



# 香山处理器 Tutorial

---

王凯帆 张紫飞

中科院计算所

2021年6月25日

# 目录

- 一. 香山是什么？
- 二. 如何使用香山？
- 三. 如何测试香山？
- 四. 如何仿真运行香山？
- 五. 如何评估香山性能？

# 目录

一. 香山是什么？

二. 如何使用香山？

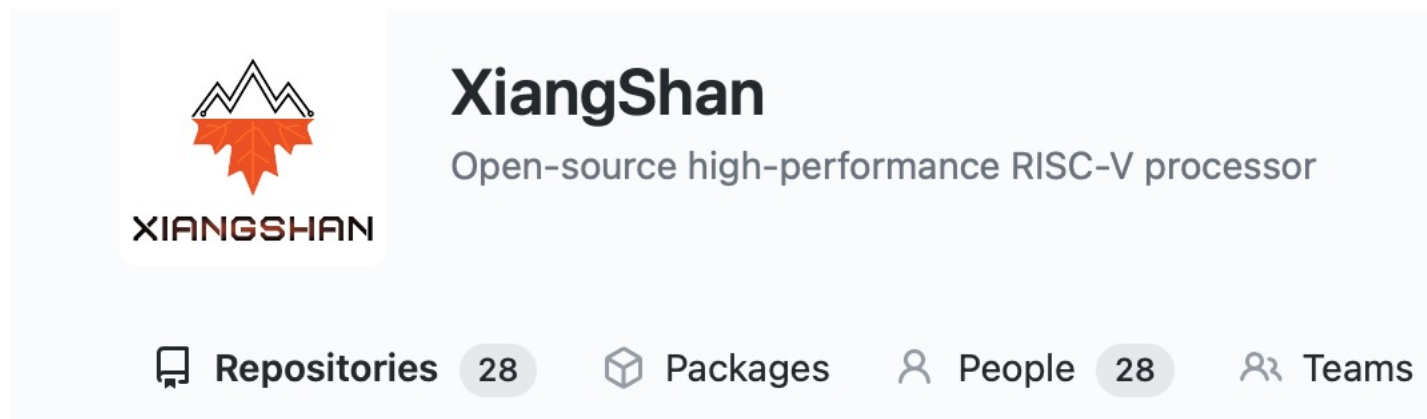
三. 如何测试香山？

四. 如何仿真运行香山？

五. 如何评估香山性能？

# 香山处理器概述

- 一款高性能 RISC-V 处理器，支持 RV64GC 指令集
- 使用 Chisel 硬件设计语言实现，参数化设计
- 实现了差分测试框架、仿真快照、检查点等基础工具
- 建立了包含设计、实现、验证在内的全开源工具敏捷开发流程



