



香山开源高性能RISC-V处理器

敏捷设计实践

包云岗

中科院计算所

2022年8月24日

大纲

一. 香山一年进展


二. 香山敏捷开发实践

三. RISC-V云上“树莓派”

四. 未来展望

香山一年进展概览

ICT 中国科学院计算技术研究所
RVA 中国开放指令生态 (RISC-V) 联盟
China RISC-V Alliance
鹏城实验室
Peng Cheng Laboratory

 **香山：开源高性能RISC-V处理器**

包云岗 唐丹 徐易难
中科院计算所
2021年6月22日

让“香山”存活30年

- **迭代优化**：保持**半年更新一代架构**的迭代优化频率
- **流片验证**：**每年2次流片**，对新架构、新功能进行验证
- **开发流程**：构建敏捷开发**基础设施**，完善敏捷开发流程
- **开源社区**：建立规范的**开源社区管理**机制
- **推广应用**：广泛应用于**工业界**，成为**学术界**的创新平台
- **资金人员**：稳定的**经费来源**与**核心开发人员**

• 微架构迭代优化

- 第一代“雁栖湖”流片成功
- 第二代“南湖”持续迭代优化
- 第三代“昆明湖”启动架构设计

• 完善开发流程与基础设施

- 增加敏捷验证工具与测试用例
- 增加支持Chisel的DFT设计
- 完善前后端协同设计流程
- 学术论文被国际顶会MICRO接收

• 开源社区建设

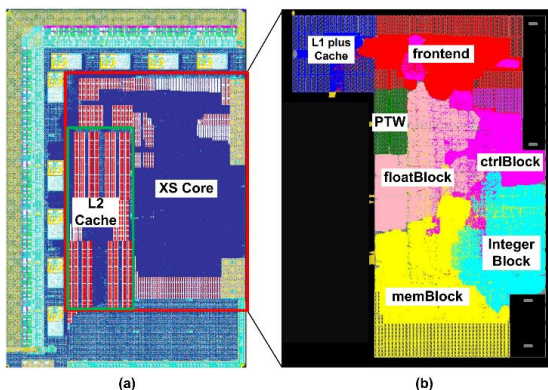
- GitHub：2900+Star，340+Fork
- 国内外的企业界与学术界开始使用香山
- 北京开源芯片研究院（开芯院）成立
- 获得支持后续开发的资金，形成联合开发团队



完成香山（雁栖湖）处理器的首次流片

- 香山处理器（单核，1MB L2配置）于2021年7月流片

流片版图

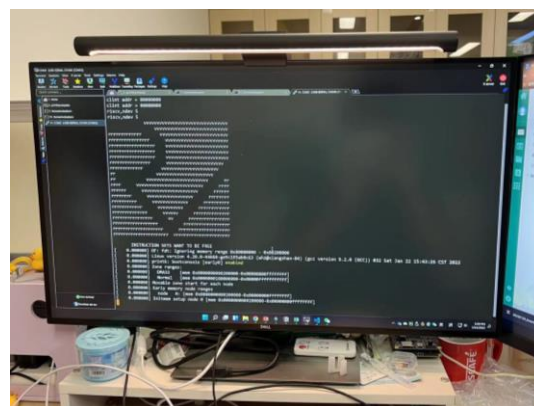


Tape-out information for the processor core	
Process Node	28nm
Die Size	8.6 mm ²
Std Cell	5.05M, 4.27 mm ²
Mem	261, 1.7mm ²
Density	66%
Cell	ULVT 1.04%, LVT 19.32%, SVT 25.19%, HVT 53.67%
Estimated Power	5W
Frequency	1.3GHz, TT85C

实测SPECint 2006: 7.03@1GHz
 实测SPECfp 2006: 7.00@1GHz

SPECint 2006 @ 1GHz		SPECfp 2006 @ 1GHz	
400.perlbench	6.14	410.bwaves	9.28
401.bzip2	4.37	416.gamess	6.59
403.gcc	6.71	433.milc	8.41
429.mcf	6.83	434.zeusmp	7.65
445.gobmk	7.92	435.gromacs	4.99
456.hmmer	5.24	436.cactusADM	3.97
458.sjeng	6.85	437.lelie3d	6.93
462.libquantum	17.71	444.namd	8.00
464.h264ref	10.91	447.dealII	10.17
471.omnetpp	5.65	450.soplex	7.03
473.astar	5.16	453.povray	7.14
483.xalanbmk	7.35	454.Calculix	2.86
		459.GemsFDTD	8.35
		465.tonto	6.42
		470.lbm	10.39
		481.wrf	7.26
		482.sphinx3	9.07

- 2022年1月回片并成功点亮，测试数据达到预期目标



南湖架构的主要改进内容

改进方向①：取指与分支预测

- 采用解耦的前端取指与分支预测架构
- 更高的吞吐率、更高的预测准确率

改进方向②：运算单元

- 支持 RISC-V 位操作 (B)、标量加密运算 (K) 扩展指令集
- 自研开源高性能 IEEE-754 浮点运算部件 FuDian
 - 开源地址：<https://github.com/OpenXiangShan/fudian>

改进方向③：功能支持

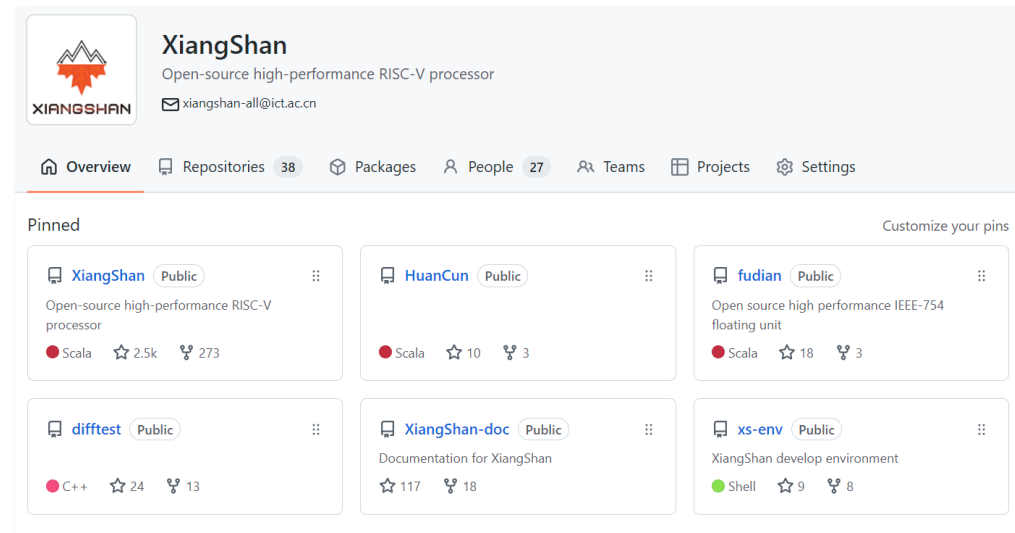
- 支持 RISC-V 物理内存保护机制 (PMP) 等安全特性
- 支持自定义的可配置物理内存属性 (PMA)

改进方向④：Load Store Unit

- 提高 TLB、L1 Cache 等结构容量
- 重构 L1 DCache 流水线设计，降低访问冲突，提高效率

改进方向⑤：L2/L3 Cache

- 自研支持 inclusive / non-inclusive 等参数可配置的开源高性能 L2/LLC 模块 HuanCun
 - 开源地址：<https://github.com/OpenXiangShan/HuanCun>



完整开发过程均在代码托管平台开源

香山处理器
前端取指架构的演进
8月25日 周四 9:00

南湖架构
后端流水线的设计与改进
8月25日 周四 9:15

南湖架构
MMU 单元的设计与实现
8月25日 周四 9:25

南湖架构
访存子系统的设计与实现
8月25日 周四 9:35

南湖架构 HuanCun
Cache 的设计与实现
8月25日 周四 9:45

南湖架构
Fudian FPU 设计与实现
8月25日 周四 9:55

基于香山实现的
标签化体系结构
8月25日 周四 10:05

南湖的最新性能评估结果

- 评估方法：利用 SPEC 的程序片段估计完整 SPEC 的运行时间与分数
 - CPU 频率 2GHz，使用 1MB L2 + 6MB L3 配置，使用 DRAMsim 模拟 DDR4-2400 的访存延迟
 - GCC 10.2.0，基于 RV64GCB 指令集，使用 O3 编译选项

SPECint 2006 @ 2GHz	
400.perlbench	19.29
401.bzip2	11.60
403.gcc	20.88
429.mcf	13.14
445.gobmk	17.21
456.hmmer	19.47
458.sjeng	16.06
462.libquantum	31.35
464.h264ref	28.93
471.omnetpp	12.24
473.astar	13.78
483.xalancbmk	20.08

SPECfp 2006 @ 2GHz	
410.bwaves	29.12
416.gamess	23.65
433.milc	15.88
434.zeusmp	18.89
435.gromacs	17.68
436.cactusADM	20.61
437.leslie3d	16.21
444.namd	24.90
447.deall	31.77
450.soplex	17.79
453.povray	30.30
454.Calculix	10.58
459.GemsFDTD	18.08
465.tonto	19.36
470.lbm	29.64
481.wrf	17.79
482.sphinx3	28.62

估计SPECint 2006: 17.80@2GHz
估计SPECfp 2006: 20.96@2GHz

产品化改造与后续架构研发

- 在北京市与中科院支持下，由**18家企业**联合发起**北京开源芯片研究院**，进行“香山”处理器核的**产品化改造**和**后续架构研发**



18家发起单位



2021年12月6日成立

合作项目1：香山v2（南湖）产品化改造

任务单位	1、处理器核微架构优化	2、处理器核接口改造	3、处理器核功耗管理设计与实现	4、处理器核验证增强	5、处理器核可测试性设计	6、处理器核前端集成和后端设计
开芯院	√	√	√	√	√	√
	√			√		√
	√		√	√		√
	√	√				
		√				√

