中国科学院计算技术研究所
INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

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

XiangShan: Industrial-Competitive RISC-V CPUs in the Era of Open-Source Chips

Yinan Xu 徐易难

Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS)

October 2025

Outline

- The Era of Open-Source Chips
- Open-Source Industrial-Competitive RISC-V Chips
- Open-Source Chip Design Tools
- Open-Source Development & Business Models
- Conclusions

Dozens of ISAs in the past 50 years

- *Over the past half-century*, **dozens** of instruction sets have appeared, **all owned by private companies**.
  - Most have *disappeared* as their companies merged or shut down.

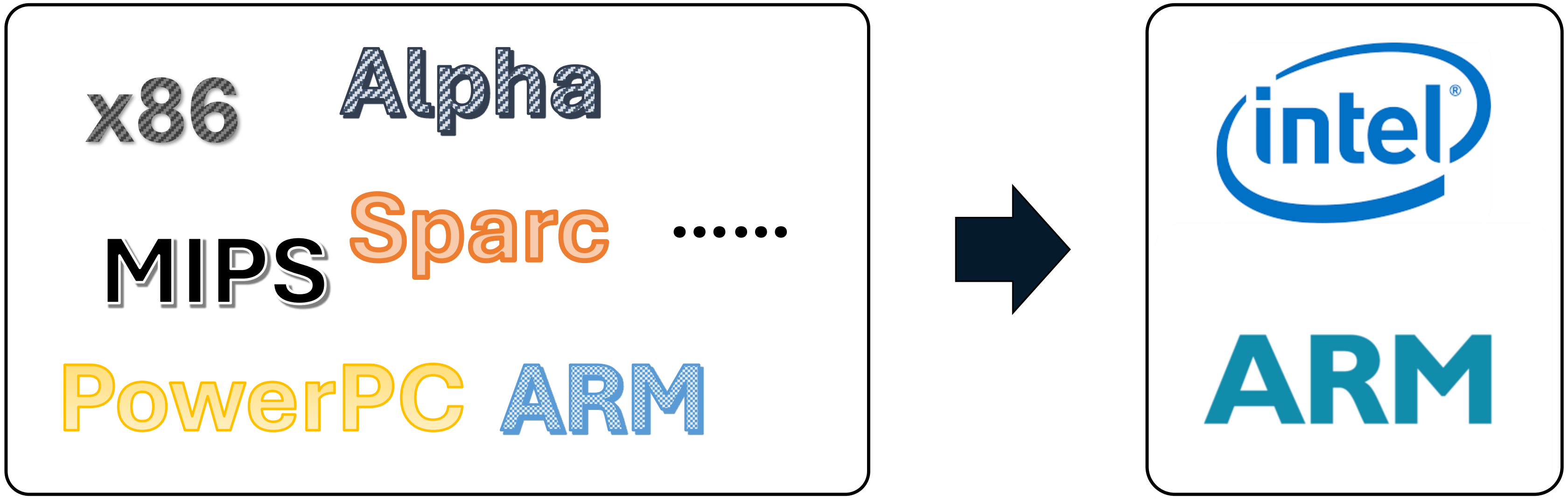
x86 Alpha

MIPS Sparc ……

PowerPC ARM

Today: CPU Market Dominated by x86/ARM

- *Over the past half-century*, **dozens** of instruction sets have appeared, **all owned by private companies**.
  - Most have *disappeared* as their companies merged or shut down.
- *Today*, **only x86 and ARM** remain as *mainstream* instruction sets, each with its own ecosystem.



x86 Alpha

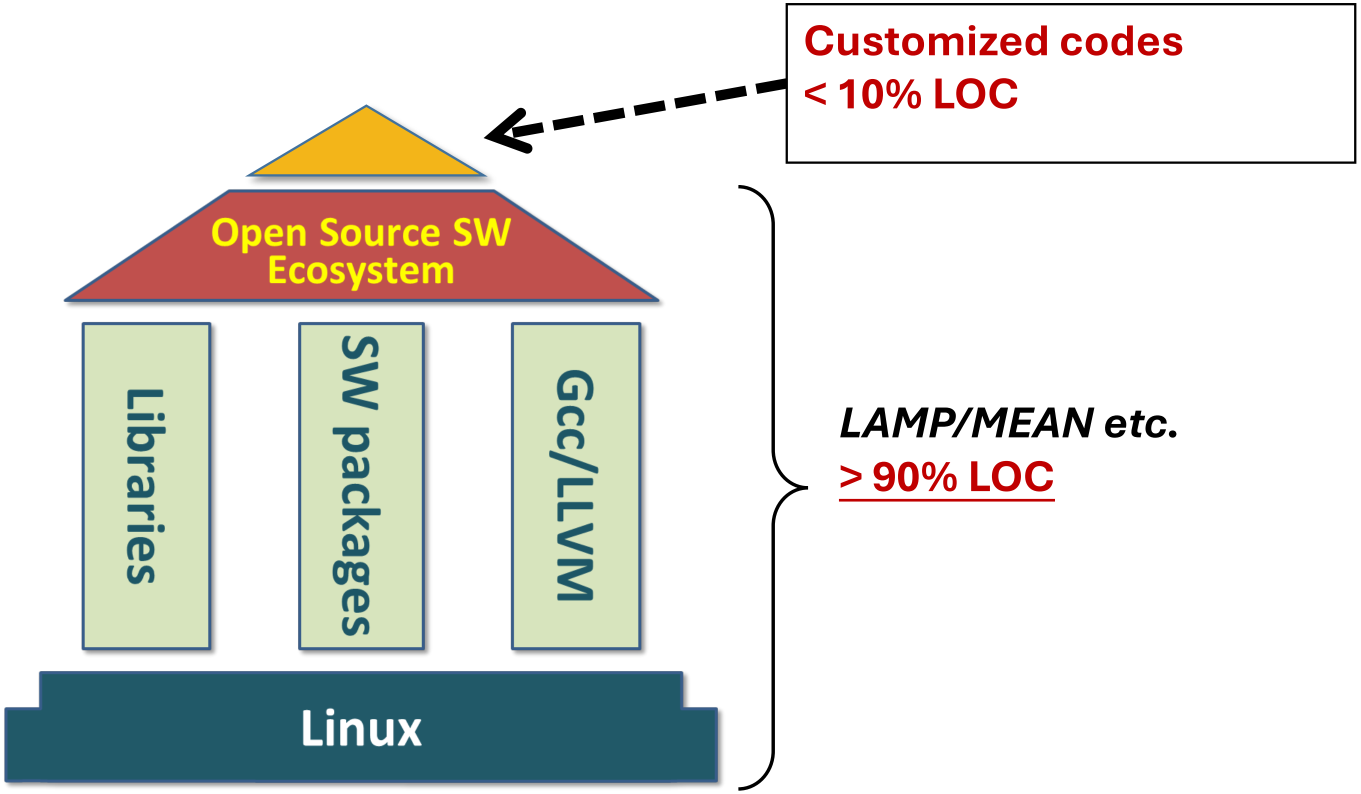
MIPS Sparc ……

PowerPC ARM

intel

ARM

Open-Source Software Ecosystem



Customized codes
< 10% LOC

Open Source SW Ecosystem

Libraries

SW packages

Gcc/LLVM

Linux

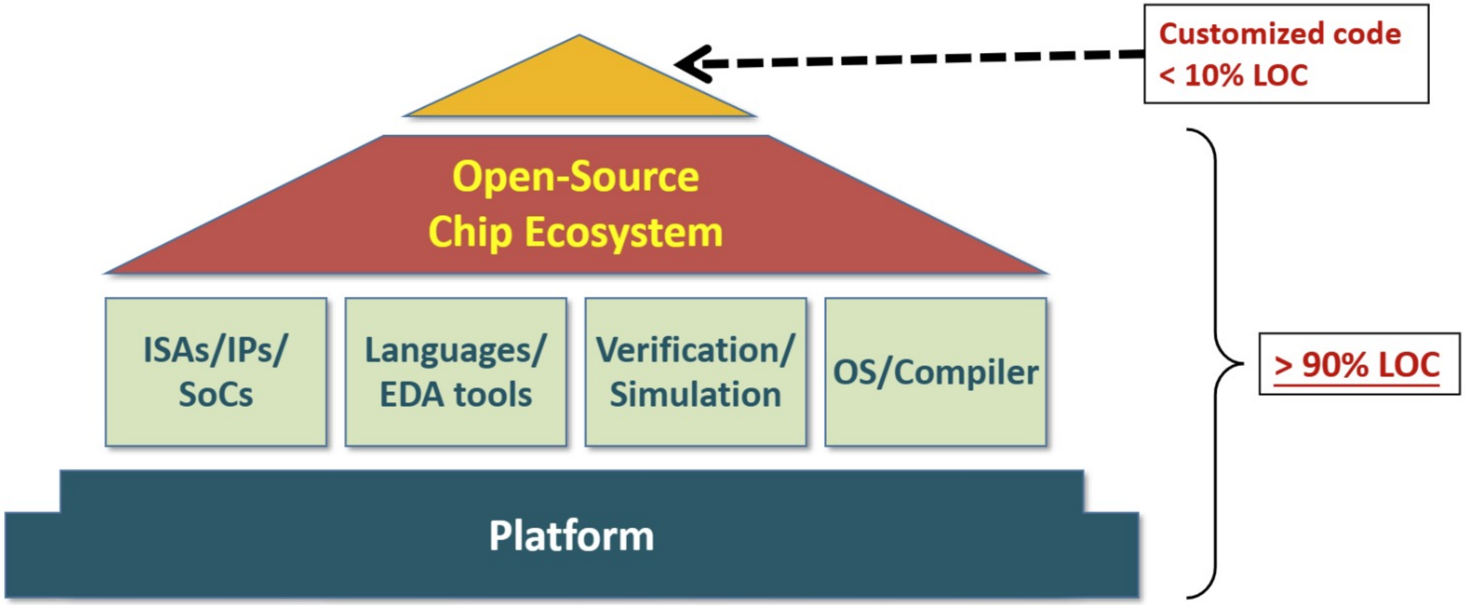
LAMP/MEAN etc.
> 90% LOC

Mirror the success of the open-source software ecosystem?

Open-Source Chip Ecosystem (OSCE)

To Lower the barrier of chip development

By saving the cost of IPs, EDA tools and engineers in chip design



* Yungang Bao, The Four Steps to An Open-Source Chip Design Ecosystem, ACM SIGARCH Visioning Workshop, June 2019

Three levels of OSCE

L1: OPEN ISA

L2: OPEN Design & Implementation

L3: OPEN Tools & Infrastructure

3 Open Tools & Infrastructure

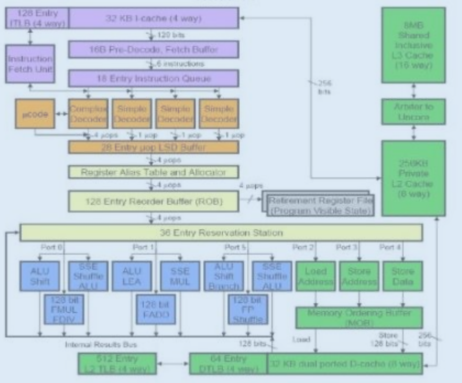
Microarch. → Engineering → EDA Tools

ISA Spec.

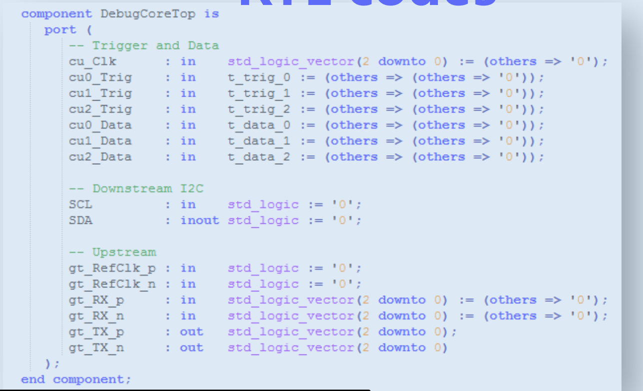
RISC-V®

1 Open ISA

Docs

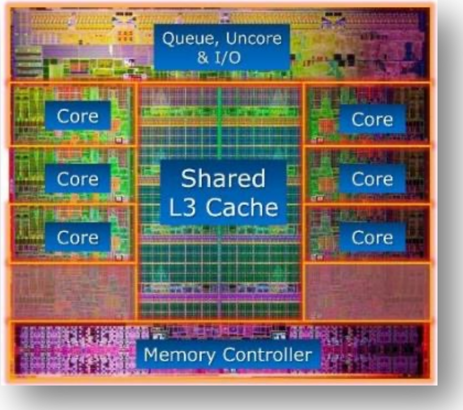


RTL codes


```
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;
```

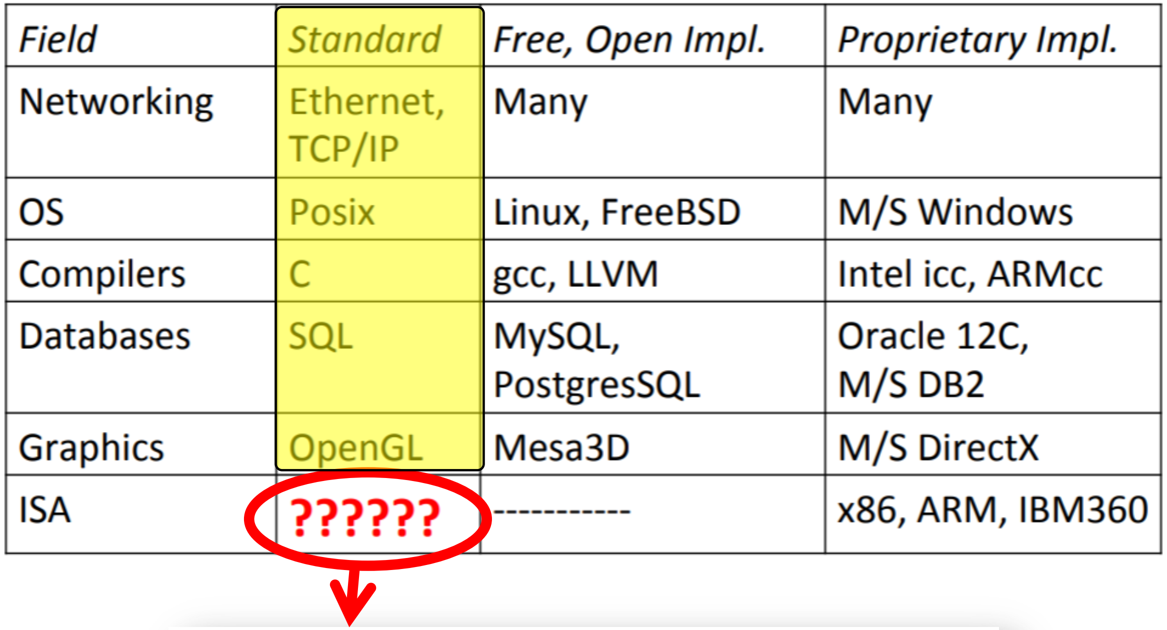
2 Open Design & Implt

Layout



Open Standards: Instruction Sets Want to be Free!

- In 2010, UC Berkeley launched the RISC-V free and open ISA.



Field	*Standard*	*Free, Open Impl.*	*Proprietary Impl.*
Networking	Ethernet, TCP/IP	Many	Many
OS	Posix	Linux, FreeBSD	M/S Windows
Compilers	C	gcc, LLVM	Intel icc, ARMcc
Databases	SQL	MySQL, PostgresSQL	Oracle 12C, M/S DB2
Graphics	OpenGL	Mesa3D	M/S DirectX
ISA	??????	-----------	x86, ARM, IBM360

RISC-V

Instruction Sets Want to be Free!

Open-Source: Democratization and Building Consensus

intel

ARM

V.S.

RISC-V

Instruction Sets Want to be Free!

- Private owned
- Monopoly

- Open development
- Open and shared ecosystem

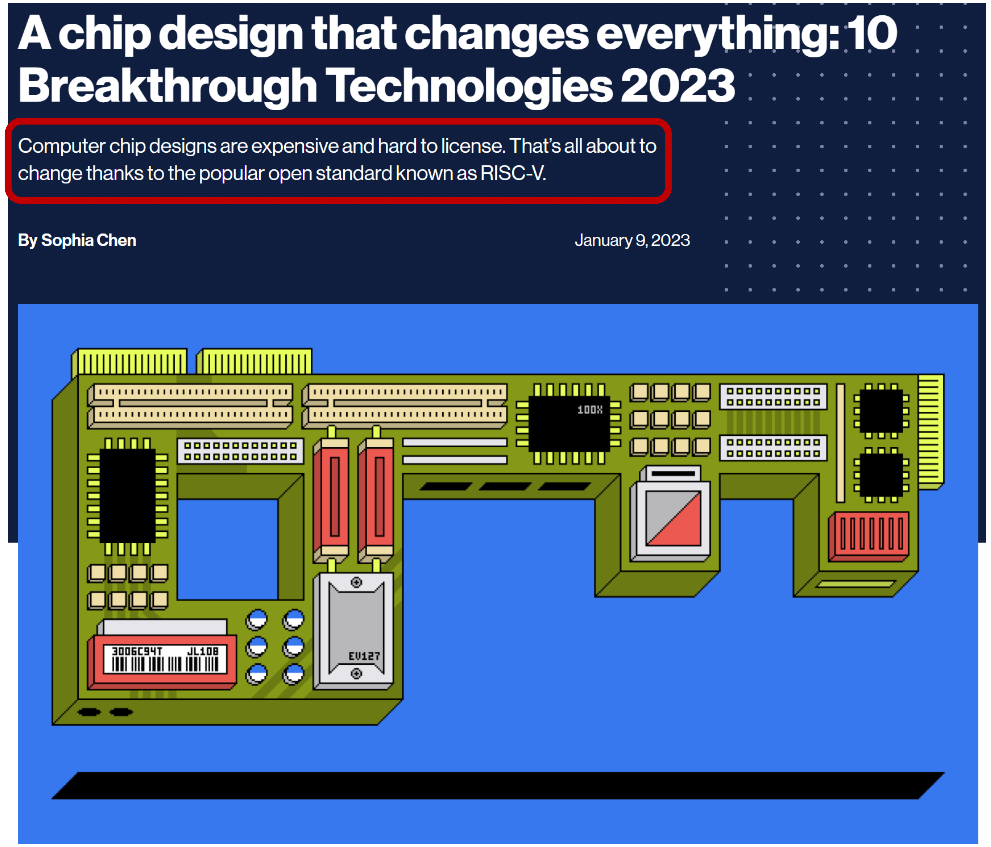
A chip design that changes everything

- **10 Breakthrough Technologies 2023**

Ever wonder how your smartphone connects to your Bluetooth speaker, given they were made by different companies? Well, Bluetooth is an open standard, meaning its design specifications, such as the required frequency and its data encoding protocols, are publicly available. Software and hardware based on open standards—Ethernet, Wi-Fi, PDF—have become household names.

Now an open standard known as RISC-V (pronounced "risk five") could change how companies create computer chips.

