XiangShan Practice:
The Path to Industrial Deployment of Open-Source High-Performance RISC-V Processor

Yungang Bao

2026.6.9

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

Open Source — fully leveraging RISC-V's openness

Prof. David Patterson
UC Berkeley

Specifications \ Designs	Free & Open Designs	Licensable Designs	Closed Designs	Products
Free & Open Spec	"Open Source"			Based on Free Open Licensed Closed
Licensable Spec				Based on Licen or Closed
Closed Spec				Based on Close Designs

Designs ("Source")

XIANGSHAN

Source: David Patterson, A New Golden Age for Computer Architecture: History, Challenges, and Opportunities, August 22, 2019

The XiangShan Project

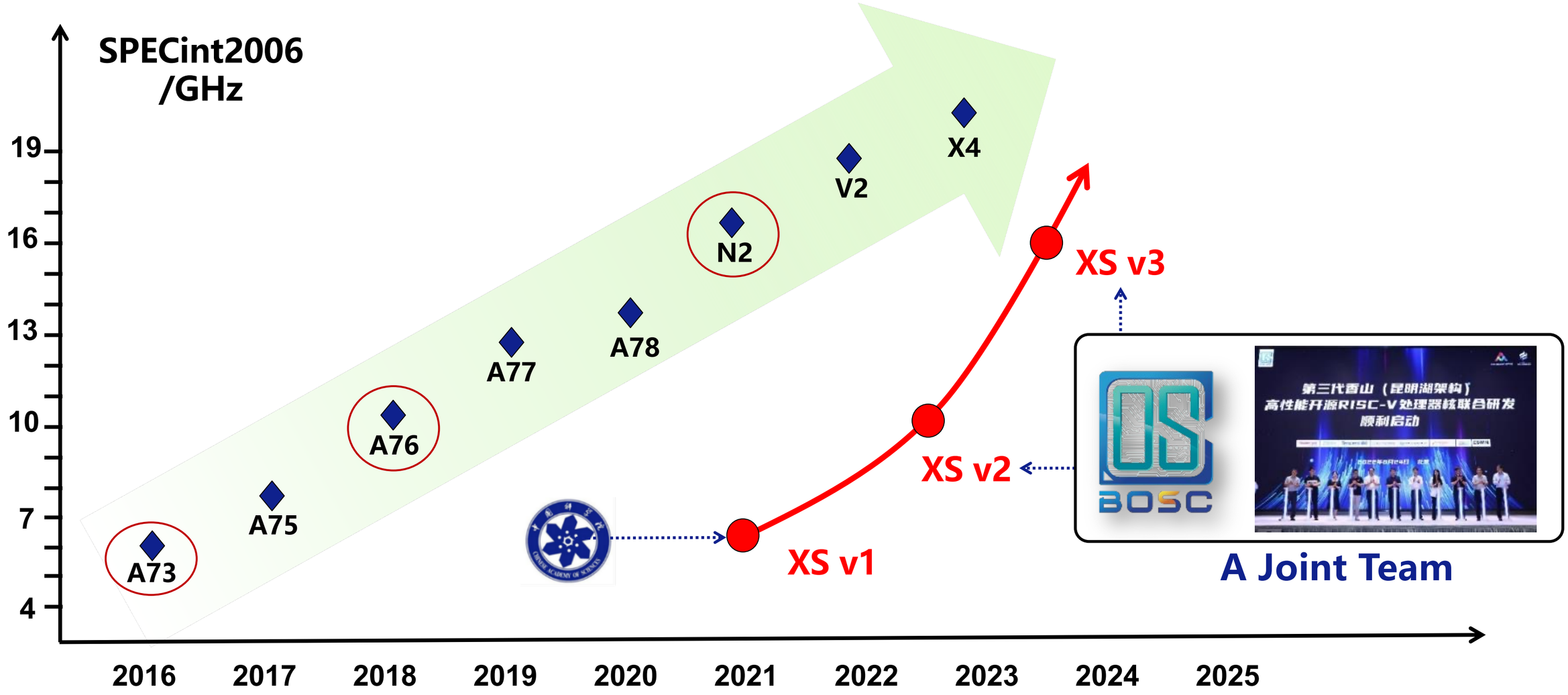
- **Goal**: Build a Linux-like open-source RISC-V processor that can be widely adopted by industry while also enabling academic research.

Linux

V.S.