1000016	STORY	前端集成和后端设计
1000015	STORY	处理器核可测试性设计
1000014	STORY	处理器核验证增强
1000060	STORY	整个南湖cpu核子系统进行ST验证
1000059	STORY	Dcache和store buffer合并在一起进行BT验证
1000058	STORY	ITLB/DTLB/L2TLB验证
1000057	STORY	memblock模块作为IT验证
1000056	STORY	执行单元验证
1000055	STORY	ctrlblock模块作为BT验证
1000054	STORY	后端处理模块作为IT验证
1000053	STORY	指令icache和取指IFU合并在一起进行BT验证
1000052	STORY	BPU和FTQ合并在一起进行BT验证
1000051	STORY	前端处理模块
1000013	STORY	处理器核功耗管理设计与实现
1000012	STORY	处理器核接口改造
1000011	STORY	处理器核微架构改造

项目任务分解及企业合作情况



合作项目2：香山v3（昆明湖）架构探索

任务 单位	处理器核设计与实现	关键宏模块定制	验证 SoC 芯片设计与实现	软件生态开发	开发板设计与实现
开芯院	√	√	√	√	√
	√			√	
	√	√		√	
	√				
	√				
	√			√	
				√	√
				√	

项目任务分解及企业合作情况

中国科学院计算技术研究所 (ICT, CAS)

- 1000028 STORY 开发板设计与实现
- 1000050 STORY OpenBoard开发板设计与实现
- 1000049 STORY ATX开发板设计与实现
- 1000027 STORY 软件生态
- 1000048 STORY TEE软件生态开发与优化
- 1000047 STORY 虚拟化软件移植与优化
- 1000046 STORY 编译器优化
- 1000045 STORY 关键应用软件移植与优化
- 1000044 STORY 设备驱动开发
- 1000043 STORY 操作系统移植
- 1000042 STORY 固件开发
- 1000026 STORY 验证SoC芯片设计与实现
- 1000041 STORY 封装设计
- 1000040 STORY 后端物理设计 (含DFT)
- 1000039 STORY SoC开发及系统级验证
- 1000038 STORY IP开发和验证
- 1000025 STORY 关键宏模块定制
- 1000037 STORY 关键宏模块的定制开发
- 1000023 STORY 处理器核设计与实现
- 1000036 STORY 处理器核功耗、面积评估、前端集成及后端设计指南
- 1000035 STORY 处理器核功能验证和性能评估平台
- 1000034 STORY Debug与Trace功能设计与实现、验证
- 1000033 STORY 可信执行环境 (TEE) 设计与实现、验证
- 1000032 STORY H扩展 (CPU虚拟化) 设计与实现、验证
- 1000031 STORY V扩展设计与实现、验证
- 1000030 STORY 微架构设计与实现
- 1000029 STORY 用于微架构设计空间探索的模拟器

大纲

一. 香山一年进展

二. 香山敏捷开发实践

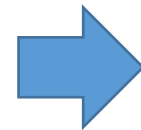
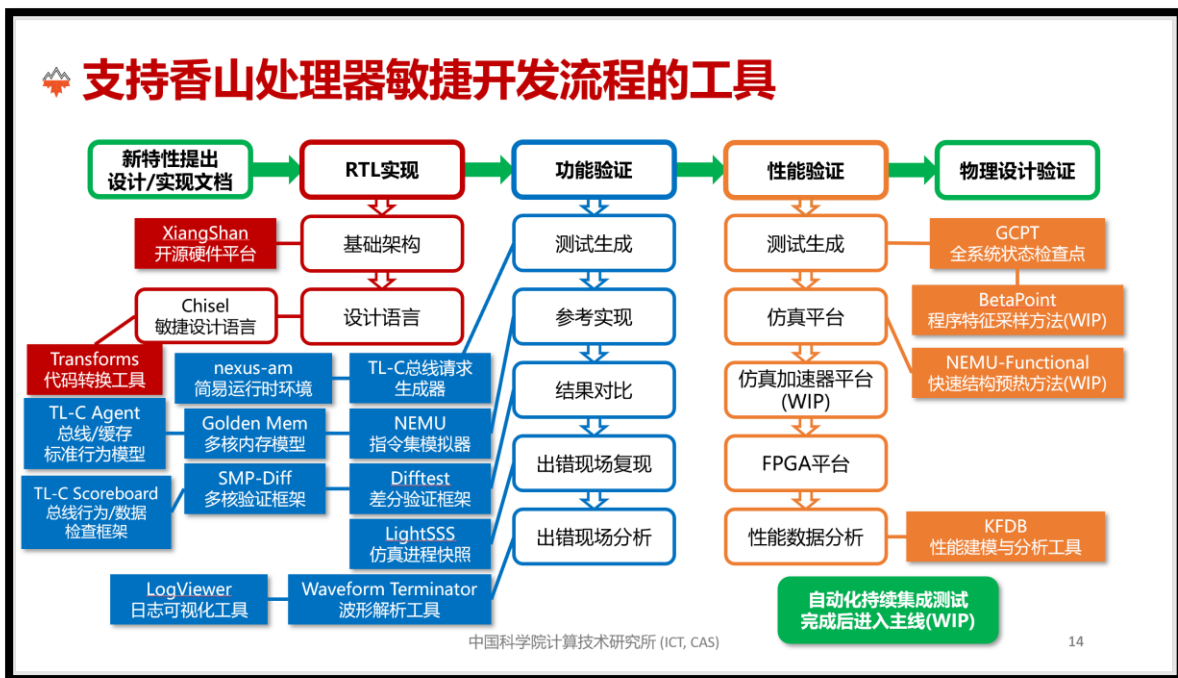
三. RISC-V云上“树莓派”

四. 未来展望



回顾：高性能处理器敏捷开发实践@2021峰会

To appear in MICRO-51, October 2022



Towards Developing High Performance RISC-V Processors Using Agile Methodology

Yinan Xu^{*†}, Zihao Yu^{*}, Dan Tang^{*†}, Guokai Chen^{*†}, Lingrui Gou^{*†}, Yue Jin^{*†}, Qianruo Li^{*†}, Xin Li^{*†}, Jiawei Lin^{*†}, Tong Liu^{*}, Zhigang Liu^{*}, Jiashan Tan^{*}, Huaqiang Wang^{*†}, Huizhe Wang^{*†}, Kaifan Wang^{*†}, Chuanqi Zhang^{*†}, Fawang Zhang^{||}, Linjuan Zhang^{*†}, Zifei Zhang^{*†}, Yaoyang Zhou^{*†}, Yike Zhou^{*}, Jiangrui Zou^{||}, Ye Cai^{||}, Dandan Huan[¶], Zusong Li[¶], Jiye Zhao[¶], Zihao Chen[§], Wei He[§], Qiyuan Quan[§], Sa Wang^{*†}, Kan Shi^{*}, Ninghui Sun^{*†} and Yungang Bao^{*†§}

^{*}State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, China

[†]University of Chinese Academy of Sciences, China

[‡]Beijing Institute of Open Source Chip, China

[§]Peng Cheng Laboratory, China

[¶]Beijing VCore Technology Co., Ltd., China

^{||}Shenzhen University, China

Abstract—While research has shown that the agile chip design methodology is promising to sustain the scaling of computing performance in a more efficient way, it is still of limited usage in actual applications due to two major obstacles: 1) Lack of tool-chain and developing framework supporting agile chip design, especially for large-scale modern processors. 2) The conventional verification methods are less agile and become a major bottleneck of the entire process. To tackle both issues, we propose MINJIE, an open-source platform supporting agile processor development flow. MINJIE integrates a broad set of tools for logic design, functional verification, performance modelling, pre-silicon validation and debugging for better development efficiency of state-of-the-art processor designs. We demonstrate the usage and effectiveness of MINJIE by building an open-source superscalar out-of-order RISC-V processor code-named XIANGSHAN with two generations using agile methodologies. We quantify the performance of XIANGSHAN using SPEC CPU2006 benchmarks and demonstrate that XIANGSHAN achieves industry-competitive performance.

Despite the benefits, agile design methodology is still of limited usage especially for practical modern designs. To further investigate the issue, we talk to chip architects, research scientists and lead engineers from twelve hyperscalers and major chip companies across multiple sectors, including data center, smartphone, communication and automobile. According to this survey, we summarize two common concerns regarding the agile methodology:

1) **Agile approaches are not quite ready for complicated processors.** Although a number of chips have been built using agile approaches, most of them are research prototypes and are relatively small or less complicated designs [1], [14], [15]. It is not clear if similar approaches can be directly applied to large-scale designs such as modern processors.

2) **The verification process is still less agile.** While modern verification methodologies such as UVM [16],

2021峰会报告PPT — <https://github.com/OpenXiangShan/XiangShan-doc/tree/main/slides>

基于香山项目深入探索硬件敏捷开发流程

**香山处理器
缓存一致性验证框架**
8月26日 周五 9:20

**DiffTest: 基于香山处理器的
敏捷验证实践**
8月26日 周五 10:40

**支持香山处理器的
性能瓶颈分析工具**
8月25日 周四 10:20

**香山分支预测器内外
快速性能评估工具**
8月26日 周五 9:40

功能验证

性能分析

基础设施

**ChiselDB: Mapping Hardware Data Structures to
Database Tables for Efficient Data Analysis**
8月27日 周六 11:00

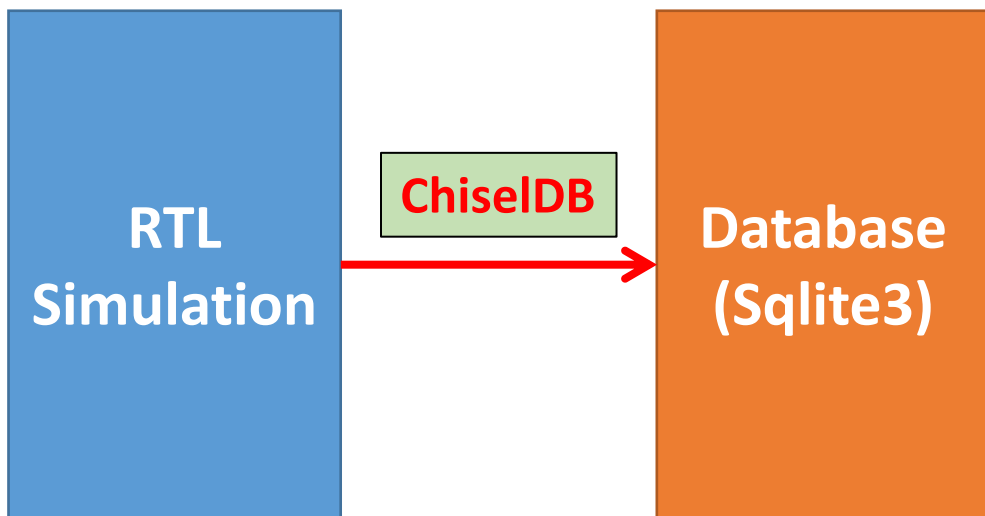
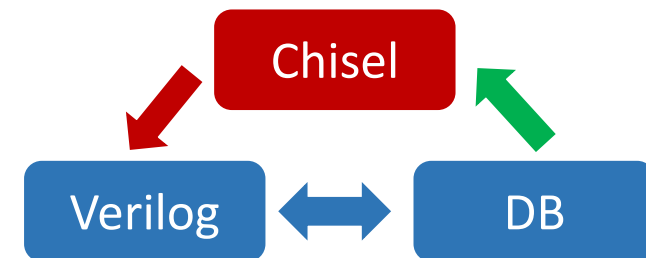
香山处理器的DFT设计
8月25日 周四 10:30