--- MIT Technology Review



A chip design that changes everything: 10 Breakthrough Technologies 2023

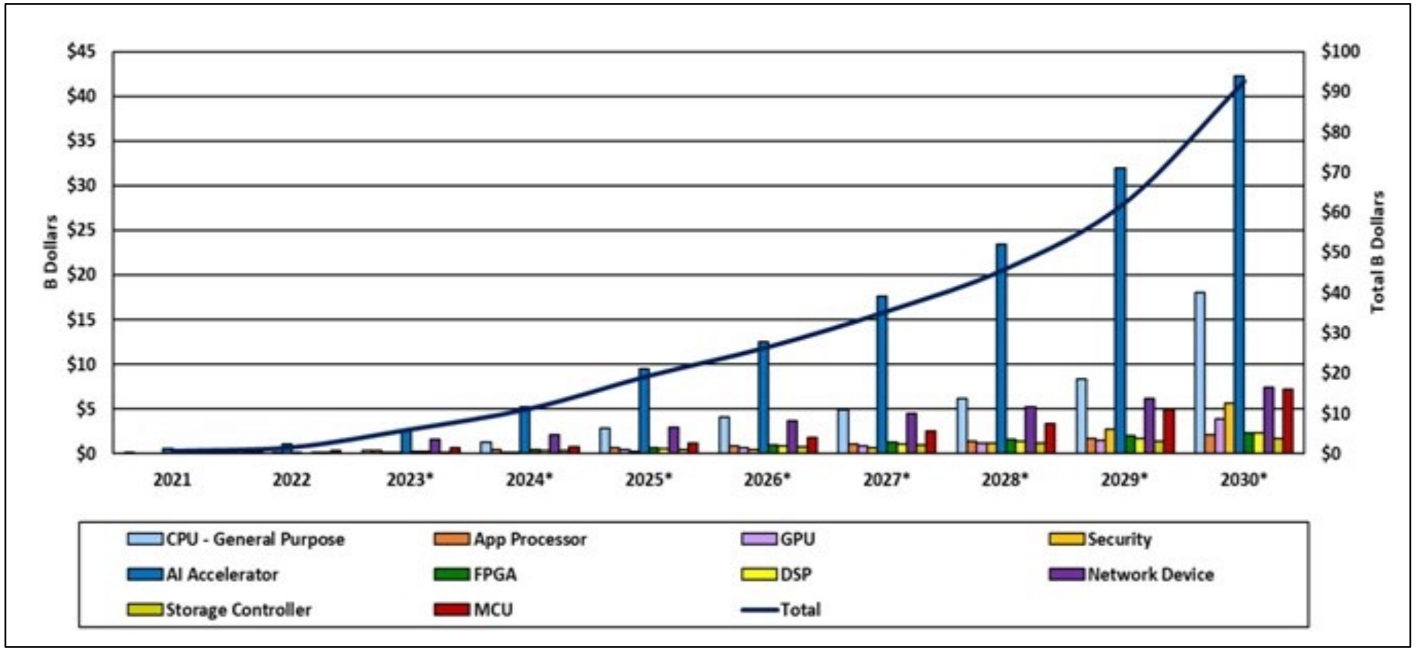
Computer chip designs are expensive and hard to license. That's all about to change thanks to the popular open standard known as RISC-V.

By Sophia Chen

January 9, 2023

RISC-V: A mainstream market (> $90 billion) by 2030

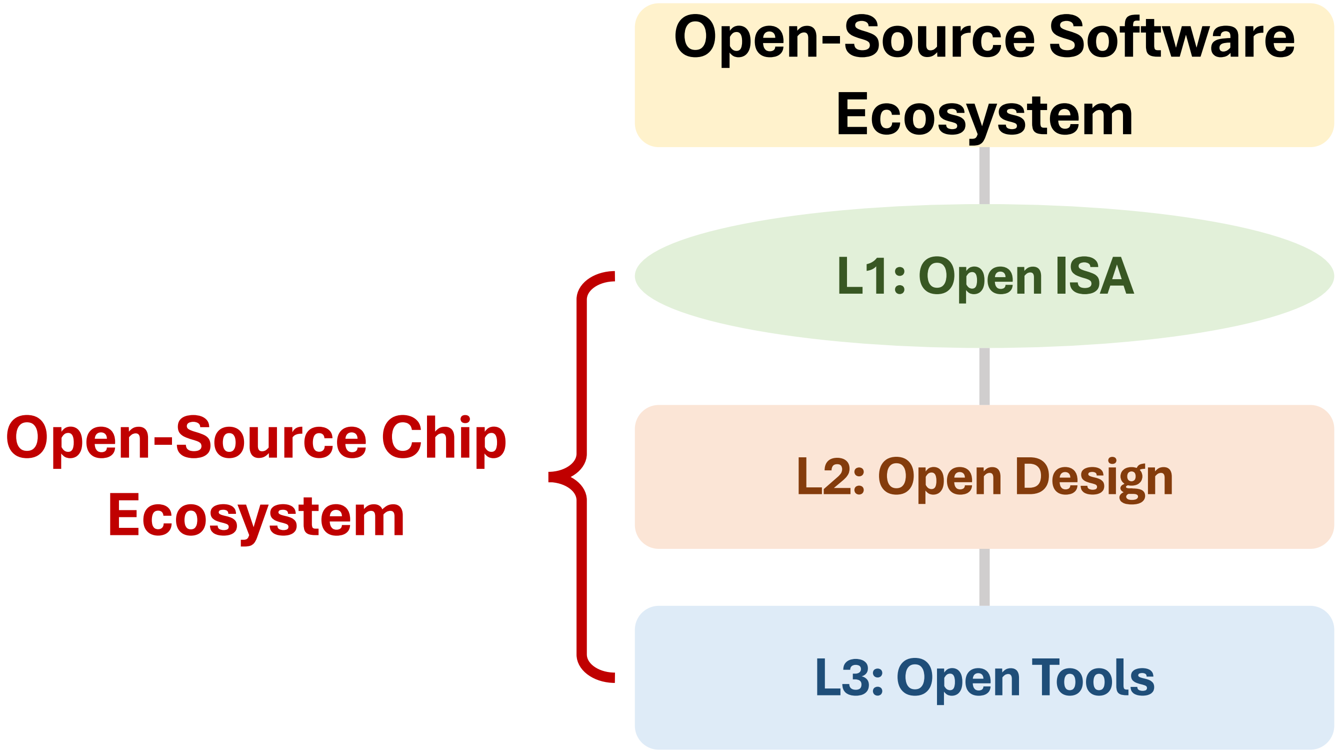
- The global market size is projected to reach **$92.7 billion** by 2030, with a compound annual growth rate of **47.4%.**



SHDgroup

Global Revenue Forecast for RISC-V SoC Chips

The XiangShan Project



Open-Source Software Ecosystem

L1: Open ISA

L2: Open Design

L3: Open Tools

Open-Source Chip Ecosystem

RISC-V®

XIANGSHAN

Outline

- The Era of Open-Source Chips
- Open-Source Industrial-Competitive RISC-V Chips
- Open-Source Chip Design Tools
- Open-Source Development & Business Models
- Conclusions

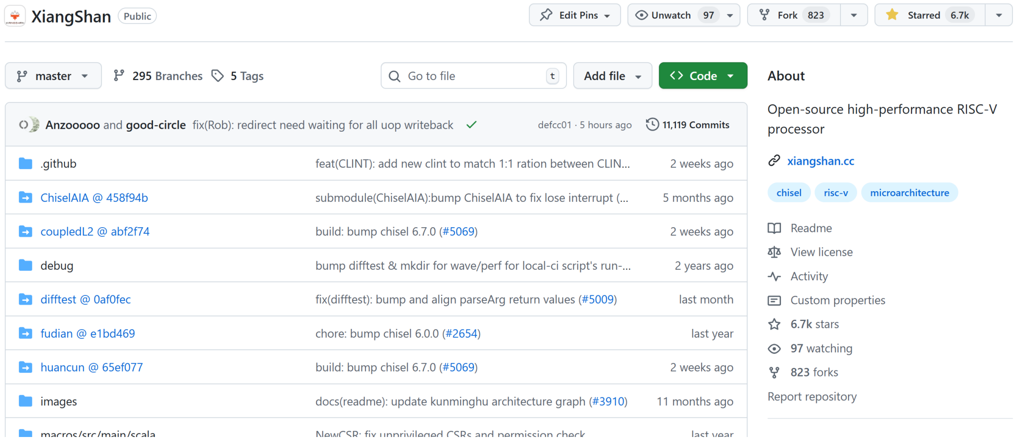
XiangShan: Open-Source High Performance Processors

L1: OPEN ISA (RISC-V)

L2: OPEN Design/Implementation

L3: OPEN Framework/Tools

Fragrant Hill in Beijing



XiangShan Public

Anzooooo and good-circle fix(Rob): redirect need waiting for all uop writeback ✓ defcc01 · 5 hours ago 11,119 Commits

.github	feat(CLINT): add new clint to match 1:1 ration between CLIN...	2 weeks ago
ChiselAIA @ 458f94b	submodule(ChiselAIA):bump ChiselAIA to fix lose interrupt (...	5 months ago
coupledL2 @ abf2f74	build: bump chisel 6.7.0 (#5069)	2 weeks ago
debug	bump difftest & mkdir for wave/perf for local-ci script's run-...	2 years ago
difftest @ 0af0fec	fix(difftest): bump and align parseArg return values (#5009)	last month
fudian @ e1bd469	chore: bump chisel 6.0.0 (#2654)	last year
huancun @ 65ef077	build: bump chisel 6.7.0 (#5069)	2 weeks ago
images	docs(readme): update kunminghu architecture graph (#3910)	11 months ago
macros/src/main/scala	NewCSR: fix unprivileged CSRs and permission check	last year

About

Open-source high-performance RISC-V processor

xiangshan.cc

chisel risc-v microarchitecture

6.7k stars

97 watching

823 forks

> 6.7K stars, > 820 forks on GitHub

The Open-Source RISC-V CPU

- **Vision:** To establish an open-source RISC-V core backbone like **Linux**, which can be widely used in *industry* and support innovative ideas from *academia*.

Linux

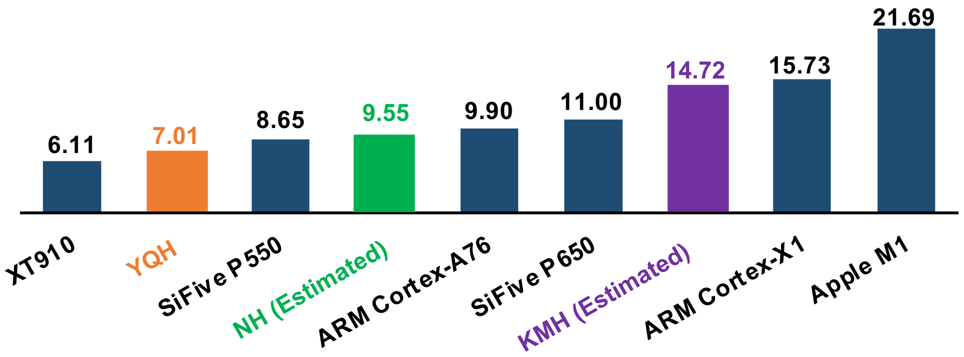
V.S.

51

XiangShan: Open-Source High Performance Processors

https://github.com/OpenXiangShan/XiangShan

- 1st generation: Yanqihu (YQH)
  - RV64GC, single-core, superscalar OoO
  - **28nm tape-out, 1.3GHz, July 2021**
  - **SPEC CPU2006 7.01@1GHz, DDR4-1600**
- 2nd generation: Nanhu (NH)
  - RV64GCBK, dual-core, superscalar OoO
  - **14nm GDSII delivery, 2GHz, 2023 Q3**
  - **Estimated** SPECint 2006 19.10@2GHz**
- 3rd generation: Kunminghu (KMH)
  - RV64GCBKHV, quad-core, superscalar OoO
  - **Advanced-node, 3GHz, 1.5x IPC of NH**
  - **Close collaboration with industrial partners**



SPECint 2006/GHz* (Proportional to IPC)

* Source: XT910@ISCA'20, SiFive, AnandTech

** Updated January 5, 2023

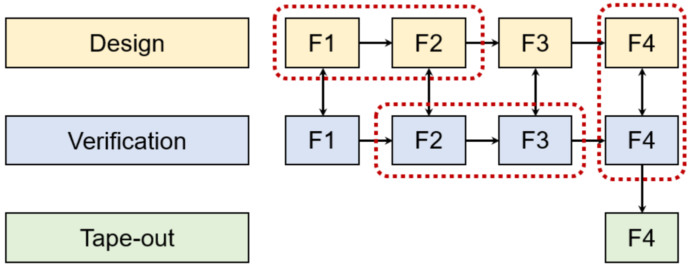
XiangShan Gen 3: Kunminghu

- **Target ARM Neoverse N2**
  - SPECCPU2006: 45@3GHz (15/GHz)
  - Vector/Hypervisor extension supported
- **A Joint Dev Team** (coordinated by BOSC)

Kunming Lake in Beijing

Highlights in XiangShan Gen 3 Kunminghu

- **Functional Enhancement**
  - Support RISC-V **Vector/Hypervisor** extension
  - Support **RVA23 profile**
  - Support interconnection based on **CHI protocol**
- **Performance Exploration**
  - Performance boost in frontend, backend, load-stor
  - Performance model calibrated with RTL
  - Workflow: **DSE on perf model => Impl. & fine tunning on RTL**
- **Functional Verification**
  - **Hierarchical verification flow** spanning system/integration/unit level + FPGA prototyping
  - Industrial-grade verification process
- **Physical Design**
  - Experienced physical design team
  - Simultaneous iteration of RTL coding based on timing evaluation



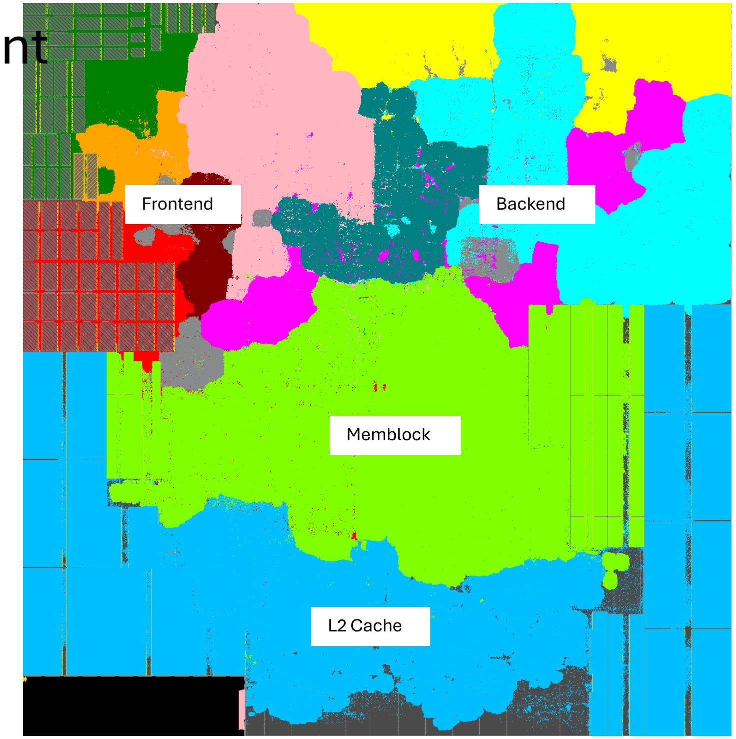
Agile Development Model

Performance Evaluation of Gen 3 Kunminghu

- Method: SPEC CPU checkpoints selected by Simpoint
  - Base: GCC 12 –O3, RV64GCB, jemalloc
  - 1MB L2 and 16MB L3
  - Simulated@3GHz with DRAMsim3 DDR4-3200

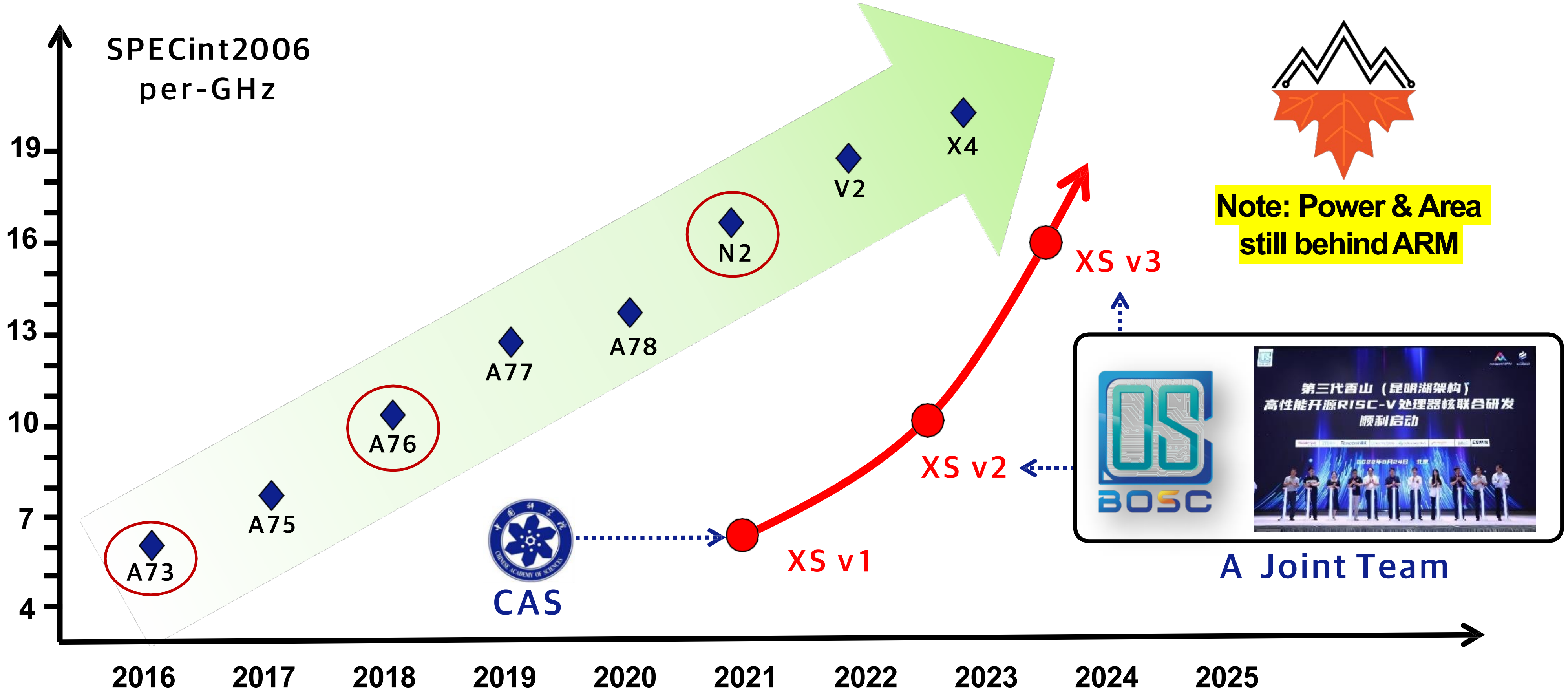
SPECint 2006 est.@ 3GHz		SPECfp 2006 est.@ 3GHz	
400.perlbench	35.88	410.bwaves	66.89
401.bzip2	25.50	416.gamess	40.89
403.gcc	46.72	433.milc	45.25
429.mcf	58.13	434.zeusmp	52.10
445.gobmk	30.26	435.gromacs	33.65
456.hmmer	41.60	436.cactusADM	46.16
458.sjeng	30.53	437.leslie3d	46.01
462.libquantum	122.50	444.namd	28.88
464.h264ref	56.57	447.dealII	73.43
471.omnetpp	39.37	450.soplex	51.99
473.astar	29.23	453.povray	53.44
483.xalancbmk	72.03	454.Calculix	16.38
GEOMEAN	**44.15**	459.GemsFDTD	37.18
		465.tonto	36.67
		470.lbm	91.24
		481.wrf	40.62
		482.sphinx3	48.57
		GEOMEAN	**44.60**

* Updated in March 2025



Floorplan of KMH V2R2 (single core)

XiangShan: Open-Source High Performance Processors

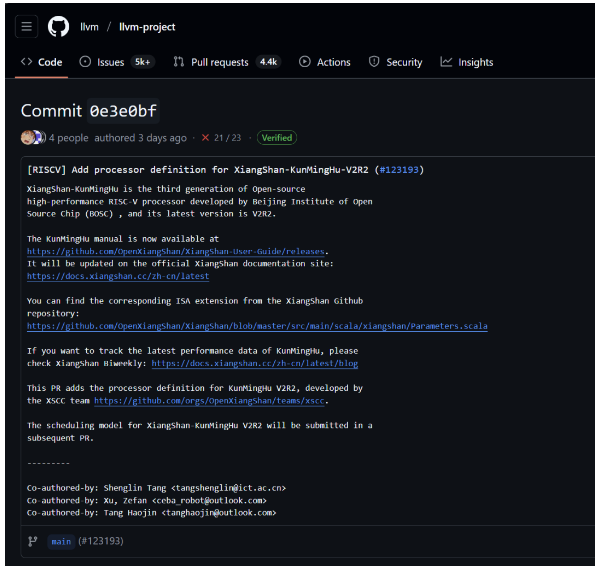


A Joint Team

Note: Power & Area still behind ARM

Compiler Support for Nanhu and Kunminghu

- LLVM & GCC support XiangShan optimizations (BOSC, ISCAS, ICT)