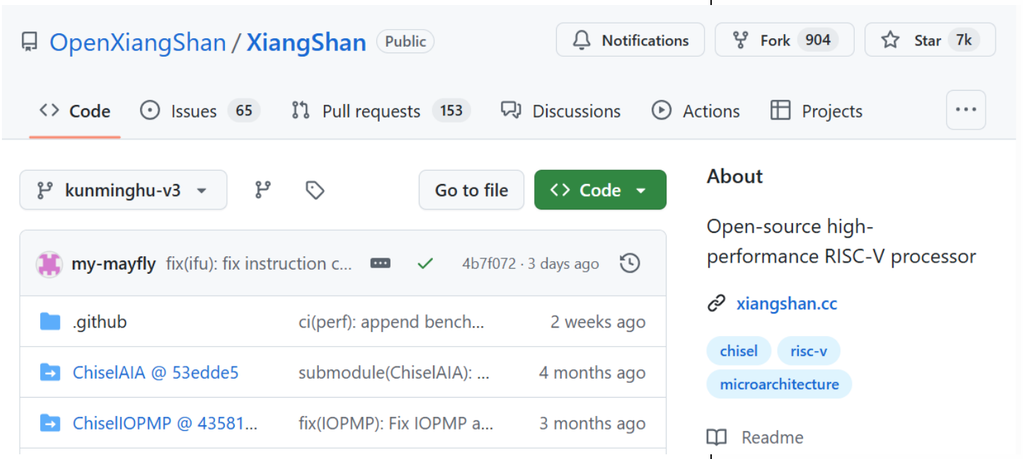
XIANGSHAN

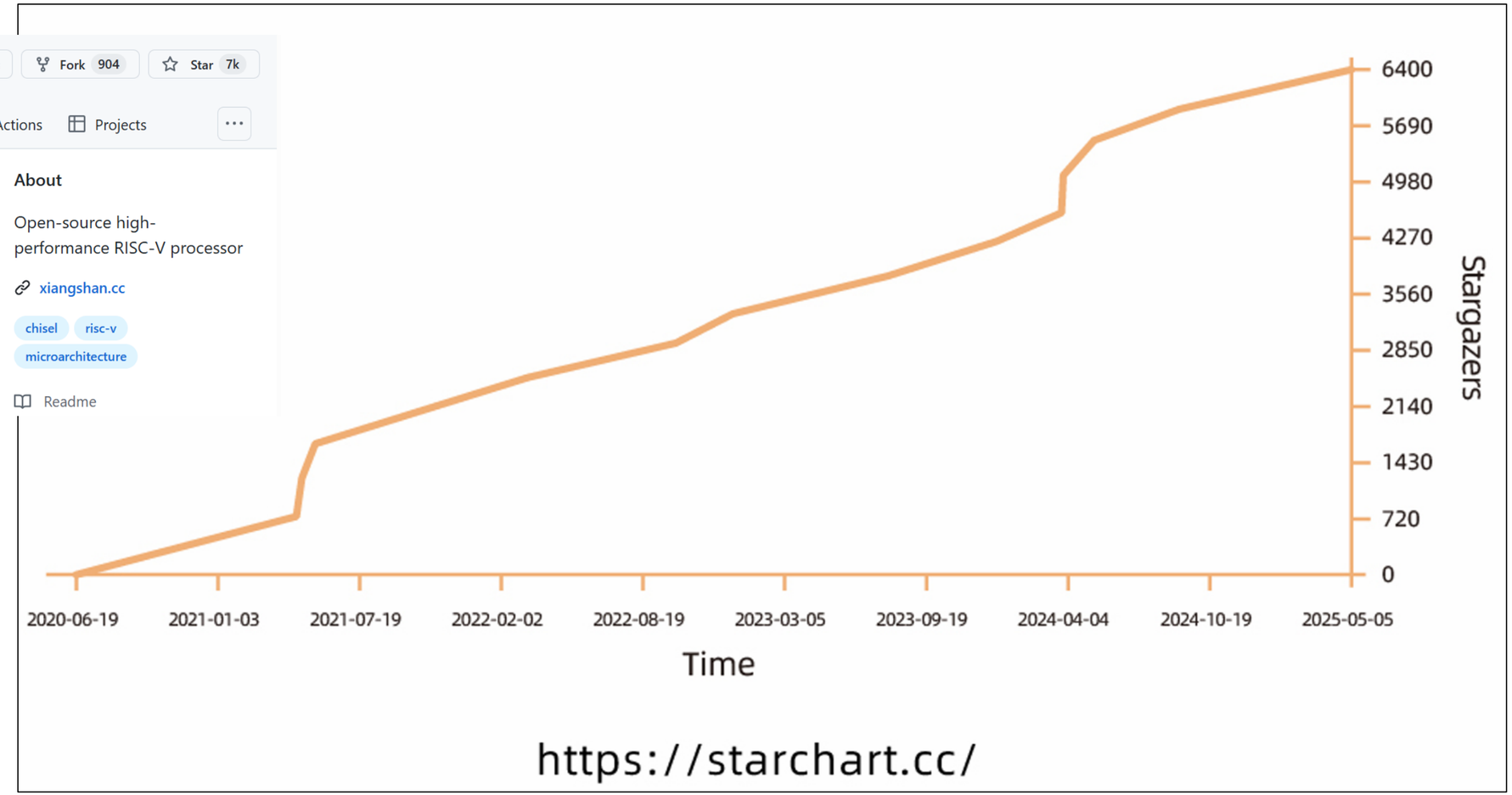
The Evolution Process of XiangShan



One of the most active open-source chip projects

- **GitHub: >7000** Stars, > **900** Forks

