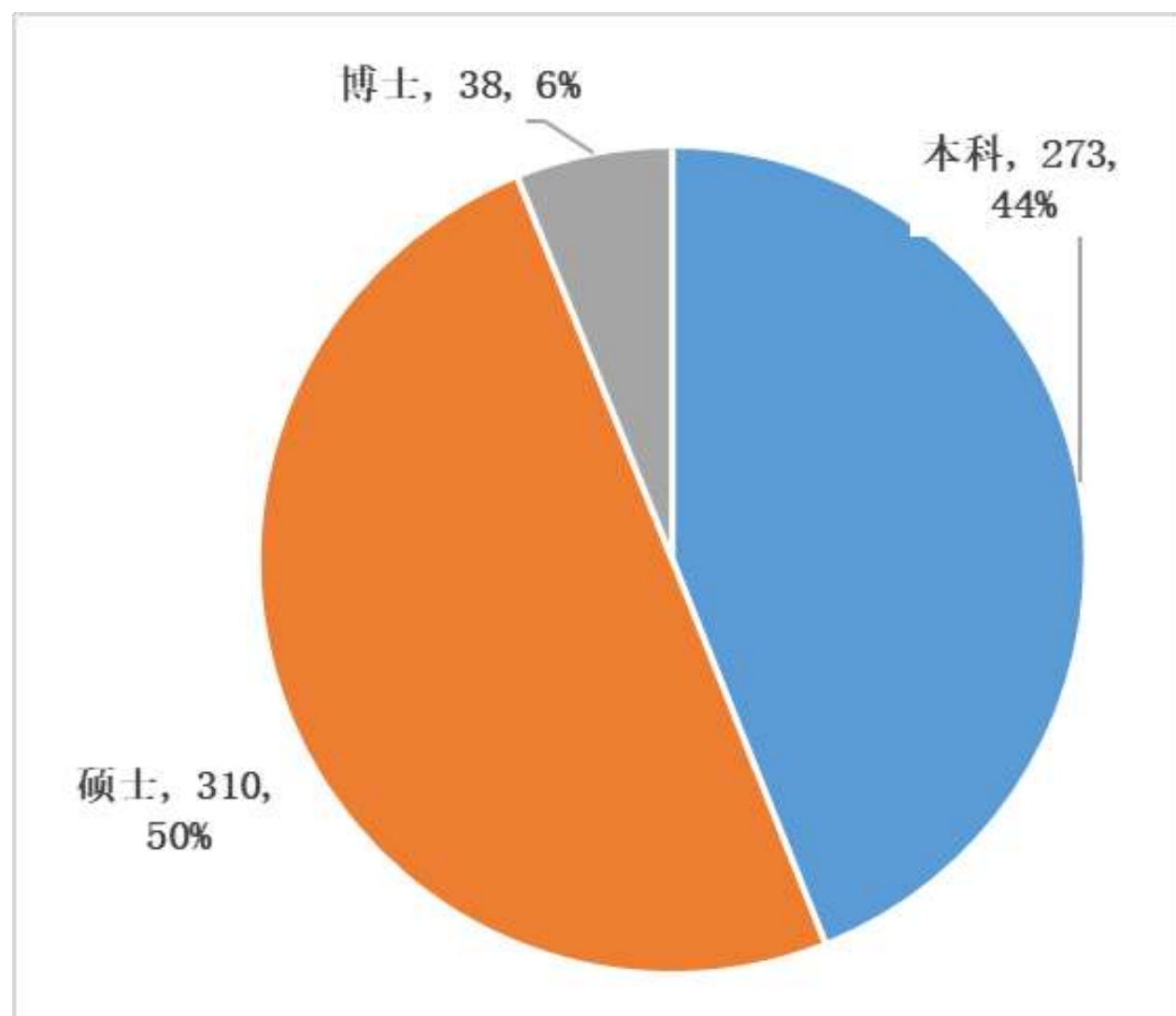
- 在校生是“一生一芯”的主力，共 625人，占比82%；较多已毕业学生报名
- 覆盖168所高校（含国外30所）
- 国科大（72人）
- 浙江大学（29人）
- 电子科技大学（24人）
- 华中科技大学（23人）
- 东南大学（23人）
- 西安电子科技大学（20人）
- 南京大学（16人）
- 北京大学（15人）
-



报名情况简析

● 年级分布：

- 本硕博各年级均有分布
- 本科生占比 44%，硕士 50%，博士 6%
- 按细分来看，研一（27%），大三（23%），研二（14%），大二（11%）



教学团队

- 工作内容

- 准备讲义和其他辅助材料；组织报告和答疑

- 组织团队

- 技术指导：余子濠
- 技术负责：洪志博（深圳大学）

- 助教团队

- 线下助教（13人）：杨烨（深大）；薛臻（鹏城实验室）；夏斌（上海科大）；王唯一（浙大）；吴莹（浙大）；杨浩泽（东南大学）；郭明鑫（中科院自动化所）；祁民浩（深大）；段震伟（中科大）；梅晓龙（海南大学）；邓海文（海南大学）；史历（上海交大）；张宇轩（西安交大）；
- 线上助教（14人）：王嵩岳（国科大）；叶从容（集美大学）；桑乾龙（武大）；潘星雨（重庆邮电）；胡轩（中科院计算所）；刘实（浙大）；苑子琦（浙大）；刘定邦（哈工大(深圳)）；周聪（天大）；肖天海（港科大）；尹承彬（上海交大）；陈春韵（南洋理工）；张行健（浙大）；唐伟康(涂鸦科技)

月	耗时	任务	运行的新程序
1	1周	完成PA2第一阶段(实现几条指令)	
	1周	单周期addi	addi指令
	1周	完成PA2第二阶段(实现完整rv32i/rv64i)	
	1周	实现更多指令, 通过同步总线访存	cpu-test, riscv-tests
	1周	完成PA2第三阶段(输入输出)	
2	0.5周	加一条自定义指令, 仿真时通过display输出字符	hello, coremark, dhrystone, microbench(时钟返回0)
	0.5周	加mcycle寄存器	timer-test, 字符版超级玛丽
	1周	完成PA3第一阶段	
	1周	加CSR, ecall	yield-test, Nanos-lite, 仙剑
	1周	加axi总线和模拟串口, 去掉自定义指令(需要握手, 类似多周期处理器)	
3	1周	完善CSR	RT-Thread
	1周	用axi访存(取指也需要握手)	
	1周	接入SoC, 从flash启动	

支撑团队

● SoC团队 (4)

- 工作内容：SoC集成和验证，前后仿，综合和时序约束等
- 预计周期：10月7日到11月7日
- 技术指导：刘彤
- 技术负责：张文迪（海南大学）
- 技术团队：谢王照琪（澳门科技），吴泽辉（华南理工），龙康杰（上海科大）

● IC后端团队 (5)

- 工作内容：实施物理设计，并生成可流片的 GDSII 版图
- 预计周期：11月7日到11月30日
- 技术指导：何伟及其团队
- 技术负责：庄楚楠（广东工业大学-新入职鹏城实验室）
- 技术团队：张书涵（海南大学），方闻绩（南航），夏丽平（华南师范），熊启（北大深）

维护学生的学习记录和组会情况

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	3.2.7	4.1.3	4.2.4	5.1	5.8	5.15	5.22	5.29	6.5	7.3	7.10	7.17	7.24	8.7	8.14		
ysyx_22040020	毛英皓	天津大学	科学与工程; 辅修: 计算机类	大四	ysyx_220020-毛英皓-天津大学-学习记录	ysyx_220020-毛英皓-天津大学-读后感		两周未更新	3.20			2022.4.11	S-关键	PA; 2021	2021	npc-c	202	2021	2022	7.10	请假		2022.8.7	请假		
ysyx_22040047	吴浩宇	东北大学	计算机科学与技术	大三	ysyx_220047-吴浩宇-东北大学-学习记录	ysyx_220047-吴浩宇-东北大学-读后感			2022.3.20		2022.4.10	请假	PA2-3	npc读	2022	2022.6.1	2022	2022.7.17	请假							
ysyx_22040091	冯浩原	中国科学院大学	计算机科学与技术	大三	ysyx_220091-冯浩原-中国科学院大学-学习记录	ysyx_220091-冯浩原的读后感	ysyx_220091-冯浩原的预学习实验报告		2022.3.20			2022.4.24	请	PA4.1												
ysyx_22040030	刘明手	东北大学秦皇岛分校	计算机类	大一	ysyx_220030-刘明手-东北大学秦皇岛分校-学习记录	ysyx_220030-刘明手的读后感	ysyx_220030-刘明手的实验报告		2022.3.20	2022.3	2022.4.10	请假	PA2													
ysyx_22040063	包子旭	河南理工大学	计算机科学与技术	大二	ysyx_220063-包子旭-河南理工大学-学习记录	ysyx_220063-包子旭的读后感			2022.3.20		2021	PA2	PA2.1	PA2.1	PA2.1	2021	2022	2022	7.3	请	2022.7.17	请	2022.8.7	请假		
ysyx_22040031	苗恒	青岛科技大学	电子信息	研一	ysyx_220031-苗恒-青岛科技大学-学习记录	ysyx_220031-苗恒的读后感	ysyx220031-苗恒-实验报告		2022.3.20			PA2	PA2.1	PA2.1	PA2.1	PA2.1	2021	2022	2022	7.3	请	2022.7.17	请	2022.8.7	请假	
ysyx_22040012	毕睿	海南大学	电子信息工程	大二	副本-ysyx_220012-毕睿-海南大学-学习记录	ysyx_220012-毕睿的读后感	ysyx_220012-毕睿的实验报告.pdf		2022.3.20	2022.4	2021	PA2.1	PA2.1	PA2.1	PA2.1	2021	2022	2022	2022	2022	2022.7.20	2022.8.7	请假		2022.8.14	请假
ysyx_22040099	蔡惠荣	太原理工大学	通信工程	大二	ysyx_220099-蔡惠荣-太原理工大学-学习记录	ysyx_220099-蔡惠荣的读后感	ysyx_220099-蔡惠荣的实验报告		2022.3.20					PA; 2021	2021	2022	2022	2022	2022	2022	npc告-	准备接入diffie		2022.8.14	请假	
ysyx_22040025	邱苒尧	东北大学	物联网工程	大二	ysyx_220025-邱苒尧-东北大学-学习记录	ysyx_220025-邱苒尧的读后感	ysyx_22040025-邱苒尧的实验报告		2022.3.20			npc-1	进行中	PA2-2;	2021	npc-2	2022.6.19	请假	npc-加	npc-流	2022.8.7	请假				
ysyx_22040034	杜奕明	太原理工大学	软件工程	大四	ysyx_220034-杜奕明-太原理工大学-学习记录	ysyx_220034-杜奕明的读后感	完成PA1, 无报告		2022.3.20			npc	进行中	PA2-1, n	2021	PA2-1	PA2-2	2022	2022.7	2022.7	2022.8.7	请假		2022.8.14	请假	
ysyx_22040094	黎梓浩	国防科技大学	软件工程	大三	ysyx_220094-黎梓浩-国防科技大学-学习记录	ysyx_220094-黎梓浩的读后感	完成至PA1.1, 无报告		2022.3.20	2022.3	2021	pa2	2-k	PA; 2022.5.2	npc-c	202	2021	2022	2022	2022	2022.7	2022.8.7	请假		2022.8.14	请假
ysyx_22040046	王凯	中国科学技术大学	计算机科学与技术	大三	ysyx_220046-王凯-中国科学技术大学-学习记录	ysyx_220046-王凯的读后感	ysyx_220046-王凯的实验报告		2022.3.20	2022.3	2021	2022.4.24	请	2021	2021	2021	2022	2021	2022	2022	7.10	请假				
ysyx_22040073	关富润	华南理工大学	集成电路工程	研一	ysyx_22040073-关富润的学习记录	ysyx_22040073-关富润的读后感	完成至PA1.1		2022.3.20		2021	2022.4.24	请	PA; 2022.5.2	npc-流水	中										
ysyx_22040096	叶创豪	杭州电子科技大学	软件工程	大三	ysyx_220096-叶创豪-杭州电子科技大学-学习记录	ysyx_220096-叶创豪的读后感	ysyx_220096-叶创豪的实验报告		2022.3.20		2022.4	PA2.1	2021	sdb	watch	2022	2021	2021	2022	2022.7	2022.7	2022.8.7	请假		2022.8.14	请假
ysyx_22040080	孙佳丰	电信科学技术研究院	电子科学与技术	研三	ysyx_220080-孙佳丰-电信科学技术研究院-学习记录	ysyx_220080-孙佳丰读后感	完成至PA1.1	一周未更新	3.20		2022.4	2022.4.24	请	PA2.2	NI	2022.5.29	2022.6.19	请假								
ysyx_22040040	祝静	天津理工大学	集成电路工程	研一	ysyx_220040-祝静-天津理工大学-学习记录	ysyx_220040-祝静的读后感	ysyx_220040-祝静的实验报告		2022.3.20			PA2.1	完成	PA	2021	2022.5.29	2021	2022.7.3	请	流水线	SoC中	参加会议		2022.8.14	请假	
ysyx_22040070	杨利民	北京工商大学	软件工程	大三	-	ysyx_220070-杨利民-北京工商大学-读后感.docx	-	等待更新中	3.20			npc	踩坑中										PA3完成			
ysyx_22040163	王晨宇	南通大学	计算机科学与技术	大二	ysyx_22040163-王晨宇-南通大学-学习记录	ysyx_22040163-王晨宇的读后感	ysyx_22040163-王晨宇的预学习实验报告		2022.3.20				PA3.3	PA3.3	2021	2022.6.5	请假		npc-中	PA3完	参加会议					
ysyx_22040068	王俊	海南大学	保研至微电子学, 研究方向:	大四	ysyx_220068-王俊-学习记录	ysyx_220068-王俊-海南大学读后感.docx	发至PA1.1 (问题有点多, 还没来得及汇总和撰写实验报告)	请假5.31	2022.3.20	缺席			PA1													
ysyx_22040188	徐铭伟	西安电子科技大学	信息与通信工程	研一	ysyx_22040188-徐铭伟-西安电子科技大学-学习记录	ysyx_22040188-徐铭伟-读后感	完成至PA1.1	两周未更新	2022.3.20	缺席																
ysyx_22040214	朱吉宏	哈尔滨工业大学	电子信息	研一	ysyx_22040214-朱吉宏-哈尔滨工业大学-学习记录	ysyx22040214-朱吉宏的读后感	ysyx_22040214-朱吉宏-哈尔滨工业大学-实验记录		2022.3.20				npc1	挂	PA	2021	npc-c	npc-挂	2022.6.1	2022.7.10	请	2022.7	2022.8.7	请假		
ysyx_22040539	周君宝	清华大学	电子科学与技术	大四	ysyx_22040539-周君宝-清华大学-学习记录		ysyx_22040539-周君宝-清华大学-实验报告	两周未更新	2022.3.20	缺席																
ysyx_22040017	解博元	东北大学秦皇岛分校	计算机科学与技术	大二	ysyx_220017-解博元-东北大学秦皇岛分校-学习记录	ysyx_220017-解博元的读后感		请假5.31	2022.3.20	2021	2022.4.10	请	因	2022.5.15	请	PA1							无进展	参加会议		
ysyx_22040067	李珍琪	国防科技大学	计算机科学与技术	大三	ysyx_220067-李珍琪-国防科技大学-学习记录	ysyx_220067-李珍琪的读后感	ysyx_220067-李珍琪的实验报告	一周未更新	2022.3.20	2021	2022.4.24	请														
ysyx_22040001	蒋雨龙	深圳大学	计算机科学与技术	大四	ysyx_220001-蒋雨龙-深圳大学-学习记录	ysyx_220001-蒋雨龙-读后感	ysyx_220001-蒋雨龙-PA实验报告	一周未更新	2021	2021	2022.4.2	请	PA2.1													
ysyx_22040178	王九龙	北京邮电大学	电子科学与技术	大三	ysyx_22040178-王九龙-北京邮电大学-学习记录	ysyx_22040178-王九龙的读后感	ysyx_22040178-王九龙的预学习实验报告	两周未更新	3.20	2022.3.27	请假					2022.5.29	请假									
ysyx_22040374	崔家贺	南开大学	软件工程	大二	ysyx_22040374-崔家贺学习记录	ysyx_22040374-崔家贺提问的智慧读后感	ysyx_22040374-崔家贺实验记录		2022.3.20	2022.3	2021	2022.4.24	请	2021	2021	2021	2022	2021	2022	2022.7	2022.7	2022.8.7	请假		2022.8.14	请假
ysyx_22040154	徐步青	东南大学	电子科学与技术	大四	ysyx_22040154-徐步青-东南大学-学习记录	ysyx_22040154-徐步青《提》与《别》读后感	ysyx_22040154-徐步青实验报告		2022.3.20			PA2.3	声卡前	PA; 2022.5.2	2022	2022.6.19	请假	npc-流	npc-流	参加会议						
ysyx_22040193	苑文强	中国科学技术大学	电子信息	研二	ysyx_22040193-苑文强-中国科学技术大学-学习记录	ysyx_22040193-《提问的智慧》与《别像弱智一样提问》读后感	ysyx_22040193-苑文强预学习实验报告		2022.3.20			2021	PA2完成, n	PA; 2021	2022.5.29	2021	2022.7.3	请	2022.7	2022.7	2022.8.7	请假				
ysyx_22040142	刘磊	上海海洋大学	计算机科学与技术	大三	ysyx_22040142-刘磊-上海海洋大学-学习记录	ysyx_22040142-刘磊-上海海洋大学-读后感	ysyx_22040142-刘磊-上海海洋大学-实验报告		2022.3.20				PA2除tr	PA2除tr	2021	2022	nemu	2022.7.10	请假							
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	ysyx_22040200-乌鑫龙-北京师范大学珠海分校-学习记录	ysyx_22040200-读后感	ysyx_22040200-实验报告	请假5.31	3.20				npc	实现跳转	PA2-1	npc-dummy	npc-chisel		npc	基本完成						
ysyx_22040414	林沛润	广东工业大学	集成电路设计与集成系统	大三	ysyx_22040414-林沛润-广东工业大学-学习记录	ysyx_22040414-林沛润读后感	ysyx_22040414-林沛润实验报告		2022.3.20	2022.3.27	请假		PA2.3	2021	npc-基础	2022.7.3	请假		2022.7	2022.7	2022.8.7	请假		2022.8.14	请假	
ysyx_22040413	聂凯	电子科技大学	电子信息工程	研二	ysyx_22040415-聂凯-电子科技大学大学-学习记录	ysyx_22040015-聂凯的读后感	ysyx_22040015-聂凯的实验报告		2022.3.20		2022.4	2022.4.24	请	2021	2022.5.22	请假										
ysyx_22040595	郑佳纯	华南理工大学	电子科学与技术	大四	ysyx_22040595-郑佳纯-华南理工大学-学习记录-1	ysyx_22040595-郑佳纯-科学地提问读后感.docx	ysyx_22040595-郑佳纯-预学习报告.docx		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	ysyx_22040301-于皓哲-沈阳工业大学-学习记录	ysyx_22040301-于皓哲-读后感.docx	ysyx_22040301-于皓哲-实验报告.docx	请假5.31	3.20	2021	2022.4	回归NE	因	PA1				2021	2022	2022.7	2022.7	2022.8.7	请假		2022.8.14	请假
ysyx_22040562	李张勋	华中科技大学	集成电路设计与集成系统	大三	ysyx_22040562-李张勋-华中科技大学-学习记录	ysyx_22040562-李张勋-提问的智慧读后感	ysyx_22040562-李张勋-预学习实验报告		2022.3.20		2022.4.2	请假	npc	2021	2021	2022.6.5	请	2021	2022.7.10	请	无进展	2022.8.7	请假			
ysyx_22040228	李国旗	齐鲁理工学院	电子信息工程	大二	ysyx_22040228-李国旗-齐鲁理工学院-学习记录	ysyx_22040228-李国旗-读后感	ysyx_22040228-李国旗-实验报告		2022.3.20				npc1-环	PA; npc-	2022.5.29	npc-cache		SoC仿	SoC仿	参加会议						
ysyx_22040221	曾明星	山东科技大学	计算机技术	研一	ysyx_22040221-曾明星-山东科技大学-学习记录	ysyx_22040221-曾明星-提问的智慧读后感	ysyx_22040221-曾明星-实验报告	已进香山	3.20	2022.4	2021	npc2-添	PA2.2	PA2.2	PA3.1	PA3.1	2021	2022	PA3.3	2022.7	2022.8.7	请假		2022.8.14	请假	
ysyx_22040140	周浩杰	浙江工业大学	计算机科学与技术	大四	ysyx_22040140-周浩杰-浙江工业大学-学习记录	如何提问与解决问题	ysyx_22040140-周浩杰-浙江工业大学-实验报告		2022.3.20				PA2-2													
ysyx_22040114	王以苏	西安电子科技大学	通信与信息系统	研三	ysyx_22040114-王以苏-西安电子科技大学-学习记录	别像弱智一样提问-读后感	ysyx_22040114-王以苏-一生一芯计划ysyx实验报告		2022.3.20				PA; 2022.5.2	2022.6.5	请	2021	2022.7.10	请假				2022.8.7	请假		2022.8.14	请假
ysyx_22040735	朱俊宸	电子科技大学	电子科学与技术	研二	ysyx_22040735-朱俊宸-电子科技大学-学习记录	ysyx_22040735-朱俊宸-读后感	ysyx_22040735-朱俊宸-实验报告		2022.3.20	2022.3.27	请假		PA; 2021	PA3-2		2021	2022	2022.7	中断前	2022.8.7	请假					
ysyx_22040203	王杰	西安电子科技大学	电子信息工程	研一	ysyx_22040203-王杰-西安电子科技大学-学习记录	《提问的智慧》与《别像弱智一样提问》读后感	ysyx-22040203-王杰-实验报告	一周未更新	3.20	2022.3.27	请	2022.4.24	请	2022.5.15	请	PA2.1										
ysyx_22040863	陈洪手	重庆大学	计算机科学与技术	大四	ysyx_22040863-陈洪手-进展记录	如何科学地提问读后感			2022.3.20																	
ysyx_22040411	曹泽文	中国科学院大学	电子科学与技术	研一	ysyx_22040411-曹泽文-中国科学院大学-学习记录	ysyx_22040411-曹泽文-中国科学院大学-读后感	ysyx_22040411-曹泽文-中国科学院大学-实验报告	请假5.31	3.20	2022.3	2021	2022.4	因	2022.5.15	请	PA1	npc-diffie	2022	npc-大	准备挂	参加会议					
ysyx_22040295	张子御	北京科技大学	计算机科学与技术	大三	ysyx_22040295-张子御-北京科技大学-学习记录	ysyx_22040295-张子御-《提问的智慧》读后感			2022.3.20				2021	npc-diff	PA2-2	npc指令添加中			Cache	2022.7	参加会议			2022.8.14	请假	
ysyx_22040422	张虎森	电子科技大学	电子科学与技术	研二	ysyx_22040422-张虎森-电子科技大学-学习记录		ysyx_22040422-张虎森-电子科技大学-实验报告		2022.3.20	缺席		PA3.2-	PA3仙创前, r	2022	2021	2021	2022.7.10	请	2022.7	2022.8.7	请假					
ysyx_22040525	李权鑫	中山大学	集成电路工程	研一	ysyx_22040525-李权鑫-中山大学-学习记录	ysyx_22040525-李权鑫-读后感	ysyx_22040525-李权鑫-实验报告	请假5.31	3.20	2022.3	2022.4.10	请	因	2022.5.1	2021	2022	2021	2022	2022.7	2022.7	2022.8.7	请假		2022.8.14	请假	
ysyx_22040737	刘成众	中国科学院大学	集成电路工程	研一	ysyx_22040737-刘成众-中国科学院大学-学习记录	《提问的智慧》读后感	刘成众-一生一芯预学习实验报告	一周未更新	3.20	2022.3.27	请															