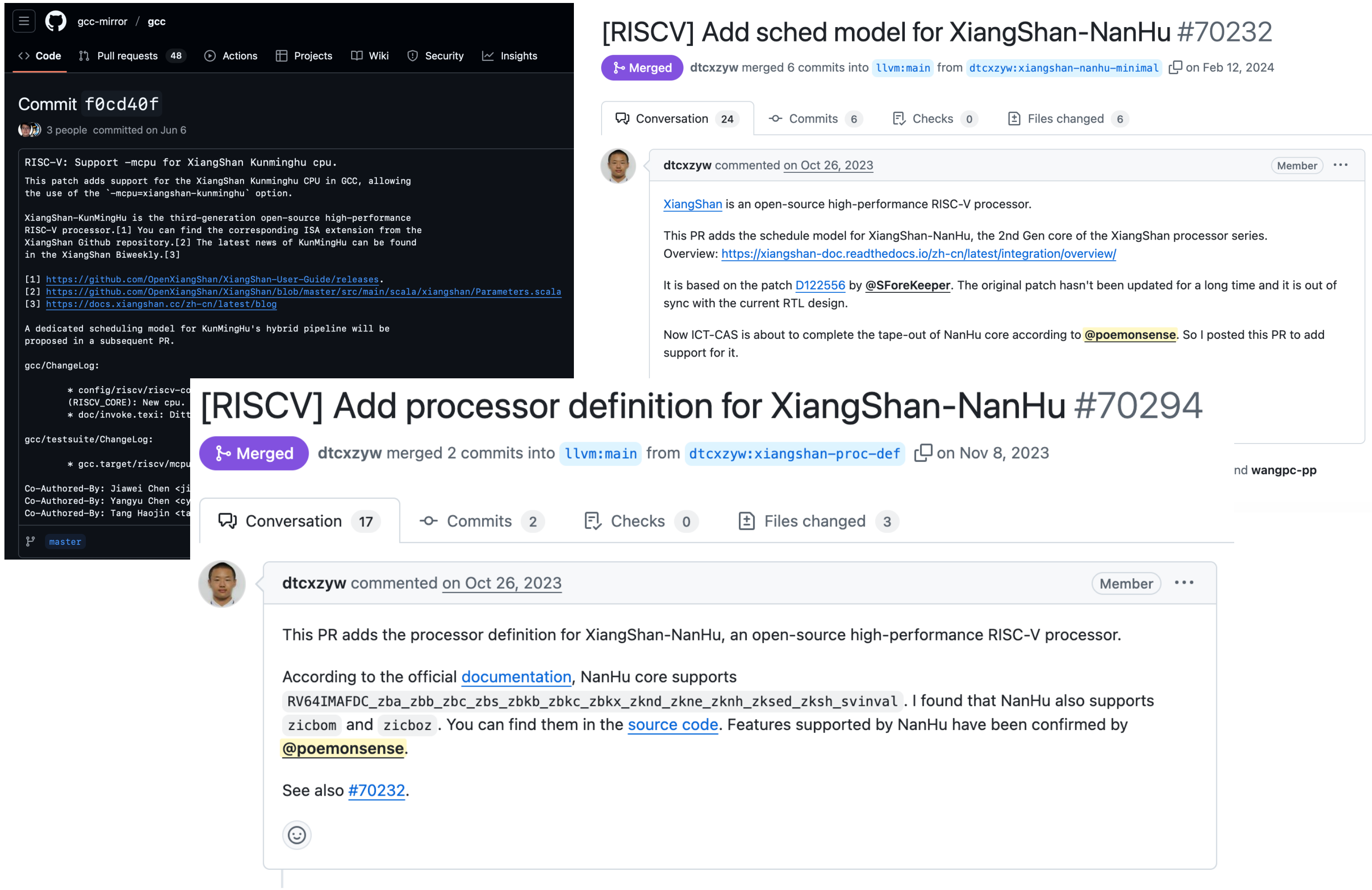


Commit 0e3e0bf

4 people authored 3 days ago · ✕ 21 / 23 · Verified

```
[RISCV] Add processor definition for XiangShan-KunMingHu-V2R2 (#123193)

XiangShan-KunMingHu is the third generation of Open-source
high-performance RISC-V processor developed by Beijing Institute of Open
Source Chip (BOSC) , and its latest version is V2R2.

The KunMingHu manual is now available at
https://github.com/OpenXiangShan/XiangShan-User-Guide/releases.
It will be updated on the official XiangShan documentation site:
https://docs.xiangshan.cc/zh-cn/latest

You can find the corresponding ISA extension from the XiangShan Github
repository:
https://github.com/OpenXiangShan/XiangShan/blob/master/src/main/scala/xiangshan/Parameters.scala

If you want to track the latest performance data of KunMingHu, please
check XiangShan Biweekly: https://docs.xiangshan.cc/zh-cn/latest/blog

This PR adds the processor definition for KunMingHu V2R2, developed by
the XSCC team https://github.com/orgs/OpenXiangShan/teams/xscc.

The scheduling model for XiangShan-KunMingHu V2R2 will be submitted in a
subsequent PR.

---------

Co-authored-by: Shenglin Tang <tangshenglin@ict.ac.cn>
Co-authored-by: Xu, Zefan <ceba_robot@outlook.com>
Co-authored-by: Tang Haojin <tanghaojin@outlook.com>
```

main (#123193)



Commit f0cd40f

3 people committed on Jun 6

```
RISC-V: Support -mcpu for XiangShan Kunminghu cpu.

This patch adds support for the XiangShan Kunminghu CPU in GCC, allowing
the use of the `-mcpu=xiangshan-kunminghu` option.

XiangShan-KunMingHu is the third-generation open-source high-performance
RISC-V processor.[1] You can find the corresponding ISA extension from the
XiangShan Github repository.[2] The latest news of KunMingHu can be found
in the XiangShan Biweekly.[3]

[1] https://github.com/OpenXiangShan/XiangShan-User-Guide/releases.
[2] https://github.com/OpenXiangShan/XiangShan/blob/master/src/main/scala/xiangshan/Parameters.scala
[3] https://docs.xiangshan.cc/zh-cn/latest/blog

A dedicated scheduling model for KunMingHu's hybrid pipeline will be
proposed in a subsequent PR.

gcc/ChangeLog:

        * config/riscv/riscv-co
        (RISCV_CORE): New cpu.
        * doc/invoke.texi: Ditt

gcc/testsuite/ChangeLog:

        * gcc.target/riscv/mcpu

Co-Authored-By: Jiawei Chen <ji
Co-Authored-By: Yangyu Chen <cy
Co-Authored-By: Tang Haojin <ta
```

master

[RISCV] Add sched model for XiangShan-NanHu #70232

Merged dtcxzyw merged 6 commits into llvm:main from dtcxzyw:xiangshan-nanhu-minimal on Feb 12, 2024

dtcxzyw commented on Oct 26, 2023 Member

XiangShan is an open-source high-performance RISC-V processor.

This PR adds the schedule model for XiangShan-NanHu, the 2nd Gen core of the XiangShan processor series.
Overview: https://xiangshan-doc.readthedocs.io/zh-cn/latest/integration/overview/

It is based on the patch D122556 by @SForeKeeper. The original patch hasn't been updated for a long time and it is out of sync with the current RTL design.

Now ICT-CAS is about to complete the tape-out of NanHu core according to @poemonsense. So I posted this PR to add support for it.

nd **wangpc-pp**

[RISCV] Add processor definition for XiangShan-NanHu #70294

Merged dtcxzyw merged 2 commits into llvm:main from dtcxzyw:xiangshan-proc-def on Nov 8, 2023

dtcxzyw commented on Oct 26, 2023 Member

This PR adds the processor definition for XiangShan-NanHu, an open-source high-performance RISC-V processor.

According to the official documentation, NanHu core supports `RV64IMAFDC_zba_zbb_zbc_zbs_zbkb_zbkc_zbkx_zknd_zkne_zknh_zksed_zksh_svinval`. I found that NanHu also supports `zicbom` and `zicboz`. You can find them in the source code. Features supported by NanHu have been confirmed by **@poemonsense**.

See also #70232.

OpenXiangShan: Empowering Architecture Research

- **An open-source, continuously developing research platform**

Your research field	XiangShan provides ...
Microarchitecture	**Performance**: An industrial-competitive, high-performance superscalar OoO microarchitecture
	Functionality: RVA23-compatible RISC-V design
	Development: mature, user-friendly design flows
	Tapeouts by the XiangShan team and leading industrial partners with real-world deployment
Chip development tool	Realistic and challenging **research problems**

An Effective Infrastructure for Research

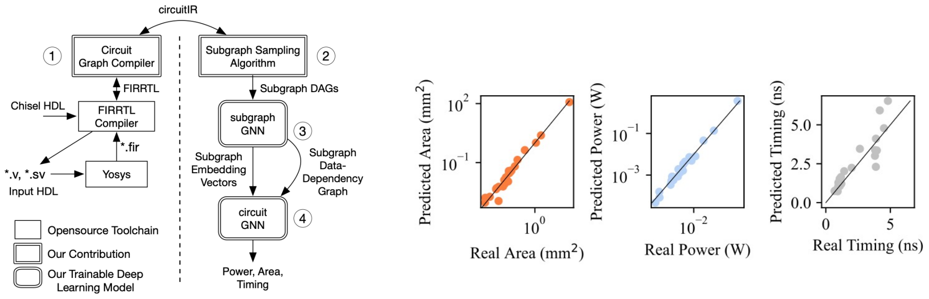
- ***Topic: Computer Architecture***
  - XiangShan: a realistic out-of-order RISC-V implementation with industry-competitive performance and an active open-source community
  - MinJie provides the toolchains
- *Microarchitecture, accelerators, novel architectures, profiling, systems, benchmarking, security, compilers, …*

midgard EPFL

Imprecise Store Exceptions, ISCA'23

Single Address Space Faas with Jord, ISCA'25

- ***Topic: Agile Chip Development***
  - XiangShan is a progressive, configurable, complicated, challenging benchmark
  - MinJie provides a good startpoint
- *HDLs, verification, performance, power, area, prototyping, DFT, synthesis, placement, routing, ECO, …*



SNS v2, MICRO'23 (Duke University)

Outline

- The Era of Open-Source Chips
- Open-Source Industrial-Competitive RISC-V Chips
- Open-Source Chip Design Tools
- Open-Source Development & Business Models
- Conclusions

Open-Source Infrastructures (Tools) for Chips

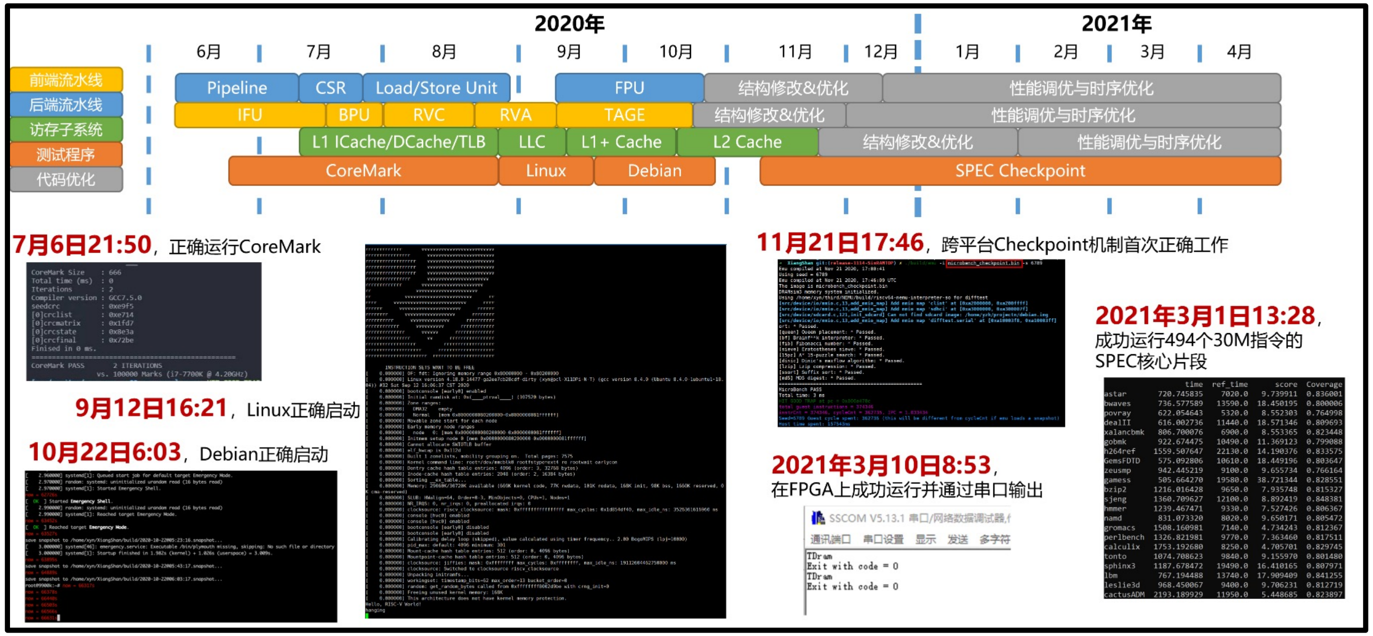
香山
芯片
敏捷开发
基础设施

XiangShan Chips

- Complex ISAs
- Microarchitecture
- PPA improvements
- ......
- Lower thresholds
- Agile development
- Parameterization
- ......

Chisel: agile hardware description language

- 2018: quantitatively comparing Chisel and Verilog, reducing code size by 80%
- 2020: completed the 1st generation of XiangShan, booting Linux within 3 months
- 2022: 67,000 lines of design code and 31,000 lines of verification code
- 2024: 214,000 lines of code in all code repositories



7月6日21:50，正确运行CoreMark

```
CoreMark Size    : 666
Total time (ms)  : 0
Iterations       : 2
Compiler version : GCC7.5.0
seedcrc          : 0xe9f5
[0]crclist       : 0xe714
[0]crcmatrix     : 0x1fd7
[0]crcstate      : 0x8e3a
[0]crcfinal      : 0x72be
Finised in 0 ms.
==================================================
CoreMark PASS       2 ITERATIONS
              vs. 100000 Marks (i7-7700K @ 4.20GHz)
```

9月12日16:21，Linux正确启动

10月22日6:03，Debian正确启动

11月21日17:46，跨平台Checkpoint机制首次正确工作

```
→ XiangShan git:(release-1114-SimRAMTOP) ✗ ./build/emu -i microbench_checkpoint.bin -s 6789
Emu compiled at Nov 21 2020, 17:08:41
Using seed = 6789
Emu compiled at Nov 21 2020, 17:46:09 UTC
The image is microbench_checkpoint.bin
DRAMsim3 memory system initialized.
Using /home/xyn/third/NEMU/build/riscv64-nemu-interpreter-so for difftest
[src/device/io/mmio.c,13,add_mmio_map] Add mmio map 'clint' at [0xa2000000, 0xa200ffff]
[src/device/io/mmio.c,13,add_mmio_map] Add mmio map 'sdhci' at [0xa3000000, 0xa300007f]
[src/device/sdcard.c,121,init_sdcard] Can not find sdcard image: /home/yzh/projectn/debian.img
[src/device/io/mmio.c,13,add_mmio_map] Add mmio map 'difftest.serial' at [0xa10003f8, 0xa10003ff]
ort: * Passed.
[queen] Queen placement: * Passed.
[bf] Brainf**k interpreter: * Passed.
[fib] Fibonacci number: * Passed.
[sieve] Eratosthenes sieve: * Passed.
[15pz] A* 15-puzzle search: * Passed.
[dinic] Dinic's maxflow algorithm: * Passed.
[lzip] Lzip compression: * Passed.
[ssort] Suffix sort: * Passed.
[md5] MD5 digest: * Passed.
==================================================
MicroBench PASS
Total time: 3 ms
HIT GOOD TRAP at pc = 0x800a478c
total guest instructions = 374346
instrCnt = 374346, cycleCnt = 362735, IPC = 1.033434
Seed=6789 Guest cycle spent: 362735 (this will be different from cycleCnt if emu loads a snapshot)
Host time spent: 157543ms
```

2021年3月1日13:28，
成功运行494个30M指令的
SPEC核心片段

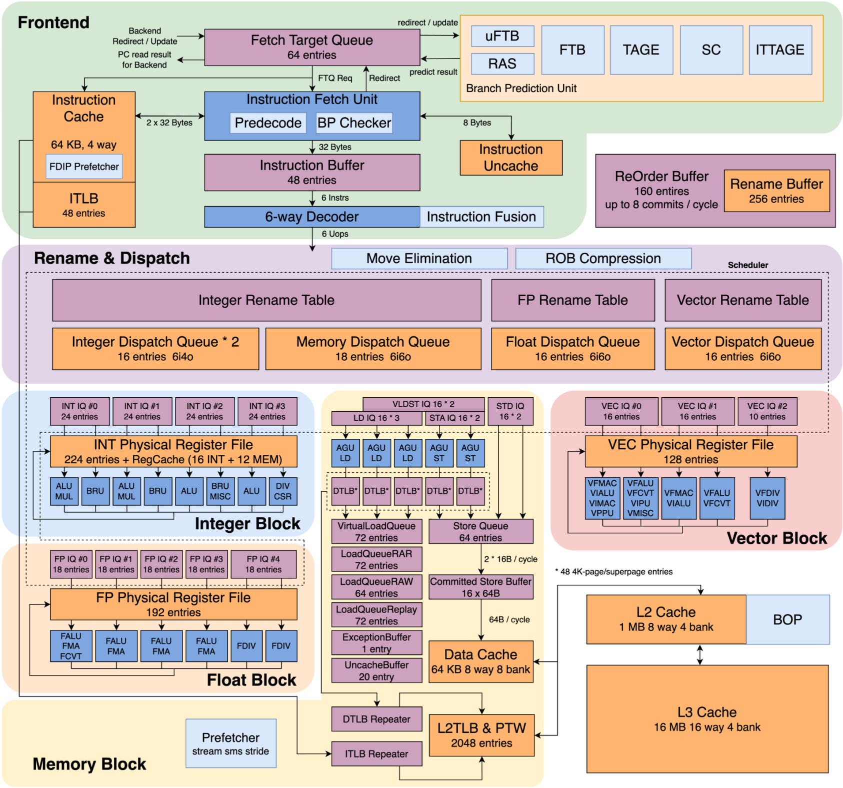
```
                 time  ref_time      score  Coverage
astar      720.745835    7020.0   9.739911  0.836001
bwaves     736.577589   13590.0  18.450195  0.800006
povray     622.054643    5320.0   8.552303  0.764998
dealII     616.002736   11440.0  18.571346  0.809693
xalancbmk  806.700076    6900.0   8.553365  0.823448
gobmk      922.674475   10490.0  11.369123  0.799088
h264ref   1559.507647   22130.0  14.190376  0.833575
GemsFDTD   575.092806   10610.0  18.449196  0.803647
zeusmp     942.445219    9100.0   9.655734  0.766164
gamess     505.664270   19580.0  38.721344  0.828551
bzip2     1216.016428    9650.0   7.935748  0.815327
sjeng     1360.709627   12100.0   8.892419  0.848381
hmmer     1239.467471    9330.0   7.527426  0.806367
namd       831.073320    8020.0   9.650171  0.805472
gromacs   1508.160981    7140.0   4.734243  0.812367
perlbench 1326.821981    9770.0   7.363460  0.817511
calculix  1753.192680    8250.0   4.705701  0.829745
tonto     1074.708623    9840.0   9.155970  0.801480
sphinx3   1187.678472   19490.0  16.410165  0.807971
lbm        767.194488   13740.0  17.909409  0.841255
leslie3d   968.450067    9400.0   9.706231  0.812719
cactusADM 2193.189929   11950.0   5.448685  0.823897
```

2021年3月10日8:53，
在FPGA上成功运行并通过串口输出

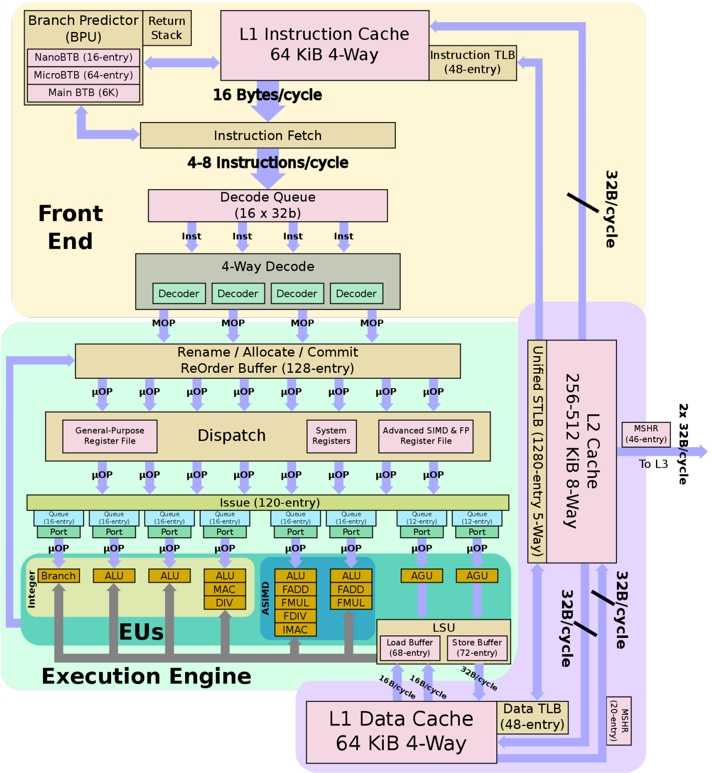
```
TDram
Exit with code = 0
TDram
Exit with code = 0
```

Better design flexibilities

Kunminghu has **212** configurable parameters, and its L2/L3 cache has **65** configurable parameters

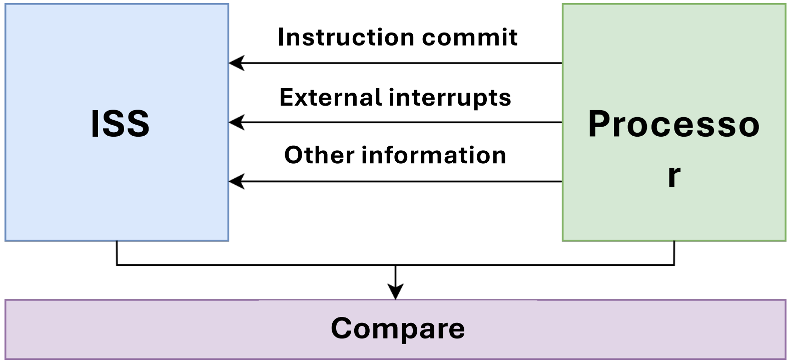


ARM A76 has **8** configurable parameters, and the DSU (L3) has **25** configurable parameters.