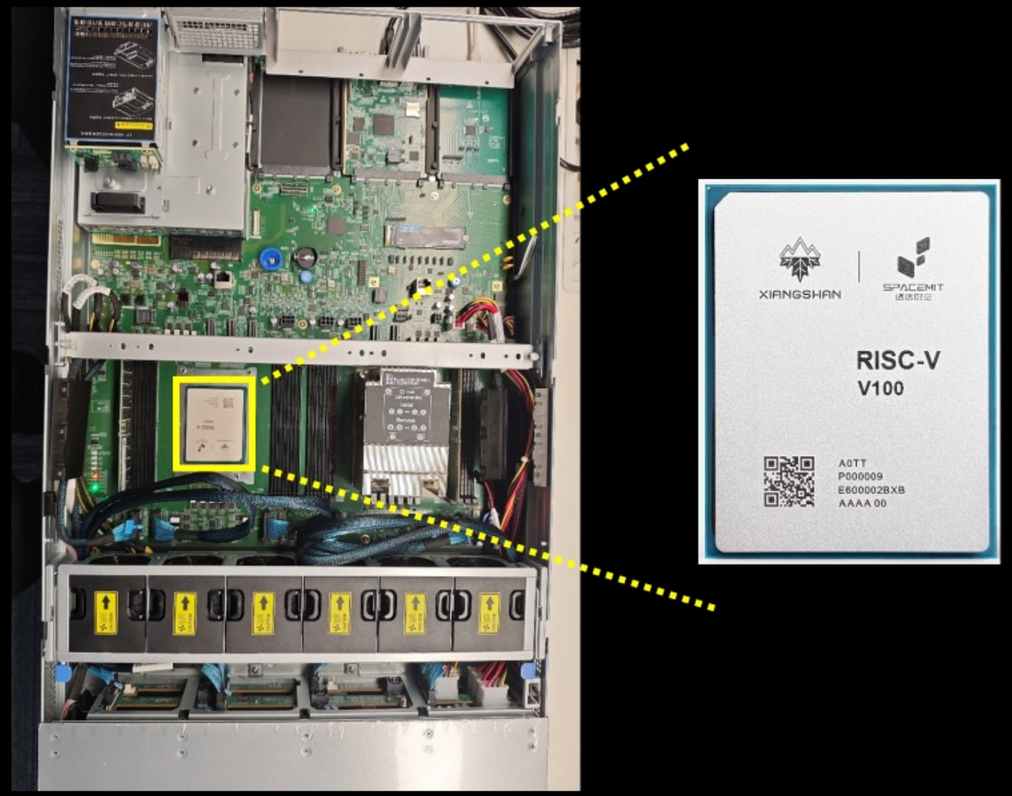




Data collected on May 20, 2026

One of the Highest Performance RISC-V Chip

XiangShan-KMH achieved 16.5/Ghz for SPECPU 2006, one of the highest performance RISC-V Chip by far.