一位学生在“一生一芯”学习过程的成长记录

	日期	计划任务	总时长	任务完成情况	卡了一段时间的bug
				naya库里现成有。尝试在cygwin环境中从源代码编译一份，编译出来的二进制	
第一周 (5天)	2月25日	安装Veril	3月28日	尝试完成MIPS的MEMU	5h 基础部分MIPS和RISC-V差距不大。实现起来的思路也大同小异
	2月26日	调	3月29日	完成MIPS的浮点计算	5h 完成浮点寄存器，可以将32位和64位浮点数据存到寄存器中
	2月27日	完	3月30日	完成MIPS的浮点	5h 修SpinalHDL
	3月1日	完成实验报	3月31日	完成MIPS的浮点	5h 修Sp
	3月3日	完	4月1日	完成CSR指令	5h 调试
第二周 (7天)	3月6日	准备并	4月2日	完成CSR指令	5h 调试
	3月7日	完成PA	4月3日	完成CSR指令	5h 调试
	3月8日	修正	4月4日	完成CSR指令	5h 调试
	3月9日	完成	4月5日	完成CSR指令	5h 调试
	3月10日	引入	4月6日	FPGA选型+裁	5h 调
第三周 (7天)	3月11日	完	4月7日	完成DDR3L内存	5h 重写go
	3月12日	进一	4月8日	完成Flash绘	5h 重写go
	3月14日	完成	4月9日	使用SFP和PMC作为外	5h 编写龙
	3月15日	完成键	4月10日	修正上电时	5h 测试
	3月16日	完成VGA及	4月11日	修正上电时	5h 学习LoongArch《
第四周 (7天)	3月17日	完成	4月12日	布线	5h 学习LoongArch《
	3月18日	完成	4月13日	布线	5h 学习LoongArch《
	3月19日	完成ctr	4月14日	打板	5h 学习LoongArch《
	3月20日	完成ftrace、	4月15日	配置Verdi环境 & 学	5h 学习LoongArch《
	3月21日	完成中	4月16日	配置Virtuoso环境 & 开	5h 学习《量
第五周 (6天)	3月22日	完成	4月17日	画CPU微架构	5h 学习《量
	3月23日	测试中断	4月18日	修正微架构	5h 学习《量
	3月24日	完成nemu与	4月19日	完成取指和RV64I	5h 修复S
	3月25日	完成nemu与	4月20日	完成RV64M	5h 修复S
	3月26日	尝试引	4月21日	修改微架构	5h 修复S
第六周 (5天)	3月27日	完成最基	4月22日	编写Testben	5h 修
			4月23日	编写Testben	5h 学习《
			4月24日	编写Testben	5h 补DIV指
			4月25日	取指部分重	5h 测试MUL和DDI
			4月26日	编写计分析	5h 编译LAB
第七周 (7天)			4月27日	编写计分析	5h 调试LA顶层
			4月28日	学习香山代	5h 调试LA的AX
			4月29日	学习香山代	5h 调试LA的AX
			4月30日		5h 调试LA的AX
			5月1日		5h 调试LA的AX
第八周 (5天)			5月2日	五一放假	4h poco烧写步骤挺多，而且不知道是不是已经烧进去了
			5月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第九周 (4天)			5月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十周 (7天)			5月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十一周 (7天)			5月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十二周 (7天)			5月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十三周 (4天)			5月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			5月31日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十四周 (7天)			6月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十五周 (7天)			6月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十六周 (7天)			6月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十七周 (7天)			6月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十八周 (7天)			6月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第十九周 (7天)			6月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			6月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十周 (7天)			7月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十一周 (2天)			7月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十二周 (7天)			7月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十三周			7月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十四周			7月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十五周			7月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			7月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十六周			7月31日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十七周			8月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十八周			8月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第二十九周			8月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十周			8月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十一周			8月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十二周			8月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			8月31日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十三周			9月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十四周			9月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十五周			9月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十六周			9月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十七周			9月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十八周			9月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			9月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第三十九周			10月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十周			10月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十一周			10月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十二周			10月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十三周			10月24日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月25日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月26日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月27日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月28日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十四周			10月29日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月30日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			10月31日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月1日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月2日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十五周			11月3日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月4日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月5日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月6日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月7日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十六周			11月8日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月9日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月10日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月11日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月12日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十七周			11月13日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月14日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月15日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月16日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月17日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十八周			11月18日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月19日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月20日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月21日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月22日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
第四十九周			11月23日		4h 感觉好像烧进去了，但是跑比特流的时候好像跑不起来
			11月24日		

第三期 “一生一芯” 成果

● 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

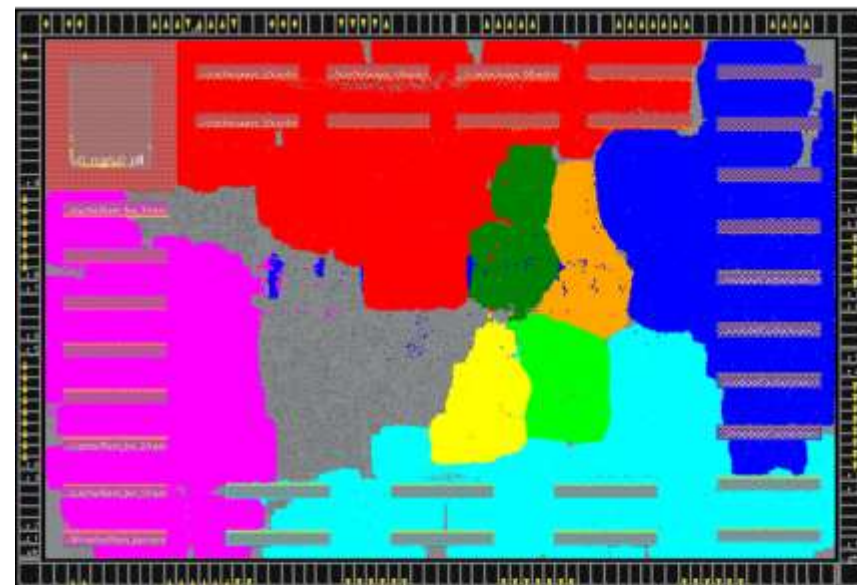
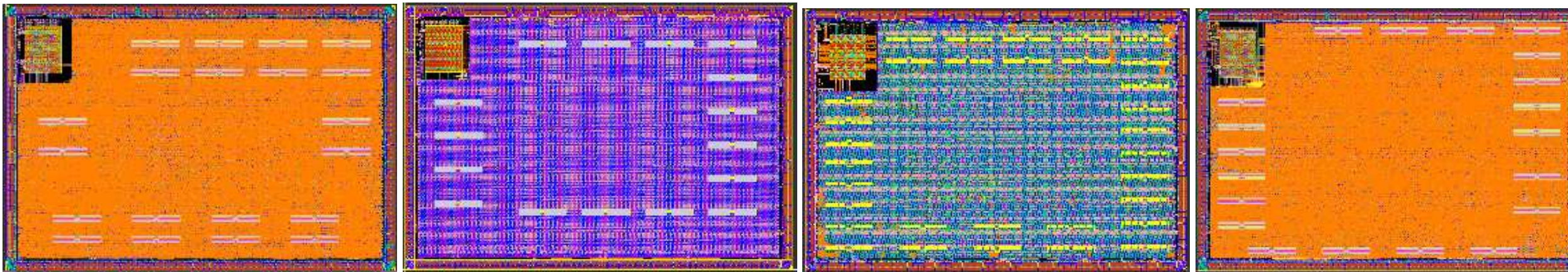
- 分支预测: 12个;
- 乱序执行: 3个;
- Chisel:16个; Verilog/sv: 31个
- Cache: 12个

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

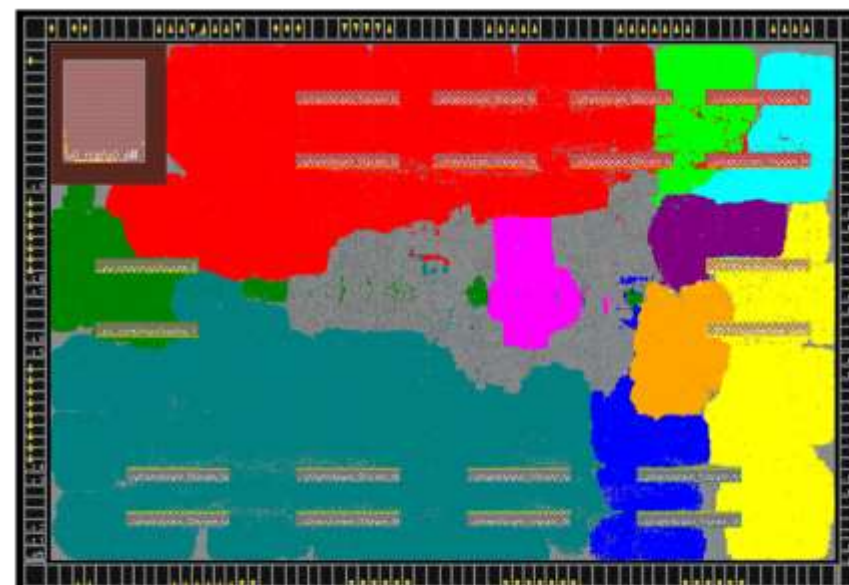
ysyx_210292	集美大学	大四	计算机科学与技术
ysyx_210191	南京理工大学	大三	计算机科学与技术
ysyx_210195	西安电子科技大学	硕士二年级	电子科学与技术
ysyx_210413	大连理工大学	研一	软件工程
ysyx_210428	沈阳工业大学	研二	电子科学与技术
ysyx_210313	电子科技大学	大三	微电子
ysyx_210302	复旦大学	研一	微电子学与固体电子学
ysyx_210184	清华大学	研二	集成电路工程
ysyx_210479	太原理工大学	大三	计算机科学与技术
ysyx_210013	西安交通大学	研二	微电子学与固体电子学
ysyx_210438	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