DiffTest: a co-simulation CPU verification framework

- **Co-simulation workflow**
  - Instructions commit/other states update
  - The simulator executes the same instructions
  - Compare the architectural states
  - Abort or continue



Basic architecture

- **Verification infrastructures for CPUs**
  - APIs for HDLs such as Chisel/Verilog
  - RTL simulators such as Verilator, VCS, Palladium
  - RISC-V ISS such as Spike, NEMU

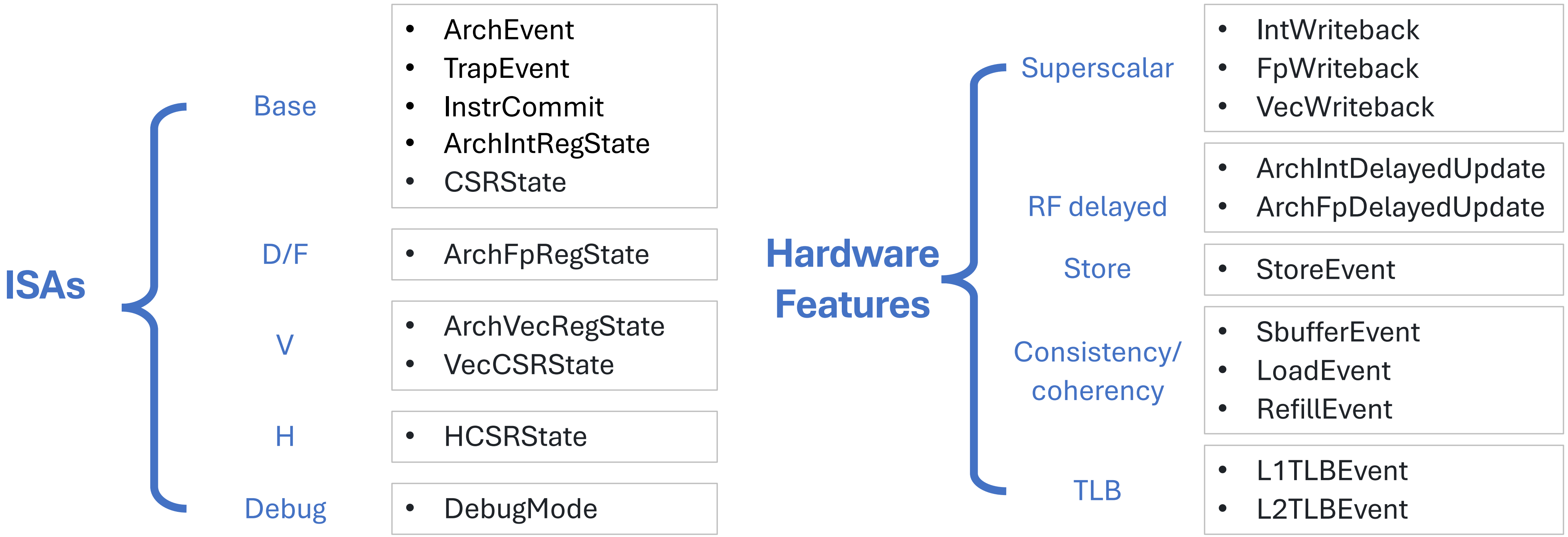
```
while (1) {
    icnt = cpu_step();
    ref_step(icnt);
    r1s = cpu_getregs();
    r2s = ref_getregs();
    if (r1s != r2s) { abort(); }
}
```

Online checking

https://github.com/OpenXiangShan/difftest

Standard interfaces for RISC-V CPU verification

- Key idea: information probes support flexible combination for different scenarios



ISAs

Base
- ArchEvent
- TrapEvent
- InstrCommit
- ArchIntRegState
- CSRState

D/F
- ArchFpRegState

V
- ArchVecRegState
- VecCSRState

H
- HCSRState

Debug
- DebugMode

Hardware Features

Superscalar
- IntWriteback
- FpWriteback
- VecWriteback

RF delayed
- ArchIntDelayedUpdate
- ArchFpDelayedUpdate

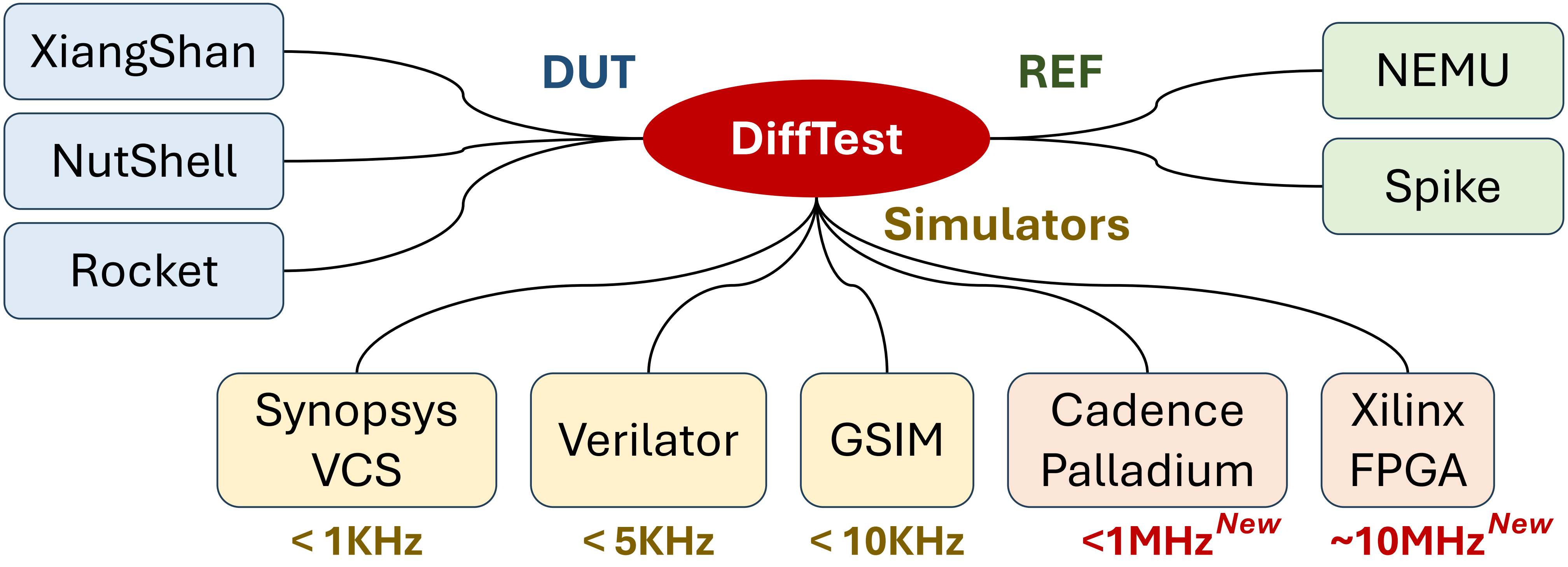
Store
- StoreEvent

Consistency/ coherency
- SbufferEvent
- LoadEvent
- RefillEvent

TLB
- L1TLBEvent
- L2TLBEvent

Accelerated co-simulation on Emulator/FPGA

- **DiffTest now supports hardware-accelerated co-simulation**



DiffTest-H: semantic-aware co-simulation acceleration

- Optimizes communication overhead for verification data packets
  - **Batch**: Reduces communication frequency
  - **Squash**: Reduces communication data volume
  - **Replay**: Maintains debugging granularity

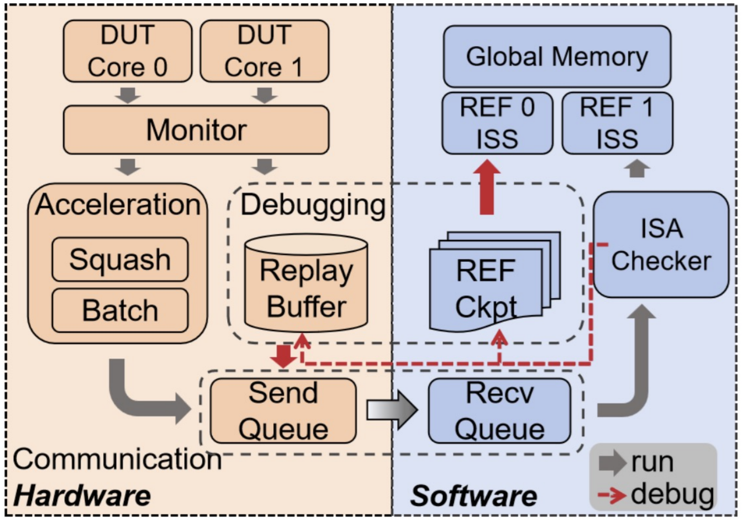
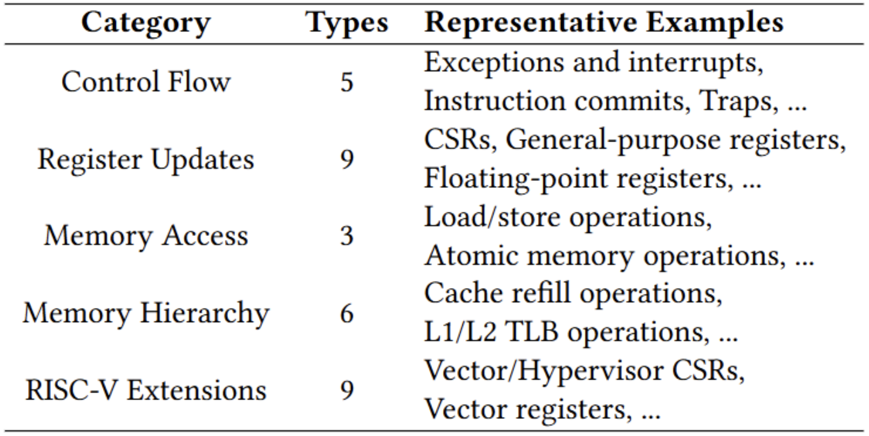


Figure: workflow



Category	Types	Representative Examples
Control Flow	5	Exceptions and interrupts, Instruction commits, Traps, ...
Register Updates	9	CSRs, General-purpose registers, Floating-point registers, ...
Memory Access	3	Load/store operations, Atomic memory operations, ...
Memory Hierarchy	6	Cache refill operations, L1/L2 TLB operations, ...
RISC-V Extensions	9	Vector/Hypervisor CSRs, Vector registers, ...

Table: Packets

DiffTest-H: Hardware-Accelerated Co-Simulation Verification

- **13.8 MHz** on FPGA, **instruction-level** debugging
- Deployed on **XiangShan**, with **151 bugs uncovered**
- **Open-Source** at *github/OpenXiangShan/difftest*

中国科学院计算技术研究所 INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES

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

香山开源处理器社区 XiangShan Open-Source Processor Community

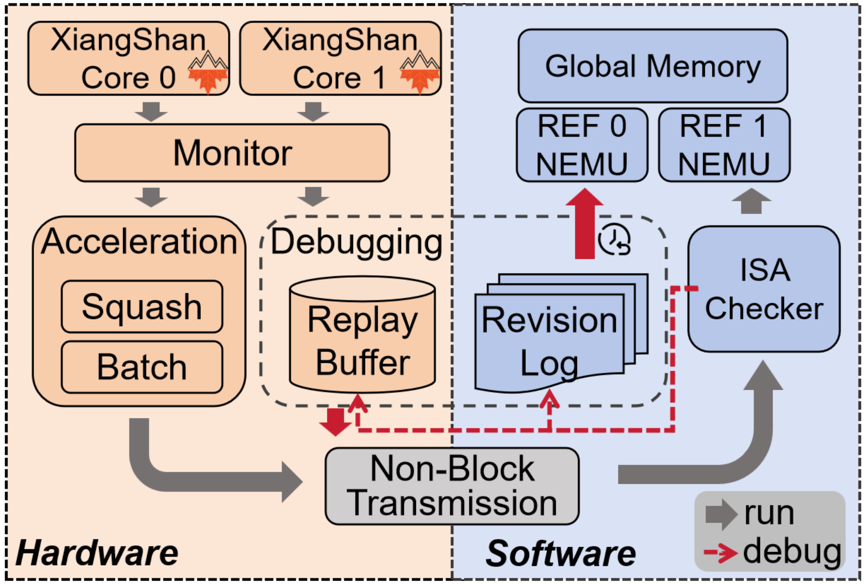
DiffTest-H: Toward Semantic-Aware Communication in Hardware-Accelerated Processor Verification

Kunlin You[1,2], Yinan Xu[1], Kehan Feng[3], Luoshan Cai[1,2], Yaoyang Zhou[3] and Yungang Bao[1,2]

[1] *State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences*

[2] *University of Chinese Academy of Sciences*

[3] *Beijing Institute of Open Source Chip*



Verilua: an easy-to-use unit-testing framework

New option for hardware verification

	Verilua	Cocotb	Fault	PyMTL/PyHGL/Chisel	UVM/SV
核心技术	VPIML	VPI	IR + 多后端	单一语言垂直整合	--
执行模型	**在线仿真+离线分析**	在线仿真	-- （元编程）	在线仿真	在线仿真
验证资产复用	**跨语言、跨场景**	**否**	跨语言	**否**	--
HDL	--	--	--	Y	Y
HVL	Y	Y	Y	Y	Y
HSE	**Y**	--	--	--	--
WAL	**Y**	--	--	--	--
学习门槛	**低** ✓	中	中	**高**	**高**

https://github.com/cyril0124/verilua

mux.sv

```
module mux(
    output wire out,
    input wire sel,
    input wire a,
    input wire b
);

assign out = sel ? b : a;

endmodule
```

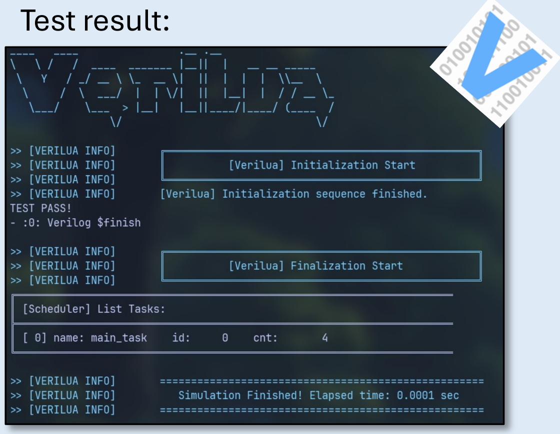
+

main.lua

```
fork {
    main_task = function()
        sim.bypass_initial()

        dut.a:set(0)
        dut.b:set(1)

        dut.sel:set(1)
        await_time(1)
        dut.out:expect(1)

        dut.sel:set(0)
        await_time(1)
        dut.out:expect(0)

        print("TEST PASS!")

        sim.finish()
    end
}
```

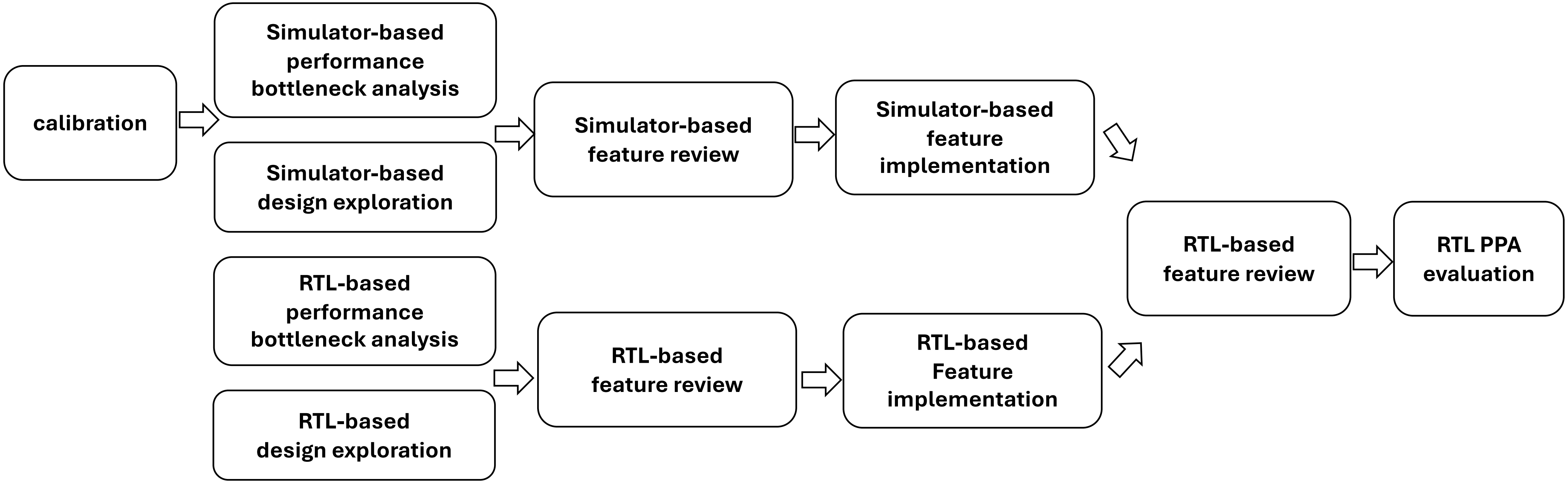
=

Test result:



```
>> [VERILUA INFO]
>> [VERILUA INFO]        [Verilua] Initialization Start
>> [VERILUA INFO]
>> [VERILUA INFO]       [Verilua] Initialization sequence finished.
TEST PASS!
- :0: Verilog $finish

>> [VERILUA INFO]
>> [VERILUA INFO]        [Verilua] Finalization Start
>> [VERILUA INFO]

 [Scheduler] List Tasks:

 [ 0] name: main_task    id:     0    cnt:       4


>> [VERILUA INFO]       ====================================================
>> [VERILUA INFO]          Simulation Finished! Elapsed time: 0.0001 sec
>> [VERILUA INFO]       ====================================================
```