ChiselDB: 高效自动地传递结构化信息

ChiselDB: Mapping Hardware Data Structures to Database Tables for Efficient Data Analysis
8月27日 周六 11:00

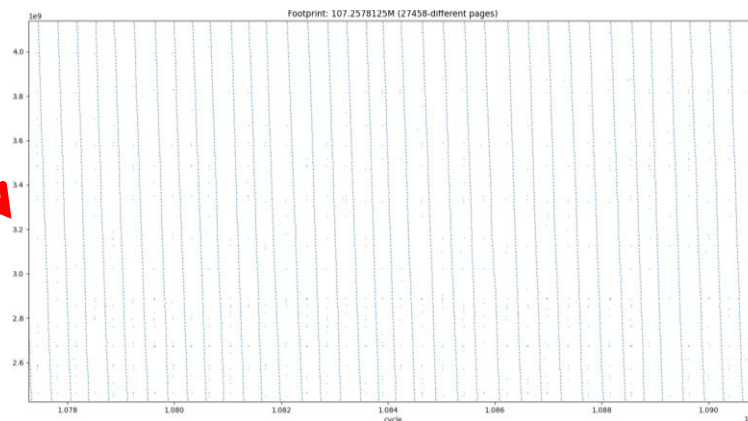
- 使用软件工程的方法，提高**结构化信息**传递的自动化程度
- 案例：针对关键微结构信息的调试方案



```

* * * [1]:(main) * sqlite3 mcf_19150000000_0_195600.db "SELECT * FROM L1MissTrace" | head -n 50
1|2147483664|0|2147487488|2147487488|4323|MissQueue
2|2147483712|0|2147488512|2147488536|4713|MissQueue
3|2147483732|0|2147494656|2147494656|5574|MissQueue
4|2147483872|0|2147495168|2147495168|5928|MissQueue
5|2147483972|0|2147495936|2147495936|6216|MissQueue
6|2147483992|0|2147496064|2147496064|6422|MissQueue
7|2147484072|0|2147490600|2147490600|6519|MissQueue
8|2147484112|0|2147491072|2147491072|6746|MissQueue
9|2147484192|0|2147491584|2147491584|7019|MissQueue
10|2147484312|0|2147488000|2147488016|7509|MissQueue
11|2147484344|0|2147488064|2147488080|7513|MissQueue
12|2147484376|0|2147488128|2147488144|7517|MissQueue
13|2147484416|0|2147488192|2147488224|7522|MissQueue
14|2147484448|0|2147488256|2147488256|7525|MissQueue
15|2147484468|0|2147487744|2147487752|7773|MissQueue
16|2147484492|0|2147487808|2147487808|7776|MissQueue
17|2147484532|0|2147487872|2147487888|7781|MissQueue
18|2147484556|0|2147487936|2147487936|7784|MissQueue
19|70026|0|2160081728|68713315192|10180|MissQueue
20|70042|0|2160081792|68713315200|10184|MissQueue
21|70074|0|2160099584|520448110492|MissQueue
22|68438|0|2158101232|1278592|10600|MissQueue
23|68446|0|2163909312|3744488|10613|MissQueue
24|70294|0|2158112960|1257664|10681|MissQueue
  
```

Debugging



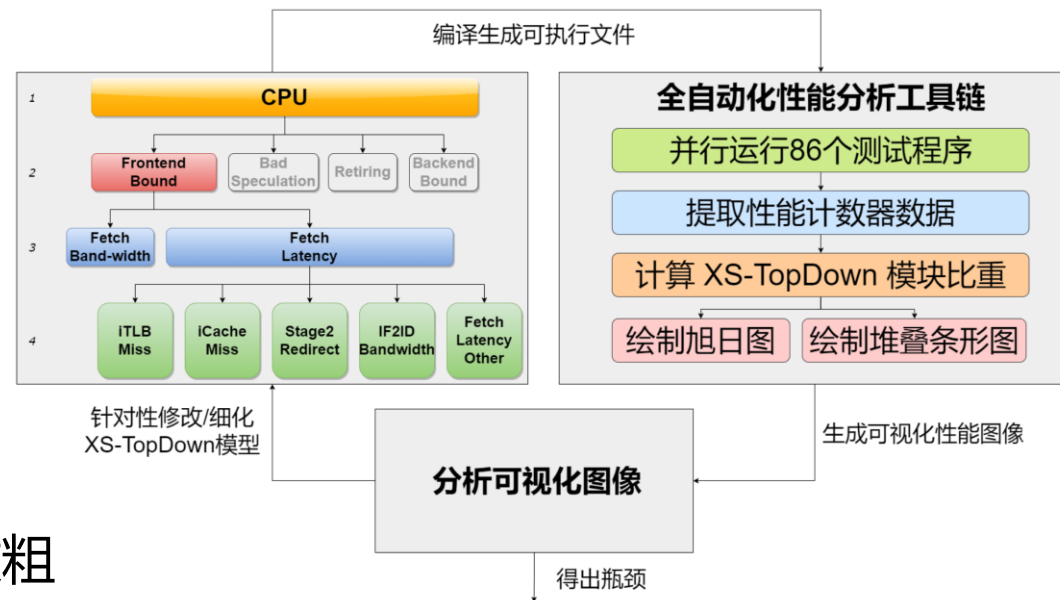
Performance Analysis



TopDown: 面向香山处理器的性能分析模型

支持香山处理器的
性能瓶颈分析工具
8月25日 周四 10:20

- 支持“性能建模 – 数据处理 – 结果分析 – 优化建模”的完整工作流程
- 针对 RISC-V 指令集完成针对性优化适配
 - 指令集差异大，现有工作不兼容 RISC-V 结构
- 针对香山处理器微结构优化性能计数器
 - 微结构差异大，现有工作无法适配香山结构特征
- 进一步细化 Top-Down 模型的分层设计
 - 香山处理器微结构复杂度高，现有模型分解粒度过粗



支持香山处理器的
TopDown 性能分析工具

分支预测器的性能建模与性能评估

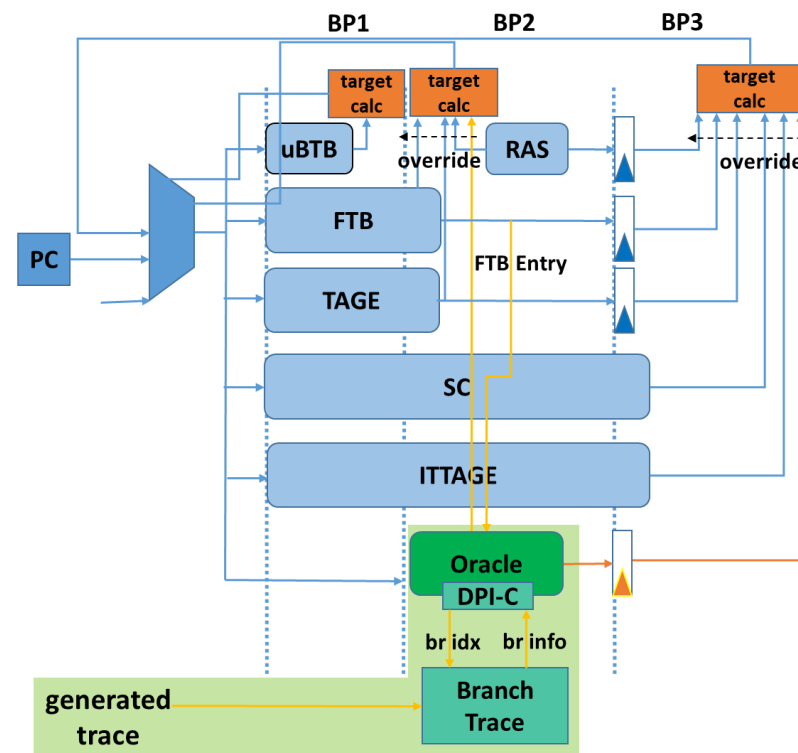
香山分支预测器内外
快速性能评估工具
8月26日 周五 9:40

• OracleBP: 搭建分支预测器的 Roofline 性能模型

- 高性能处理器内部模块关联度大，难以预测性能瓶颈
- 准确的分支预测器性能建模能指导处理器的优化迭代
- 关键：动态获得正确的分支路径并覆盖 RTL 预测结果

• BPUTester: 面向分支预测器的性能评估框架

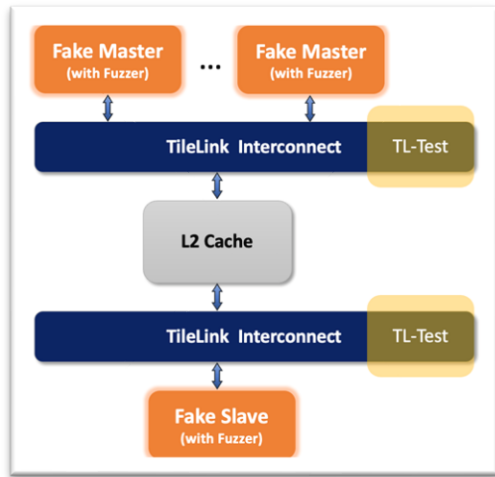
- 分支预测器内部微结构复杂，模拟器对齐成本高
- 系统级仿真速度慢，难以胜任分支预测器的细节评估
- 关键：剥离分支预测器 RTL，并搭建单元级交互模型