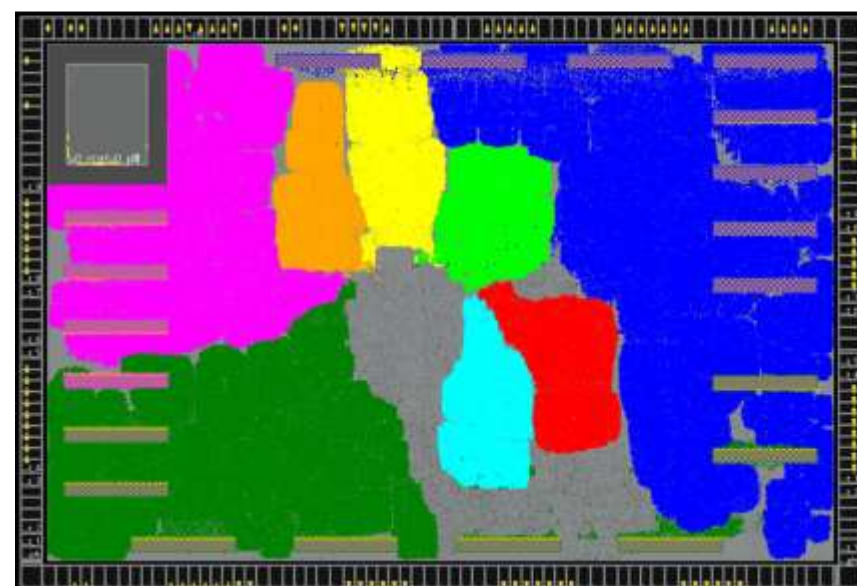
SoC_1

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



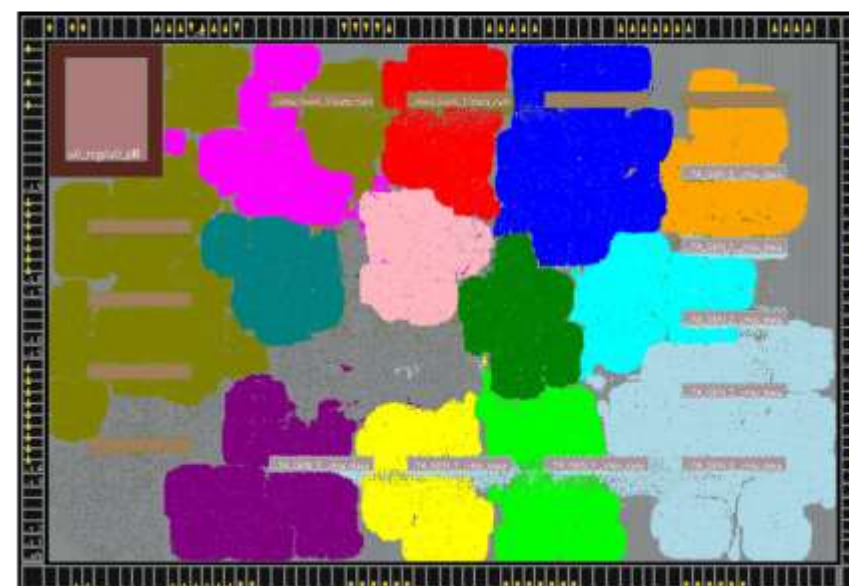
SoC_2

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



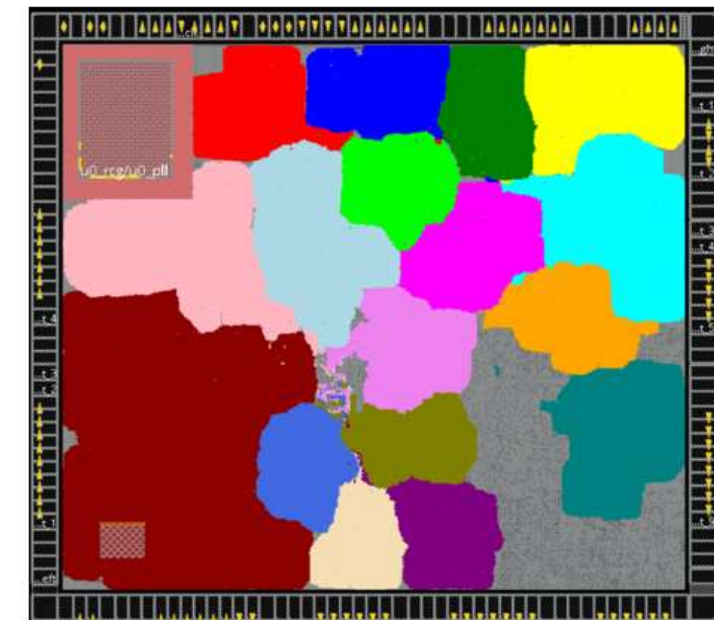
SoC_3

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



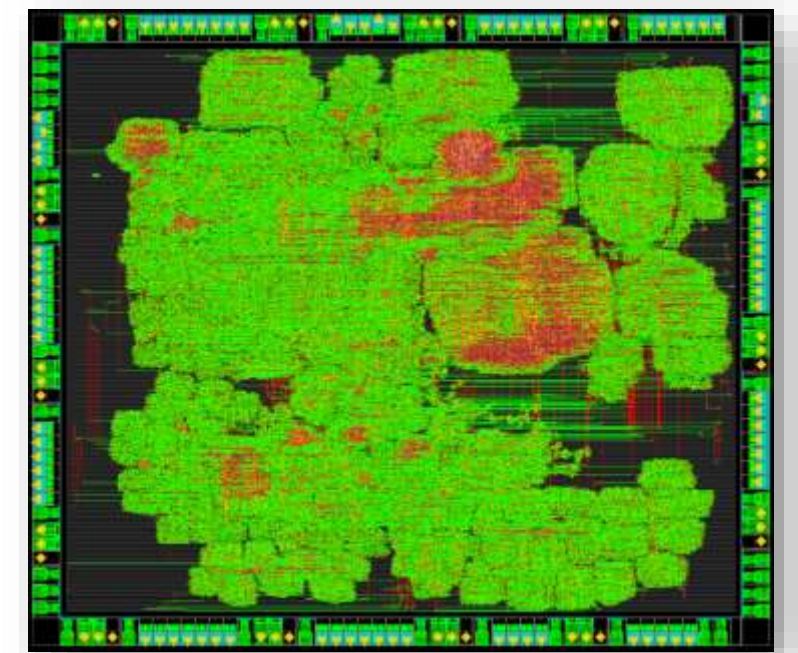
SoC_4

- 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

第三期 “一生一芯” 学生学习心得分享

- 项目组安排了8位同学的分享报告
 - **不同点：来自不同学校、不同年级、不同专业**
 - **相同点：之前均未设计过处理器**
 - 高泽宇，中国科学院大学，报名时大三，电子信息工程
 - 徐鑫，山东交通学院，报名时大一，电子信息工程
 - 孙际儒，南京大学，报名时大一，计算机科学与技术
 - 栗金伦，太原理工大学，报名时大二，水利
 - 于皓哲，沈阳工业大学，报名时研一，电子科学与技术
 - 许立达，中科院微电子所，报名时博一，微电子
 - 张文迪，海南大学，报名时研二，电子与通信工程，负责SoC集成和验证
 - 庄楚楠，广东工业大学，应届毕业，微电子，负责后端物理设计

03 总结回顾

两种能力的提升：

主动学习能力

- 硬件语言verilog的学习
- 计算机基础的学习
- 开发环境的学习
- 指令集的学习
-

用科学的方法快速解决问题的意识和能力

```
1111 1111 1111 0000 000 0001 010011 01=0+1 10000
0000 0000 0010 0000 000 0010 010011 02=0+2
0000 0000 0001 0000 000 0011 010011 03=0+3
1000 0000 0010 0001 011 0010 010011 04=1 (255+1) 9028213
0000 0000 0010 0001 011 0010 010011 05=2(05) 208213
0000 0000 0001 0001 101 0010 010011 06=3+1
0100 0000 0001 0001 101 0011 010011 07=3+1+1
0000 0000 0001 0000 000 0001 010011 01=10000
0000 0000 0001 0000 000 0010 010011 02=10010
0000 0000 0010 0001 000 0010 110011 03=10001
000000 00010 0001 000 0011 010011 04=
00000000011 0001 000 0011 000011 05=
```



心得3 – 工程经验的积累

经验一：“先完成，后完美”原则指导任务规划

– 在没有讲义的情况下，自己分解任务、规划步骤，并评估自己的规划

初版规划	最终规划
单周期addi	单周期addi
实现更多指令，通过简单总线访问	加diff-test
加axi总线，跑通所有cpu-tests	实现更多指令，通过简单总线访问
实现dummy的流水	加axi总线和模拟串口
实现所有cpu-test的流水(五级流水线)	加mcycle寄存器

- 初版规划：没有验证新功能正确性，急于加入流水线→bug层出不穷，难以排查
- 最终规划：运行新测试保证功能正确性，再提升性能→进度进展顺利



高泽宇@中国科学院大学
电子信息工程, 报名时大三

徐鑫@山东交通学院
电子信息工程, 报名时大一

三、“一生一芯”带来的能力提升

能够独立探索

- 1天上手Verilog
- 独立弄懂AXI
- 能够灵活运用Verilator
 - 修正计组实验的测试框架
 - 链接自己的项目

知识不是老师教会的
而是自己学会的

零基础

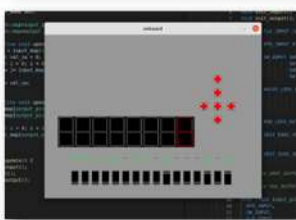


自学

将学习经验反馈
到课程建设



驾驭



孙际儒@南京大学
计算机科学与技术, 报名时大一

一、工欲善其事，必先利其器

^[1] 香山 开发过程中使用大量的高效开发的工具，包括但不限于
Verilator (高速仿真器)
NEMU(指令集模拟器)，
Diffest(差分验证框架)...

用了“一生一芯”建议的开发工具，我认识到原来优秀的工具可以对开发起到如此大的提升。现在我在开发的时候首先会想到完善基础设施。

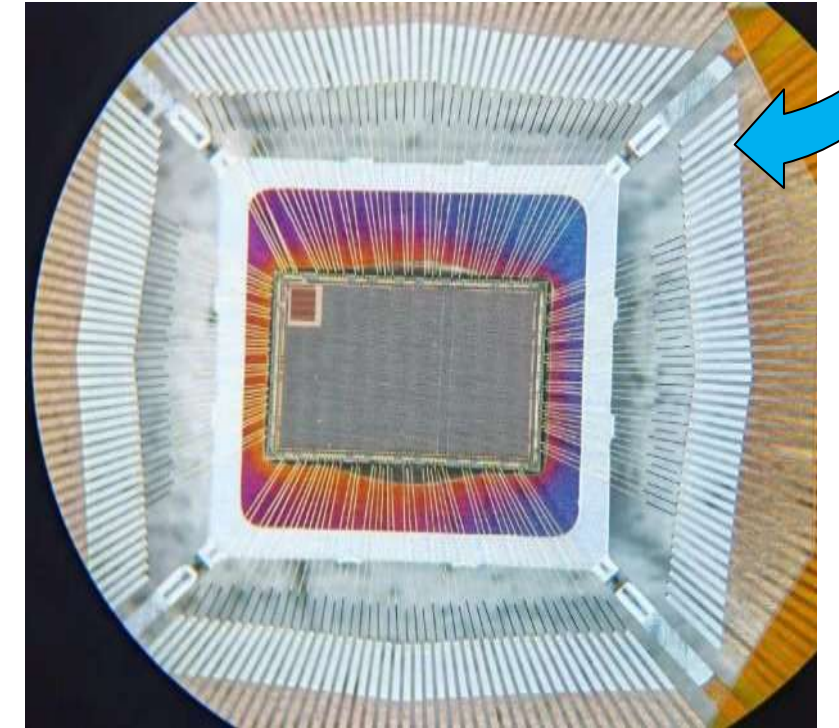
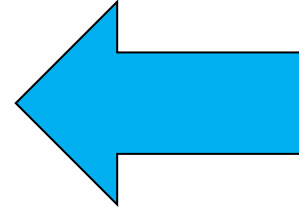
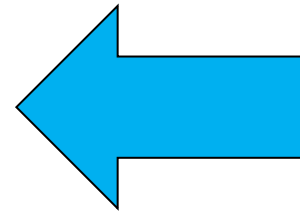
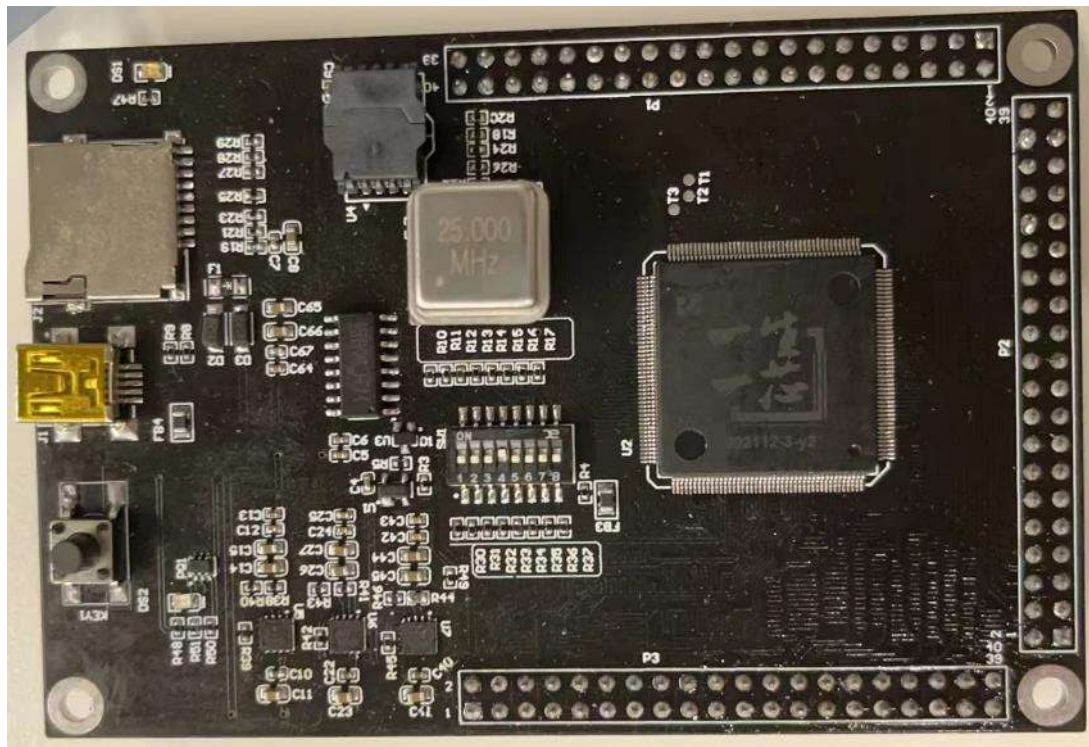
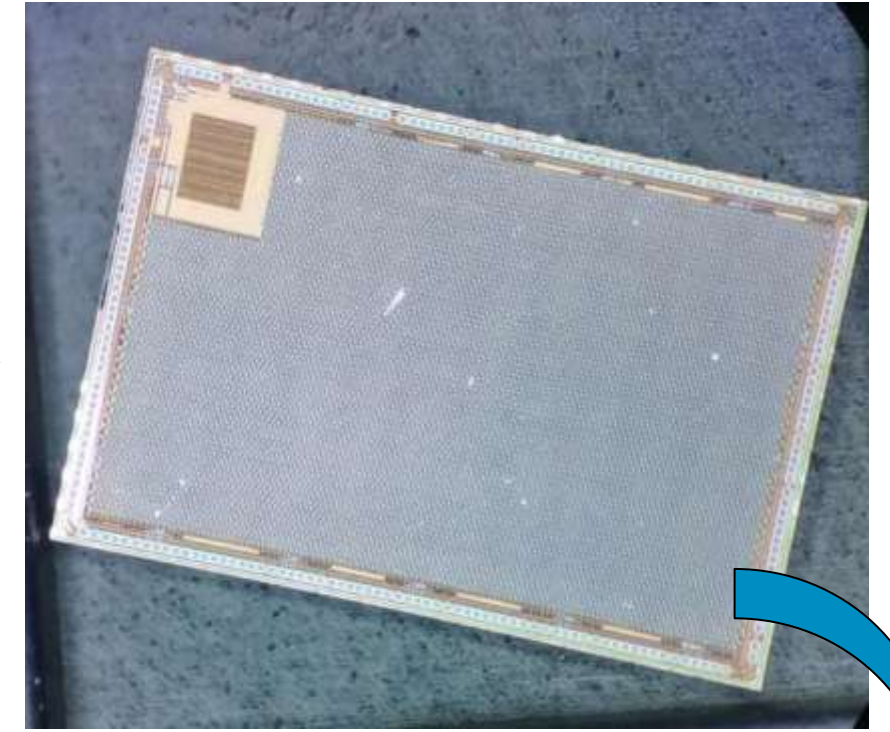
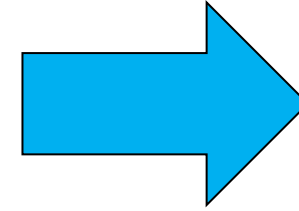
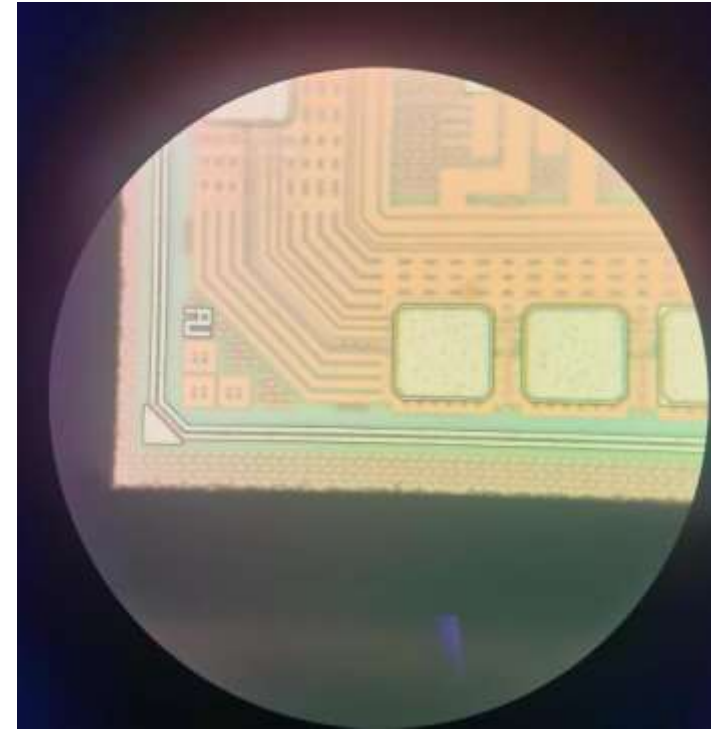
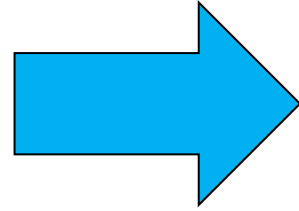
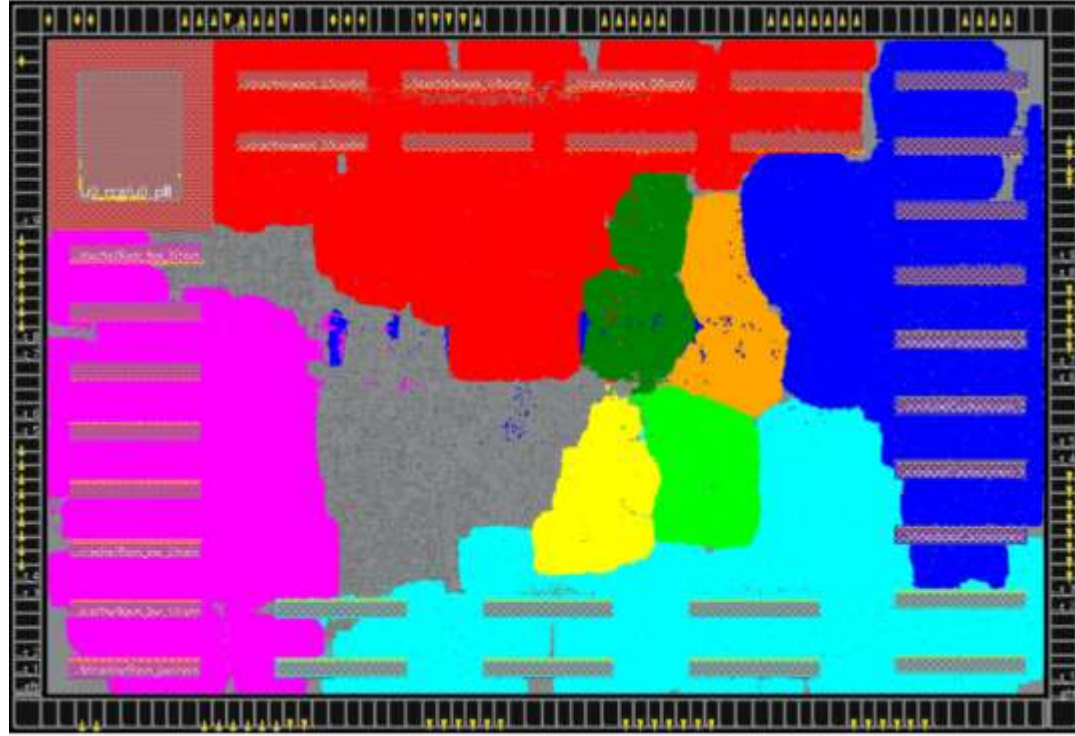
Verilator仿真coremark的速度
比Vivado快约300倍