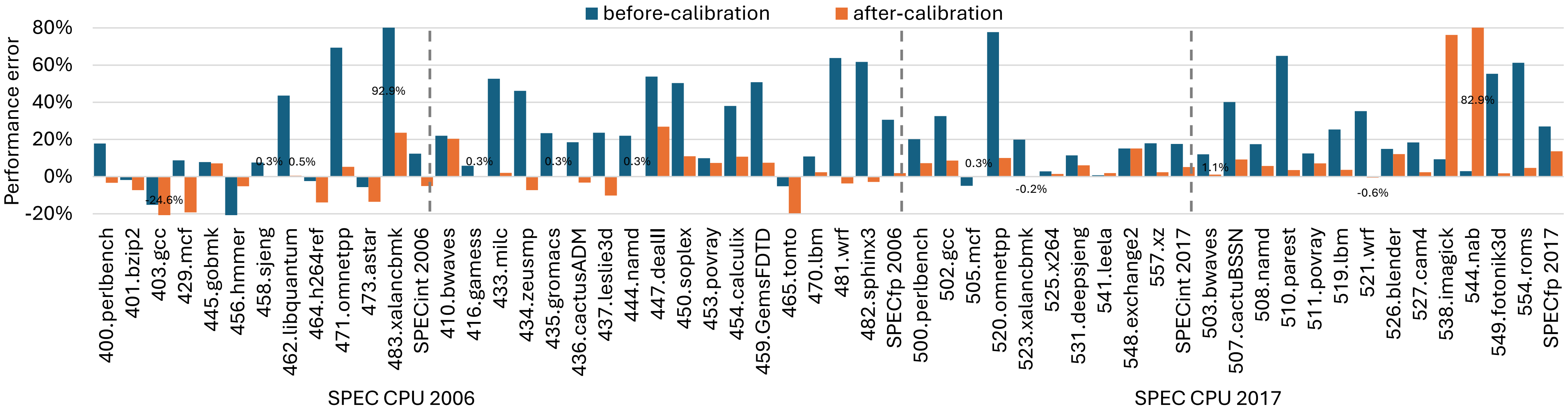
XS-GEM5: Calibrated Gem5 Simulator


calibration
Simulator-based performance bottleneck analysis
Simulator-based design exploration
Simulator-based feature review
Simulator-based feature implementation
RTL-based performance bottleneck analysis
RTL-based design exploration
RTL-based feature review
RTL-based Feature implementation
RTL-based feature review
RTL PPA evaluation

https://github.com/OpenXiangShan/GEM5

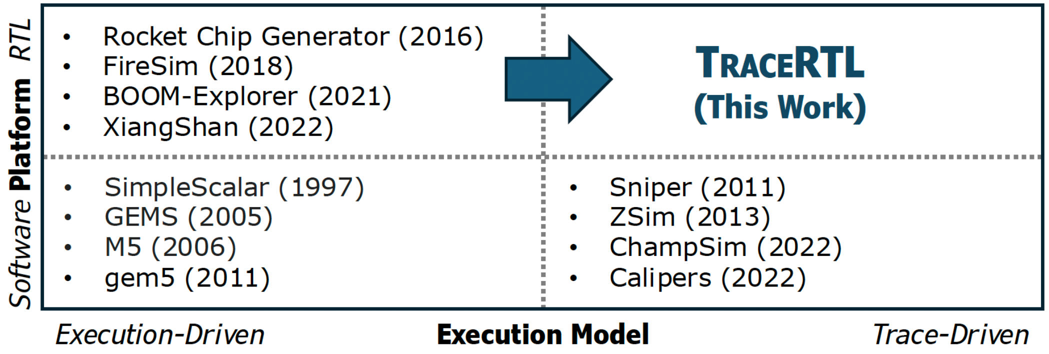
XS-GEM5: Calibrated Gem5 Simulator

- Based on GEM5, calibration was completed for XiangShan
- The overall error with RTL (KunminghuV2) in SEPC06 is less than 3%
  - The error of a single benchmark is less than 5%
  - Currently being used for architecture exploration of the KunminghuV3



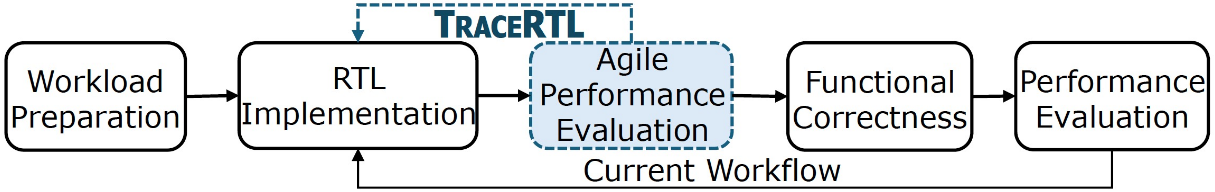
TraceRTL: Trace-driven RTL CPU Model

- Applying mature architecture exploration techniques *from simulators to RTL*
- **Insight**: Architecture exploration **only requires essential performance** components; complete functional correctness is *unnecessary*
  - Performance evaluation is performed before ensuring CPU functional correctness



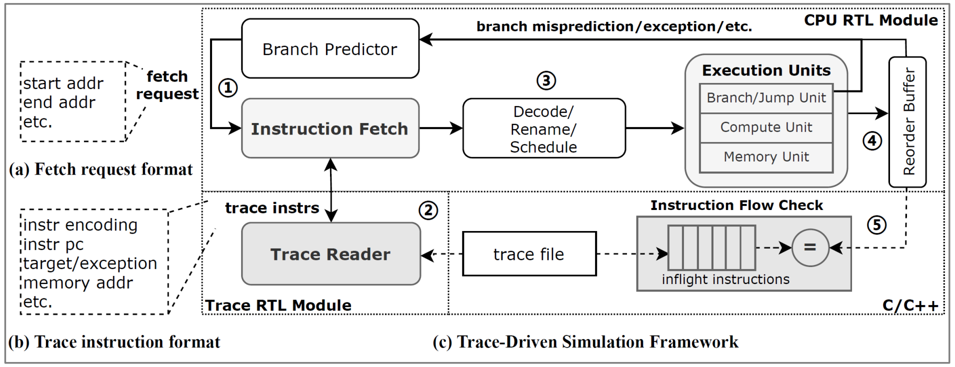
Platform	Execution-Driven	Trace-Driven
RTL	Rocket Chip Generator (2016) FireSim (2018) BOOM-Explorer (2021) XiangShan (2022)	**TRACERTL (This Work)**
Software	SimpleScalar (1997) GEMS (2005) M5 (2006) gem5 (2011)	Sniper (2011) ZSim (2013) ChampSim (2022) Calipers (2022)

Execution Model

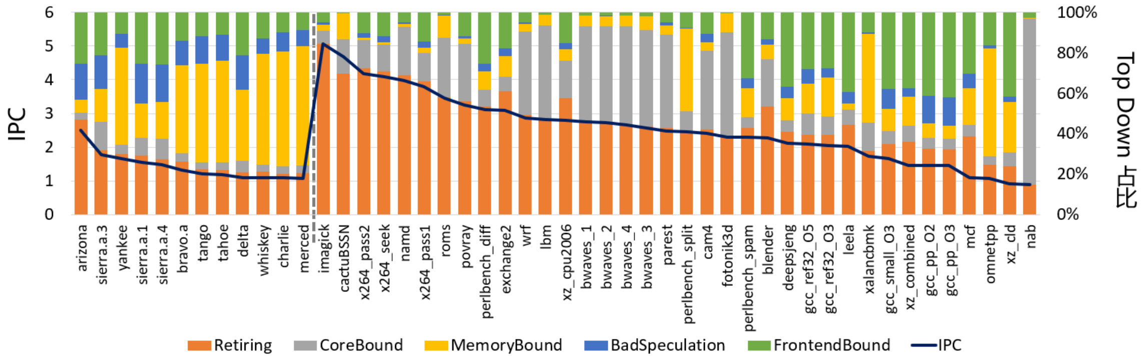


TraceRTL: Trace-driven RTL CPU Model

- Automated, low-intrusion modification of **a trace-driven XiangShan**
  - **Eliminating functional dependencies**: Chisel, circuitry, architecture, functional abstraction, performance sensitivity, etc.
  - **Eliminating performance errors**: The impact of lost information in the trace, including addresses, data, etc.



Trace-driven CPU from XiangShan



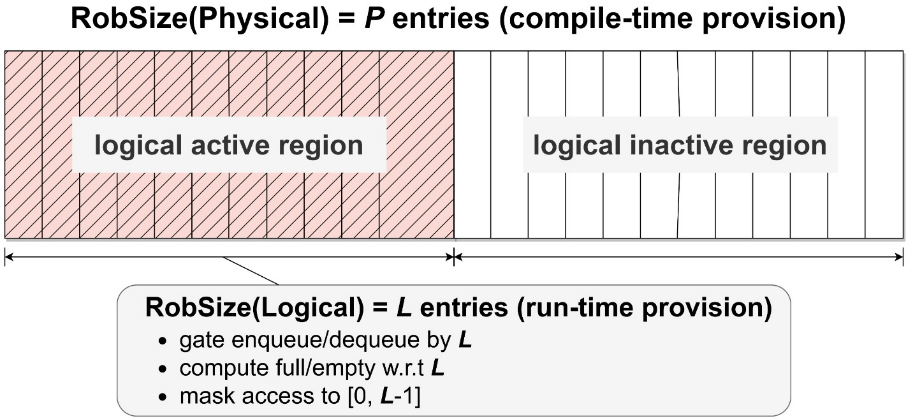
Top-Down differences between Google Datacenter Workloads and SPEC CPU

FPGA accelerated design space exploration

- **Design Space Exploration (DSE)**: Searching the CPU design parameter space to find the PPA balance point
- Due to several *limitations*, all current work is performed on a simulator.
  - **Speed Limitation**: Single simulation speed should be fast enough.
  - **Parameter Limitation**: Simulators offer flexible parameter tuning, while RTL is challenging

FastDSE: FPGA + logic/physical parameter decoupling

- FastDSE: FPGA Acceleration + Logic/Physical Parameter Decoupling
  - Stable operation at 50MHz
  - accelerating the DSE process by 68.7 times.

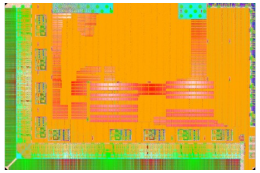

RobSize(Physical) = *P* entries (compile-time provision)
logical active region
logical inactive region
RobSize(Logical) = *L* entries (run-time provision)
• gate enqueue/dequeue by *L*
• compute full/empty w.r.t *L*
• mask access to [0, *L*-1]

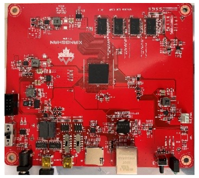
Logic/Physical Parameter Decoupling

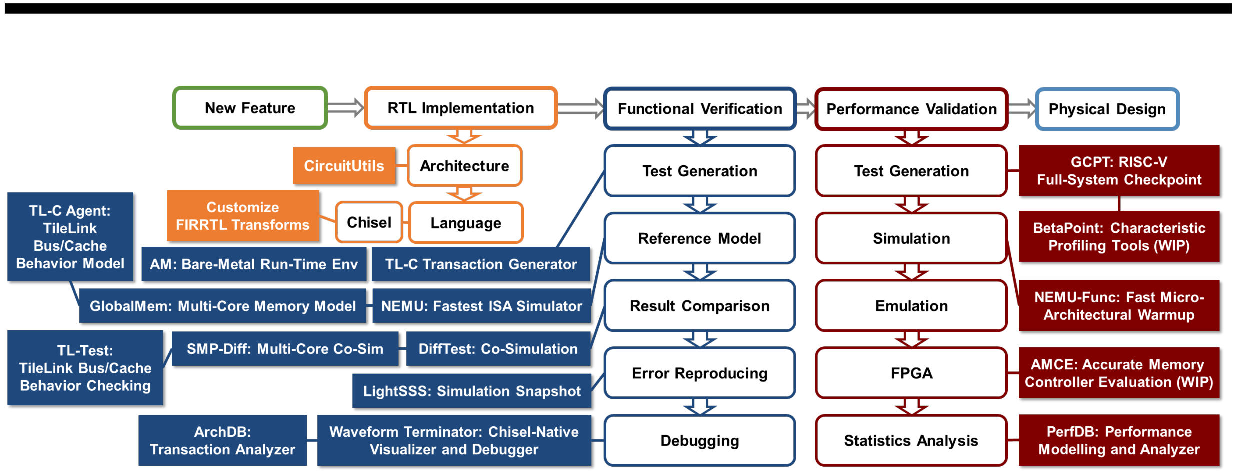
- One physical param
- Multiple logical parameters
- Adjust dynamically

MinJie: Open & Agile Development Toolchain

- It supports a new model of collaborative chip development based on open source, continuously building a team of over 600 people (the largest in the world).
  - Selected as one of the 12 IEEE MICRO Top Picks







New Feature → RTL Implementation → Functional Verification → Performance Validation → Physical Design

RTL Implementation: Architecture, Language

CircuitUtils

Customize FIRRTL Transforms

Chisel

TL-C Agent: TileLink Bus/Cache Behavior Model

AM: Bare-Metal Run-Time Env

TL-C Transaction Generator

GlobalMem: Multi-Core Memory Model

NEMU: Fastest ISA Simulator

TL-Test: TileLink Bus/Cache Behavior Checking

SMP-Diff: Multi-Core Co-Sim

DiffTest: Co-Simulation

LightSSS: Simulation Snapshot

ArchDB: Transaction Analyzer

Waveform Terminator: Chisel-Native Visualizer and Debugger

Functional Verification: Test Generation, Reference Model, Result Comparison, Error Reproducing, Debugging

Performance Validation: Test Generation, Simulation, Emulation, FPGA, Statistics Analysis

GCPT: RISC-V Full-System Checkpoint

BetaPoint: Characteristic Profiling Tools (WIP)

NEMU-Func: Fast Micro-Architectural Warmup

AMCE: Accurate Memory Controller Evaluation (WIP)

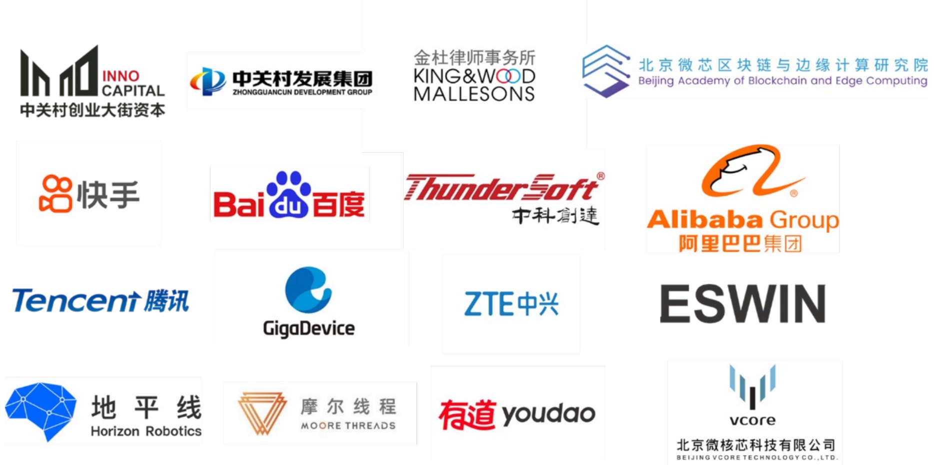
PerfDB: Performance Modelling and Analyzer

Outline

- The Era of Open-Source Chips
- Open-Source Industrial-Competitive RISC-V Chips
- Open-Source Chip Design Tools
- Open-Source Development & Business Models
- Conclusions

Development Model for Open-Source Chips

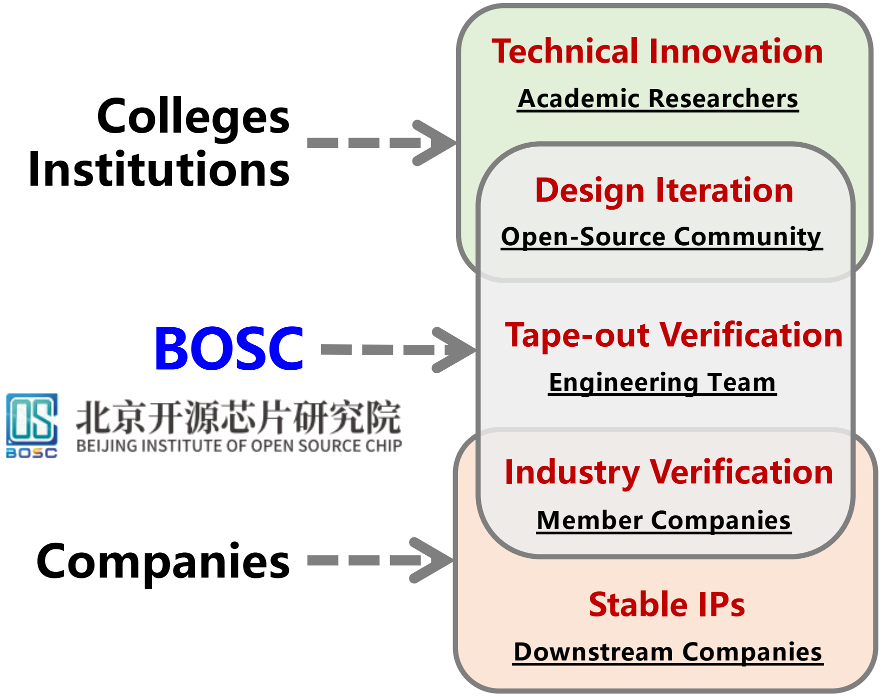
- With the support of the Beijing Municipal Government and the Chinese Academy of Sciences, **16 companies** jointly launched the **Beijing Institute of Open Source Chip (BOSC)** to accelerate the industrial development of Xiangshan.
- BOSC has assembled a team of **>500 people**, becoming one of the largest RISC-V CPU core R&D teams in the world.