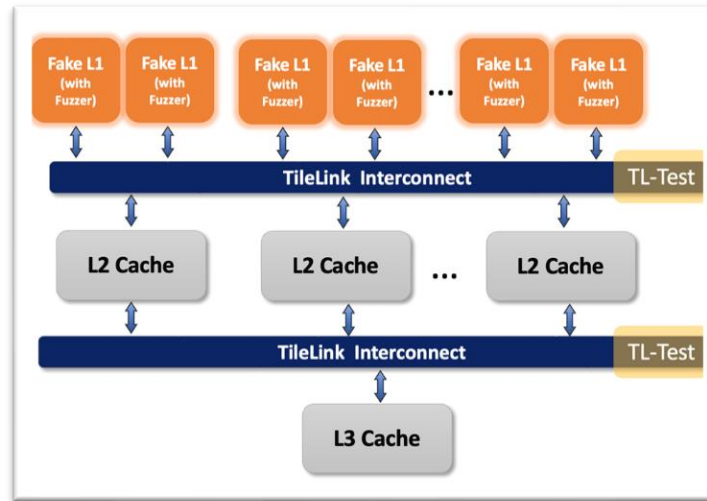
分支预测器与 OracleBP 架构

TL-Test: 基于 TileLink 总线的缓存敏捷验证框架

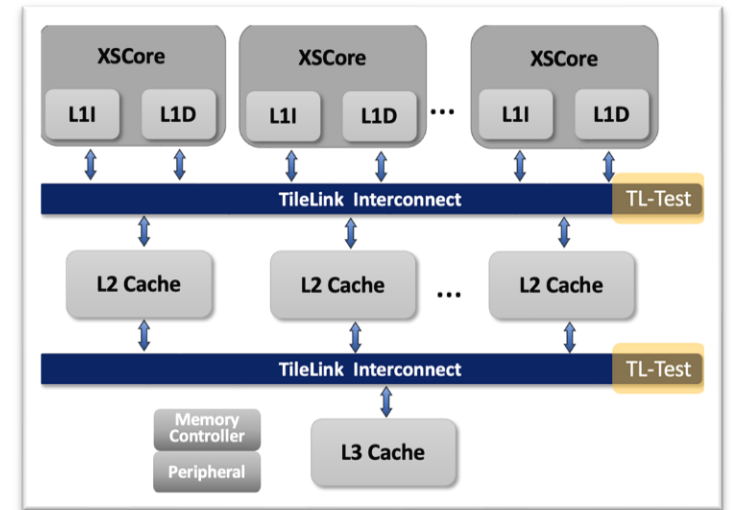
- 随着 RISC-V 处理器性能与复杂度提升，缓存一致性验证越来越重要
- TL-Test 有效支撑香山南湖双核 SoC 的缓存一致性验证



① 单元级验证



② 子系统级验证



③ SoC 级验证

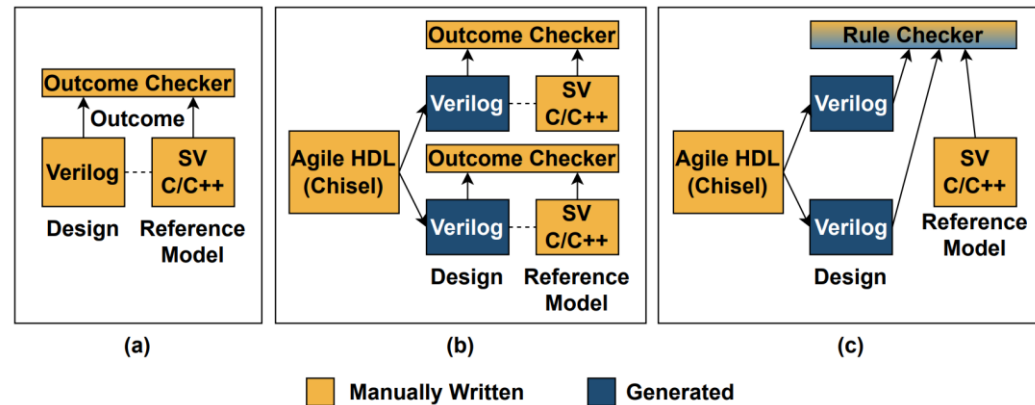


DiffTest: 通用的指令级在线差分验证框架

DiffTest: 基于香山处理器的敏捷验证实践
8月26日 周五 10:40

基本流程

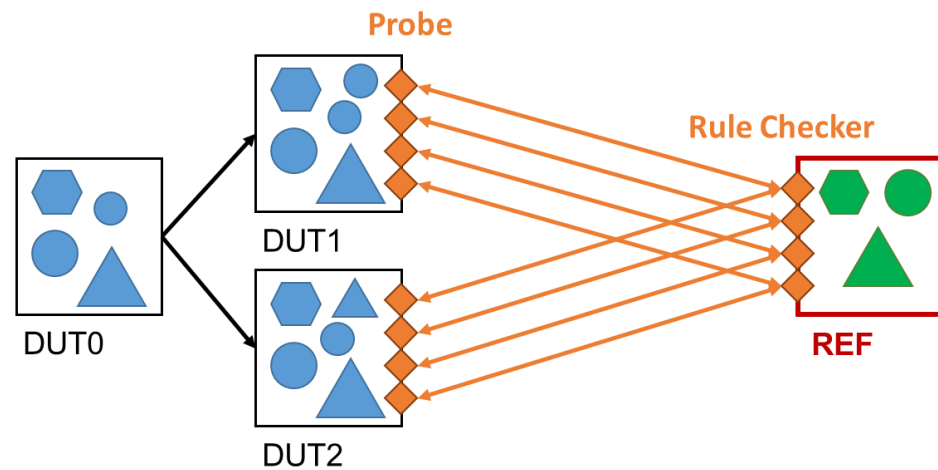
- 处理器仿真产生指令提交/其他状态更新
- 模拟器执行相同的指令
- 比较两者状态，报错或继续



不同的验证框架搭建方式

DRAV: 如何更好地支持 RISC-V 处理器敏捷验证

- 更多的处理器、更复杂的场景、更快速的迭代
- 使用 Diff-rule 描述设计规范所允许的行为
- 使用 Probe 完成微结构信息的传递



降低验证框架与 REF 的复用成本

Chisel-based Design的 DFT 实现及挑战

香山处理器的DFT设计

8月25日 周四 10:30

兼容业界主流的 DFT 实现流程

- 较业界主流的DFT实现流程增加从Chisel编译成RTL的步骤
- Chisel编译出的RTL利于工具解析, 更好支持DFT RTL Flow
- 香山处理器使用DFT RTL Flow完成设计实现

DFT 实现流程前移的尝试, Chisel-based DFT

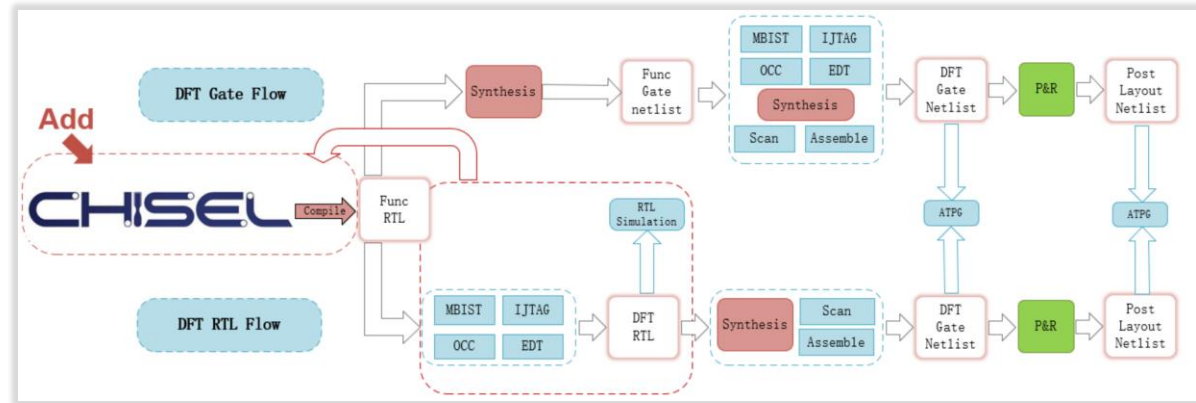
- 南湖全新MBIST架构: Chisel-based的Sharedbus Interface
- Chisel-based的BISR Chain For Memory Repair
- Chisel-based的OCC电路改造 For Global Tree&Local Tree

香山处理器 DFT 实现流程的演进

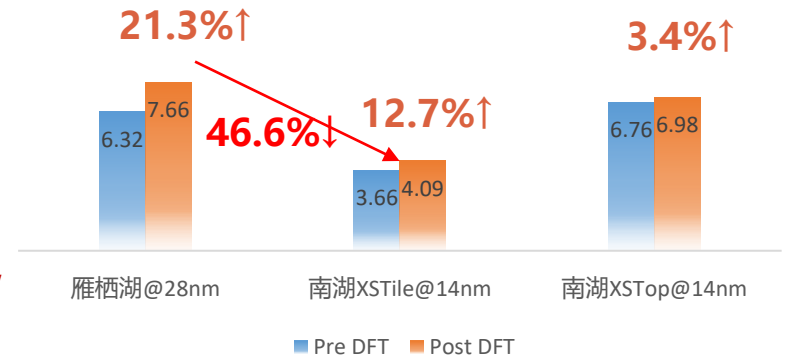
- **挑战:** DFT Chisel Flow颠覆目前商业EDA体系
- **愿景:** 让DFT设计更加敏捷: DFT Gate Flow → DFT RTL Flow → DFT Chisel Flow