<https://github.com/OpenXiangShan>

# 项目结构

debug: 一些跑测试的常用命令，写成了脚本

scripts: 仿真使用的脚本，比如SRAM替换

src: 结构与测试代码

- main/scala

- xiangshan: 香山结构设计

- bus & util...: 其他内容

- test: 单元测试、仿真测试、diff test等内容

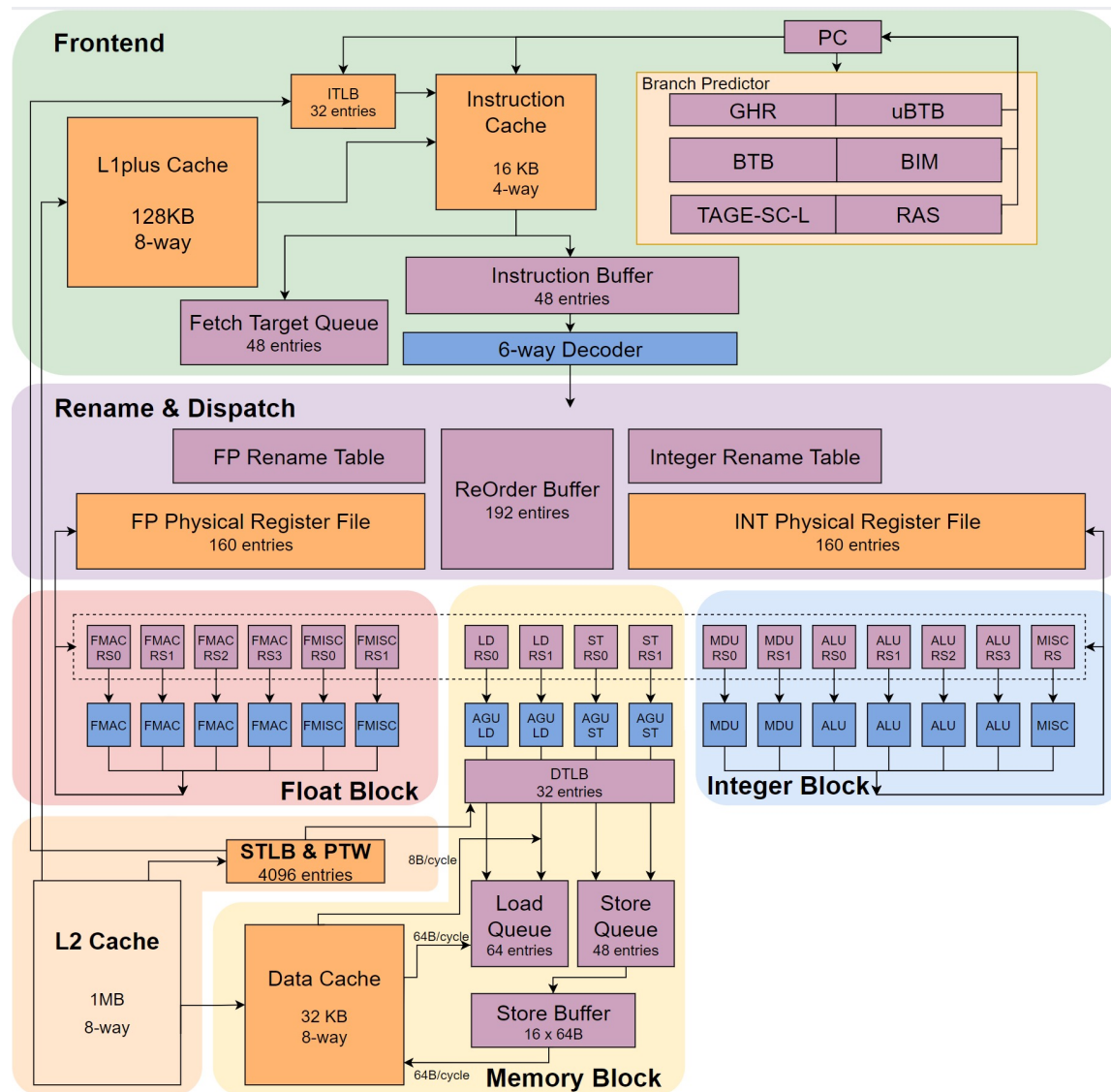
ready-to-run: 预先编译好的nemu-so，和一些workload

submodule

📁	.github/workflows
➔	api-config-chipsalliance @ fd8df11
➔	berkeley-hardfloat @ 8031f85
➔	block-inclusivecache-sifive @ 715d...
➔	chiseltest @ 6a2e177
📁	debug
📁	fpga
📁	project
📁	ready-to-run
➔	rocket-chip @ c7da610
📁	scripts
📁	src
➔	timingScripts @ c5c6162
📁	tools/readmemh

# 逻辑结构

- 前端
  - IFU, BPU...
- 后端
  - Decode, Rename, Rob, Dispatch
  - Integer Block, Float Block
- 访存
  - Memory Block
- 缓存
  - I\$, D\$, L1+\$, L2\$, TLB, Prefetcher