SPECCPU 2006 @ XSCC1.1 base

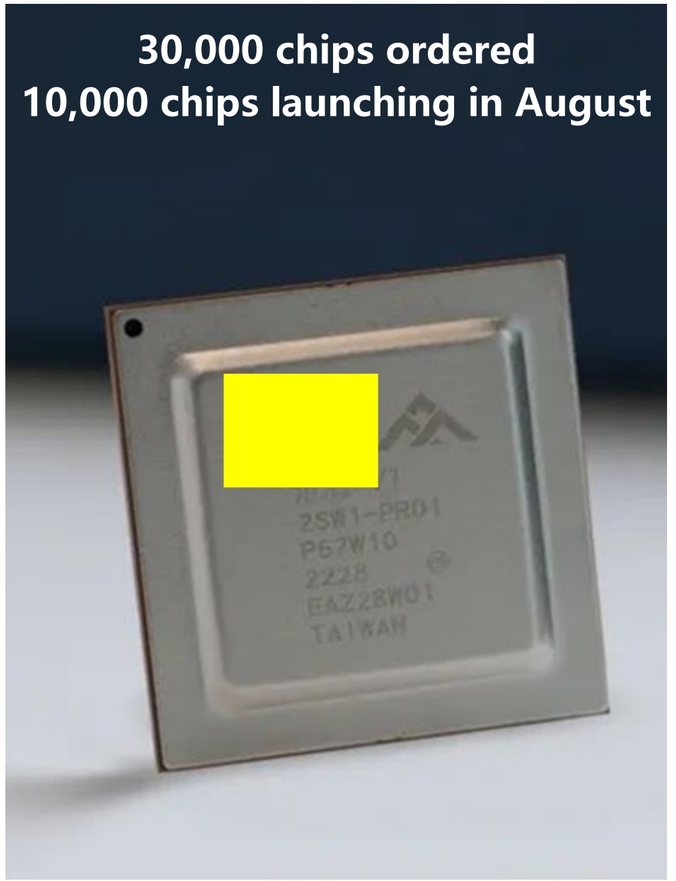
	Benchmark	Score/GHz
INT	400.perlbench	12.14
	401.bzip2	7.66
	403.gcc	14.05
	429.mcf	15.84
	445.gobmk	9.92
	456.hmmer	20.09
	458.sjeng	10.89
	462.libquantum	89.14
	464.h264ref	19.34
	471.omnetpp	11.46
	473.astar	8.40
	483.xalancbmk	22.15
	SPECInt	15.31
FP	410.bwaves	22.84
	416.gamess	13.19
	433.milc	20.16
	434.zeusmp	18.79
	435.gromacs	10.49
	436.cactusADM	24.21
	437.leslie3d	17.86
	444.namd	11.63
	447.dealII	23.82
	450.soplex	18.25
	453.povray	18.81
	454.calculix	9.90
	459.GemsFDTD	19.49
	465.tonto	12.57
	470.lbm	41.20
	481.wrf	15.66
	482.sphinx3	15.27
	SPECfp	17.36
SPECCPU 2006		16.48

Mass Deployment

MooreThreads GPGPU in 10K-chip datacenters Integrates XiangShan-NH as control CPU