与开源EDA、国产EDA厂商的联动, 共建开源开放的DFT设计平台

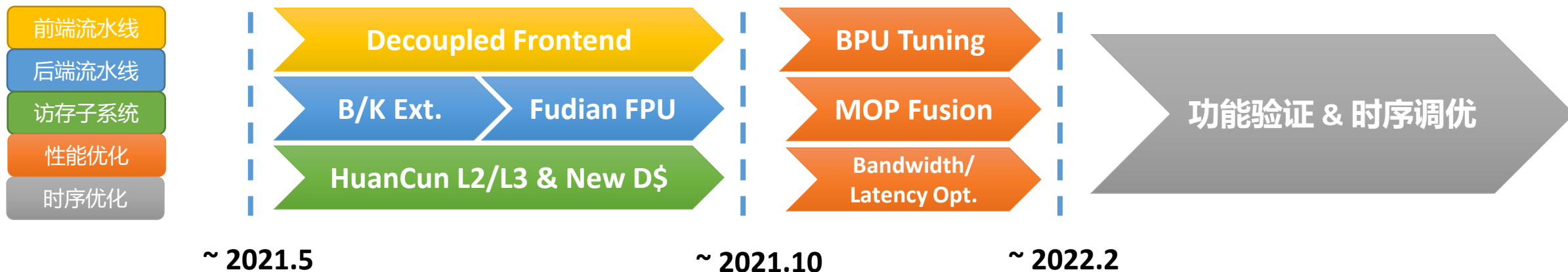
DFT实现流程全景图



雁栖湖与南湖DFT面积增长对比



南湖架构开发时间线



- 南湖架构开发过程中的大量时间花费在**时序调优和功能验证上**
 - 1.3GHz @ 28nm → 2GHz @ 14nm
 - 高复杂度的新架构设计提升了验证成本

时序的前后端协同优化迭代

- 降低整体TNS/WNS的数量级

- 降Fanin, 模块间及内部关键路径进行Retiming, 打拍, 复位实现等时序优化手段

- 优化物理综合时序逻辑

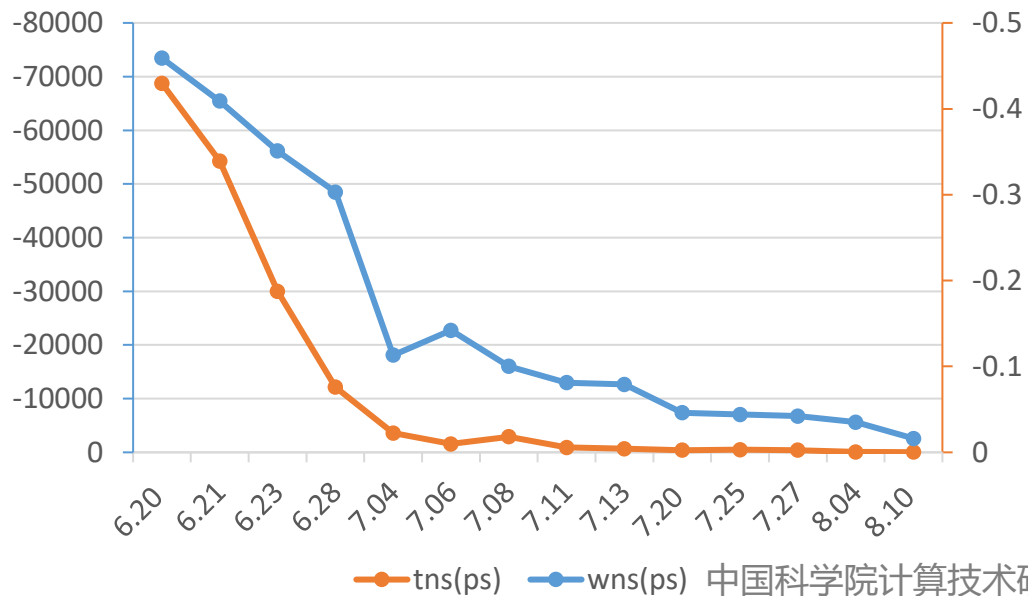
- 解Fanout, 根据物理综合的结果, 将大扇出的寄存器进行复制等操作解Fanout