# 目录

一. 香山是什么？

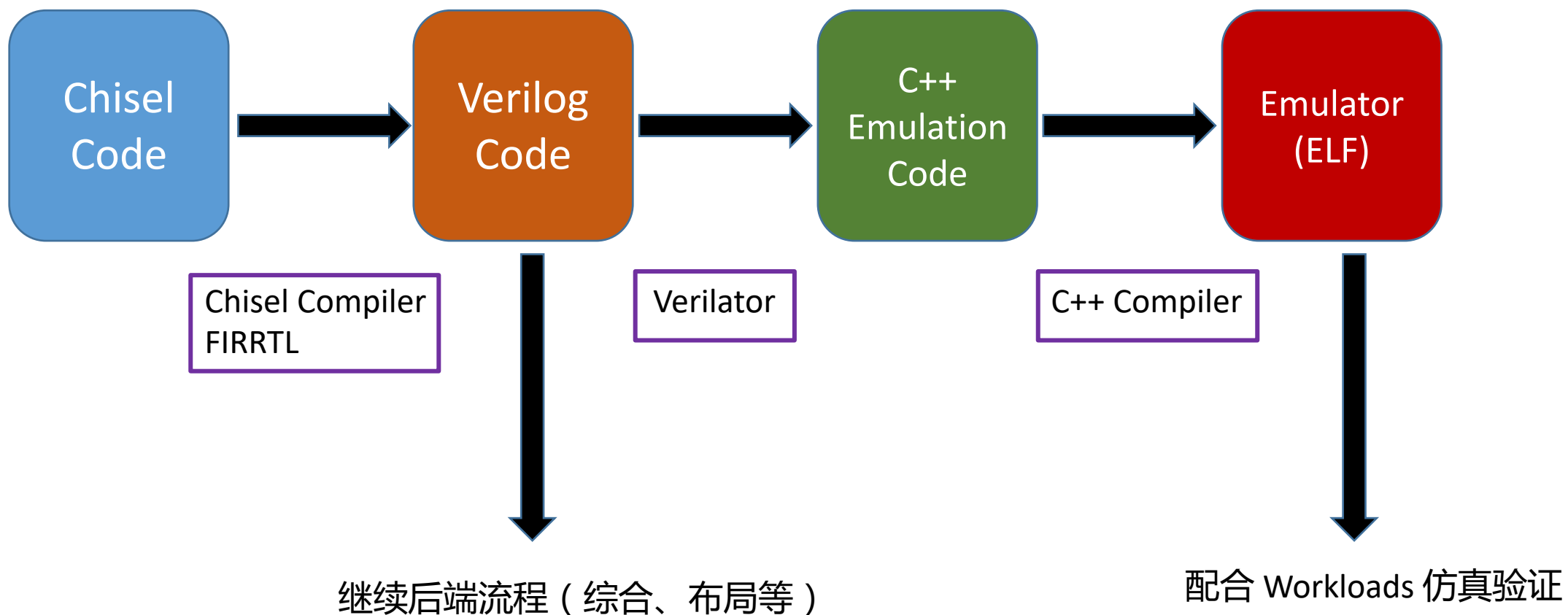
二. 如何使用香山？

三. 如何测试香山？

四. 如何仿真运行香山？

五. 如何评估香山性能？

# Chisel 使用流程





# 参数系统

- Config :
  - [src/main/scala/xiangshan/Parameters.scala](#)
  - 隐式传参
  - 控制核内的大部分配置：
    - 分支预测器的规模和组合
    - 各种队列的大小，如重定序缓冲，保留站，Store Buffer，Cache和TLB的大小
  - [src/main/scala/top/Configs.scala](#)
  - default config: 默认配置，编译时间较长
  - minimal config: 迷你配置，编译时间较短
  - make emu **CONFIG=MinimalConfig**