SpaceMIT edge-server chip integrates XiangShan-KMH

An internet company‘s next-gen chip integrates 4-core XiangShan-KMH



XiangShan Family Open-Source Projects

XiangShan·KMH

XiangShan·NH

XiangShan·AI

XiangShan·NoC

XIANGSHAN

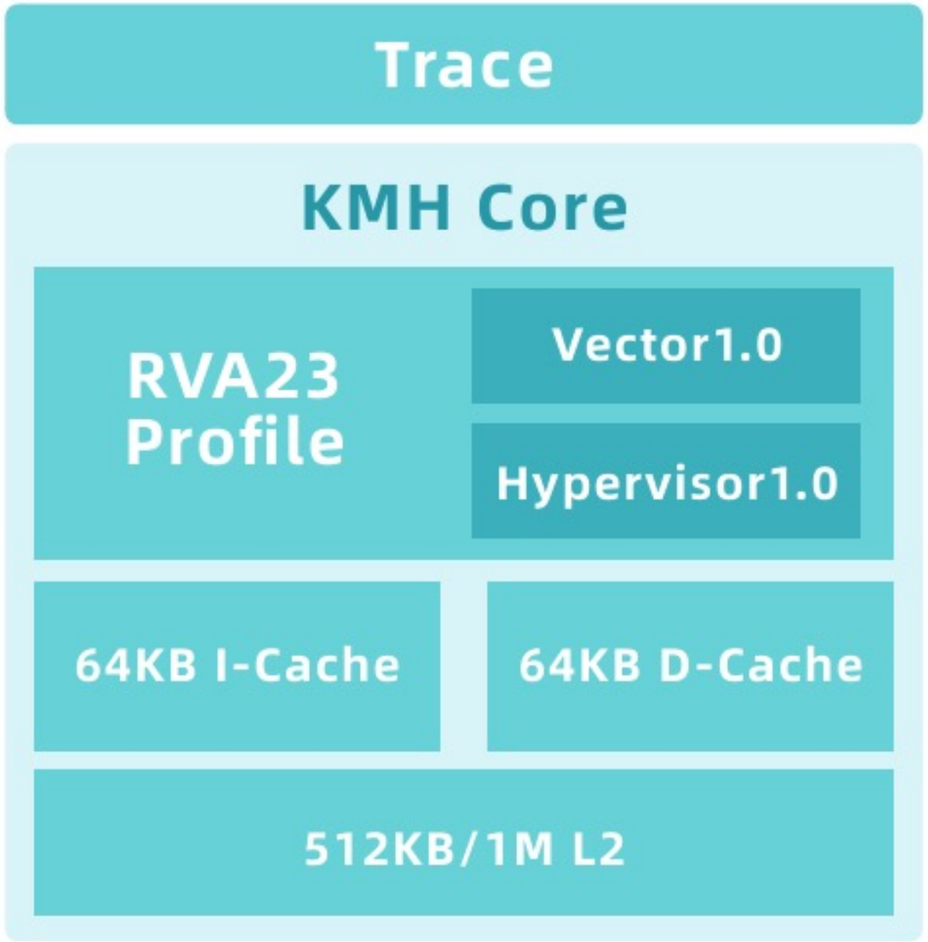
openxiangshan.cc

XiangShan
(KMH + NoC-Mesh)

XiangShan
(NH + NoC-Ring)

(1) XiangShan • KMH

Core Architecture and Features



Advanced Architecture

Fully compliant with the RISC-V RVA23 Profile standard ensures broad software ecosystem compatibility

Scalability

Supports flexible core configurations from 1 to 128 cores, with scalable multi-die and multi-socket options.

Performance Optimizations

Integrates a TAGE-based branch prediction algorithm and data prefetching mechanism. Combined with advanced microarchitectural techniques, it significantly reduces latency and improves instruction throughput.