- 优化模块间逻辑耦合深度

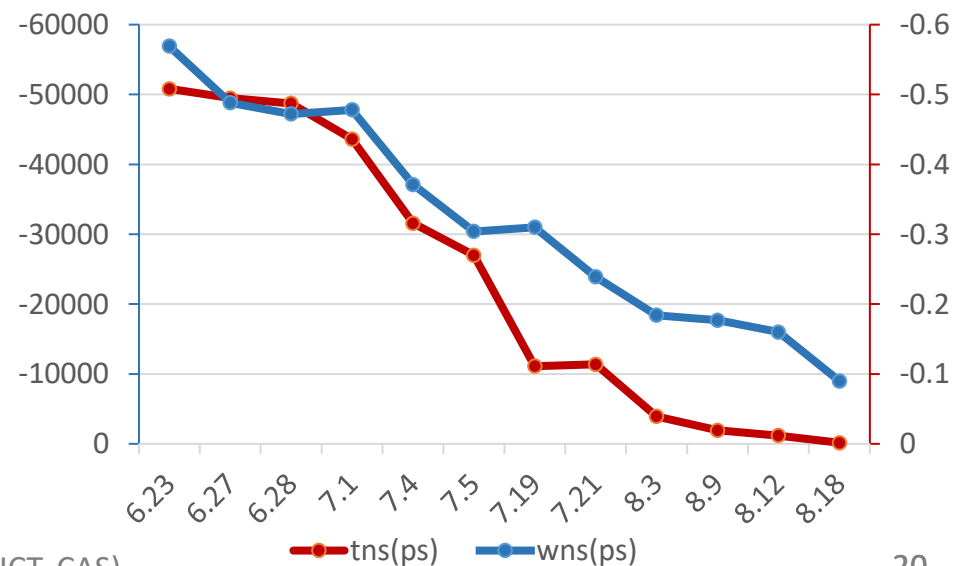
- 解长走线带来的transition过大问题

综合结果收敛情况

XSTile DC综合迭代



XSTile DCG综合迭代



Floorplan迭代

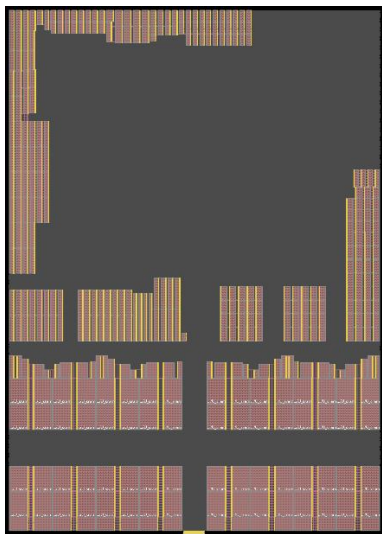
迭代的总体思路

保证绕线
无问题

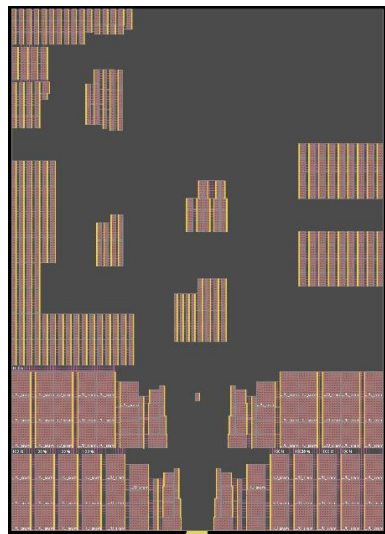
不断降低时序
的WNS和TNS

符合整体
数据流走向

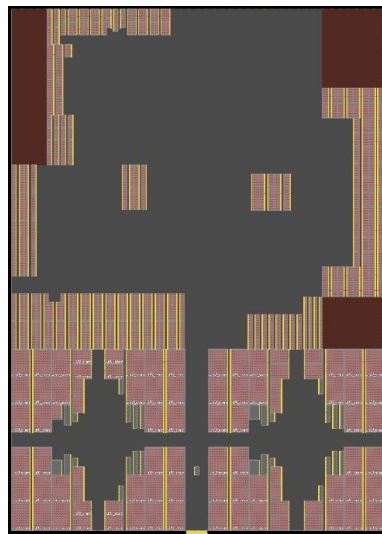
Floorplan V1



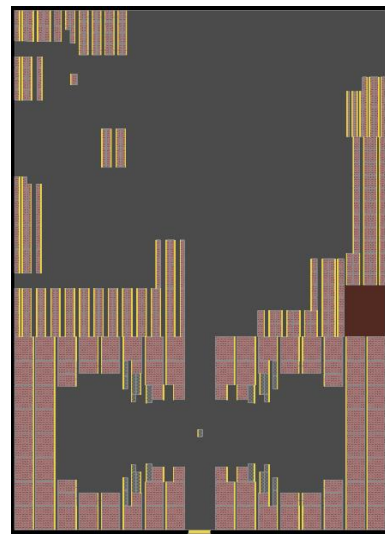
Floorplan V8



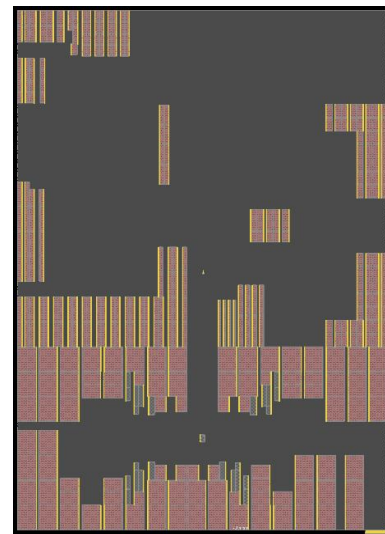
Floorplan V25



Floorplan V36



Floorplan V43



经过43版Floorplan的优化挑选出最优版本进行后续迭代

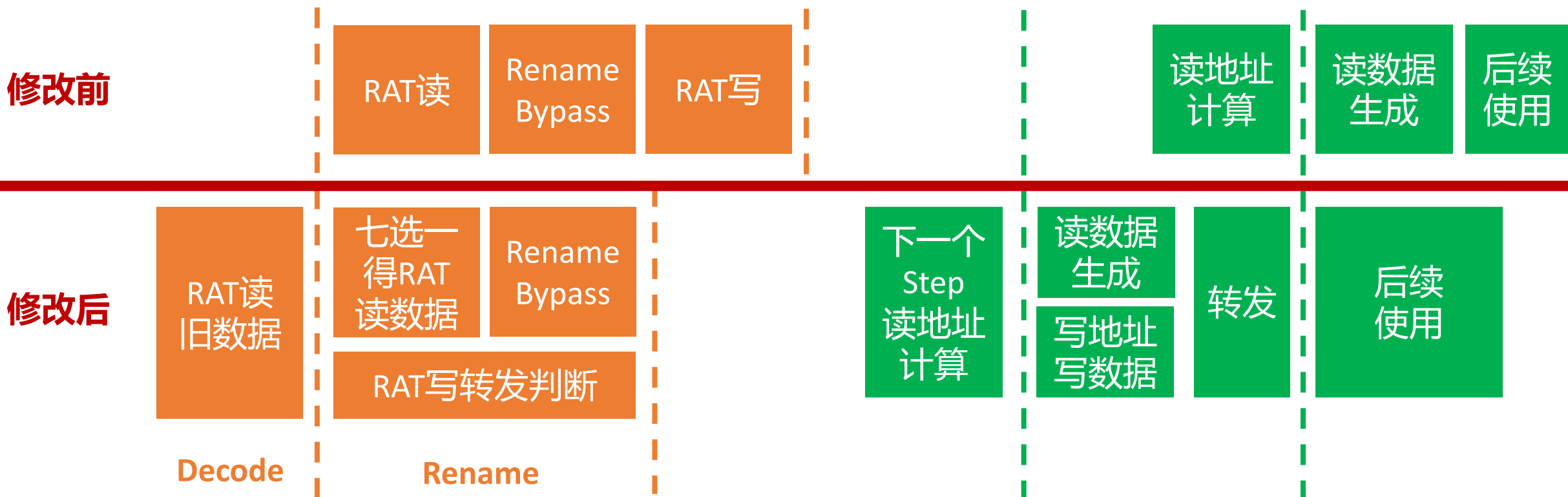
时序问题①：寄存器扇入过大，逻辑级数过长

重新组织微结构流水线形式

- 特点：部分信号可提前产生、延后使用
- 优化：融合多个逻辑流水级

消除队列数据存储的读延迟

- 特点：指针移动范围 (Step) 受限
- 优化：读数据只从特定几项中选择

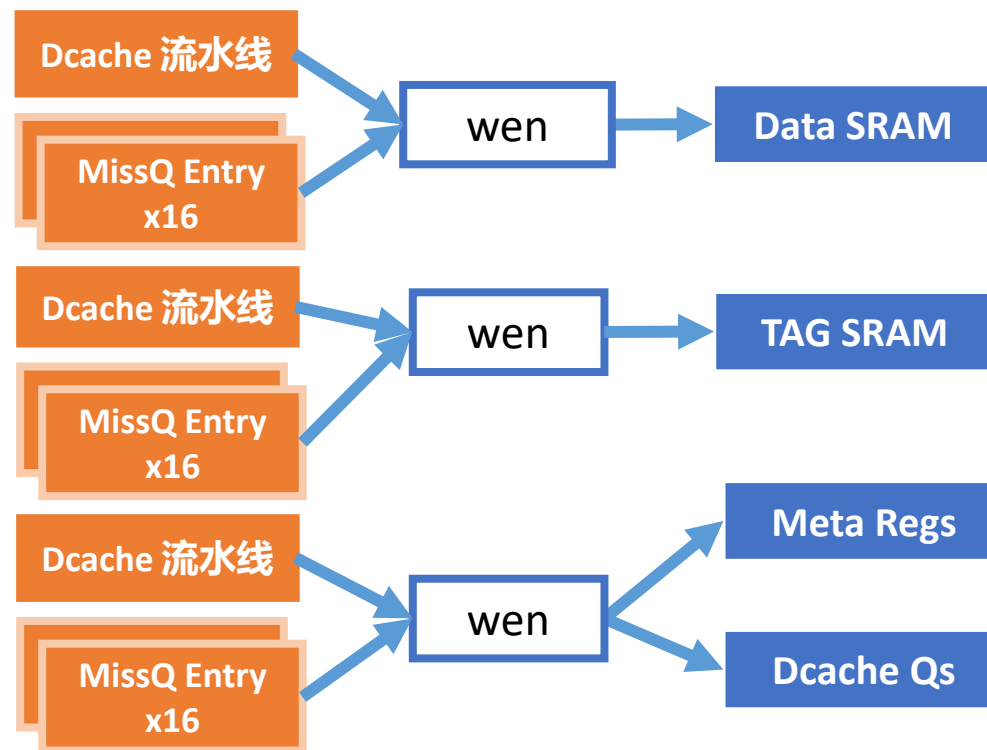
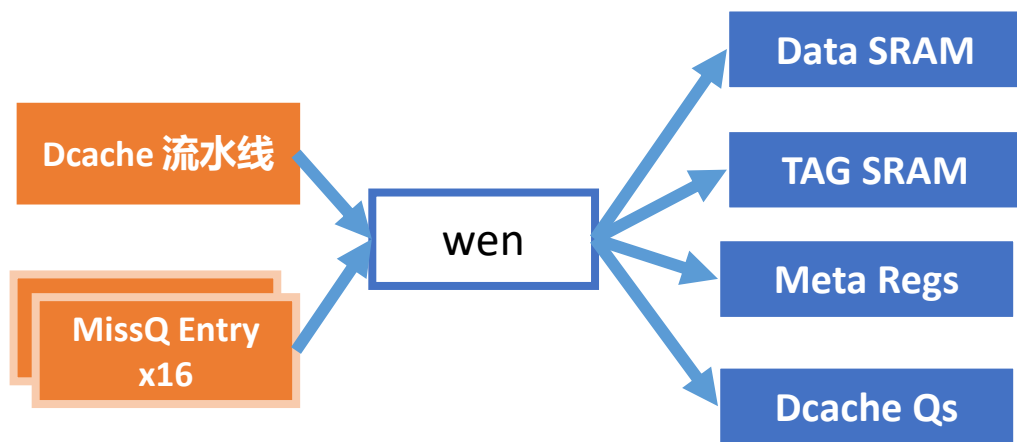


时序问题②：寄存器扇出过大

设计阶段分离控制流与数据流，规避大驱动逻辑

- 控制通路逻辑级数优化，如：驱动宽数据的控制信号以寄存器为直接起点
- 数据通路操作流水化，如：多来源的数据写入寄存器堆时加一级锁存流水

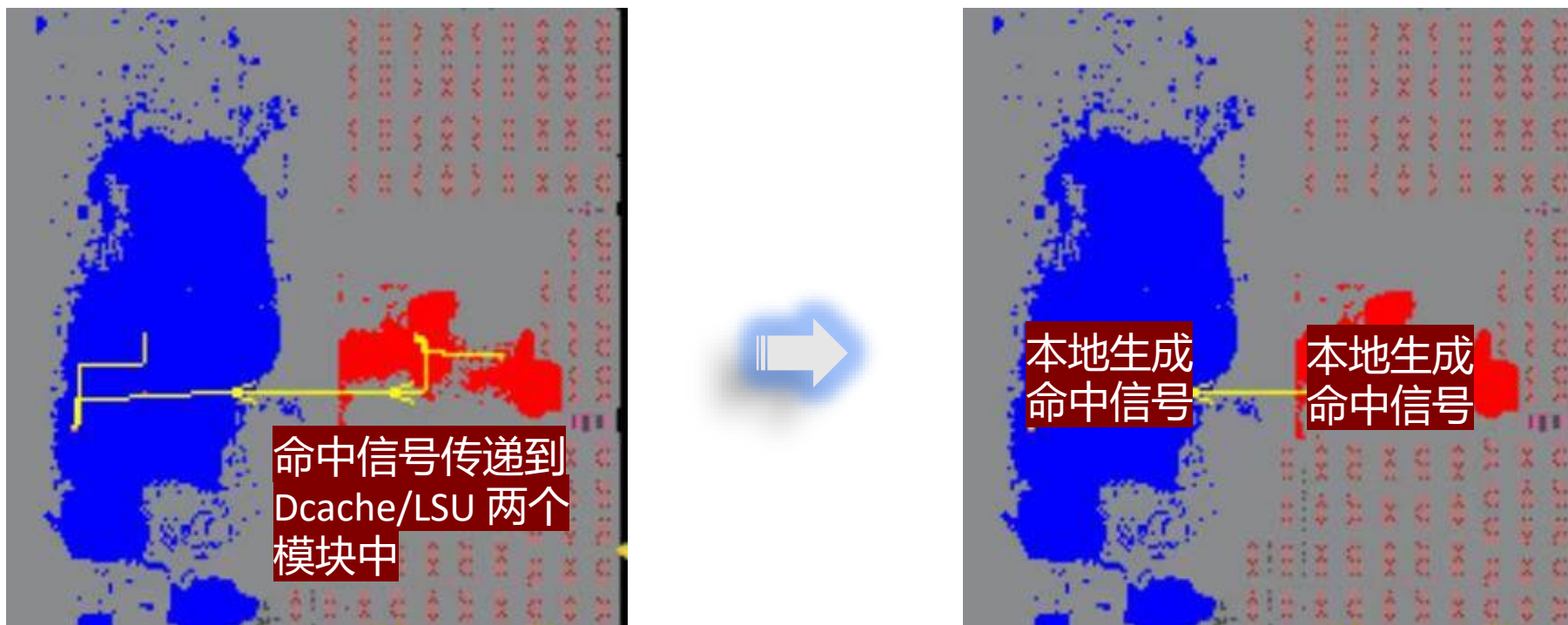
复制寄存器，切分扇出



时序问题③：模块间逻辑耦合过深，长距离绕线

- 设计阶段慎重考虑如何使用跨模块的信号
 - 降低模块间功能耦合程度，结合芯片布局情况进行跨模块优化
- 针对多模块使用的同义控制信号，在模块本地分别搭建生成逻辑

Place & Route
示意图



南湖架构的开发经验与思考

- **教训：时序优化周期长，但前后端协同优化介入过晚**
 - 应尽早启动前后端沟通协调与优化迭代，在设计阶段规避大多数时序问题
- **思考：探索面向高性能CPU的前后端协同设计流程**
 - 积累前后端协同的CPU优化迭代经验，**实现know-how开源开放**
 - 创新前后端结合的复杂芯片优化工具，**实现基础设施开源开放**
 - 搭建前后端融合的跨层优化能力体系，**提升流程自动化程度**



大纲

一. 香山一年进展

二. 香山敏捷开发实践

三. RISC-V云上“树莓派”

四. 未来展望

SERVE 开源处理器芯片系统级设计验证云平台

- **目标**：为开发者提供 “云上RISC-V计算系统”
- **架构**：通用x86服务器集群 + 自研异构FPGA服务器 + 容器云
 - ✓ 关键技术：持续集成的开发流程、FPGA虚拟化部署、云端IDE
 - ✓ 开源仓库：RISC-V计算系统基础软硬件代码、自建用户代码托管
- **开放共享**：第三方算力接入 + 可定制设计验证流程