[1] <https://github.com/OpenXiangShan>

栗金伦@太原理工大学
水利, 报名时大二

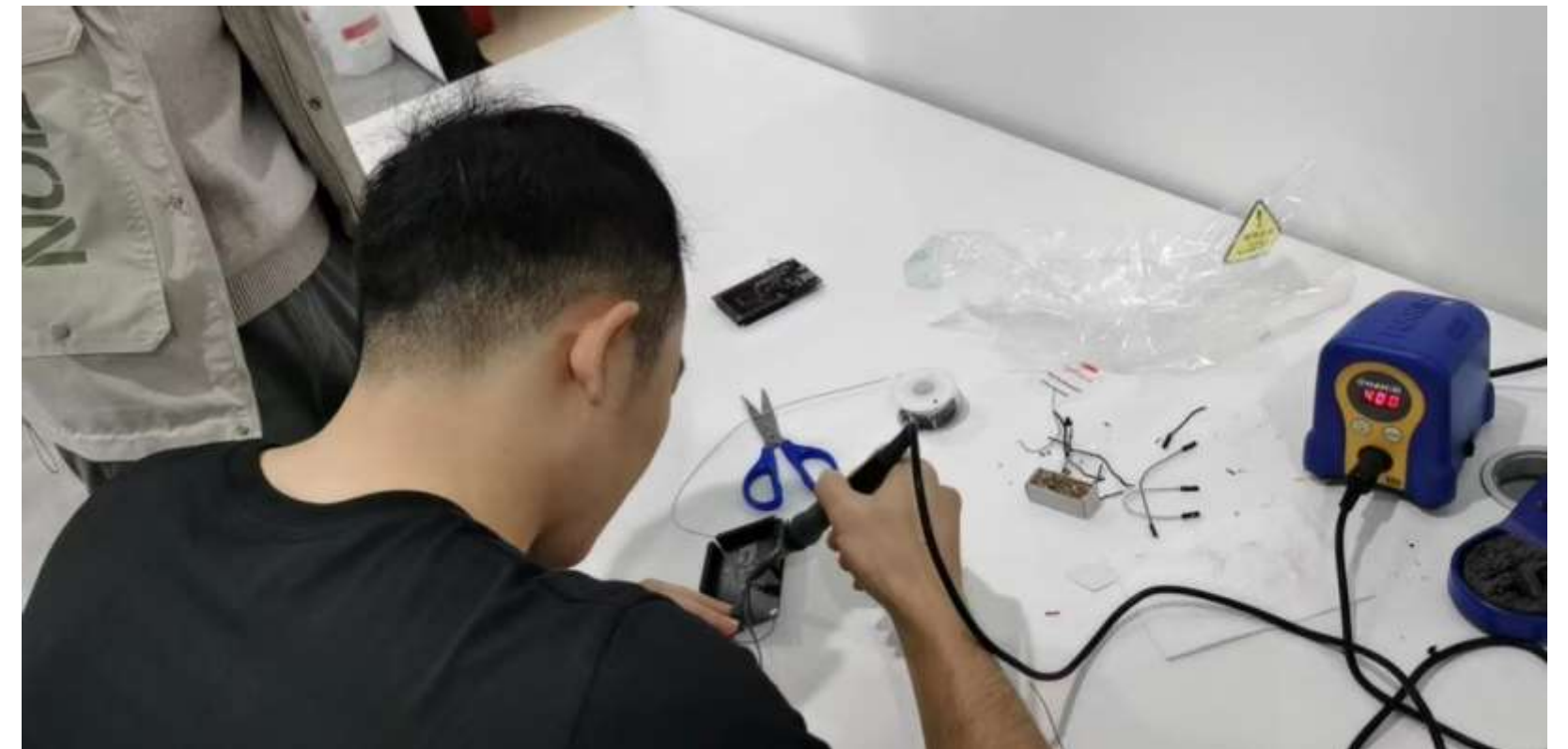
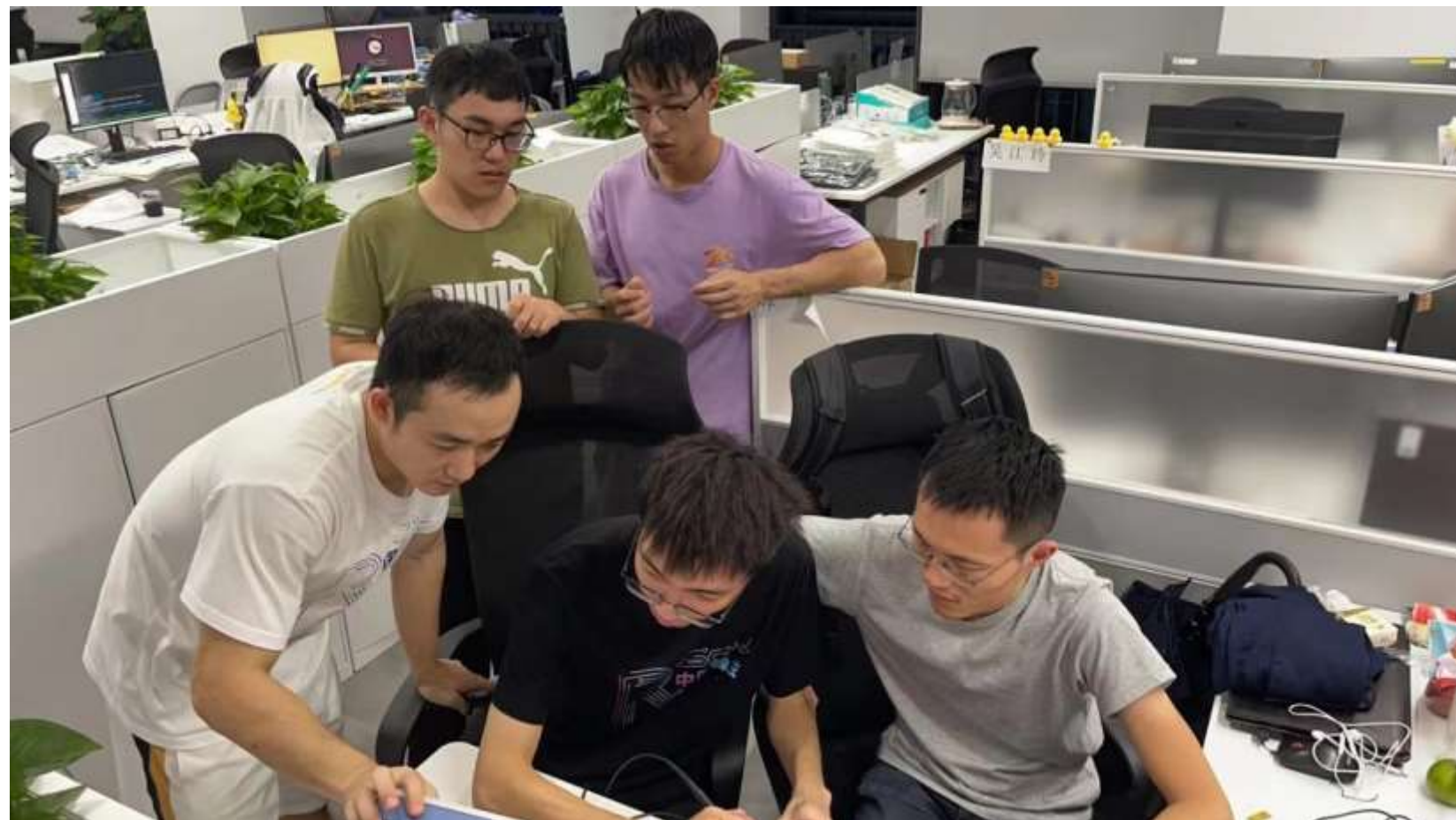
第三期处理器芯片和板卡展示



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(3fd0000), 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

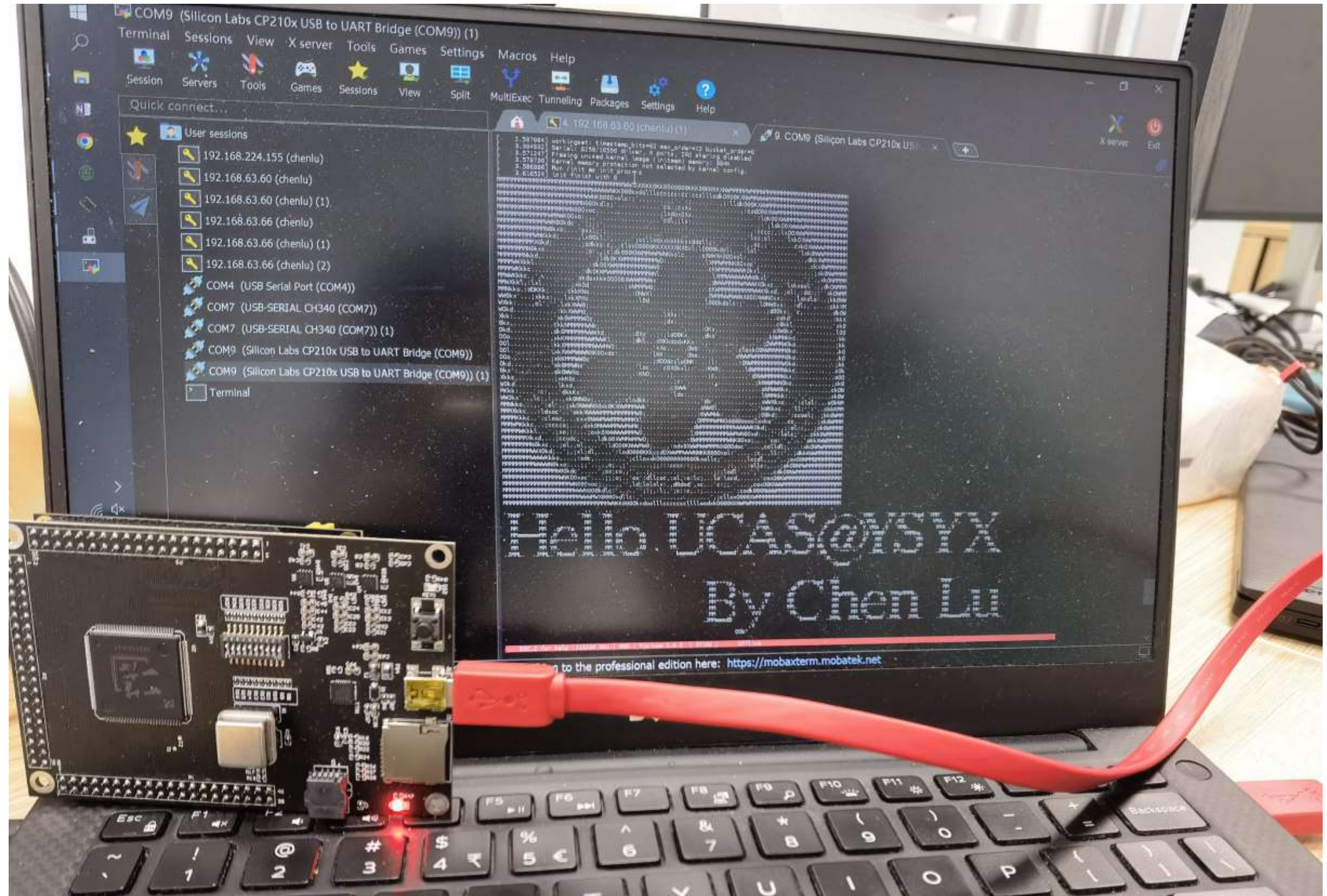
```
Load finished
Exec app...
heap: [0x80022590 - 0x86422590]

\ | /
- 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
```


第三期学生单人作品展示

- 陈璐@南京大学, 报名时大三, 计算机科学与技术
 - 从Flash中加载Linux并启动, 展示中科院logo



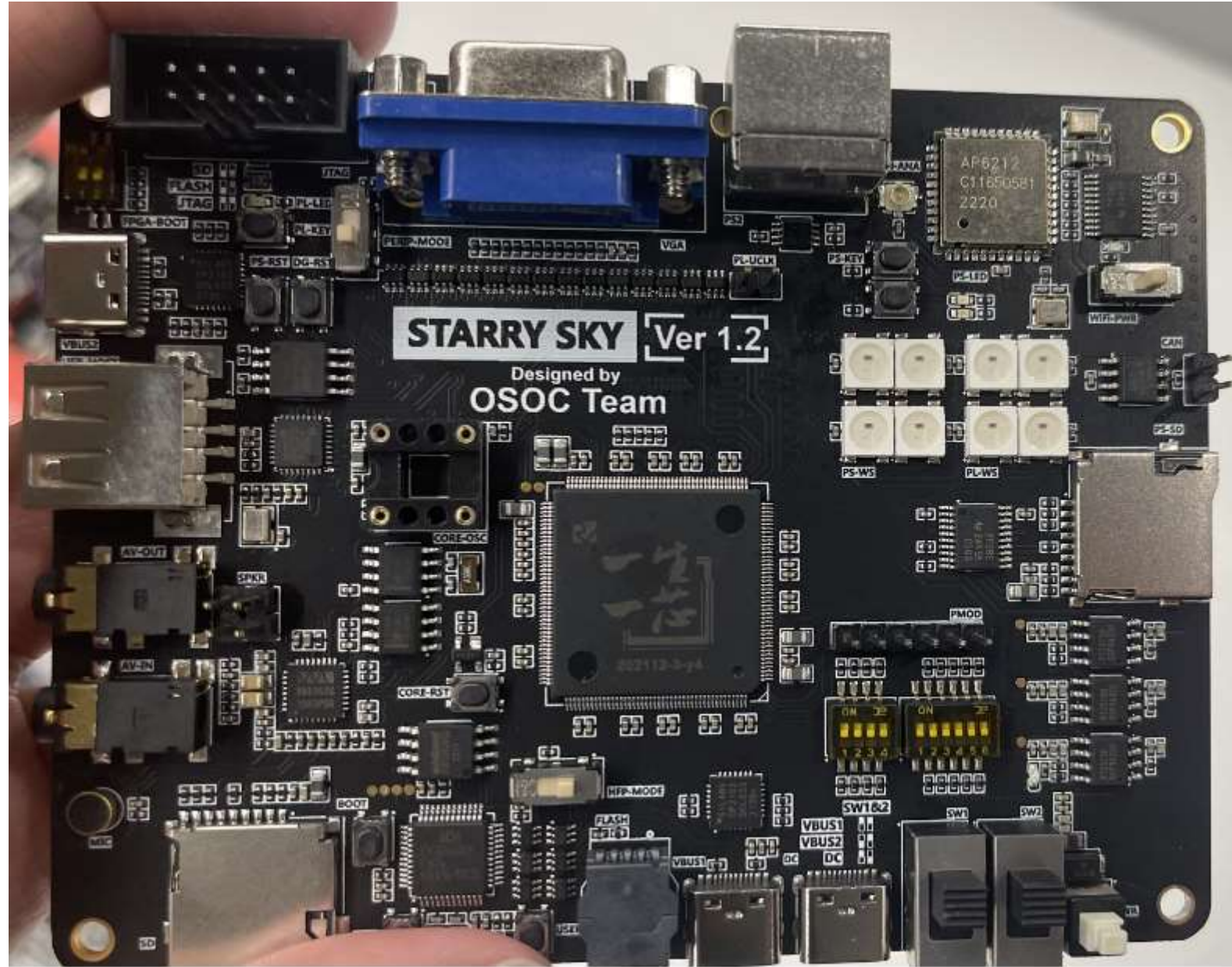
第三期学生单人作品展示(2)

- 唐浩晋@中国科学院大学, 报名时大三, 电子信息工程
 - 在Linux上运行字符版2048游戏



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

板卡设计和发放

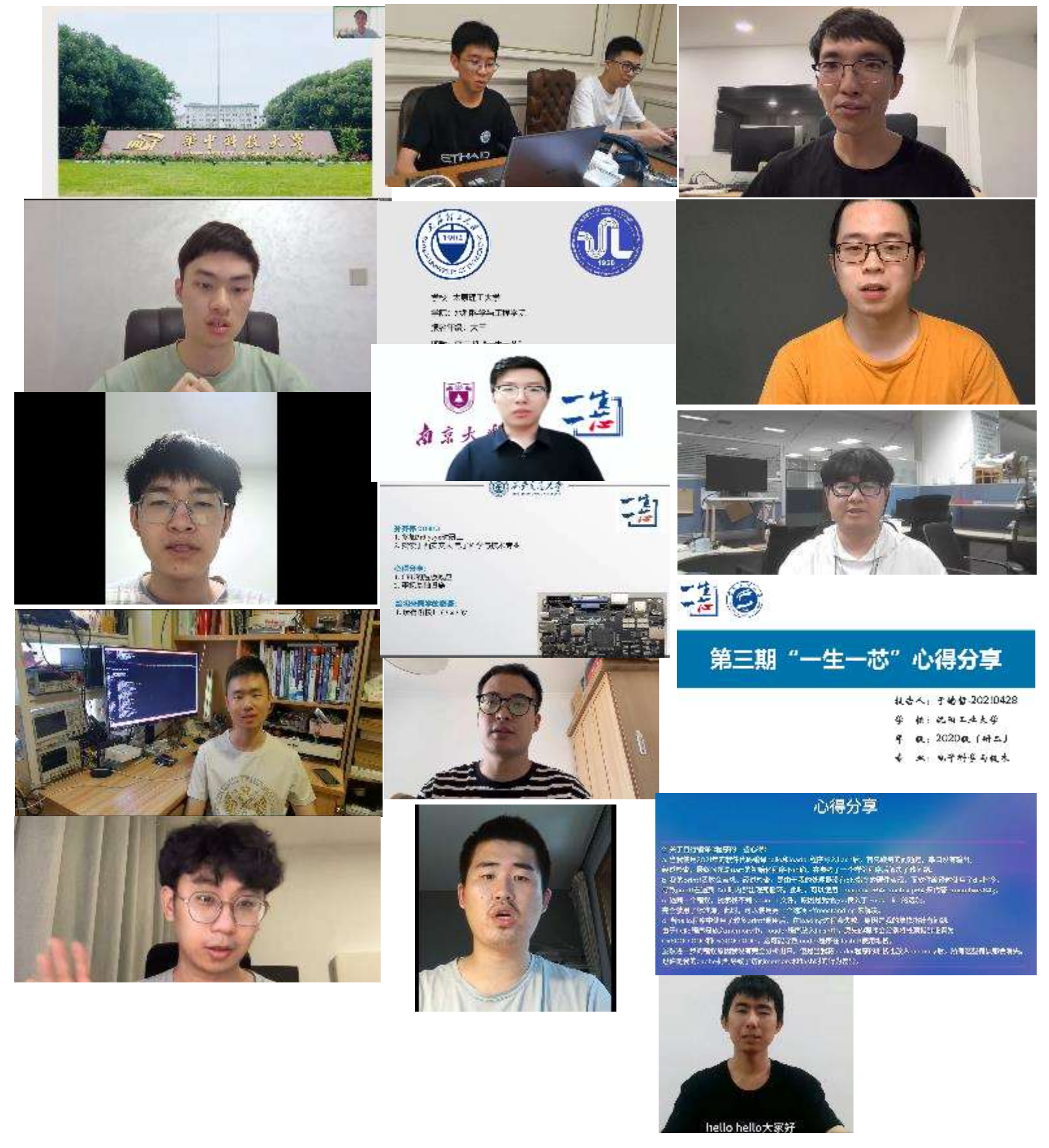
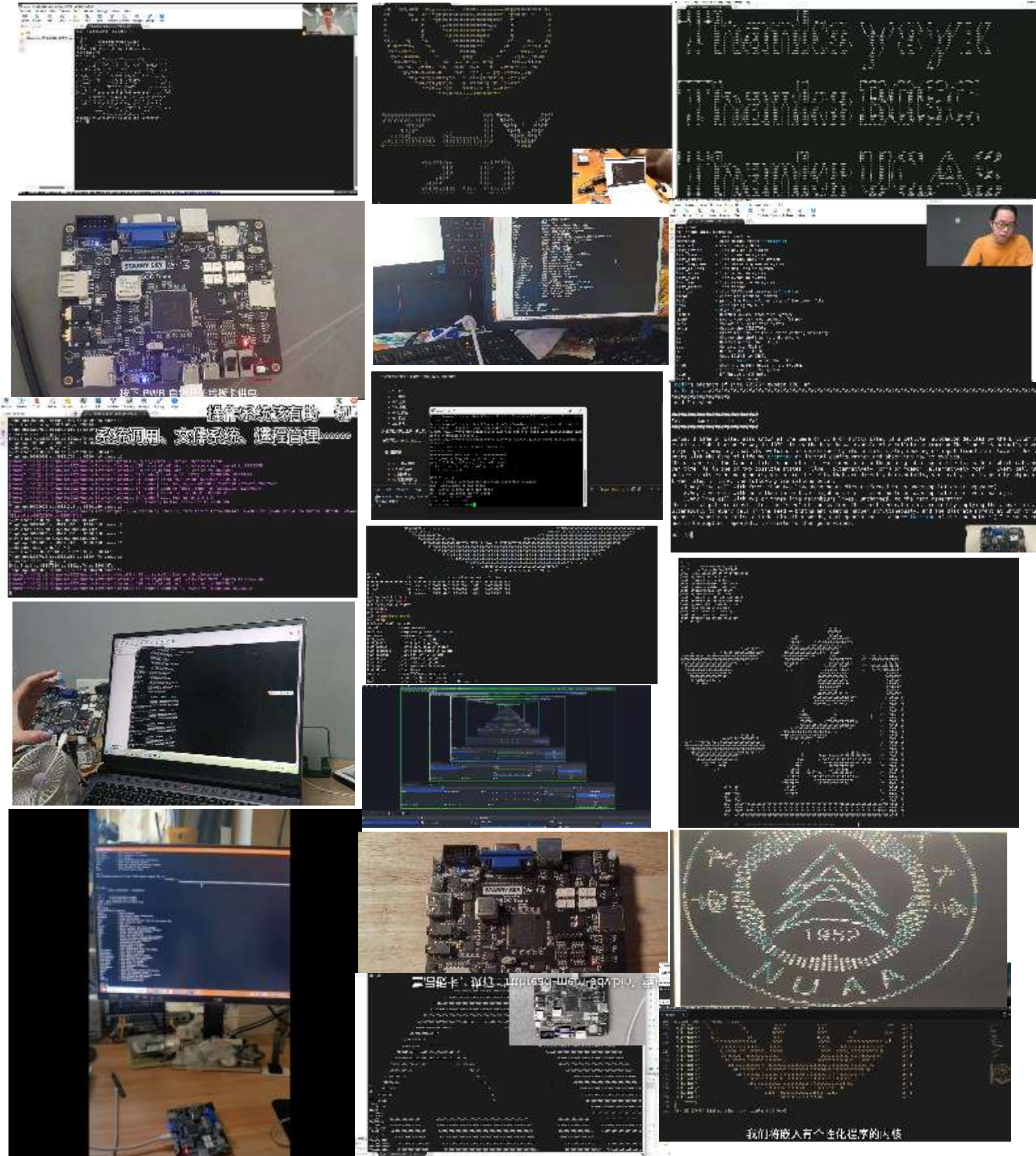