16 Founding Members

Dec. 6, 2021

Tiered Rolling Open-Source Model



Colleges Institutions

BOSC

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

Companies

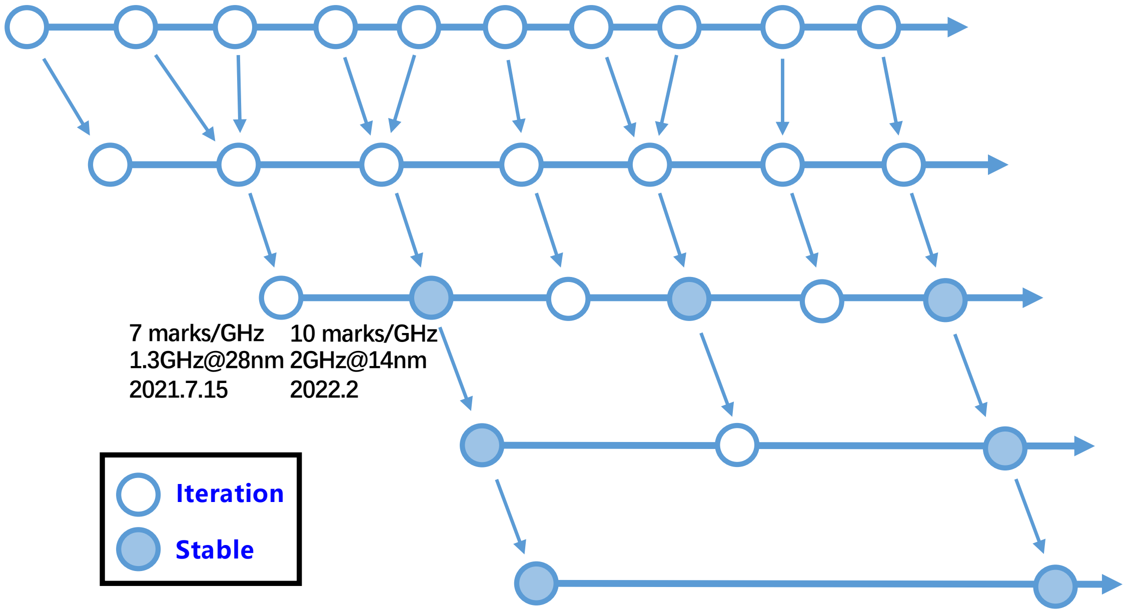
Technical Innovation
Academic Researchers

Design Iteration
Open-Source Community

Tape-out Verification
Engineering Team

Industry Verification
Member Companies

Stable IPs
Downstream Companies



7 marks/GHz
1.3GHz@28nm
2021.7.15

10 marks/GHz
2GHz@14nm
2022.2

○ Iteration
● Stable

Chip products based on XiangShan

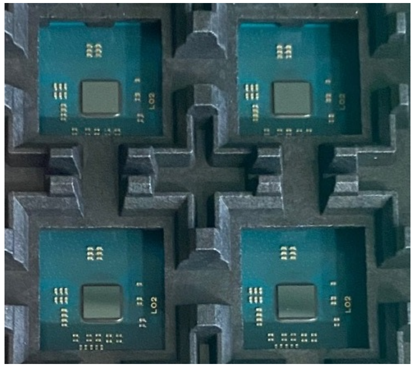
Kunminghu V2

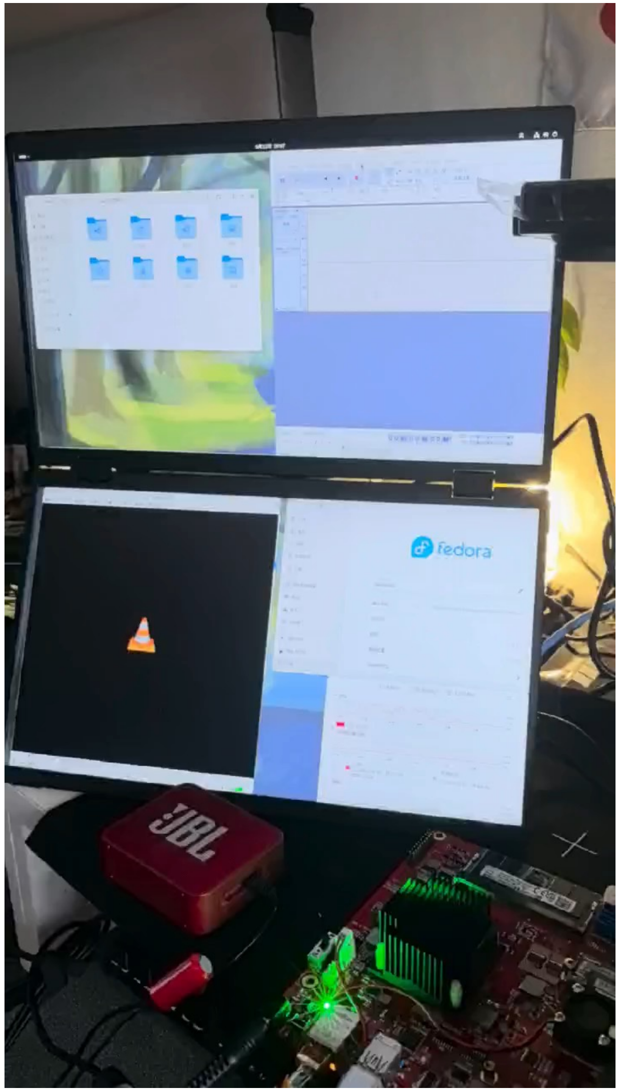
User	Product	Tapeout Time
A	**8-core video codec**	**2025/9**
B	**64-core server**	**2025/9**
C	**128-core server**	**2025/10月**
D	128-core server	2026/3
B	8-core client SoC	2026/3
E	16-core	2026/3
F	64-core server	2026/3
G	4-core client SoC	2026/6
H	128-core server	2026/12

Nanhu

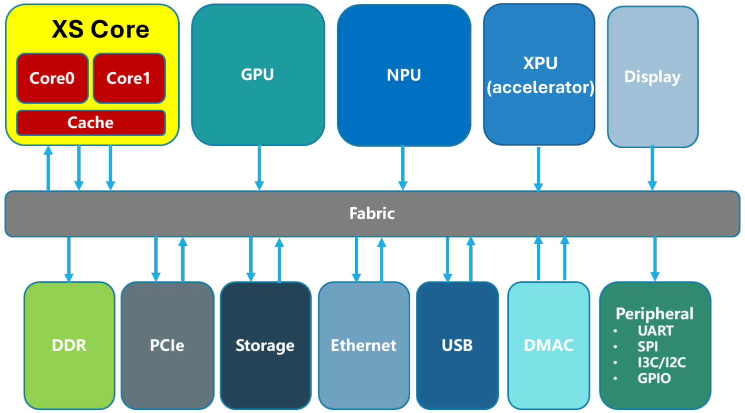
User	Product	Tapeout Time
I	**GPGPU**	**2024/production**
J	**GPGPU**	**2024/production(>10k)**
K	4-core FPGA ctrl.	2025/12
L	4-core sec. ctrl.	2025/12
M	4-core	2025/12
N	4-core router ctrl.	2026/3

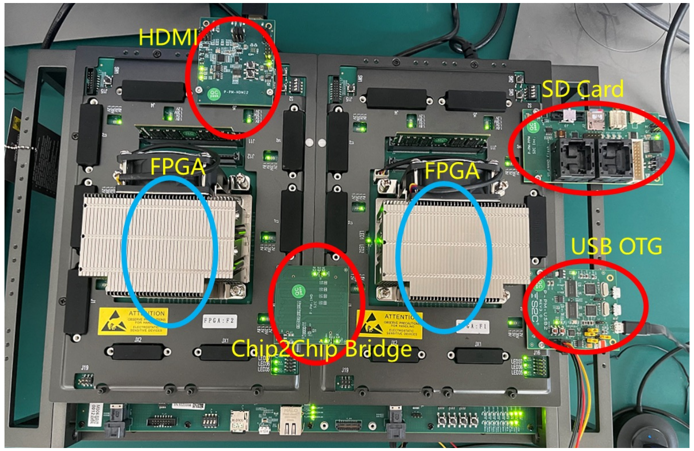
XiangShan Nanhu Chips

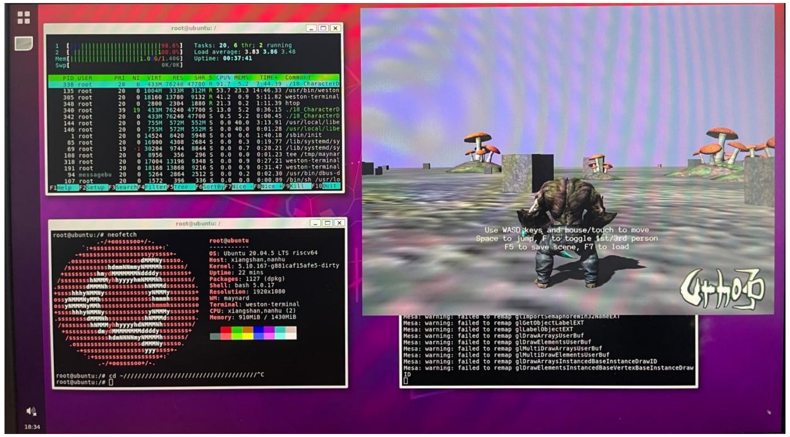




FPGA Prototype in Only Two Weeks





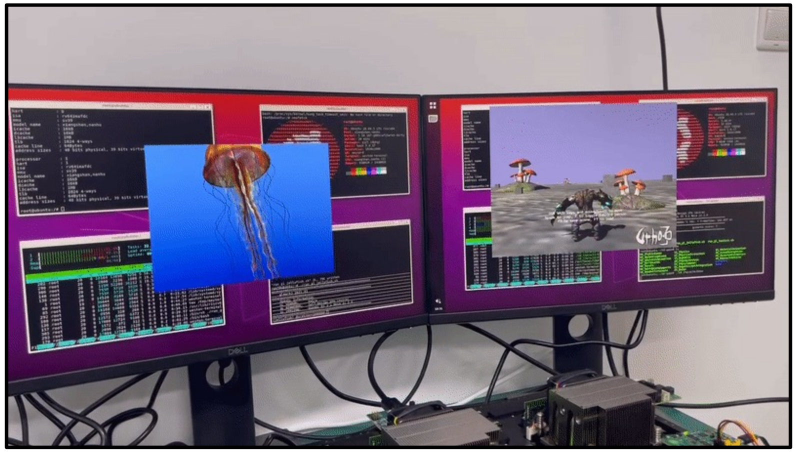


```
root@ubuntu: /
  1  [||||||||||||||||||||||||98.6%]   Tasks: 20, 6 thr; 2 running
  2  [||||||||||||||||||||||100.0%]   Load average: 3.83 3.86 3.48
  Mem[||||||||||||||||||1.00G/1.40G]   Uptime: 00:37:41
  Swp[                        0K/0K]

  PID USER      PRI  NI  VIRT   RES   SHR S CPU% MEM%   TIME+  Command
  338 root       20   0  433M 76240 47700 R 91.7  5.2  7:44.39 ./18_CharacterD
  135 root       20   0 1004M  333M  312M R 53.7 23.3 14:46.33 /usr/bin/weston
  305 root       20   0 18160 13780  9132 R 41.2  0.9  5:11.82 weston-terminal
  348 root       20   0  2800  2304  1880 R 21.3  0.2  1:11.39 htop
  340 root       39  19  433M 76240 47700 S 13.0  5.2  0:36.15 ./18_CharacterD
  342 root       20   0  433M 76240 47700 S  0.5  5.2  0:00.45 ./18_CharacterD
  144 root       20   0  755M  572M  552M S  0.0 40.0  3:13.91 /usr/local/libe
  146 root       20   0  755M  572M  552M S  0.0 40.0  0:01.28 /usr/local/libe
    1 root       20   0 14524  8420  5948 S  0.0  0.6  1:40.18 /sbin/init
   85 root       20   0 16900  4308  2684 S  0.0  0.3  0:19.77 /lib/systemd/sy
   69 root       19  -1 30204  9744  8844 S  0.0  0.7  0:28.21 /lib/systemd/sy
  108 root       20   0  8956   356   296 S  0.0  0.0  0:01.23 tee /tmp/maynar
  318 root       20   0 17004 13196  9348 S  0.0  0.9  0:27.21 weston-terminal
  191 root       20   0 18168 13868  9216 S  0.0  0.9  0:31.47 weston-terminal
   94 messagebu  20   0  5264  2864  2512 S  0.0  0.2  0:02.30 /usr/bin/dbus-d
  107 root       20   0  1572   396   336 S  0.0  0.0  0:00.09 /bin/sh /usr/lo
F1Help  F2Setup F3SearchF4FilterF5Tree  F6SortByF7Nice -F8Nice +F9Kill  F10Quit
```

```
root@ubuntu: /
root@ubuntu:/# neofetch
                                        root@ubuntu
                                        -----------
                                        OS: Ubuntu 20.04.5 LTS riscv64
                                        Host: xiangshan,nanhu
                                        Kernel: 5.10.167-g881caf15afe5-dirty
                                        Uptime: 22 mins
                                        Packages: 1127 (dpkg)
                                        Shell: bash 5.0.17
                                        Resolution: 1920x1080
                                        WM: maynard
                                        Terminal: weston-terminal
                                        CPU: xiangshan,nanhu (2)
                                        Memory: 910MiB / 1430MiB

root@ubuntu:/# cd ~///////////////////////////////////^C
root@ubuntu:/# 
```

```
Use WASD keys and mouse/touch to move
Space to jump, F to toggle 1st/3rd person
F5 to save scene, F7 to load
```

```
Mesa: warning: failed to remap glImportSemaphoreWin32NameEXT
Mesa: warning: failed to remap glGetObjectLabelEXT
Mesa: warning: failed to remap glLabelObjectEXT
Mesa: warning: failed to remap glDrawArraysUserBuf
Mesa: warning: failed to remap glDrawElementsUserBuf
Mesa: warning: failed to remap glMultiDrawArraysUserBuf
Mesa: warning: failed to remap glMultiDrawElementsUserBuf
Mesa: warning: failed to remap glDrawArraysInstancedBaseInstanceDrawID
Mesa: warning: failed to remap glDrawElementsInstancedBaseVertexBaseInstanceDraw
ID
```



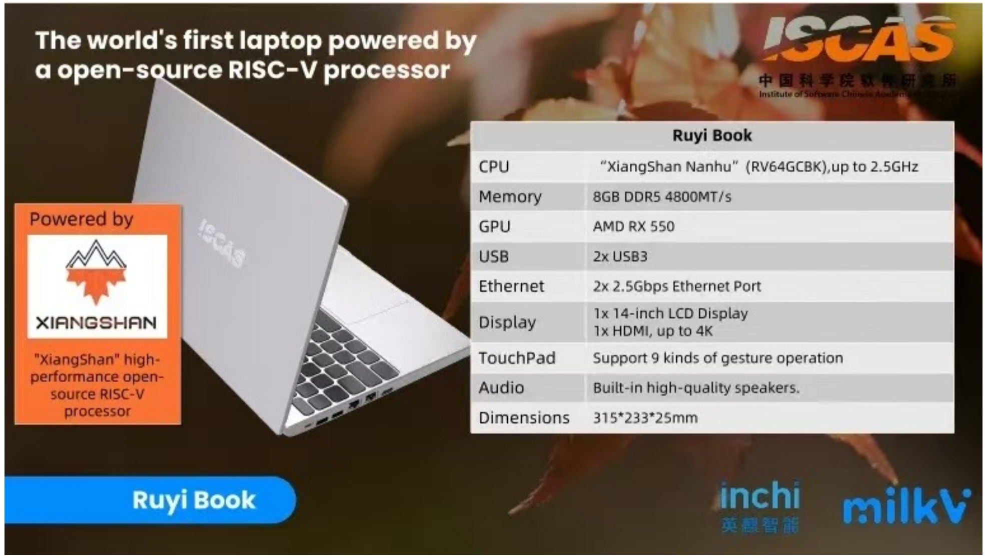
Source: Xinchen Technology

Real Chip of XiangShan Nanhu

- **Test chip was back in October 2023**
  - Successfully brought up Linux and working with PCIe device (GPU, Ethernet, USB..)

Nanhu test board @ RVSC'24

Running Desktop Linux & Cloud Game*


The world's first laptop powered by a open-source RISC-V processor
ISCAS
中国科学院软件研究所
Institute of Software Chinese Academy of Sciences
Powered by
XIANGSHAN
"XiangShan" high-performance open-source RISC-V processor
Ruyi Book
inchi
英麒智能
milkV

Ruyi Book	
CPU	"XiangShan Nanhu" (RV64GCBK),up to 2.5GHz
Memory	8GB DDR5 4800MT/s
GPU	AMD RX 550
USB	2x USB3
Ethernet	2x 2.5Gbps Ethernet Port
Display	1x 14-inch LCD Display 1x HDMI, up to 4K
TouchPad	Support 9 kinds of gesture operation
Audio	Built-in high-quality speakers.
Dimensions	315*233*25mm

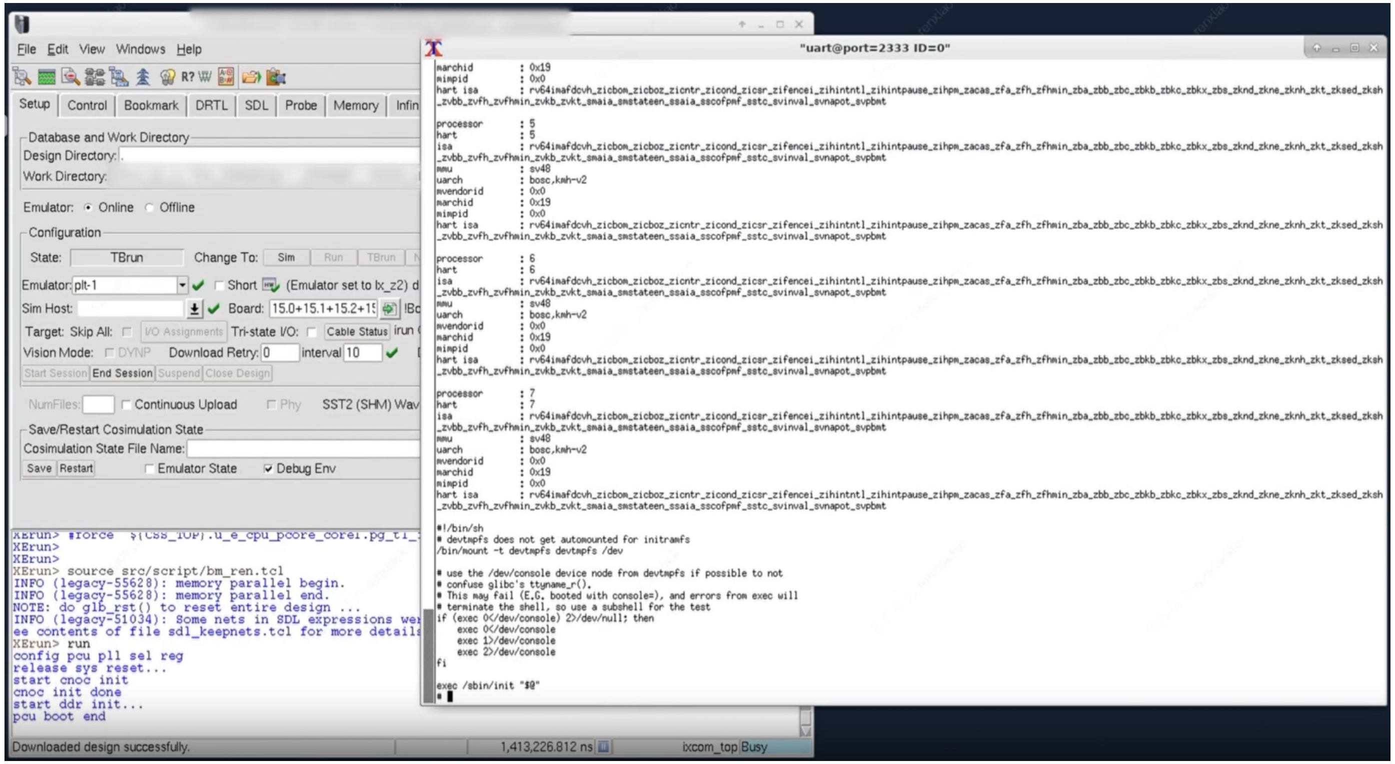
Ruyi XiangShan Book

Design by ISCAS, Inchi, MilkV

* Genshin Impact · Cloud, only ~ 1 fps, just for fun

Linux Boot on 8-Core XiangShan Kunminghu V2

- Lanxin Computing has successfully launched Linux on an 8-core SoC built on the Kunminghu V2



Database and Work Directory
Design Directory: .
Work Directory:

Emulator: ● Online ○ Offline

Configuration
State: TBrun
Emulator: plt-1 ✓ □ Short (Emulator set to lx_z2) d
Sim Host: ✓ Board: 15.0+15.1+15.2+1 !Bc
Target: Skip All: □ Tri-state I/O: □ irun
Vision Mode: □ DYNP Download Retry: 0 interval 10 ✓