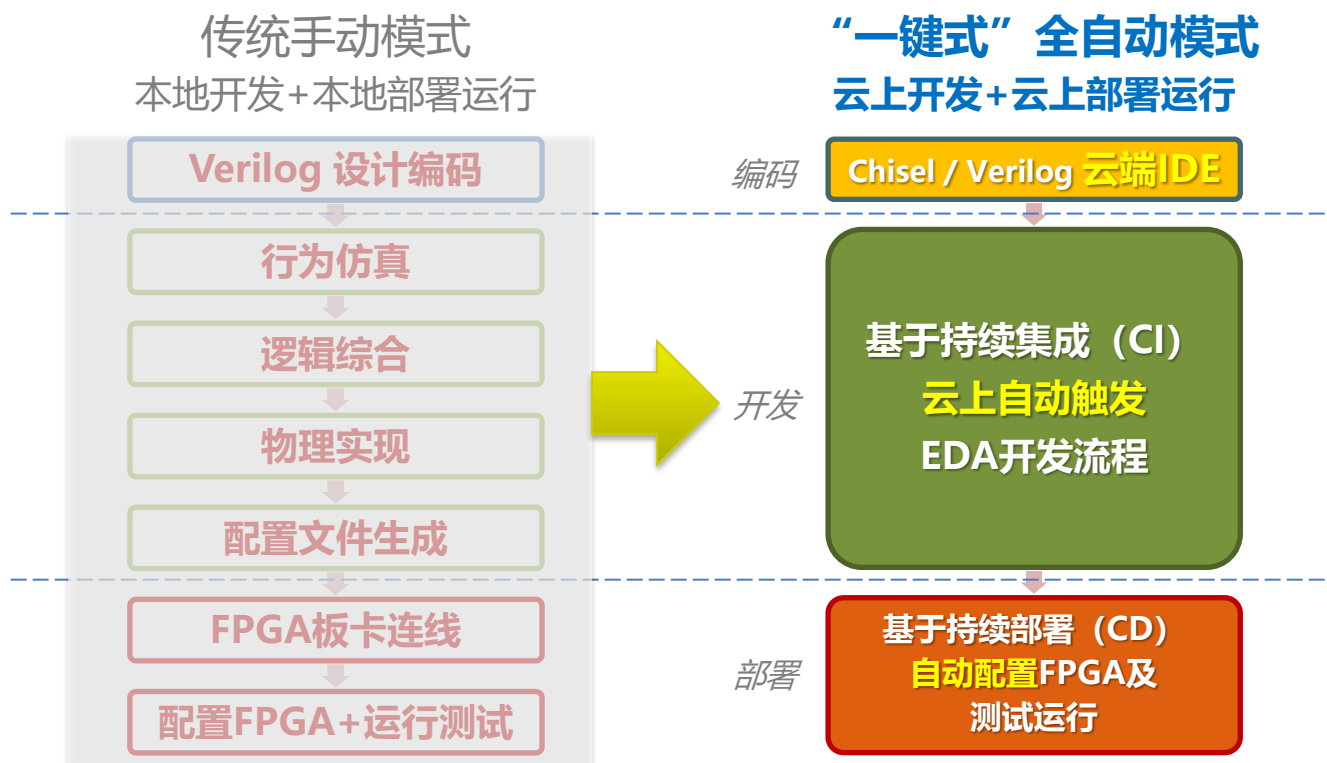


欢迎扫码获取信息
并注册使用



SERVE 基于持续集成/持续部署的设计验证流程

- 通过**云端敏捷服务**，加快RISC-V开源计算系统的设计验证迭代速度
- 借鉴“云原生”软件开发理念，提出基于持续集成/持续部署(CI/CD)技术的**“自动化硬件系统级开发”流程与模式**

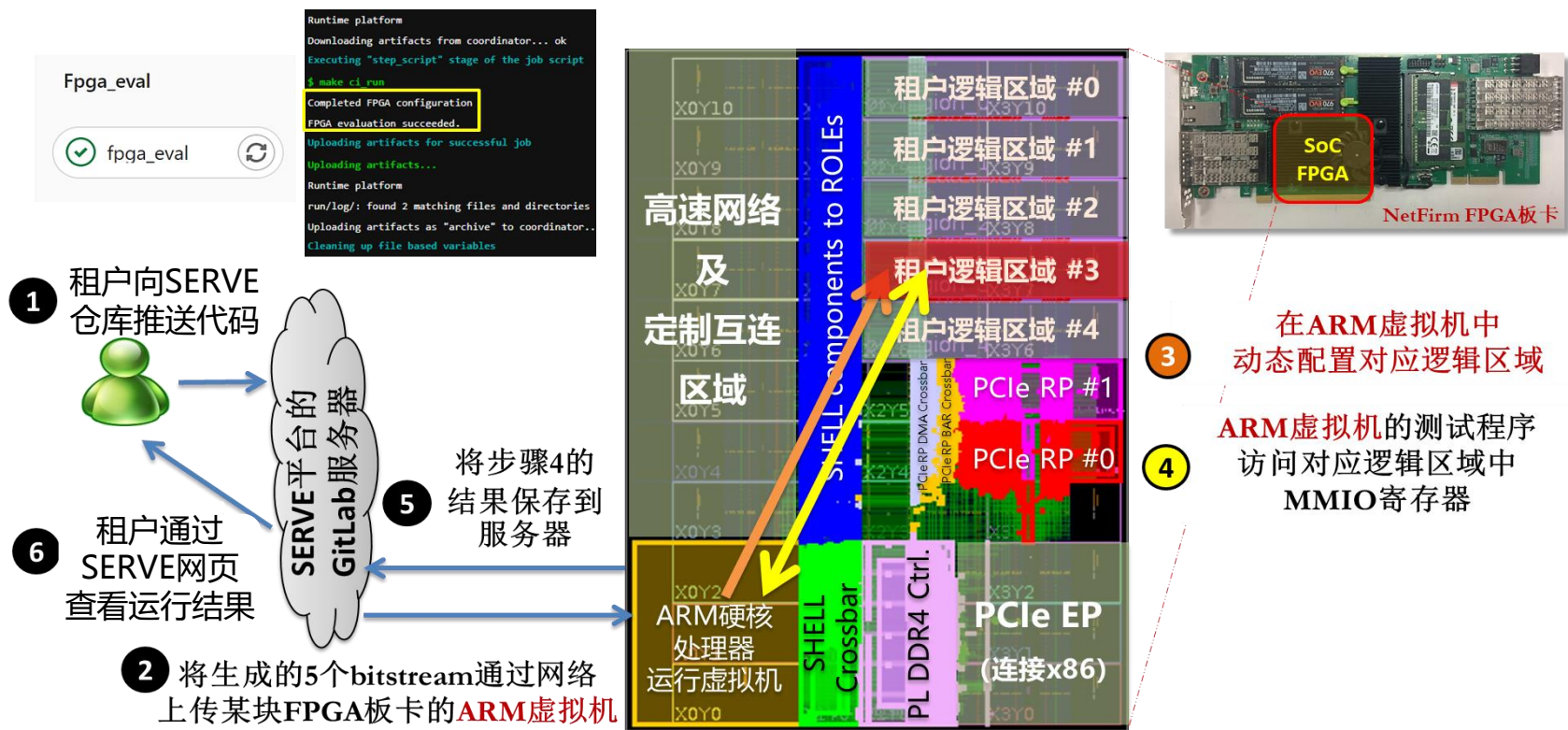


SERVE平台用户体验

- **显著降低**用户开发环境构建时间
- **节省**用户开发时间
- 用户并发数**提高一倍**

FPGA虚拟化支持多用户并发部署

- **核心技术:** FPGA逻辑区域划分与动态部分可重构 + ARM通用处理器虚拟化
- **实现效果:** 单节点FPGA利用率提升四倍





RISC-V计算系统软硬件代码仓库

- 提供用户：RV处理器核硬件、SoC基础外设组件、固件软件、系统软件、应用软件等**软硬件全栈基础设施**
- 实现云端开发、云端部署、云端评测，**降低二次开发门槛**

应用程序

User Applications

操作系统

Debian

Fedora

软件所傲侠

工具链

GNU Toolchain (gcc, gdb, etc.)

标准程序库

glibc

系统内核

Linux Kernel

引导程序

BBL/OpenSBI + U-Boot

处理器核

果壳 (NutShell)

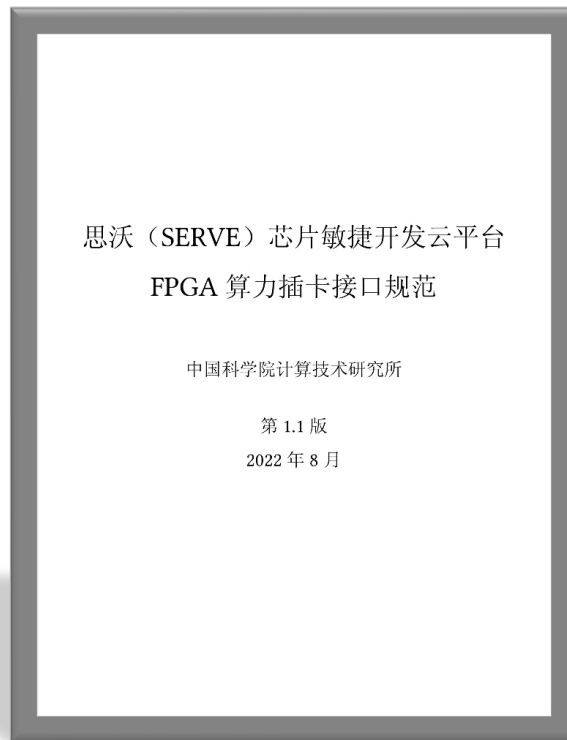
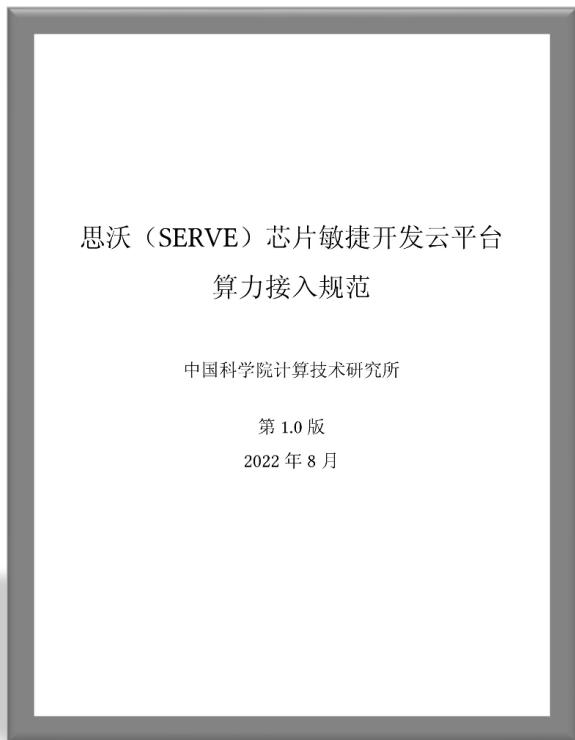
Rocket-chip

Name	Last commit	Last update
bootstrap	Deleted code information related to ipx...	2 months ago
build_scripts	Deleted code information related to ipx...	2 months ago
fpga	Add serve_i's bootscr, try CI of serve_i	2 months ago
software	Create a new open source main branch.	2 months ago
.gitignore	Create a new open source main branch.	2 months ago
.gitlab-ci.yml	Add serve_i's bootscr, try CI of serve_i	2 months ago
.gitmodules	Create a new open source main branch.	2 months ago
Makefile	Create a new open source main branch.	2 months ago



SERVE 规范制定：算力接入与FPGA算力插卡接口

愿景：开放共建、资源共享



欢迎了解!
共建共享!