搭载“一生一芯”芯片的“星空”板卡



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



五、建设高标准、低坡度的 学习流程

第四~六期 (2022年2月~今)

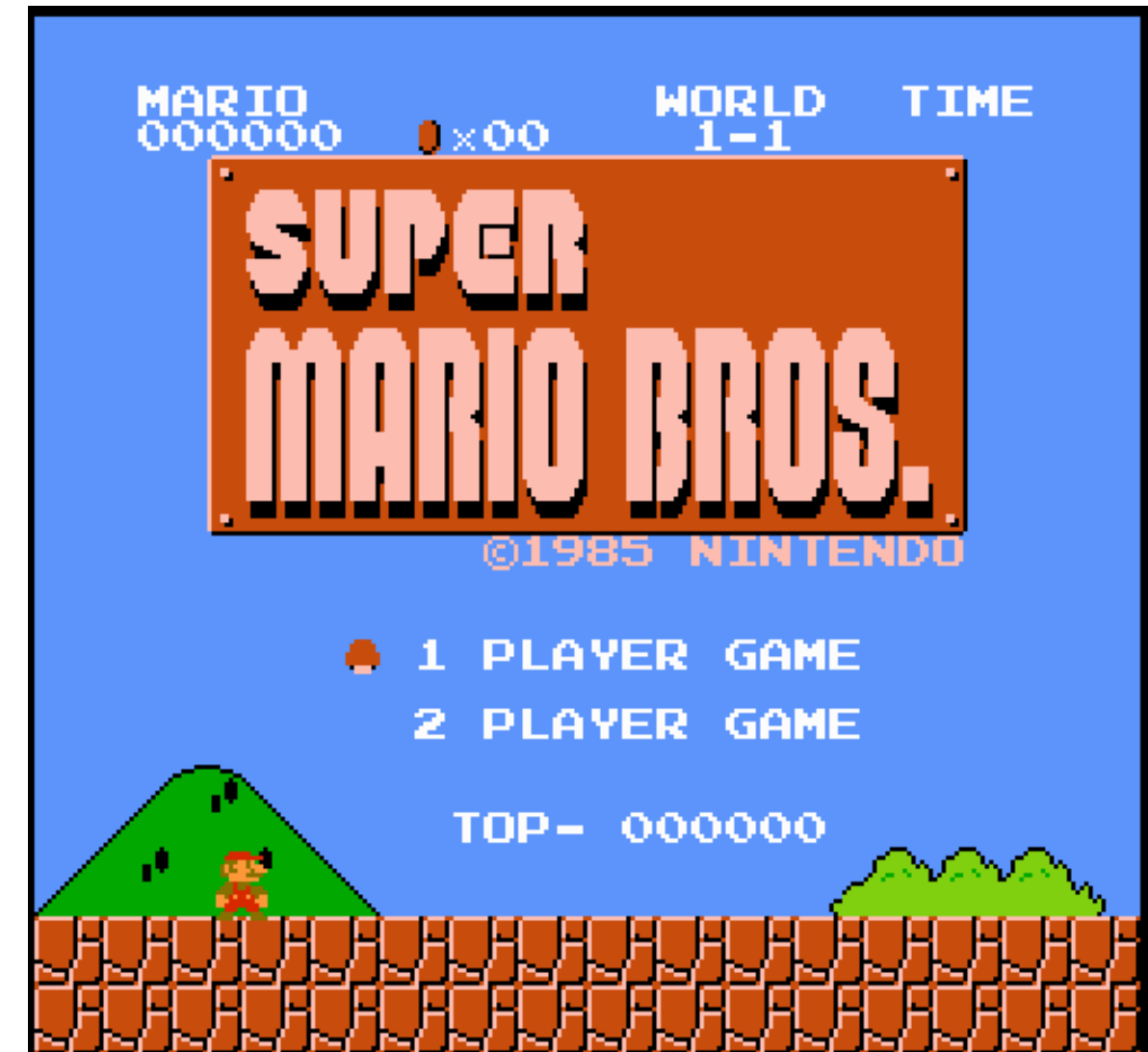
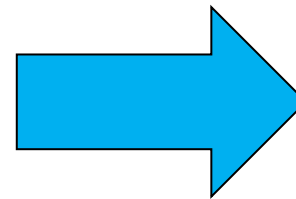
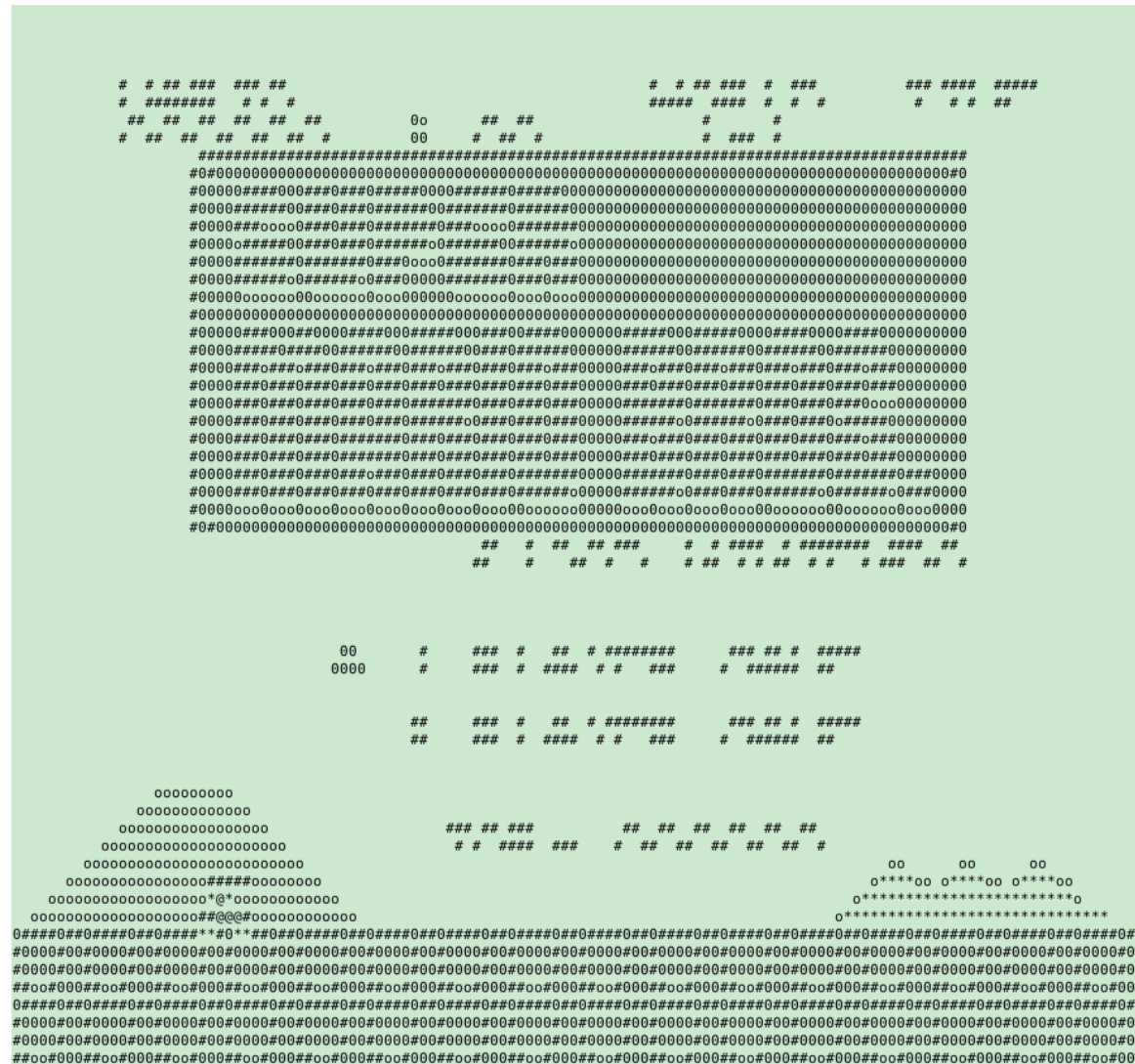
第四/五期

- 提高学习指标: RV64IM启动自制OS软件栈, 运行游戏 “仙剑奇侠传”
- 加入南京大学的PA实验, **强化软硬件协同的训练**
 - 自行实现模拟器/编译程序/编写自制OS和运行时环境/搭建仿真和测试环境...
- **完善讲义, 共26万字**(含PA部分)
 - 预学习/B(aseline)阶段/A(dvanced)阶段, 指导学生学习
 - 螺旋上升的融合式计算机系统设计
- **培养和流片分离**, 学习不设截止日期
- 取消组队, 只允许单人参加
- 第五期**新增教学视频和课件**

第四期学习成果

- 新增VGA, 提升展示效果

第三期的 字符版本



B阶段学习成果: 运行80年代游戏《超级玛丽》

第四期学习成果

- 新增VGA, 提升展示效果



A阶段学习成果:
运行90年代游戏《仙剑奇侠传》



S阶段学习成果（部分）：
运行00年代游戏《CLANNAD》

报名无需费用, 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
 - 📺 [B站直播](#) | [录播链接](#)
- 答疑: 每周日19:00~20:00 (通过预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](#) | [📖 复习C语言](#)
- [📺 从C语言到二进制程序](#) | [📖 T](#) | [📖](#)
- [📺 程序的执行和模拟器](#) | [📖 R](#) | [📖 P](#) | [📖](#) | [📖 搭建verilator仿真环境](#) | [📖 数字电路基础实验](#)
- [📺 NEMU代码导读](#) | [📖 T](#) | [📖](#) | [📖 完成PA1](#)

B阶段

- [📺 RISC-V指令集](#) | [📖 R](#) | [📖 P](#) | [📖](#) | [📖 支持RV64IM的NEMU](#)
- [📺 程序的机器级表示](#) | [📖 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

余子豪



中国科学院大学



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

一生一芯 官方文档

关于项目 | 学习讲义 | 课件准备 | 其它资料 | 导航

预学习阶段

预学习概述

如何科学地提问

Linux系统安装和基本使用

复习C语言

搭建verilator仿真环境

数字电路基础实验

完成PA1

提交预学习答疑申请

Linux系统安装和基本使用

📌 安装一个Linux操作系统

我们复用PA讲义的内容, 请大家根据PA0安装Linux操作系统.

📌 获取"一生一芯"框架代码

当你阅读PA0讲义, 并进行到获取PA框架代码的部分, 将会有提示框指引你回到此处讲义内容.

首先请在github上添加一个ssh key, 具体操作请STFW, 然后通过以下命令获取一生一芯的框架代码:

```
git clone -b ysyx2204 git@github.com:OSCPU/ysyx-workbench.git
```

获取后, 你就可以回到PA讲义的相应位置, 继续阅读了. 不过你还需要注意:

- 请把 `ysyx-workbench` 作为PA讲义中的项目目录, 即将PA讲义中的 `ics2022` 看成是 `ysyx-workbench`
- 修改 `ysyx-workbench/Makefile` 中的字母和姓名时, 请使用"一生一芯"的字母和真实姓名

这种来回跳转的做法可能会给你带来一些麻烦, 但我们之所以这样做, 是希望把文档看作代码来管理. 我们期望做到类似"一生一芯"讲义像代码讲义的效果, 因此我们在PA讲义中尽可能少地提到"一生一芯", 而把一生一芯的相关内容放到一生一芯本身的讲义中. 如果不遵守这条原则, 不仅会使得我们维护讲义时感到困难, 而且大家阅读讲义时也不知道应该到哪里寻找相关的内容.

! 安装系统是独立解决问题的最简单的训练

如果你是第一次安装并使用Linux, 你可能会遇到非常多的问题. 不用担心, 因为全世界都在使用Linux, 因此你遇到的问题, 很大概率别人也遇到过. 在互联网上搜索关键字, 很大概率就能找到解决方案.

📌 树立正确的价值观, 接受最大程度的训练

一生一芯 官方文档

关于项目 | 学习讲义 | 课件准备 | 其它资料 | 导航

TA的合集和视频列表 > [加载中] 第五期 "一生一芯"

当前: 19个视频 | 1-7条

课程主页 <https://ysyx.oscc.cc/docs/>

默认排序

升序排序

一生一芯 概述 [第五期 "一生一芯" 计划 - P1]

1:52

2022-6-8

Linux入门教程 [第五期 "一生一芯" 计划 - P2]

1:10

2022-9-12

计算机系统的状态机模型 [第五期 "一生一芯" 计划 - P3]

8:54

2022-9-17

从C语言到二进制程序 [第五期 "一生一芯" 计划 - P4]

2:18

2022-9-25

程序的执行和模拟器 [第五期 "一生一芯" 计划 - P5]

2:48

2022-10-8

NEMU代码导读 [第五期 "一生一芯" 计划 - P6]

2:49

2022-10-9

一生一芯 概述 [第五期 "一生一芯" 计划 - P7]

4:38

2022-10-15

程序的机器级表示 [第五期 "一生一芯" 计划 - P8]

1:15

2022-10-22

RISC-V指令集 [第五期 "一生一芯" 计划 - P9]

3:42

2022-10-28

Abstract Machine裸机运行时环境 [第五期 "一生一芯" 计划 - P10]

1:21

2022-11-5

ELF文件和链接 [第五期 "一生一芯" 计划 - P11]

1:02

2022-11-13

工具和基础设施 [第五期 "一生一芯" 计划 - P12]

1:02

2022-11-20

一生一芯 概述 [第五期 "一生一芯" 计划 - P13]

1:52

2022-11-27

请读者阅读 [第五期 "一生一芯" 计划 - P14]

3:15

2022-12-8

异常处理 [第五期 "一生一芯" 计划 - P15]

9:51

2022-12-16

计算机系统软件栈 [第五期 "一生一芯" 计划 - P16]

2:10

2022-12-19

总结回顾 [第五期 "一生一芯" 计划 - P17]

1:11

2022-12-25

SoC计算机系统 [第五期 "一生一芯" 计划 - P18]

1:13

2022-11-20

一生一芯 概述 [第五期 "一生一芯" 计划 - P19]

1:52

2022-11-27

请读者阅读 [第五期 "一生一芯" 计划 - P20]

3:15

2022-12-8

异常处理 [第五期 "一生一芯" 计划 - P21]

9:51

2022-12-16

计算机系统软件栈 [第五期 "一生一芯" 计划 - P22]

2:10

2022-12-19

总结回顾 [第五期 "一生一芯" 计划 - P23]

1:11

2022-12-25

SoC计算机系统 [第五期 "一生一芯" 计划 - P24]

1:13

2022-11-20

B站账号: 一生一芯-视频号

第五期 一生一芯 | 周六 19:00~21:00 | Sat 08 Oct 2022 08:37:32

课程主页 <https://ysyx.oscc.cc/docs/>

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

展开的结果不好阅读

使用代码格式化工具

课程主页 <https://ysyx.oscc.cc/docs/>

一生一芯 官方文档

关于项目 | 学习讲义 | 课件准备 | 其它资料 | 导航

预学习阶段

预学习概述

如何科学地提问

Linux系统安装和基本使用

复习C语言

搭建verilator仿真环境

数字电路基础实验

完成PA1

提交预学习答疑申请

Linux系统安装和基本使用

📌 安装一个Linux操作系统

我们复用PA讲义的内容, 请大家根据PA0安装Linux操作系统.