# 目录

一. 香山是什么？

二. 如何使用香山？

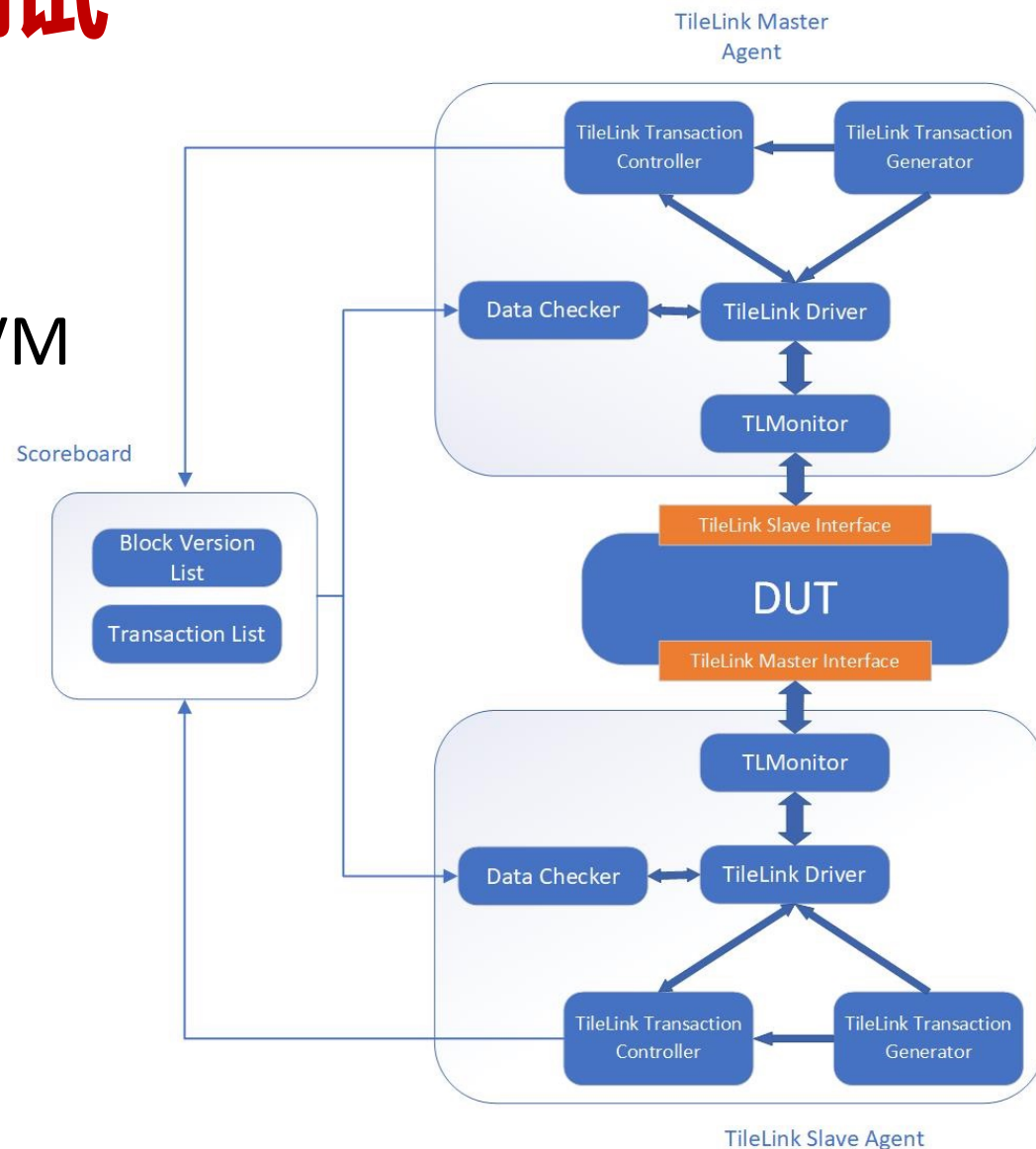
三. 如何单元测试香山？

四. 如何仿真运行香山？

五. 如何评估香山性能？

# 单元测试

- ChiselTesters2
  - 使用ChiselTester2可以进行单元测试
- Agent-Faker: 基于ChiselTester2的类UVM的单元测试
  - DCache
  - L2Cache
  - TLB