SERVE



极狐
GITLAB



博云®
BoCloud

中科曙光
SUGON



Corigine
芯启源



S2C
EDA SOLUTIONS

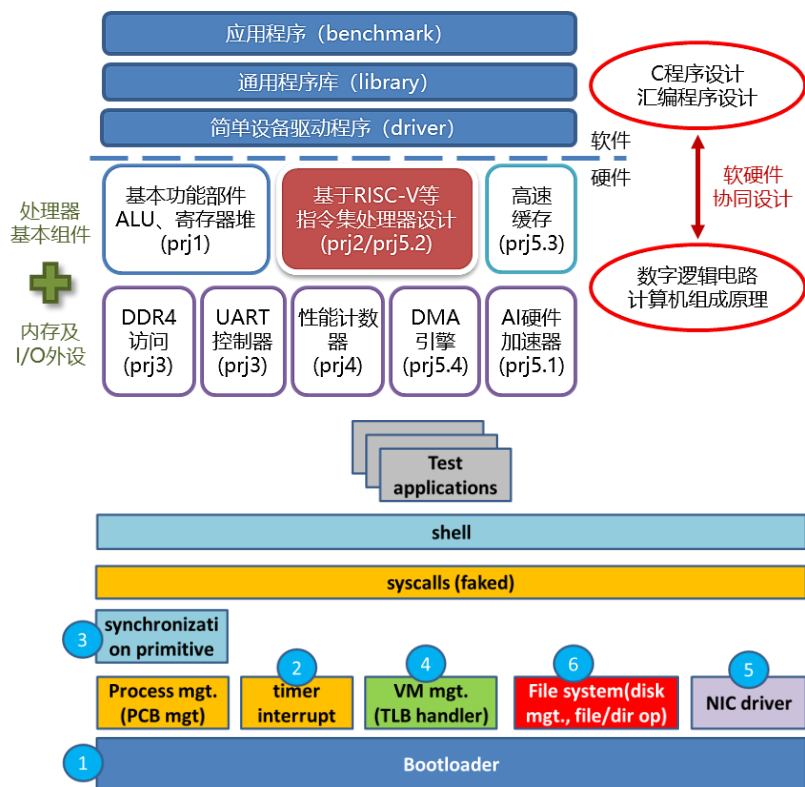
首批合作伙伴 (排名不分先后)

中国科学院计算技术研究所 (ICT, CAS)

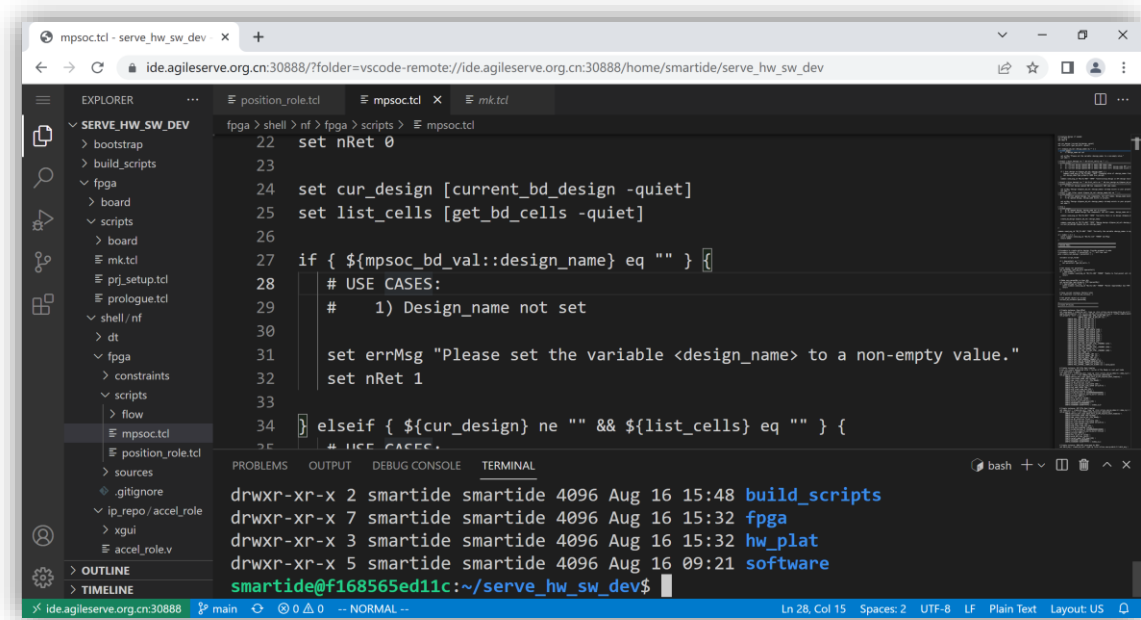
场景1：处理器芯片教学实验项目

- 基于FPGA构建开源开放的计算机组成原理与操作系统配套实验环境
- 基于云端IDE支撑实验项目云上 “发布-开发-测试-优化” 全周期管控

计算机组成原理
配套实验
(框架开源)



操作系统
配套实验



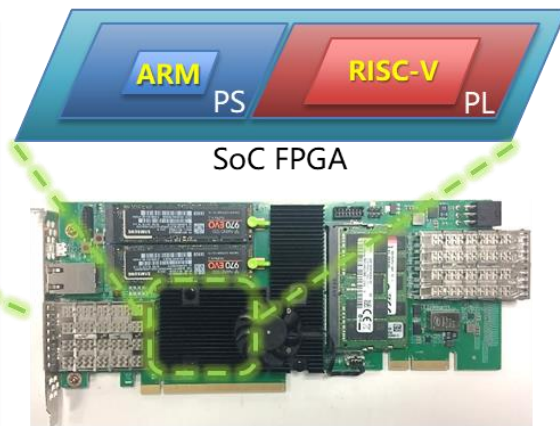
基于SmartIDE的云端代码编辑及开发环境



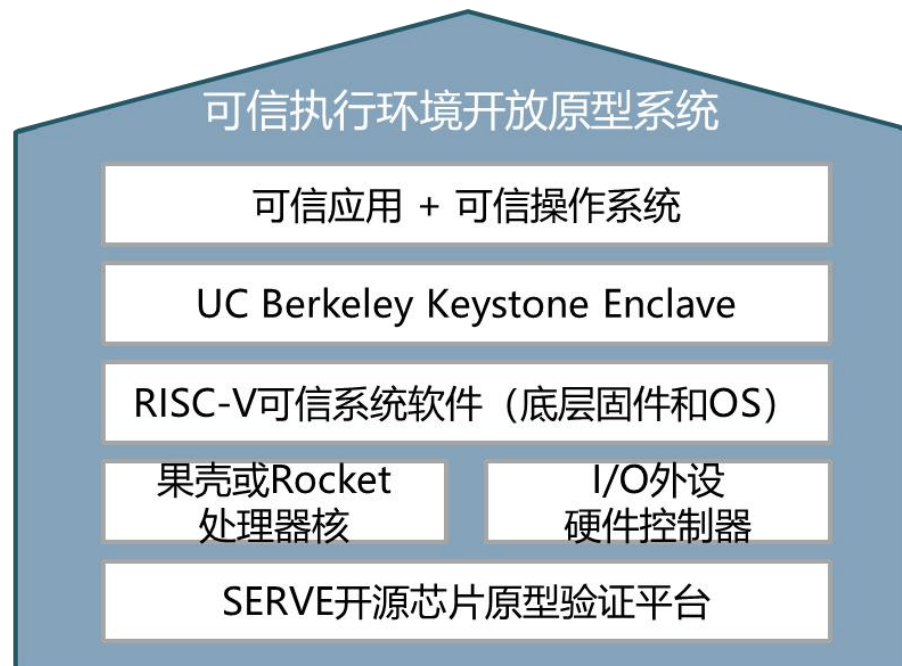
SERVE

场景2: RISC-V可信执行环境开放研究

- 支持Keystone可信执行环境 (TEE) 软硬件全栈环境
- 允许用户基于此环境深入开展可信应用相关研究



单节点FPGA板卡



SERVE



Keystone

<https://keystone-enclave.org>

欢迎使用SERVE并共同创造更多场景!

大纲

一. 香山一年进展

二. 香山敏捷开发实践

三. RISC-V云上“树莓派”

四. 未来展望

让“香山”存活~~30年~~ 29年

- 迭代优化：保持**半年更新一代架构**的迭代优化频率
- 流片验证：**每年2次流片**，对新架构、新功能进行验证
- 开发流程：构建敏捷开发**基础设施**，完善敏捷开发流程
- 开源社区：建立规范的**开源社区管理**机制
- 推广应用：广泛应用于**工业界**，成为**学术界**的创新平台
- 资金人员：稳定的**经费来源**与**核心开发人员**



学术界与工业界的桥梁





香山开源高性能RISC-V处理器

敏捷设计实践

谢谢!
欢迎加入!