📌 获取"一生一芯"框架代码

当你阅读PA0讲义, 并进行到获取PA框架代码的部分, 将会有提示框指引你回到此处讲义内容.

首先请在github上添加一个ssh key, 具体操作请STFW, 然后通过以下命令获取一生一芯的框架代码:

```
git clone -b ysyx2204 git@github.com:OSCPU/ysyx-workbench.git
```

获取后, 你就可以回到PA讲义的相应位置, 继续阅读了. 不过你还需要注意:

- 请把 `ysyx-workbench` 作为PA讲义中的项目目录, 即将PA讲义中的 `ics2022` 看成是 `ysyx-workbench`
- 修改 `ysyx-workbench/Makefile` 中的字母和姓名时, 请使用"一生一芯"的字母和真实姓名

这种来回跳转的做法可能会给你带来一些麻烦, 但我们之所以这样做, 是希望把文档看作代码来管理. 我们期望做到类似"一生一芯"讲义像代码讲义的效果, 因此我们在PA讲义中尽可能少地提到"一生一芯", 而把一生一芯的相关内容放到一生一芯本身的讲义中. 如果不遵守这条原则, 不仅会使得我们维护讲义时感到困难, 而且大家阅读讲义时也不知道应该到哪里寻找相关的内容.

! 安装系统是独立解决问题的最简单的训练

如果你是第一次安装并使用Linux, 你可能会遇到非常多的问题. 不用担心, 因为全世界都在使用Linux, 因此你遇到的问题, 很大概率别人也遇到过. 在互联网上搜索关键字, 很大概率就能找到解决方案.

📌 树立正确的价值观, 接受最大程度的训练

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
- 课程主页 - 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>

助教团队(部分)

- 选拔学生担任助教, 支撑超过百名同学的学习
 - 引导学生思考, 而不是直提供解决方案
 - 给学生最大的成长机会

部分助教名单(2023年6月时)

序号	姓名	学校	年级
1	苗金标	中国科学技术大学	研二
2	段震伟	中国科学技术大学	研三
3	刘汉章	太原理工大学	大三
4	曹勋	中国科学技术大学	研二
5	杨海帆	浙江工商大学	大四
6	曹世洋	中国科学技术大学	研二
7	倪仁涛	东北大学	研一
8	魏人	兰州大学	大四
9	吴佳宾	青岛大学	研一
10	陈璐	中国科学院大学	博一
11	栗金伦	太原理工大学	大四



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

助教团队(第四/第五期)

- 助教工作: 技术答疑, 组织组会和考核, 了解学生的学习进展
- 第四期助教
 - 线下助教: 邓海文(海南大学), 段震伟(中国科学技术大学), 刘一鸣(华中科技大学), 梅晓龙(海南大学)
 - 线上助教: 陈泱宇(重庆大学), 冯浩原(中国科学院大学), 李志锐(厦门大学), 刘知杭(四川建筑职业技术学院), 罗昊洋(北京大学), 倪仁涛(东北大学), 栗金伦(太原理工大学), 曾广森(华南理工大学)
- 第五期助教
 - 线下助教: 曹勋(中国科学技术大学), 曹世洋(中国科学技术大学), 邓海文(海南大学), 段震伟(中国科学技术大学), 刘汉章(太原理工大学), 卢非凡(西安财经大学), 马建露(北京大学), 梅晓龙(海南大学), 苗金标(中国科学技术大学)
 - 线上助教: 陈泱宇(重庆大学), 李志锐(厦门大学), 罗昊洋(北京大学), 倪仁涛(东北大学), 栗金伦(太原理工大学), 孙际儒(南京大学), 魏人(兰州大学), 吴佳宾(青岛大学), 徐鑫(山东交通学院), 曾广森(华南理工大学), 赵博涵(杭州电子科技大学)

设置代码考核环节

- 考核软硬件系统观和工具使用等技能
- 考核独立解决问题的能力

所以

我们来玩真的: 在线调试考核, 通过才能获得流片机会

- 助教在你提交的项目中随机注入3个bug 😈
 - 涵盖硬件, 软件和环境
- 你需要在30分钟内排除bug 😈
 - 强迫大家理所有细节, 吸收科学的调试方法
 - 参考代码, 抱大腿, “看波形就够” 都没戏 😈
 - 抓紧每一次调试锻炼的机会
 - 突击是没用的, 考的就是真本事 😈
 - 抱一次大腿 = 少一次训练

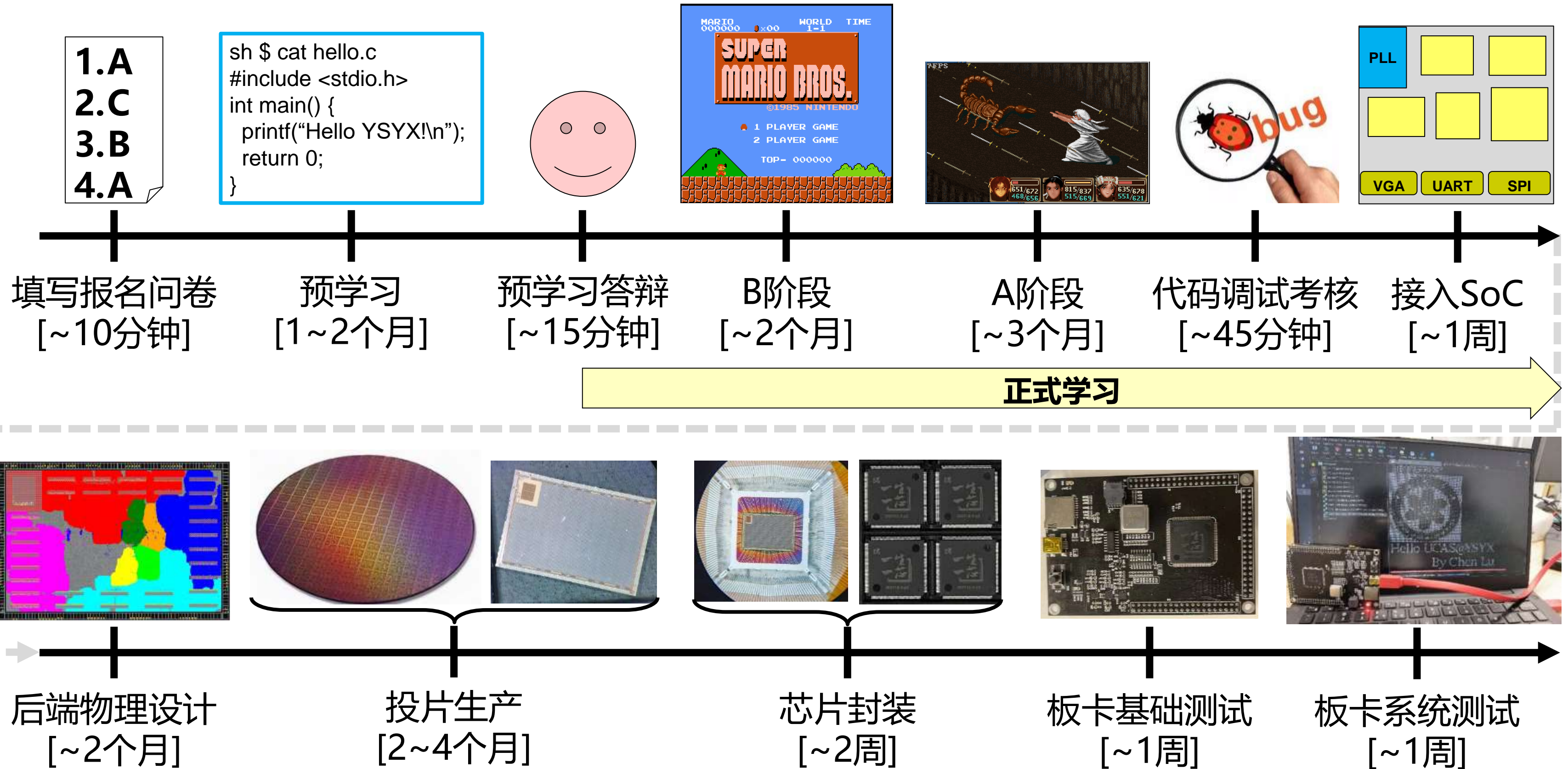


一下子Accepted不了吧

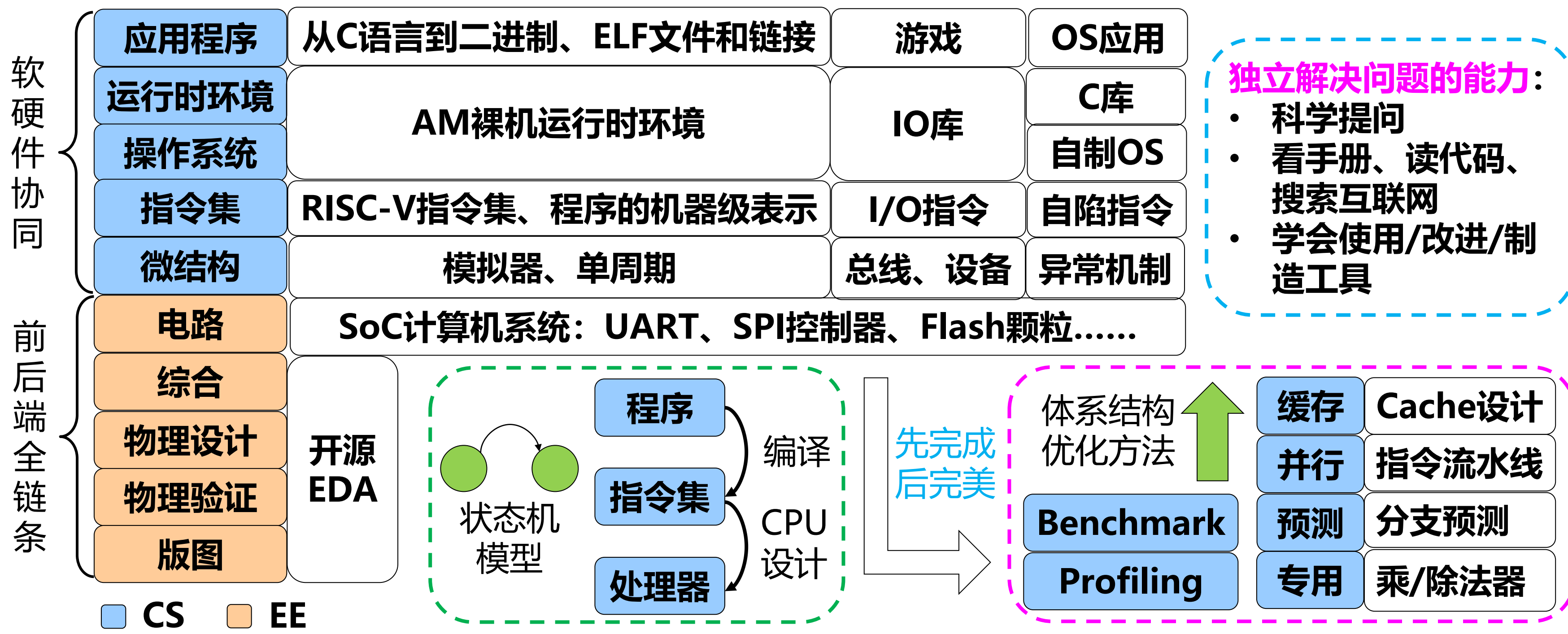
这和大家将来遇到的实际问题非常接近

- 如果你能解决实际问题, 你就是专家

第四/五期学习路线图



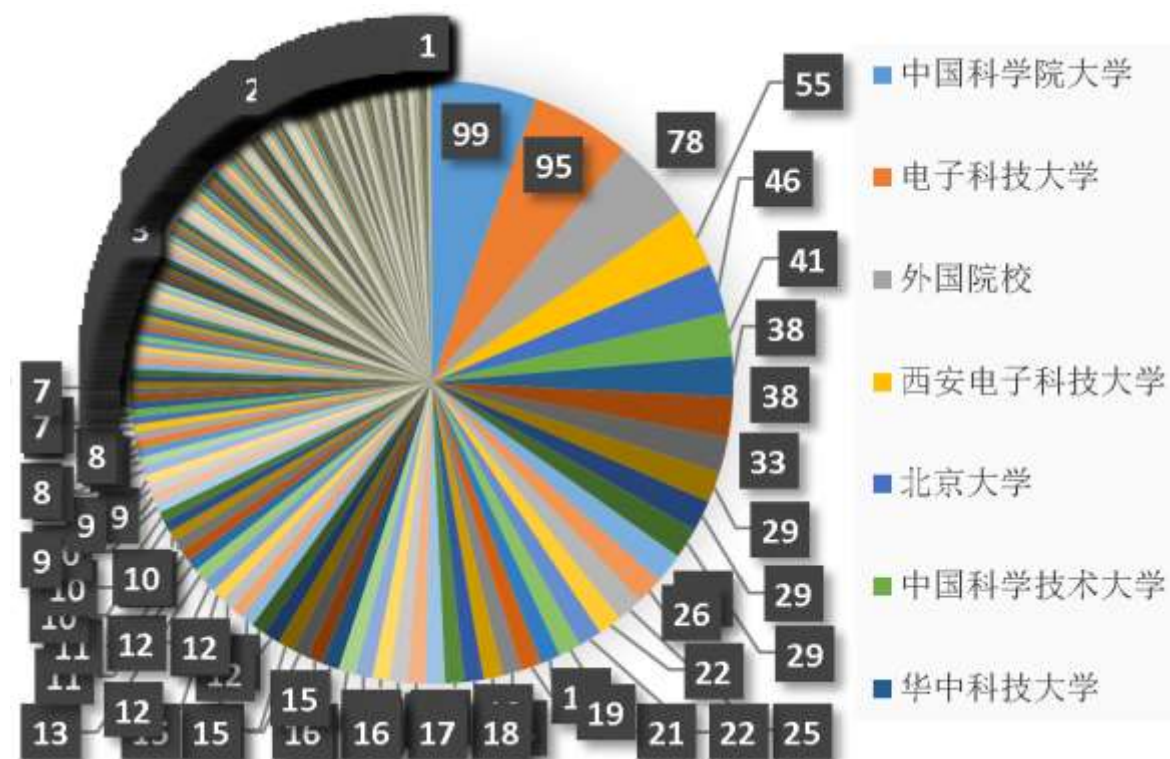
知识图谱



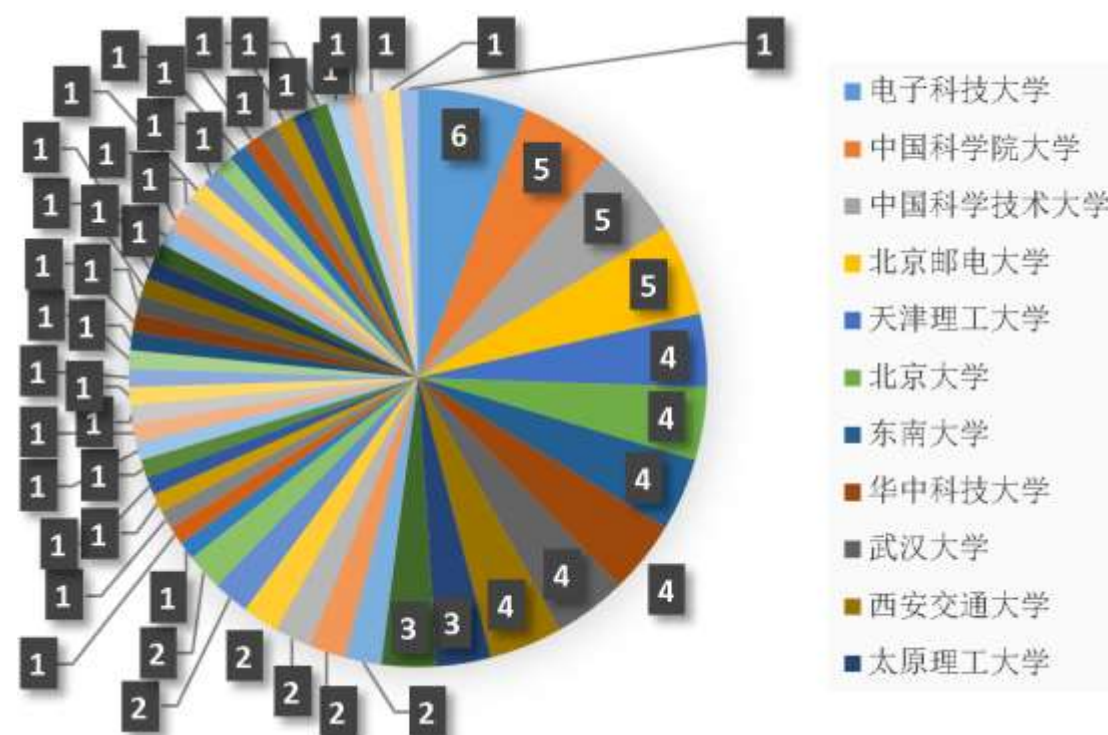
解决工程问题的基础设施: lint工具、sanitizer、printf、trace、gdb、波形、profiler.....
使用正确的模式写出好代码 (不言自明、不言自证)、测试、断言

第四期 “一生一芯” 学习情况统计(2023年2月)

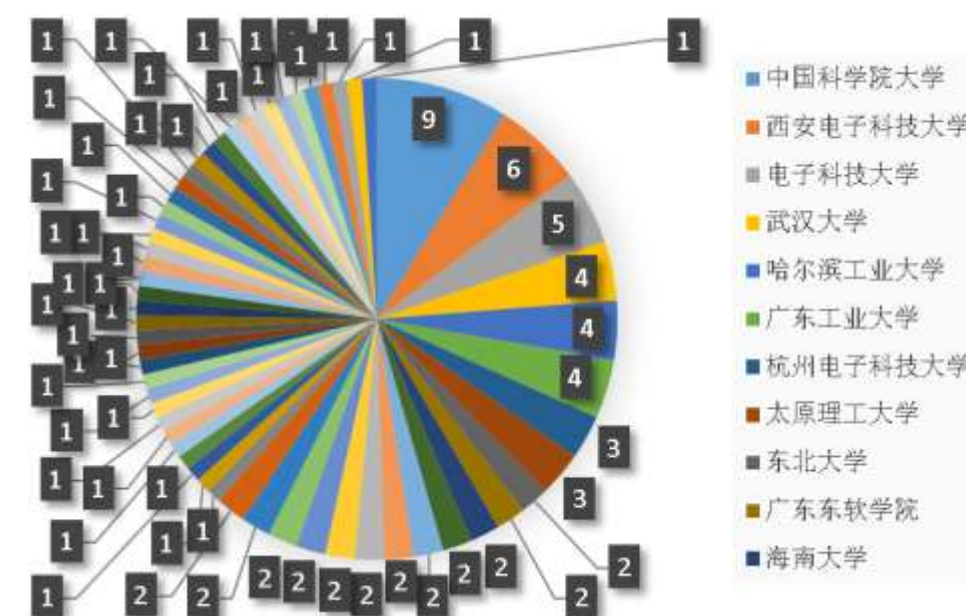
报名(1753人)