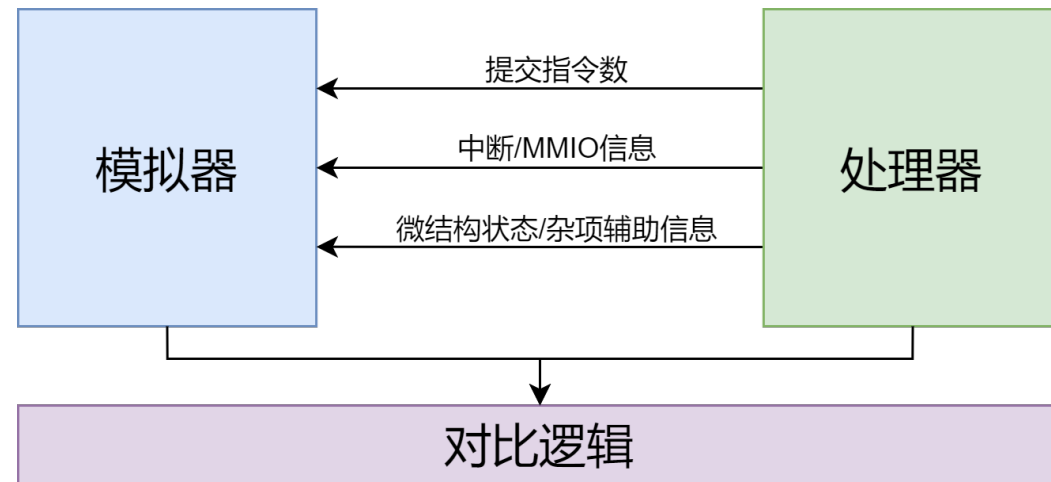


# 目录

- 一. 香山是什么？
- 二. 如何使用香山？
- 三. 如何单元测试香山？
- 四. 如何仿真运行香山？
- 五. 如何评估香山性能？

# 仿真框架

- 仿真工具: Verilator
- 差分测试框架：
  - 与模拟器 NEMU 做运行时比对
- 模拟外设: uart , sd卡等等



```
while (1) {  
    icnt = cpu_step();  
    nemu_step(icnt);  
    r1s = cpu_getregs();  
    r2s = nemu_getregs();  
    if (r1s != r2s) { abort(); }  
}
```

在线验证机制

# 仿真框架

- 波形
  - 使用GTKwave查看
  - Wave Terminator: 分析处理波形的工具
- 快照:
  - Snapshot & LightSSS
  - 保存现场，恢复执行
- 日志:
  - 模块级地控制日志打印
  - LogViewer 日志查看器

# NEMU

- NEMU 是一执行速度媲美 QEMU 的解释器
- 两种模式：执行模式和Difftest模式
- 执行模式
  - make ISA=riscv64 XIANGSHAN=1 -j8
- Difftest模式
  - make ISA=riscv64 XIANGSHAN=1 SHARE=1 -j8
- 香山处理器使用 NEMU 作为对照

# Nexus-AM

- AM (Abstract Machine) 向程序提供了裸机运行时环境
- 提供最简单的运行时环境
- 提供printf , memcpy等常用库函数
- 支持裸机环境下运行coremark, dhrystone等工作load
- 编译(自定义)程序裸机执行
- 执行裸机程序
  - make ISA=riscv64-xs # 编译workload
  - path/to/emu -i coremark.bin # 手动使用emu执行程序



# 目录

- 一. 香山是什么？
- 二. 如何使用香山？
- 三. 如何单元测试香山？
- 四. 如何仿真运行香山？
- 五. 如何评估香山性能？

# 性能评估

- 性能计数器
  - 性能计数器统计
    - 仿真：每个模块可以自由打印计数器
    - FPGA：自定义CSR寄存器
  - 性能计数器分析：脚本&可视化分析工具
- SPEC测试框架
  - SimPoint大规模并行测试

**Lab Time !**

# Github 搜索 “XiangShan”



Tutorial动手教程



Linux Kernel的创建



Light SSS的使用



使用AM生成自定义workload



香山 Tutorial 讨论群

**感谢** 

北京微核芯科技有限公司  
BEIJING VCORE TECHNOLOGY CO., LTD.

**提供产业经验、联合完成结构设计及物理设计**

**招募香山处理器二期联合开发合作伙伴**



北京微核芯科技有限公司  
BEIJING VCORE TECHNOLOGY CO., LTD.



**ESWIN**

优矽科技

**欢迎更多伙伴加入！**

**联系人：李迪 13716821562**

**敬请批评指正！**