NumFiles: □ Continuous Upload □ Phy SST2 (SHM) Wav

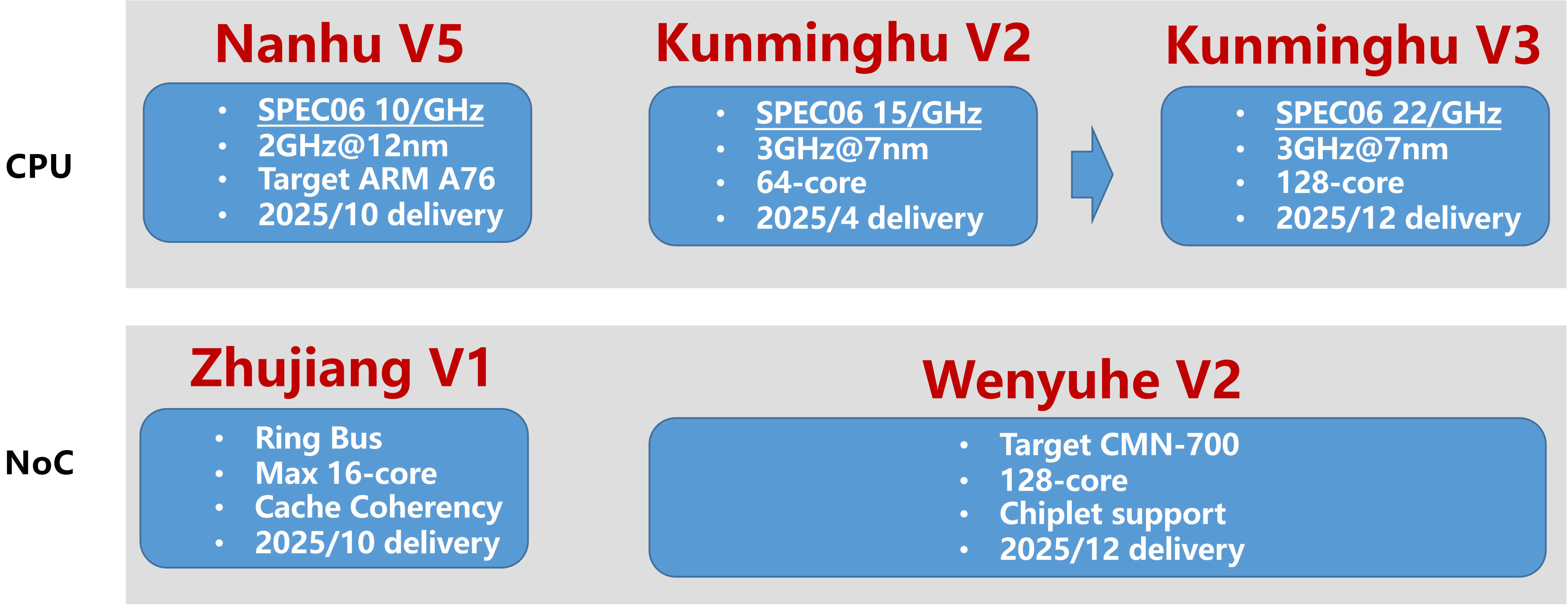
Save/Restart Cosimulation State
Cosimulation State File Name:
□ Emulator State ☑ Debug Env

```
XErun> #force  ${CSS_TOP}.u_e_cpu_pcore_core1.pg_t1_
XErun>
XErun>
XErun> source src/script/bm_ren.tcl
INFO (legacy-55628): memory parallel begin.
INFO (legacy-55628): memory parallel end.
NOTE: do glb_rst() to reset entire design ...
INFO (legacy-51034): Some nets in SDL expressions wer
ee contents of file sdl_keepnets.tcl for more details
XErun> run
config pcu pll sel reg
release sys reset...
start cnoc init
cnoc init done
start ddr init...
pcu boot end
```

Downloaded design successfully.
1,413,226.812 ns
ixcom_top Busy

```
marchid         : 0x19
mimpid          : 0x0
hart isa        : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt

processor       : 5
hart            : 5
isa             : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt
mmu             : sv48
uarch           : bosc,kmh-v2
mvendorid       : 0x0
marchid         : 0x19
mimpid          : 0x0
hart isa        : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt

processor       : 6
hart            : 6
isa             : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt
mmu             : sv48
uarch           : bosc,kmh-v2
mvendorid       : 0x0
marchid         : 0x19
mimpid          : 0x0
hart isa        : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt

processor       : 7
hart            : 7
isa             : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt
mmu             : sv48
uarch           : bosc,kmh-v2
mvendorid       : 0x0
marchid         : 0x19
mimpid          : 0x0
hart isa        : rv64imafdcvh_zicbom_zicboz_zicntr_zicond_zicsr_zifencei_zihintntl_zihintpause_zihpm_zacas_zfa_zfh_zfhmin_zba_zbb_zbc_zbkb_zbkc_zbkx_zbs_zknd_zkne_zknh_zkt_zksed_zksh
_zvbb_zvfh_zvfhmin_zvkb_zvkt_smaia_smstateen_ssaia_sscofpmf_sstc_svinval_svnapot_svpbmt

#!/bin/sh
# devtmpfs does not get automounted for initramfs
/bin/mount -t devtmpfs devtmpfs /dev

# use the /dev/console device node from devtmpfs if possible to not
# confuse glibc's ttyname_r().
# This may fail (E.G. booted with console=), and errors from exec will
# terminate the shell, so use a subshell for the test
if (exec 0</dev/console) 2>/dev/null; then
    exec 0</dev/console
    exec 1>/dev/console
    exec 2>/dev/console
fi

exec /sbin/init "$@"
# ▮
```

Roadmap 2025: 3 CPU Compute Systems



CPU

Nanhu V5

- SPEC06 10/GHz
- 2GHz@12nm
- Target ARM A76
- 2025/10 delivery

Kunminghu V2

- SPEC06 15/GHz
- 3GHz@7nm
- 64-core
- 2025/4 delivery

Kunminghu V3

- SPEC06 22/GHz
- 3GHz@7nm
- 128-core
- 2025/12 delivery

NoC

Zhujiang V1

- Ring Bus
- Max 16-core
- Cache Coherency
- 2025/10 delivery

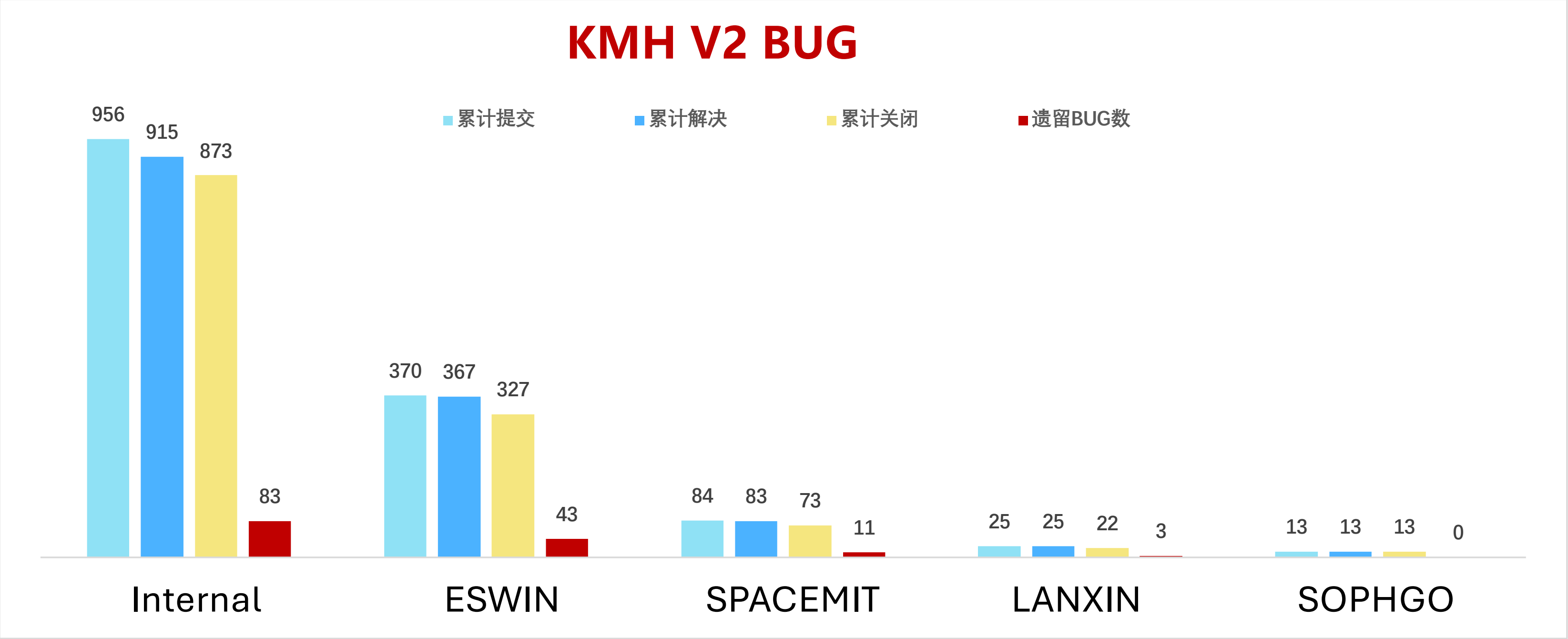
Wenyuhe V2

- Target CMN-700
- 128-core
- Chiplet support
- 2025/12 delivery

84

Advantages of open-source collaborative development

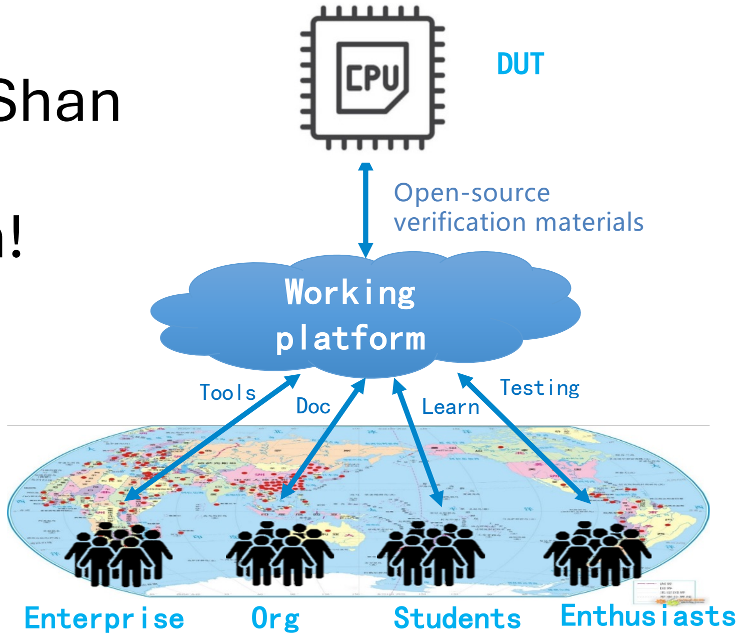
- Traditionally, **test cases are *valuable private assets*** of chip companies.
- With open source, it can be directly deployed within different companies.
  - A total of 1,467 bugs have been fixed, of which **492 (33.4%) were submitted by companies.**



85

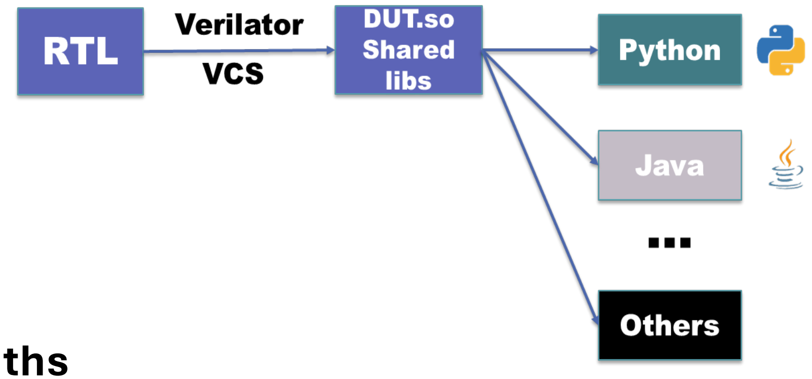
UnityChip Verification: Crowd-Sourcing Verification

- Chip verification using an open-source crowdsourcing model
- Bringing together software and hardware engineers
- A verification campaign launched on XiangShan
- Let 10,000 people participate in verification!
- https://open-verify.cc/en/



UnityChip: A Croudsourcing Platform for Chip Verification

- **Features: Involve software engineers**
  - Enable crowdsourcing
  - Support multi-languages
  - Be compatible with UVM
- **Effectiveness**
  - 5 **undergraduate** students found **10 bugs** **in 2 months**
- **Easy-to-Use**
  - Students not familiar with Linux and Python
  - Learn to use tools in **5 days**
  - Start adding test cases after another **10 days**
- **UnityChip Competition for XiangShan**
  - Task: Unit Test for IFU, BPU, ITLB, etc.
  - Scan the QR code for more information



GitHub Link

Open Problems on Chip Development Infrastructures

- Sharing ***the real-world infrastructure challenges*** faced by the XiangShan project

Problems

- **Hardware Descriptions**
- **Functional Verification**
- **Performance Improvements**

Directions

- **Architecture**
- **Software Engineering**
- **EDA**
- **PL/Sys/…**

Participation

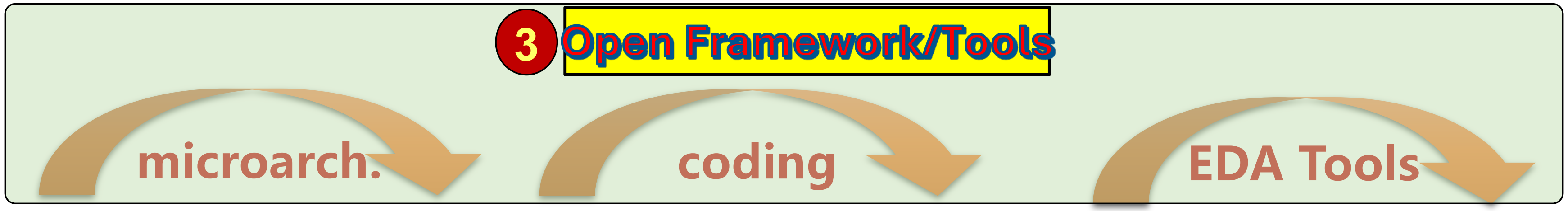
- **Discussions**
- **Interns**
- **Full-time**
- **Research/Industrial Collaborations**

Outline

- The Era of Open-Source Chips
- Open-Source Industrial-Competitive RISC-V Chips
- Open-Source Chip Design Tools
- Open-Source Development & Business Models
- Conclusions

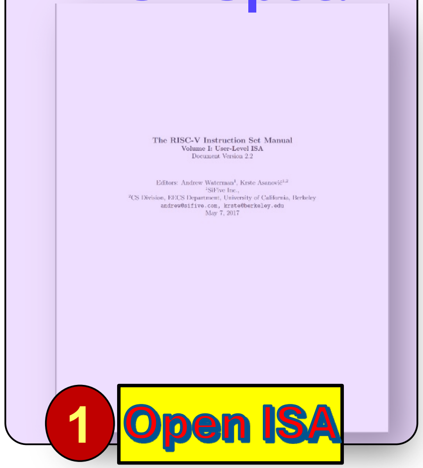
XiangShan achieves L2.5

- **L1: OPEN** ISA
- **L2: OPEN** Design/Implementation
- **L3: OPEN** Framework/Tools



3 Open Framework/Tools

ISA Spec.

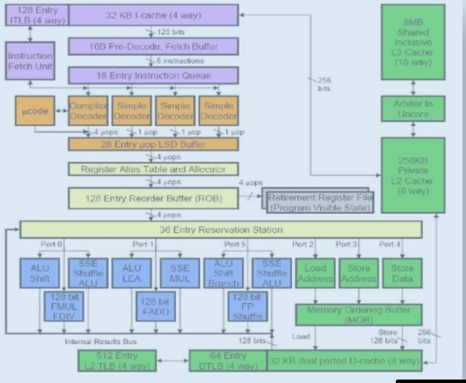


The RISC-V Instruction Set Manual
Volume I: User-Level ISA
Document Version 2.2

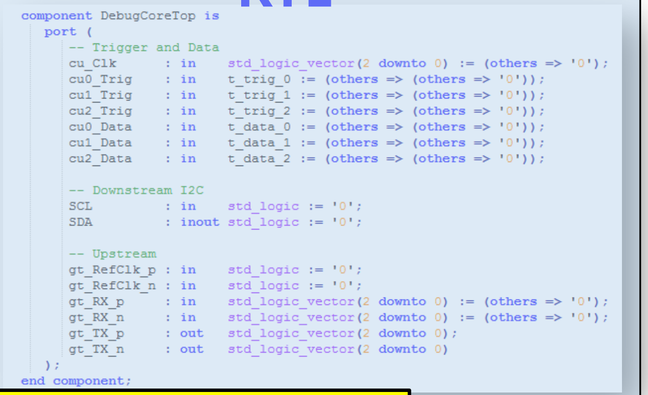
Editors: Andrew Waterman[1], Krste Asanović[1,2]
[1]SiFive Inc.,
[2]CS Division, EECS Department, University of California, Berkeley
andrew@sifive.com, krste@berkeley.edu
May 7, 2017

1 Open ISA

Docs



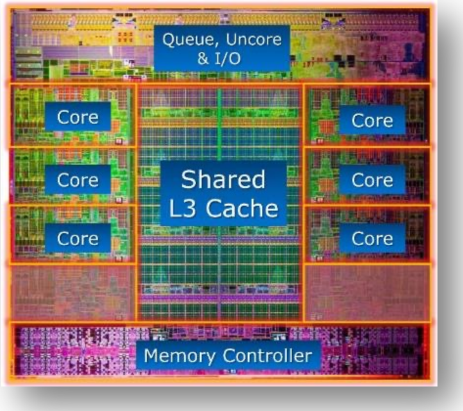
RTL



```
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;
```

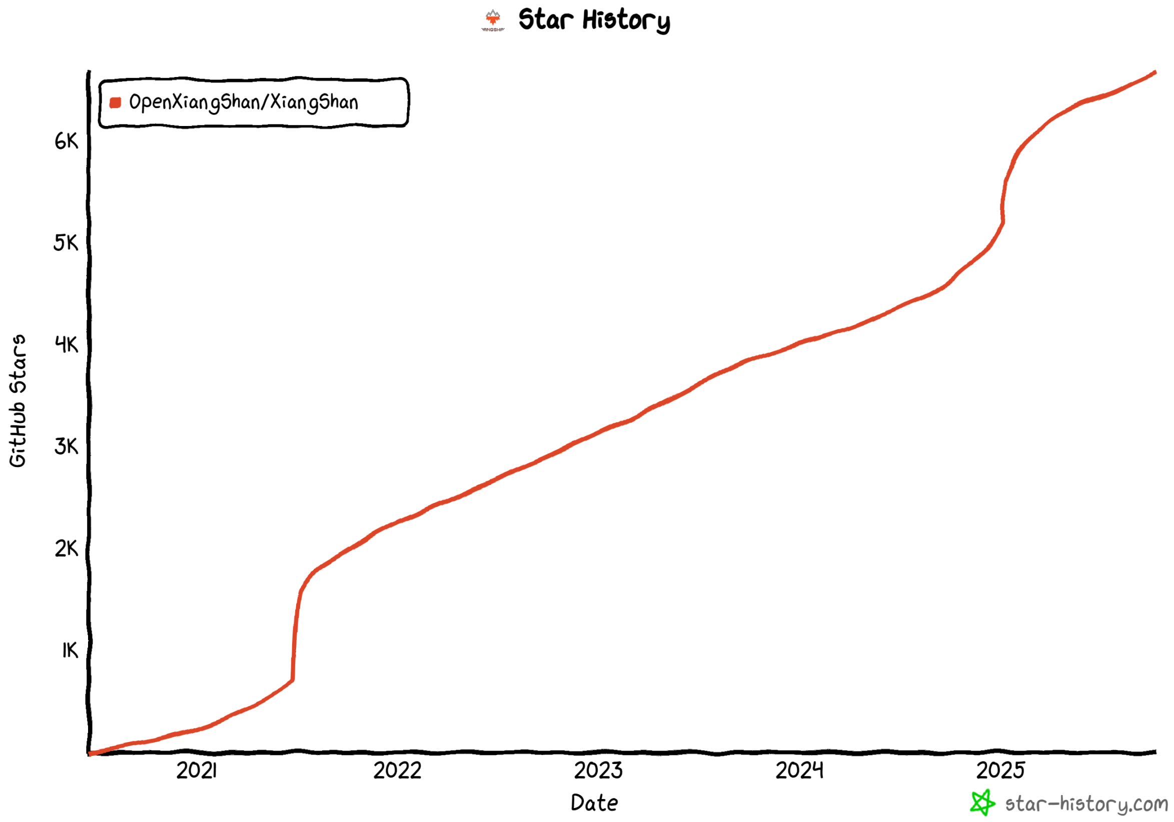
2 Open Design/Implt

Layout



One of the Most Popular Open-Source Chip Projects

- **GitHub: >6700** Stars, > 800 Forks



Data collected on Oct 10, 2025

15+ XiangShan Tutorials Around the World

2023

- ASPLOS, Vancouver, Canada
- RVSC, Beijing, China
- MICRO, Toronto, Canada

2024

- HPCA, Edinburgh, Scotland
- ASPLOS, San Diego, USA
- RVSC, Hangzhou, China
- MICRO, Austin, USA

2025

- HPCA, Las Vegas, USA
- ASPLOS, Rotterdam, Netherlands
- RVSE, Paris, France
- ISCA, Tokyo, Japan
- APPT, Athens, Greece
- RVSC, Shanghai, China
- ***MICRO, Seoul, Korea***

2026

- HPCA, Sydney, Australia
- More coming …

Welcome old and new friends!