学习到B阶段(98人)

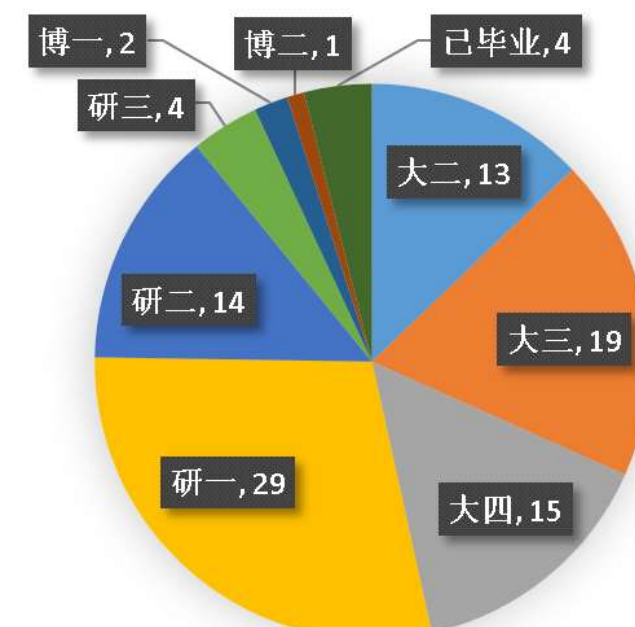
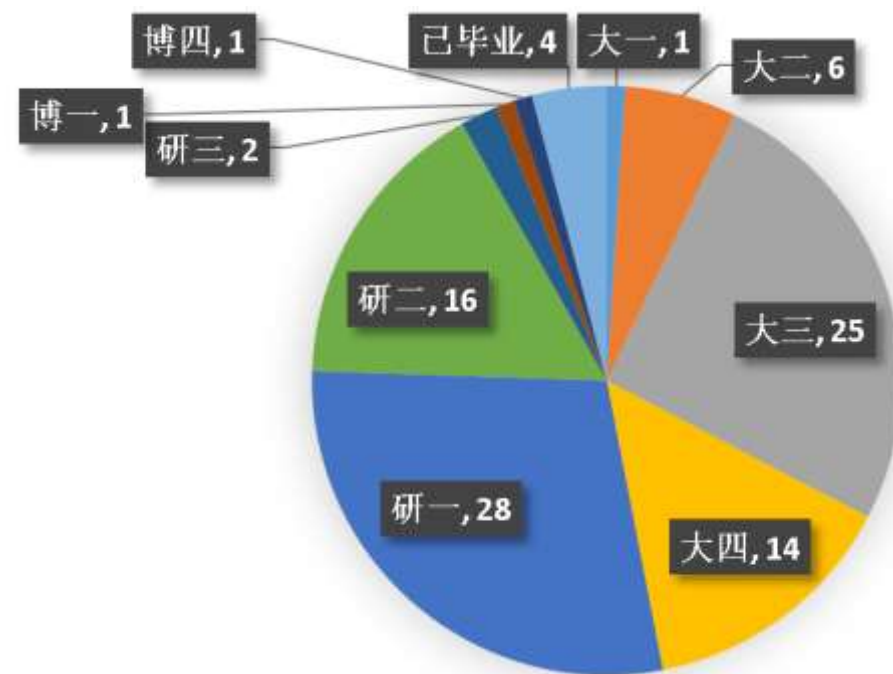
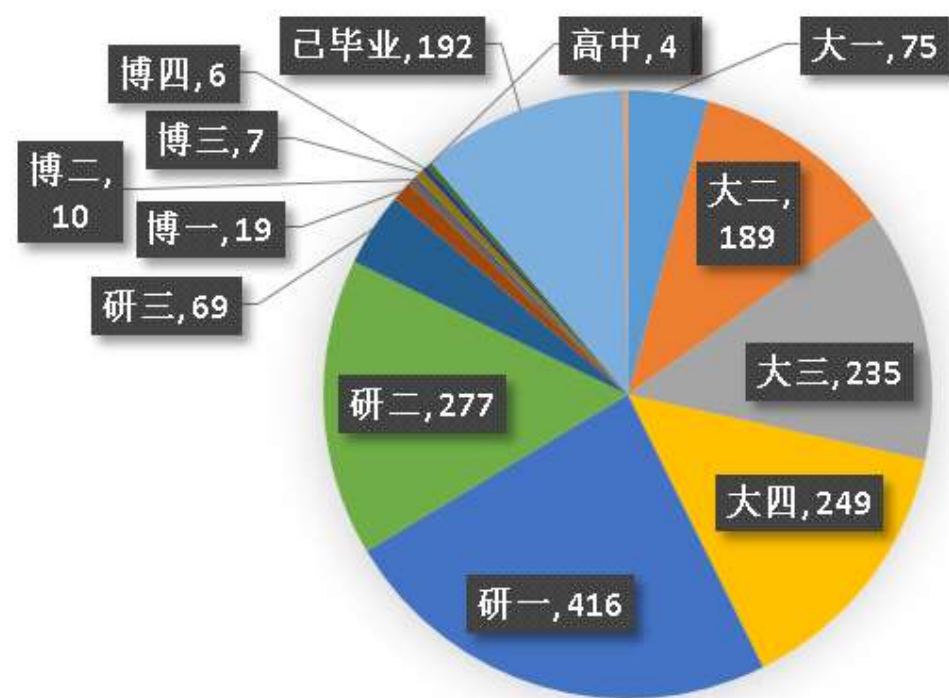


学习到A阶段(101人)



高校分布

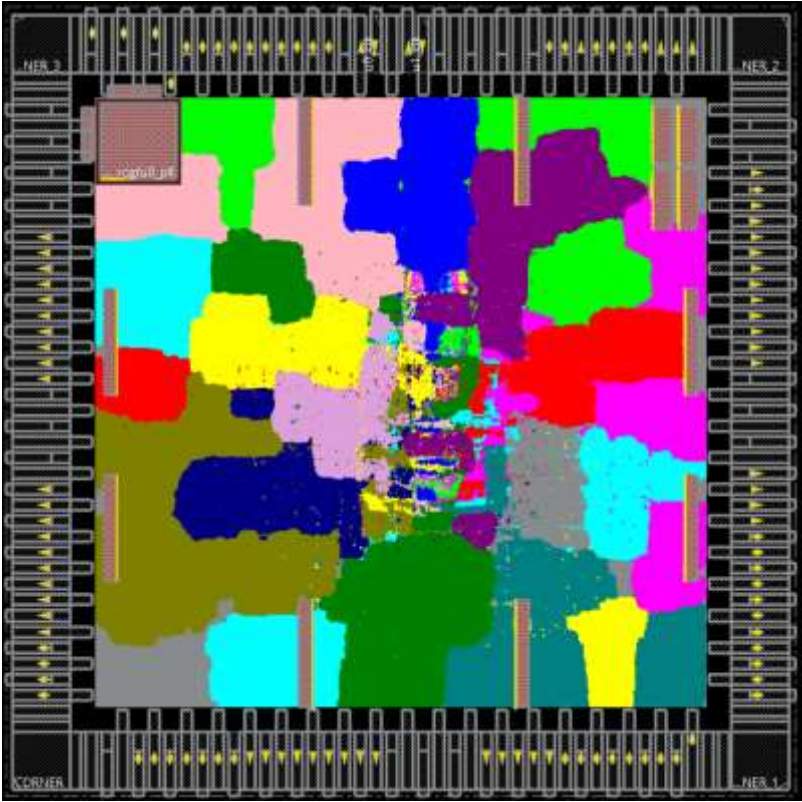
年级分布



数据整理：“一生一芯”学习成长追踪小组

第四期 “一生一芯” 流片名单

序号	学号	学校	专业	年级 (报名时)
1	ysyx_22040053	南京大学	计算机科学与技术	大二
2	ysyx_22040066	南京大学	计算机科学与技术	大二
3	ysyx_22040656	中国计量大学	计算机科学与技术	大二
4	ysyx_22040163	南通大学	计算机科学与技术	大二
5	ysyx_22040091	中国科学院大学	计算机科学与技术	大三
6	ysyx_22040596	华南理工大学	电子科学与技术	大四
7	ysyx_22041812	南方科技大学	电子科学与技术	研零6月
8	ysyx_22040127	东南大学	人工智能	研一
9	ysyx_22040654	福州大学	集成电路工程	研一
10	ysyx_22040978	中国科学院大学	电子信息	研一
11	ysyx_22041514	杭州电子科技大学	计算机技术	研一
12	ysyx_22040213	中国科学院大学	微电子学与固体电子学	研二
13	ysyx_22040561	北京大学	集成电路工程	研二
14	ysyx_22041461	四川大学	微电子科学与工程	大二
15	ysyx_22040886	北京理工大学	电子信息	大三
16	ysyx_22050228	东北大学	电子科学与技术	大三
17	ysyx_22050920	杭州电子科技大学	电子科学与技术	大四
18	ysyx_22040501	哈尔滨工业大学	信息与通信工程	研一
19	ysyx_22050133	北京大学	机械	研二



第一批流片SoC

- ysyx_040053
- ysyx_040066
- ysyx_040091
- ysyx_040127
- ysyx_040163
- ysyx_040213
- ysyx_040561
- ysyx_040596
- ysyx_040654
- ysyx_040656
- ysyx_040978
- ysyx_041514
- ysyx_041812

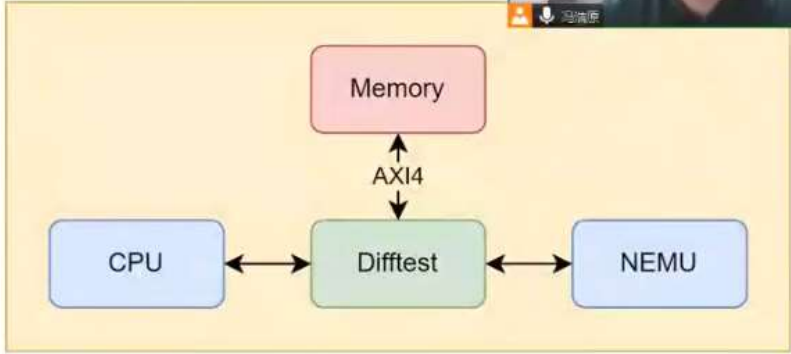
第四期 “一生一芯” 学生学习心得分享

- 冯浩原@中国科学院大学, 计算机科学与技术, 报名时大三
 - 大四时在北京开源芯片研究院实习, 成为 “香山” 团队的技术骨干
 - <https://www.bilibili.com/video/BV1C54y1T7hw>



心得分享①：理解系统

- “一生一芯” 仿真系统
 - 逐步建立了对系统的认识, 增加调试的信心与效率
 - 仔细阅读源码和手册说明
 - 通过阅读源码, 了解 NEMU 和 Spike 的对比机制
 - 接入 ysyxSoC 后, 为什么无法从 SRAM 中读出数据
- “香山” 团队中的学习、工作
 - 处理器架构和验证框架复杂很多
 - 主要负责 MMU 模块的开发
 - 涉及到和前端取指、后端访存、内存访问的交互
 - 包含 L1 TLB、L2 TLB、PMP、PMA 等子模块
 - 实践一生一芯的系统观念
 - 从整体入手, 关注前端取指和后端访存对 MMU 的查询行为
 - 关注细节, 理清 MMU 对于页表查询请求的处理过程
 - 仔细阅读源码, 不放过每一处内容, 对模块充分掌握



“一生一芯” 仿真系统示意图

端口	说明
output [127:0] Q	读数据
input CLK	时钟
input CEN	使能信号, 低电平有效
input WEN	写使能信号, 低电平有效
input [5:0] A	读写地址
input [127:0] D	写数据

YsyxSoC 中对 SRAM 行为的描述



第四期 “一生一芯” 学生学习心得分享

成长2: Debug能力的提升——difftest的应用

- 将difftest的思想应用到其他科研项目中
- 帮助我快速验证算法的正确性, 节省大量时间

nemu中的difftest

```
Welcome to riscv64-NEMU!
For help, type "help"
gpr[0] different after executing inst at pc = 80000000, right = 0 wrong = 1
pc 0x0000000000000004
s0 0x0000000000000000 ra 0x0000000000000000 sp 0x0000000000000000 gp 0x0000000000000000
t0 0x0000000000000000 t1 0x0000000000000000 t2 0x0000000000000000
s0 0x0000000000000001 s1 0x0000000000000000 a0 0x0000000000000000 a1 0x0000000000000000
a2 0x0000000000000000 a3 0x0000000000000000 a4 0x0000000000000000 a5 0x0000000000000000
a6 0x0000000000000000 a7 0x0000000000000000 s2 0x0000000000000000 s3 0x0000000000000000
s4 0x0000000000000000 s5 0x0000000000000000 s6 0x0000000000000000 s7 0x0000000000000000
s8 0x0000000000000000 s9 0x0000000000000000 s10 0x0000000000000000 s11 0x0000000000000000
t3 0x0000000000000000 t4 0x0000000000000000 t5 0x0000000000000000 t6 0x0000000000000000
mtvec :0000000000000000
mepc :0000000000000000
mstatus:0000000000001000
mcause :0000000000000000
[src/cpu/cpu-exec.c:241:cpu_exec] nemu: Abort at pc = 0x0000000000000000
```

SpMV算法科研项目中的difftest

```
check row 0, ans = 17.000000, res = 17.000000, pass
check row 1, ans = 35.000000, res = 35.000000, pass
check row 2, ans = 21.000000, res = 10.000000, wrong
check row 3, ans = 16.000000, res = 16.000000, pass
check row 4, ans = 56.000000, res = 56.000000, pass
check row 5, ans = 33.000000, res = 33.000000, pass
check row 6, ans = 18.000000, res = 18.000000, pass
```

2023/2/3

博学 慎思 明辨 笃行

4

2023/02/03 13:59:25

曾广森@华南理工大学

电子科学与技术, 报名时大四

收获: 将学到的方法和思想应用到科研项目

<https://www.bilibili.com/video/BV1sY41167kE/>

体会2: KISS

- Keep It Simple, Stupid 从易到难、逐步推进
- 将复杂的目标进行分解, 为自己做好阶段性的规划
- 勤写代码勤踩坑



我的任务分解方案



王晨宇@南通大学

计算机科学与技术, 报名时大二

收获: 学会分解复杂目标, 逐步解决新问题

<https://www.bilibili.com/video/BV1g24y1G7Hw/>

第六期新增B阶段流片, 提升A阶段指标

	第三期	第四/五期	第六期(B阶段)	第六期(A阶段)
ISA	RV64I	RV64IM	RV32E	RV64IMAC
处理器	流水线	流水线	流水线	流水线
乘除法器	-	有	-	有
异常	M模式ecall	M模式ecall	M模式ecall	M/S/U所有异常
分页	-	-	-	有
中断	时钟	时钟	-	时钟, 软件, 外部
总线	AXI4	AXI4	AXI4	AXI4
Cache	-	4KB指令+4KB数据	-	16KB指令+16KB数据
IPC	-	-	-	高于X
主频	-	-	-	高于Y
目标程序	RT-Thread	自制OS+仙剑	RT-Thread	Linux内核+发行版
系统软件	-	模拟器, AM, 自制OS	模拟器, AM	模拟器, AM, 自制OS
环境和工具	-	DiffTest, trace, 仿真环境.....	DiffTest, trace, 仿真环境.....	DiffTest, trace, 仿真环境.....

学习阶段划分

具有相关
项目经验



SIG小组

向量扩展

安全机制

高性能设计

低功耗设计

全链条: PA+OS+结构设计+SoC+IC后端+PCB板

可以流畅看B站的开源处理器芯片



组队攻关

单人学习

预学习(2个月)

- 1) Linux安装和学习
- 2) 学习C语言
- 3) 搭建仿真环境
- 4) 学习数字电路, RTL
- 5) 学习模拟器基础

入学
答辩

S: 专家 (3个月)

- 1) 乱序多发射
- 2) 性能优化

流片

A: 进阶 (3个月)

- 1) 五级流水线+Cache
- 2) 可运行自制OS和仙剑
- 3) 可运行Linux发行版

流片

B: 基础 (3个月)

- 1) 五级流水线
- 2) 可运行RT-Thread
和超级玛丽

流片

正式学习

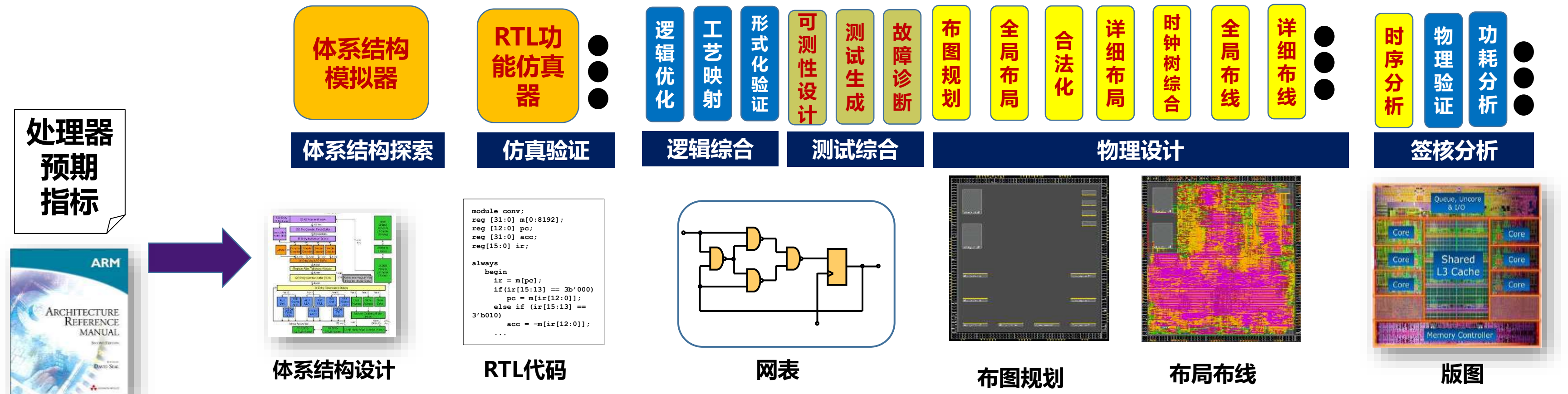
第六期新增开源EDA

- 开源EDA团队设计了多个点工具(黄色部分)[蓝色部分是将来工作]
- 大家是开源EDA工具的首批用户
 - 7月：综合(yosys)+时序分析工具
 - 10月：发布其他点工具, 可线下学习后端流程
 - 12月：芯片设计云平台上线, 基于开源EDA工具进行可流片设计

开源芯片技术生态论坛（原“一生一芯”技术论坛）

Session2: 开源EDA

8月25日14:00-15:15, 北京市香格里拉饭店三层翡翠厅



第六期 “一生一芯” 火热报名中

扫码或登录 **ysyx.org** 点击 “立即参与”



官方网站 **ysyx.org**

六、“一生一芯”社区活动

第六期新增入学大礼包

- 通过入学答辩，进入正式学习，获得**入学大礼包**一份！
- 内容丰富，专属于 **“一生一芯”** 正式学员

入学大礼包



定制文化衫一件



“一生一芯”徽章

“一生一芯”
正式学员学号



定制双肩包一个

毕业成果

- 完成 “一生一芯” 学习，可获得芯片、板卡以及证书
- 随附毕业大礼包（具体内容待定）

毕业收获



芯片



板卡



认证证书

积分兑换周边

- 参加志愿活动，积累积分，可兑换周边
- 大家可以发起提案，增加新的文化周边
- 提案被采纳后，有额外奖励

积分任务（部分）

提议新的文化周边

参与“一生一芯”调研问卷

“一生一芯”官方录用文章

“一生一芯”官方录用分享视频

“一生一芯”官方录用报告视频

“一生一芯”分享报告

组织“一生一芯”巡回演讲

完成B/A阶段学习的认证

任务内容持续更新中



(效果图)

可兑换周边（部分）

“一生一芯”文化衫

RISC-V文化衫

“一生一芯”帆布包

“一生一芯”双肩包

更多周边陆续上线，敬请期待



SIG小组 – 组队攻坚感兴趣的项目

开源芯片技术生态论坛（原“一生一芯”技术论坛），Session3: “一生一芯” SIG小组
8月25日15:15-15:40, 北京市香格里拉饭店三层翡翠厅

● 已成立的SIG小组

- 高性能体系结构模拟器——设计用于体系结构探索的模拟器
- 高性能RTL仿真器——目标性能优于Verilator
- 开源处理器核与外围IP——为“一生一芯” SoC提供丰富功能
- 开源芯片数据集——收集并运行现有的开源芯片项目, 为开源EDA提供数据集
- 国际交流与翻译——传播开源芯片国际最新动态

● 有想法但还没开设的SIG小组(需要有组长)

- 看B站——设计一颗可流畅看B站的开源处理器芯片
- 芯片上天——与航天院所合作, 学习设计高可靠的处理器芯片, 所设计芯片产品有机会在轨验证!
- 二进制翻译——在RISC-V处理器上玩windows游戏
- 大家如果有新想法, 可联系我们进行立项讨论, 通过后可提供经费支持!