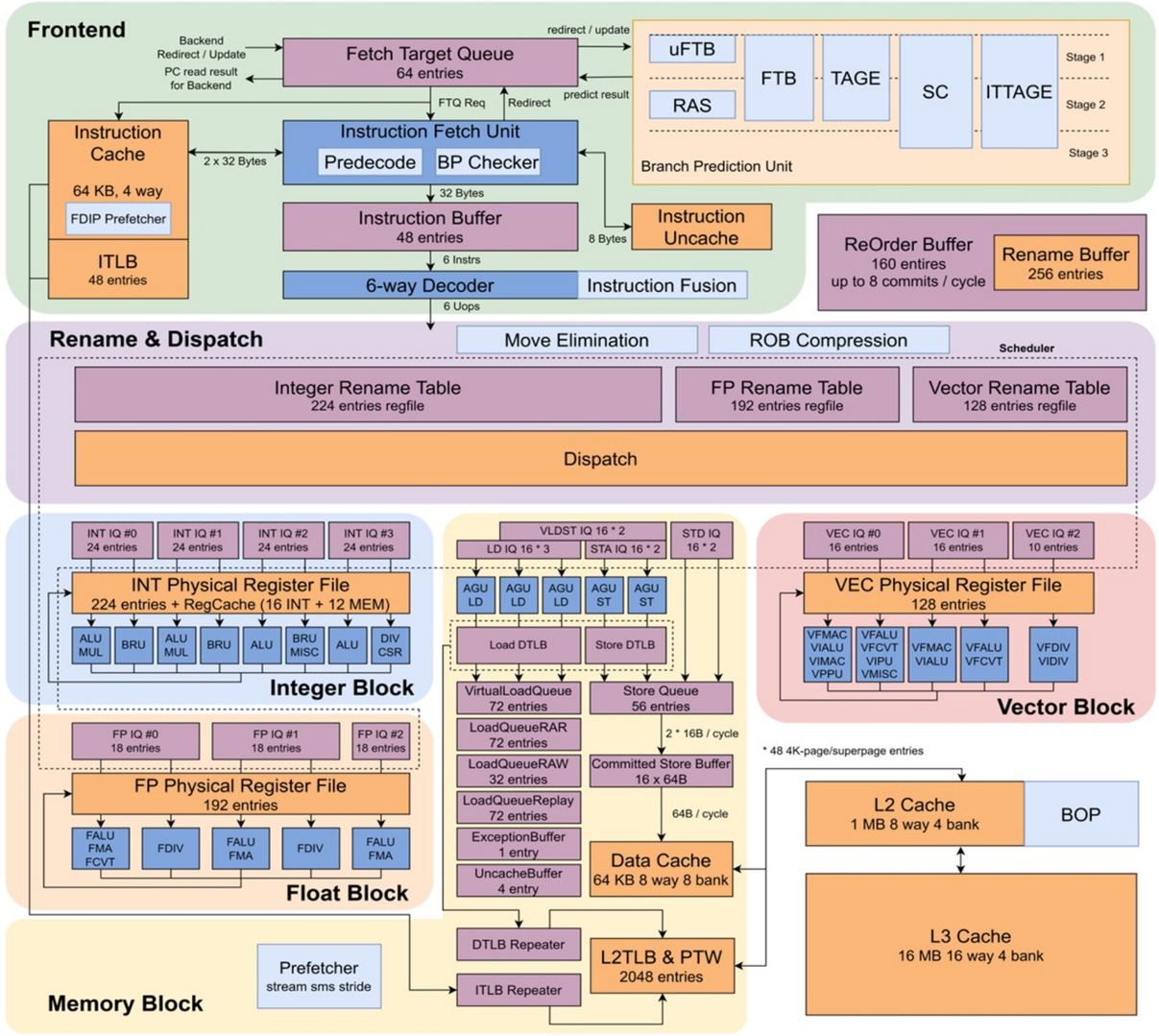
Security

Hardware-level support for core, interrupt, and I/O isolation, along with memory encryption, making it ideal for confidential computing scenarios including secure processes, containers, and confidential virtual machines.

XiangShan · KMH

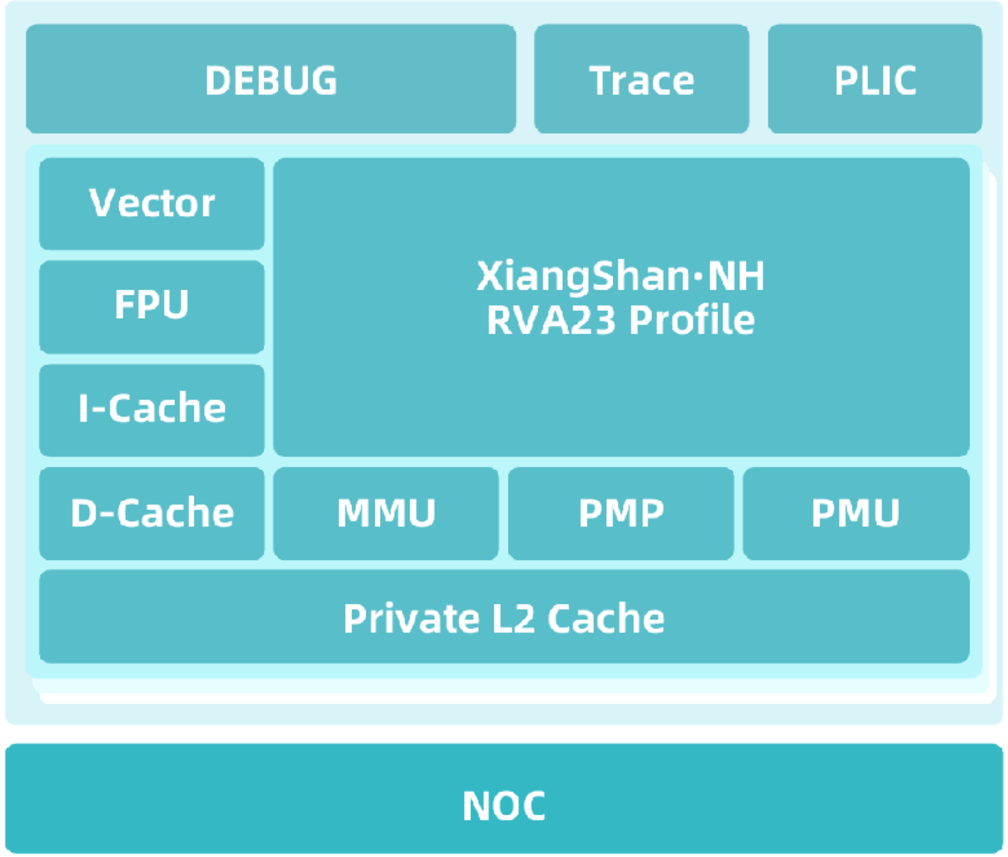
Xiangshan GitHub
https://github.com/OpenXiangShan

Feature	XiangShan·KMH（2025）
Rename width	6
ROB size	256+
ALUS	4
L1 instr cache	64KB
L1 data cache	64KB
L2 cache	1024KB
L3 cache	Up to 16MB
NoC support	Y
ITLB	48
DTLB	48
L2 TLB	1024
Vector	Y
Virtualization	Y
ECC support	Y
PMA/PMP support	Y
Debug support	Y
External interface	AXI4/CHI
ISA	RVA 23
Frequency@Process	3GHz@7nm
SPECINT2006/GHz	15/GHz



(2) XiangShan • NH

Core Architecture and Features



Advanced Architecture

Superscalar, 4-wide decode, 15-stage pipeline, fully compliant with the RVA23 Profile — laying a solid foundation for powerful computing.