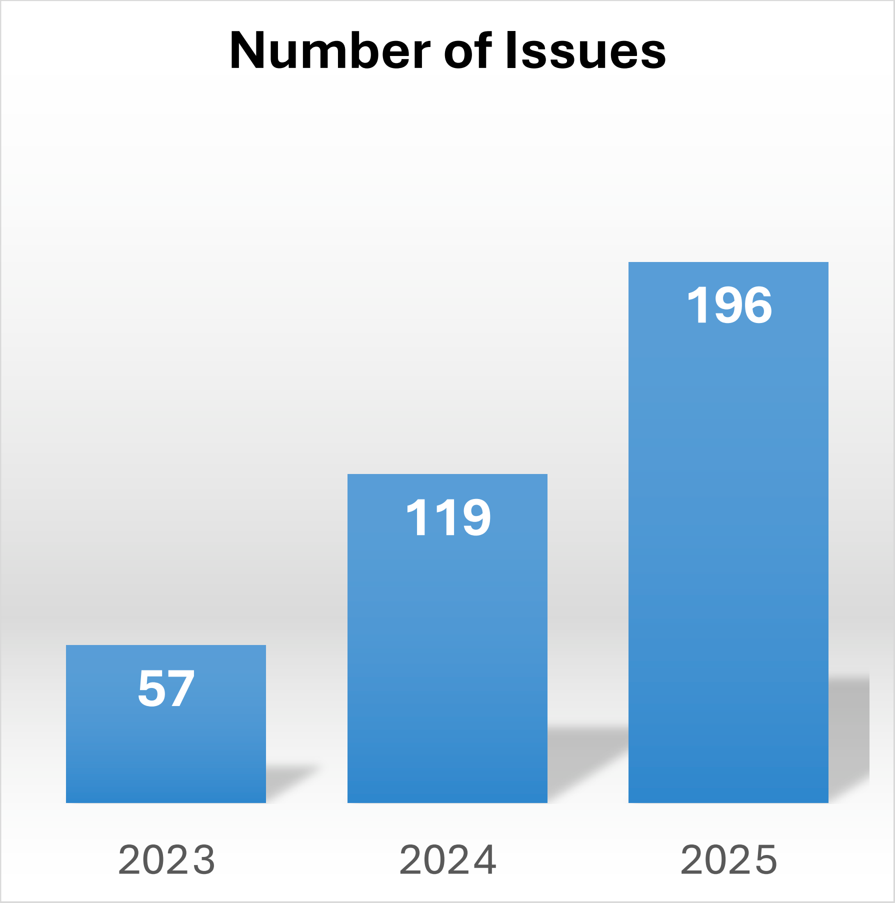
XiangShan Open-Source Community Conference

The 4th RISC-V Summit China 2024
Hangzhou, China

The 5th RISC-V Summit China 2025
Shanghai, China

93

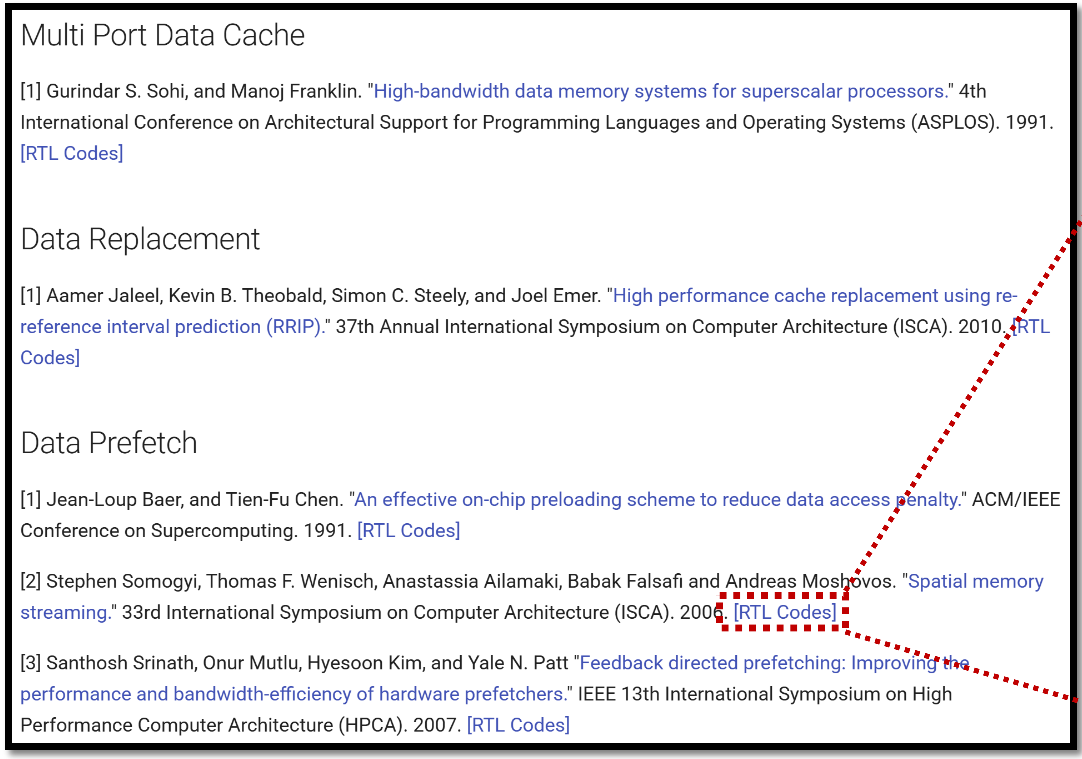
GitHub Issues & Discussions



- Time to First Response for Issues
  - Average: 25 hours
  - Median: 13 hours
- Feel free to create an issue on GitHub

Acknowledgments in XiangShan

- A list outlines 32 techniques used in the XiangShan RTL codes
- **https://docs.xiangshan.cc/acknowledgments/**



Multi Port Data Cache

[1] Gurindar S. Sohi, and Manoj Franklin. "High-bandwidth data memory systems for superscalar processors." 4th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS). 1991. [RTL Codes]

Data Replacement

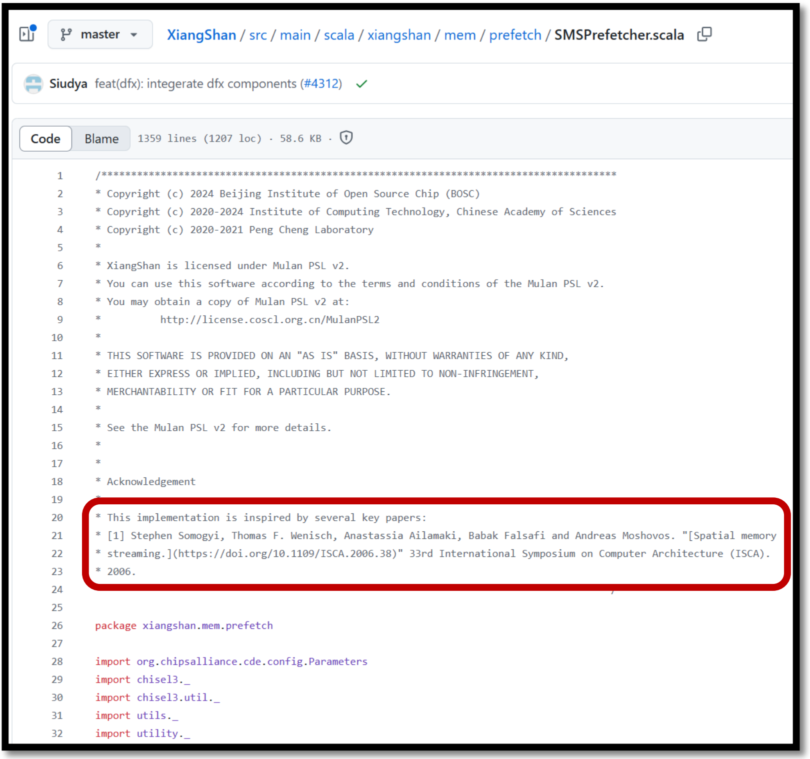
[1] Aamer Jaleel, Kevin B. Theobald, Simon C. Steely, and Joel Emer. "High performance cache replacement using re-reference interval prediction (RRIP)." 37th Annual International Symposium on Computer Architecture (ISCA). 2010. [RTL Codes]

Data Prefetch

[1] Jean-Loup Baer, and Tien-Fu Chen. "An effective on-chip preloading scheme to reduce data access penalty." ACM/IEEE Conference on Supercomputing. 1991. [RTL Codes]

[2] Stephen Somogyi, Thomas F. Wenisch, Anastassia Ailamaki, Babak Falsafi and Andreas Moshovos. "Spatial memory streaming." 33rd International Symposium on Computer Architecture (ISCA). 2006. [RTL Codes]

[3] Santhosh Srinath, Onur Mutlu, Hyesoon Kim, and Yale N. Patt "Feedback directed prefetching: Improving the performance and bandwidth-efficiency of hardware prefetchers." IEEE 13th International Symposium on High Performance Computer Architecture (HPCA). 2007. [RTL Codes]

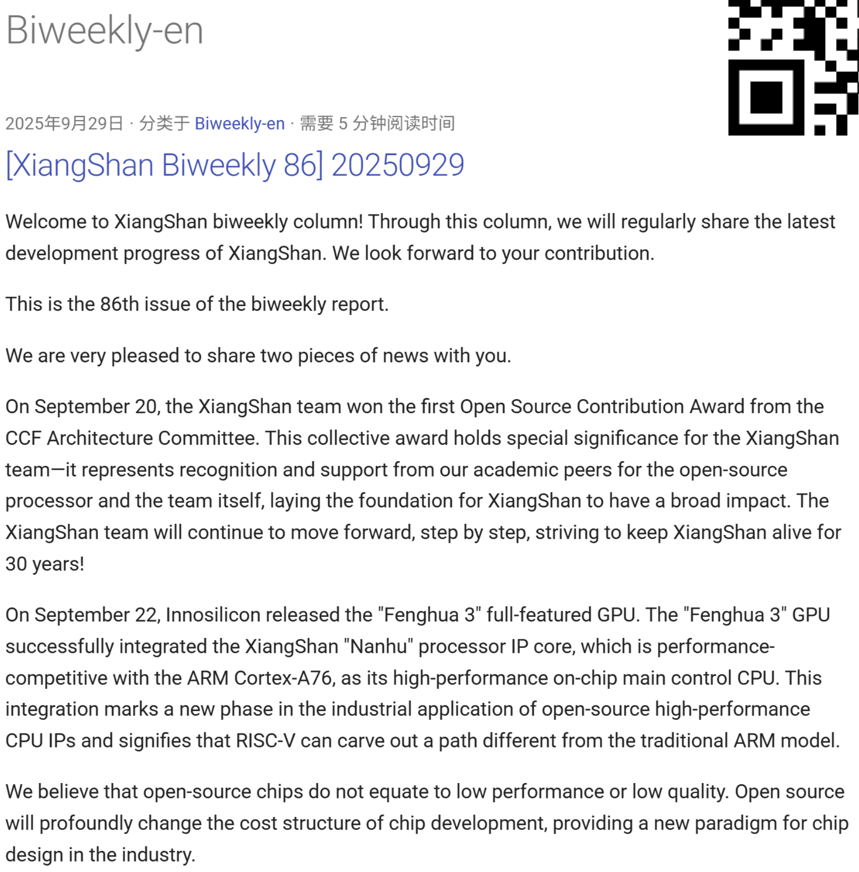


master
XiangShan / src / main / scala / xiangshan / mem / prefetch / SMSPrefetcher.scala
Siudya feat(dfx): integerate dfx components (#4312) ✓
Code
Blame
1359 lines (1207 loc) · 58.6 KB ·

```
/*************************************************************************************
* Copyright (c) 2024 Beijing Institute of Open Source Chip (BOSC)
* Copyright (c) 2020-2024 Institute of Computing Technology, Chinese Academy of Sciences
* Copyright (c) 2020-2021 Peng Cheng Laboratory
*
* XiangShan is licensed under Mulan PSL v2.
* You can use this software according to the terms and conditions of the Mulan PSL v2.
* You may obtain a copy of Mulan PSL v2 at:
*          http://license.coscl.org.cn/MulanPSL2
*
* THIS SOFTWARE IS PROVIDED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OF ANY KIND,
* EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO NON-INFRINGEMENT,
* MERCHANTABILITY OR FIT FOR A PARTICULAR PURPOSE.
*
* See the Mulan PSL v2 for more details.
*
*
* Acknowledgement
*
* This implementation is inspired by several key papers:
* [1] Stephen Somogyi, Thomas F. Wenisch, Anastassia Ailamaki, Babak Falsafi and Andreas Moshovos. "[Spatial memory
* streaming.](https://doi.org/10.1109/ISCA.2006.38)" 33rd International Symposium on Computer Architecture (ISCA).
* 2006.


package xiangshan.mem.prefetch

import org.chipsalliance.cde.config.Parameters
import chisel3._
import chisel3.util._
import utils._
import utility._
```

Biweekly Report in English

- our recent progress and performance data
- https://docs.xiangshan.cc/zh-cn/latest/blog/category/biweekly-en/



Biweekly-en

2025年9月29日 · 分类于 Biweekly-en · 需要 5 分钟阅读时间

[XiangShan Biweekly 86] 20250929

Welcome to XiangShan biweekly column! Through this column, we will regularly share the latest development progress of XiangShan. We look forward to your contribution.

This is the 86th issue of the biweekly report.

We are very pleased to share two pieces of news with you.

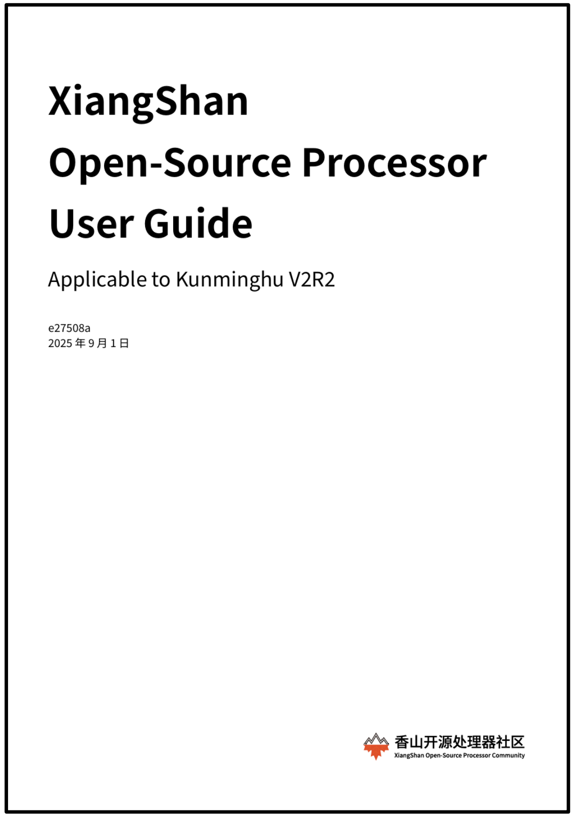
On September 20, the XiangShan team won the first Open Source Contribution Award from the CCF Architecture Committee. This collective award holds special significance for the XiangShan team—it represents recognition and support from our academic peers for the open-source processor and the team itself, laying the foundation for XiangShan to have a broad impact. The XiangShan team will continue to move forward, step by step, striving to keep XiangShan alive for 30 years!

On September 22, Innosilicon released the "Fenghua 3" full-featured GPU. The "Fenghua 3" GPU successfully integrated the XiangShan "Nanhu" processor IP core, which is performance-competitive with the ARM Cortex-A76, as its high-performance on-chip main control CPU. This integration marks a new phase in the industrial application of open-source high-performance CPU IPs and signifies that RISC-V can carve out a path different from the traditional ARM model.

We believe that open-source chips do not equate to low performance or low quality. Open source will profoundly change the cost structure of chip development, providing a new paradigm for chip design in the industry.

User Guide

- For Software Developers and Hardware Integrators
- https://docs.xiangshan.cc/projects/user-guide/en/late



XiangShan

Open-Source Processor

User Guide

Applicable to Kunminghu V2R2

e27508a
2025 年 9 月 1 日

香山开源处理器社区
XiangShan Open-Source Processor Community

Design Document

- Detailed Docs on Microarchitecture and Modules
- https://docs.xiangshan.cc/projects/design/en/latest/

XiangShan
Open-Source Processor
Design Document

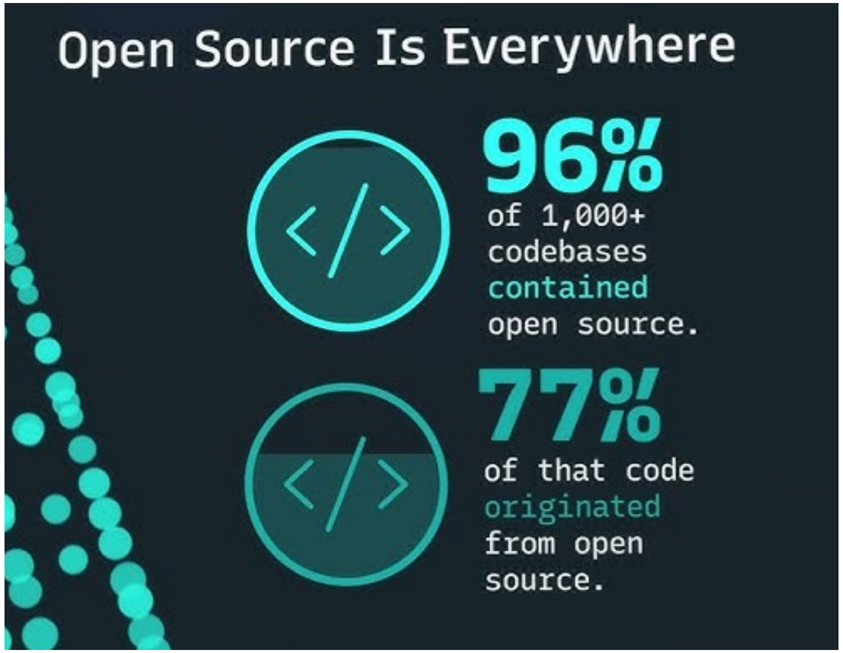
Applicable to Kunminghu V2R2

52fc49d
2025 年 7 月 15 日

香山开源处理器社区
XiangShan Open-Source Processor Community

Open-Source: From Software to Hardware

- **96% software** codebases contain **open-source code** (overall 77%)
- *In the future, the proportion of **open-source IP in the chip industry will inevitably break through zero**, and will continue to increase.*



Open Source Is Everywhere

96% of 1,000+ codebases contained open source.

77% of that code originated from open source.

The 2024 Open Source Security and Risk Analysis (OSSRA) Report | Synopsys

OpenXiangShan: Together for a Shared Future

- Feel free to contact us through email or file issues on GitHub!
  - all@xiangshan.cc
  - https://github.com/OpenXiangShan/XiangShan

XiangShan Home page

XiangShan Document

XiangShan Biweekly Report

XiangShan User Guide

XiangShan Design Document