开源社区贡献 – RVFA认证考试中文翻译

翻译小组斟酌用词



缪宇颺



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

Chapter 1: RISC-V Overview

Conventions used in this file:

- Heading 1 is used to mark the beginning of a course page.
- Heading 2 is used for subtitles within a page.
- Bold** for references to buttons or menu options, and first sentences in bullet points.
- Bold Italics** for introducing new terms.

According to L^AT_EX Author's Guide:

- Bold Consolas Font (black)** for source code.
- Bold Consolas Font (brown)** for file and folder names.
- Bold Consolas Font (Dark Blue)**: Text typed at the command line.
- Bold Consolas Font (green)**: Output.
- [Hyperlinks](#) are left in GoogleDocs' format.

Learning Objectives

- By the end of this chapter, you should be able to:
- Understand what RISC-V is and what it is not.
 - Identify the characteristics of an ISA to decide whether it is a CISC or a RISC type ISA.
 - Describe the history of RISC-V.
 - Identify the most notable differences between RISC-V and the leading commercial ISAs, like x86 and ARM.
 - Understand the structure and operation of RISC-V International.
 - Analyze the documentation of RISC-V specifications.
 - Explore ways to contribute to the RISC-V effort.

Chapter Introduction

In this chapter, you will get acquainted with RISC-V. Please note that you are expected to already be familiar with computer architecture and to have some exposure to some specific ISA. We will provide some refreshers on basic terms, but this is certainly not an introduction to computer architecture.

Among the topics we will cover, you will learn about the history of RISC-V and how it differs from today's dominating ISAs. You will also get the chance to browse through the vast documentation of RISC-V, and you will learn how the documents are organized.

You will be introduced to RISC-V International and the RISC-V ecosystem, and you will learn about quite a few different ways you may contribute to the RISC-V community.

Let us get started!

History of RISC-V: The Free and Open ISA

RISC-V (pronounced "risk-five") is an open standard instruction set architecture (ISA) based on principles, enabling a new era of processor innovation through open standard collaboration.

RISC stands for Reduced Instruction Set Computer, a computer architecture proposed in the 1970s, based on simplicity, as opposed to current microprocessors at the time, dubbed Complex Computers, or CISC. The RISC architecture was born in an academic environment, so that for simplicity and efficiency, proposing a series of features that dramatically opposed the CISC architecture, which was motivated by commercial interests at the time. RISC is the opposite of CISC in many ways.

第 1 章: RISC-V 概述

本文中使用的约定文件:

- 1级标题用于标记课程页面的开始。
- 2级标题用于页面内的子标题。
- 楷体** 用于按铃或菜单选项的引用, 以及要点中的第一句话。
- 楷体斜体** 用于引入新术语。

根据 L^AT_EX 作者指南:

- 楷体 Consolas 字体 (黑色)** 用于源代码。
- 楷体 Consolas 字体 (棕色)** 用于文件和文件夹名称。
- 楷体 Consolas 字体 (深蓝色)** 用于命令行中的文本。
- 楷体 Consolas 字体 (绿色)** 用于输出。
- [超链接](#) 保留为 GoogleDocs 的格式。

学习目标

- 到本章结束时, 您应该能够:
- 了解什么是 RISC-V, 什么不是。
 - 识别 ISA (instruction set architecture, 指令集体系结构) 的特征以决定它是 CISC 还是 RISC 类型的 ISA。
 - 描述 RISC-V 的历史。
 - 确定 RISC-V 与领先的商业 ISA (如 x86 和 ARM) 之间最显著的区别。
 - 了解 RISC-V 国际基金会的组织结构和运作模式。
 - 分析 RISC-V 规范文档。
 - 探索为 RISC-V 工作做出贡献的方法。

章节介绍

在本章中, 您将开始了解 RISC-V。请注意, 在阅读前您应该已经熟悉计算机体系结构并接触过某些特定的 ISA。我们将提供一些基本术语的复习, 但很显然这不是对计算机体系结构的介绍。

在我们将涵盖的主题中, 您将了解 RISC-V 的历史以及它与当今主导地位的 ISA 有何不同。您还将有机会浏览 RISC-V 文档, 并了解这些文档的组织方式。

您将了解 RISC-V 国际基金会和 RISC-V 生态系统, 并且您将了解可以为 RISC-V 社区做出贡献的多种不同方式。

让我们开始吧!

RISC-V 的历史: 自由开放的 ISA

RISC-V ("risk-five") 是一种基于 RISC 规范的开放标准指令集架构 (ISA)。通过开放标准协作促成了它的诞生。

指令集计算机。这是一种 1980 年代初期提出的基于简单性的计算机体系结构, 与当时被称为微处理器 (CISC) 的现代微处理器相对。RISC 架构诞生于学术环境, 因此在设计上力求简洁, 与当时受商业利益驱动的 CISC 哲学截然不同。RISC 在许多方面与 CISC 截然不同。RISC CPU 有几个寄存器和很多指令, 其中大部分指令可以访问内存, 而 RISC CPU 有很多寄存器, 但只有少数指令可以访问内存。内存访问仅限于一些加载和存储指令。

原文和译文

姓名	学校/单位	年级
缪宇颺	中科院计算所	- (组长)
苗金标	中国科学技术大学	研二
段震伟	中国科学技术大学	研三
刘汉章	太原理工大学	大三
曹勋	中国科学技术大学	研二
杨海帆	浙江工商大学	大四
曹世洋	中国科学技术大学	研二
倪仁涛	东北大学	研一
魏人	兰州大学	大四
吴佳宾	青岛大学	研一
陈璐	中国科学院大学	博一
栗金伦	太原理工大学	大四

月度报告和技术论坛

- **月度报告**：7月开始，每月第一个周末，汇报项目进展，举行入学仪式等
- **技术论坛**：每年组织 1~2次技术论坛（独立或联合其他主题）
 - **活动预告**：8月25日，开源芯片技术生态论坛（原“一生一芯”技术论坛）
 - 地点：北京市香格里拉饭店三层翡翠厅
 - 包含专家邀请报告，“一生一芯”技术分享等

祝贺北京一零一中学的烟雨松同学于2023年2月通过预学习答辩，成为“一生一芯”计划**第一位**进入正式学习的高中生！



包老师报告



第一届论坛大合影



第一位高中生正式学员

暑期夏令营活动及学员去向



2022年6月-8月，“一生一芯”暑期夏令营
地点：北京开源芯片研究院

学习深造	就业
卡耐基梅隆大学	北京开源芯片研究院
佛罗里达大学	华为海思
香港科技大学	比特大陆
中科院计算所	北京嘉楠捷思
中科院微电子所	北京微核芯
清华大学	进迭时空
复旦大学	中科院计算所
南京大学	鹏城国家实验室
浙江大学

2023年度暑期夏令营

开源芯片技术生态论坛
(原“一生一芯”技术论坛)
Session4: “一生一芯”技术分享
8月25日16:00-17:20
北京市香格里拉饭店三层翡翠厅

- 时间：7月~8月，上海处理器技术创新中心
- 近距离认识芯片，高密度学习芯片

姓名	学校	专业	年级
烟雨松	北京一零一中	-	高三
曾宇航	东华理工大学	计算机科学与技术	大一
张宇驰	太原理工大学	电气工程及其自动化	大一
袁永强	西安邮电大学	电子信息工程	大二
潘岩	哈尔滨工业大学	计算机科学与技术	大二
孟沛	电子科技大学	软件工程	大二
王翩	上海科技大学	计算机科学与技术	研零
唐德宇	中国科学技术大学	软件工程	研一
丁旻昊	中南大学	电子科学与技术	研一
赵树钰	中国科学技术大学	软件工程	研一



烟雨松



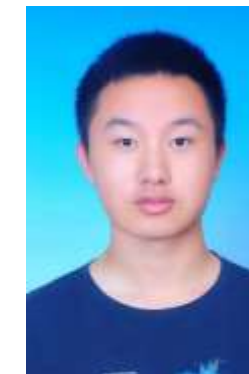
曾宇航



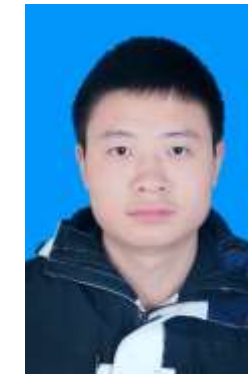
张宇驰



袁永强



潘岩



孟沛



王翩



唐德宇



丁旻昊



赵树钰

“一生一芯”官网及公众号



对写文章、视频剪辑、新媒体运营感兴趣的同学
欢迎加入宣传SIG小组
发送简历: ysyx@bosc.ac.cn



“一生一芯” 校园行巡回演讲活动

- 2023年3月至今，“一生一芯”深入校园，开展多场巡讲活动
- 秋季学期，我们的行程还会继续，尽情期待！
 - 如果想邀请“一生一芯”去你的学校，欢迎同学们随时联系项目组

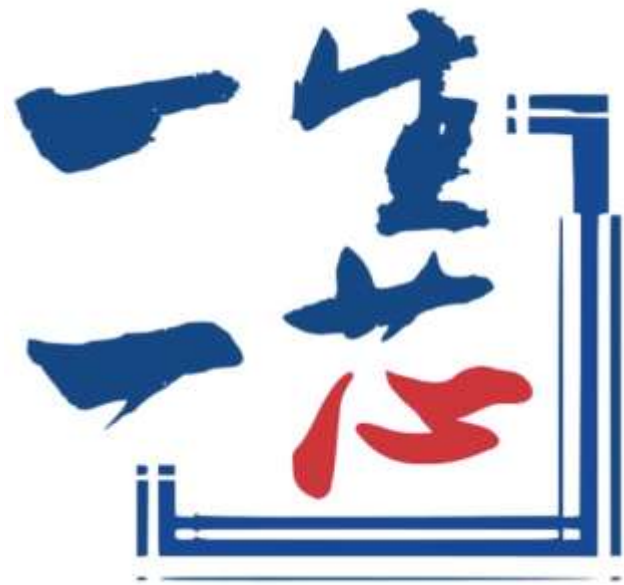
时间	巡讲学校	巡讲内容
3月19日	北京工业大学	宣讲
4月13日	北京科技大学	宣讲+教学交流
5月17日	东北大学（秦皇岛分校）	宣讲
5月25日	青岛大学	宣讲+教学交流
6月10日	天津工业大学	宣讲
6月14日	天津理工大学	宣讲+教学交流
6月16日	太原理工大学	宣讲

更多活动正在排期中，欢迎同学们邀请



“一生一芯”计划

- 硅上教学：融合本科阶段EE和CS专业知识点的实践项目
- 战略意义：为攻关卡脖子领域培养人才，输送到企业和开源社区
- 公益培养：学习资源全公开，免费学习；在校生免费流片



设计人生中的
第一颗处理器芯片



官方网站
ysyx.org



“一生一芯”公众号



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

七、常见问题

大家会问到的问题 - 1

■ 加入“一生一芯”需要具备哪些条件？

- **零基础即可加入，不论年级、专业和学校；在校生和已毕业都可以报名学习**
- 如果具备计算机或电路方面的基础知识，那当然更好，但并不强制，因为“一生一芯”的预学习和B阶段也会安排相关的学习内容。
- **在关键节点上指引大家，但并不会手把手教学。**比如，讲义告诉大家掌握 verilog，会推荐教程，但是并不会安排详细的课堂教学，因为这方面教材和在线课程已经很多

■ 流片条件？

- **达到指标，通过答辩考核，即可进入流片，会尽量安排最近一批次班车（涉及拼片）**
- 原创性和提升性，代码是自己写的，且参与“一生一芯”有较大提升（或社区服务）
- 流片只支持在校生（应届毕业8.31前提交）；已毕业可提交费用流片（视乎面积）
- **回馈社区：获得流片后，建议参加社区服务（助教和一生一芯的开源项目、科普等）**

大家会问到的问题 - 2

■ 参与方式？

- 以远程参加为主，部分同学会被邀请到现场来做助教（可报名）
- 每周一次会议，查看进展并答疑
- 同学们需要每 1~3 天记录一次学习过程，便于我们指导和跟踪大家工作过程
- 大家需要保持 Git commit, Git log 也是我们跟踪大家学习过程的重要依据

■ 研究生是否可以参加？ 研究生是在校生，若是国内高校，可以参与学习和流片

■ 非在校生是否可以参加？

- 可以参加学习流程，但无法免费流片（现有经费仅可用于支持国内在校生成流片）
- 刚毕业或已毕业但即将入学的同学，视为在校生成（比如9月开学，或毕业的8.31前）

■ 参加 “一生一芯”，是否还可以参加其他比赛？

- “一生一芯”注重的是人才培养，不与任何比赛冲突，大家参与 “一生一芯”写的代码，可以拿去参加比赛。如果获奖，非常希望大家能在致谢中，cite 一下 “一生一芯”

大家会问到的问题 - 3

■ 是否可以基于现有开源芯片项目修改后提交？

- 不可以，“一生一芯”是一个训练项目，每一行代码必须都是自己写的才行

■ 是否可以提高指标？

- 可以，鼓励同学们冲刺更高指标

■ 是否可以降低指标？

- 流片指标不会降低，没有达成流片指标者不能进入流片

■ 是否可以组队参加？

- 不可以，“一生一芯”是培养性项目，我们希望大家都能获得最大程度的训练
- 组队参加，仅在必要情况下，面向基础较好的同学，针对一部分特定主题和目标的开源项目开放，例如组队做一个以太网IP，此时数字和模拟都需要不同的队员加入
- 大家完成培养，进入社区，或者Sig小组以及开源IP项目，自然就可以跟其他同学组队了

大家会问到的问题 - 4

■ 参加“一生一芯”的名额，以及流片的名额

- 参加“一生一芯”的名额，目前没有限制，且随到随学。极端情况下，受限于支撑团队的规模，我们会结合报名人数和助教团队的情况，综合考虑后确定。

■ 是否达到流片标准就一定可以流片？

- 不一定。达到流片标准是最基础要求，还需评审通过后方可流片
- 极端情况下，如果达到流片标准的数量大幅高于预期，拟评审后择优流片
- **关于流片名额和班车等，项目组承诺Best effort，但并不做确定性保证（目前看来问题不大）**

■ 如果被误判作弊怎么办？

- 判断是否作弊的依据是大家日常提交的学习记录，以及Git log
- 如果误判，可向项目组提起申诉，我们会组织二轮专家评审来确定

■ 参与“一生一芯”期间，是否可以中间暂停一段时间后再继续？

- 可以，大家可以根据自己的知识基础和时间，适当调整计划；但需要事先声明和请假

大家会问到的问题 - 5

■ 是否一定要每周汇报，并按每 1~3 天更新一次学习记录？

- 对于能力较好，可冲刺更高指标的同学，可以自己安排
- 对于能力较弱，需要提升的同学，还是请按照如上节奏推进
- 此外，Git log，无论对于哪些同学，都是需要的

■ 跟不上队怎么办？

- 一生一芯提供的Schedule，只是一个参考的时间区段。大家可以根据自己的基础能力和可分配的时间，逐步推进就好。我们依旧会安排答疑

■ 答疑可以指导哪些问题？

- 会指明方向，告诉大家思路。比如告诉大家需要用Git，但是不会教Git
- 希望大家先主动探索，不到万不得已情况（Bug卡了一礼拜没进展），不会指导太细
- 很多材料课本上和网上都有，大家只要花一些时间去搜索和实践，是可以自己解决的
- **主动探索和动手实践的能力，才是整个“一生一芯”计划培养的核心**

大家会问到的问题 - 6

- **“一生一芯”一定要使用 Chisel 吗?**
 - 不要求一定使用 Chisel, 大家可以自己选定语言, 项目组都会给大家答疑
 - VHDL 暂时还不支持, 多语言的话, SoC集成的时候也会复杂一些
- **“一生一芯”一定要基于 RISC-V?**
 - 是的, RV64
- **是否可以报名参加助教、SoC 和 IC 后端的工作**
 - 可以。SoC 和 IC 后端很重要, 需要训练一段时间, 才能进入项目
- **Cache的规格有多大?**
 - 接入SoC时会跟大家讲
- **报名了第4期, 是否还需要报名第5期?**
 - 不需要, 可以按照第4期的讲义继续学习
 - 将来A阶段讲义会调整指导方式, 但具体的学习任务不会大幅调整