Memory Model

Fully supports the memory consistency model, both single-core and multi-core scalability, ensuring high efficiency and stability for multitasking.

Microarchitecture Design

Supports out-of-order issue with 4 rename registers; integrates 4 ALUs, 3 branch units, 2 multiply units, and 1 division unit, delivering robust computational capability.

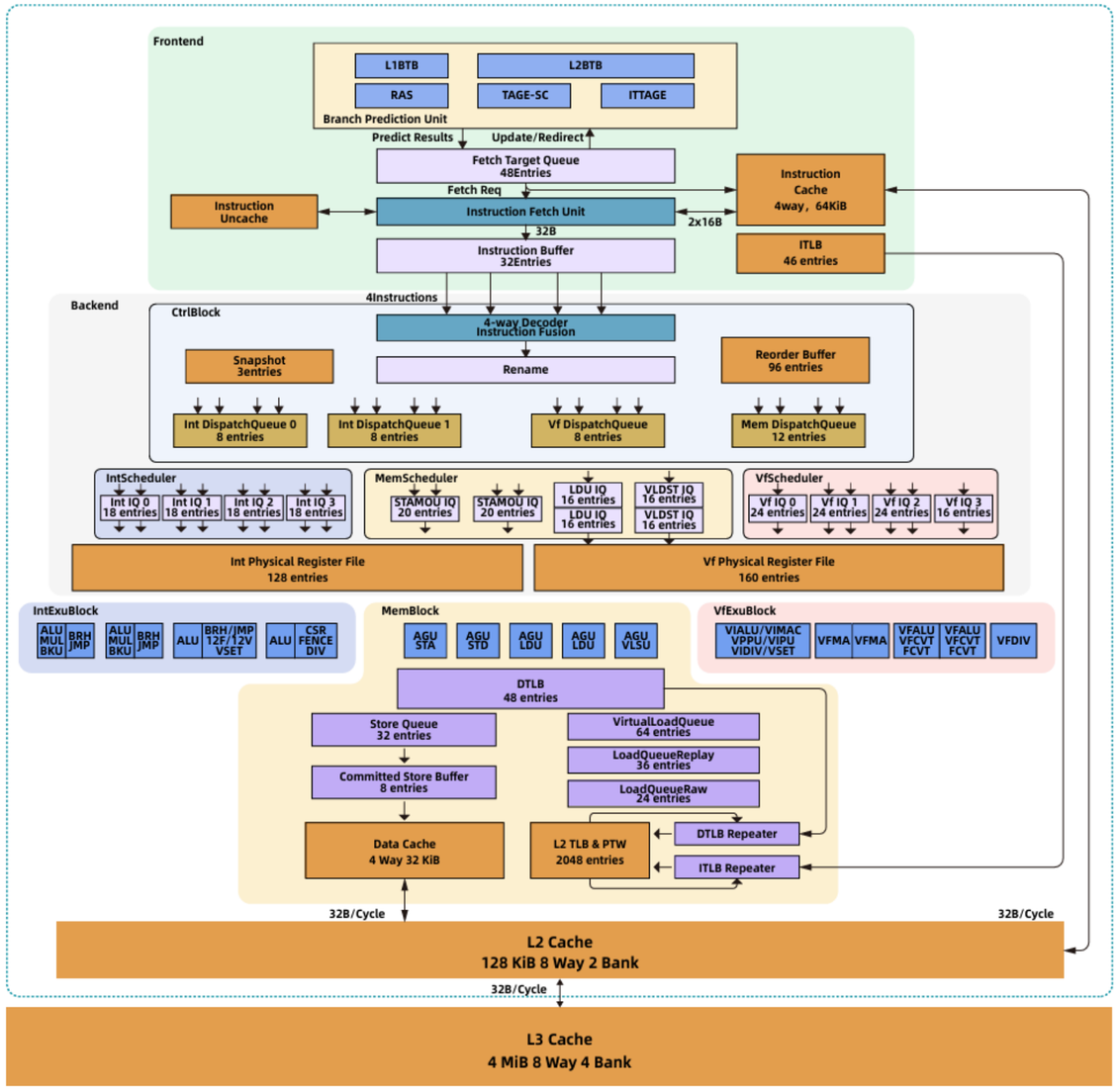
Standards and Extensions

Fully compatible with the IEEE 754-2008 floating-point standard and fully supports the RISC-V vector extension, significantly improving data processing efficiency.

XiangShan • NH

Xiangshan GitHub
https://github.com/OpenXiangShan

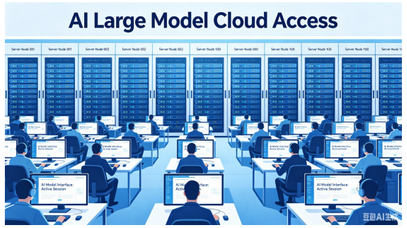
Feature	XiangShan·NH（2023）
Rename width	4
ROB size	192
ALUS	4
L1 instr cache	32KB/64KB
L1 data cache	64KB
L2 cache	128KB/256KB
L3 cache	Up to 4MB
NoC support	N
ITLB	48
DTLB	48
L2 TLB	2048
Vector	Y
Virtualization	N
ECC support	Y
PMA/PMP support	Y
Debug support	Y
External interface	AXI4
ISA	RV64GCBKV
Frequency@Process	2GHz@12nm
SPECINT2006/GHz	10/GHz



(3) XiangShan · AI

- XiangShan empowers AI computing through two paths: **high-throughput computing IP** and **low-latency AI CPUs** that unify general-purpose computing with AI inference

Open Source AI IP for High Throughput Applications



Open Source AI CPU for Low Latency Applications

Open Source AI IP for High Throughput Applications

- Xiangshan high-throughput AI IP integrates E6 CPU control core, RVV vector core, AME matrix core, intelligent DMA, and software-programmable SRAM.

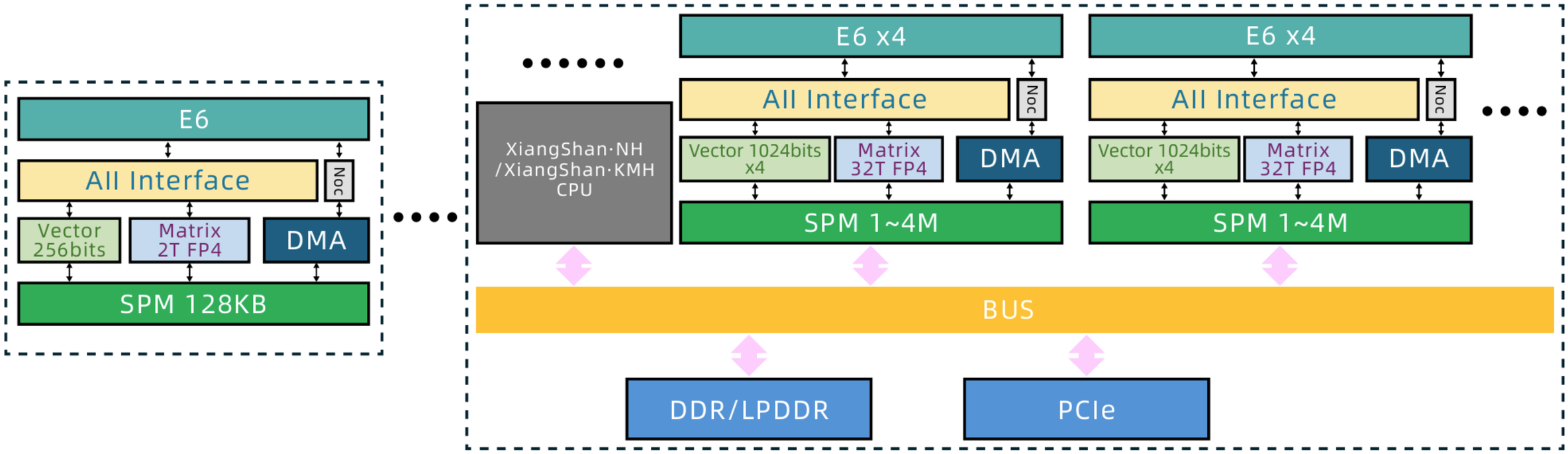
- **Adaptable to Diverse Computing Requirements**

  Single-core FP4 compute: configurable 2-32 TFLOPS

  Single-core FP8 compute: configurable 1-16 TFLOPS

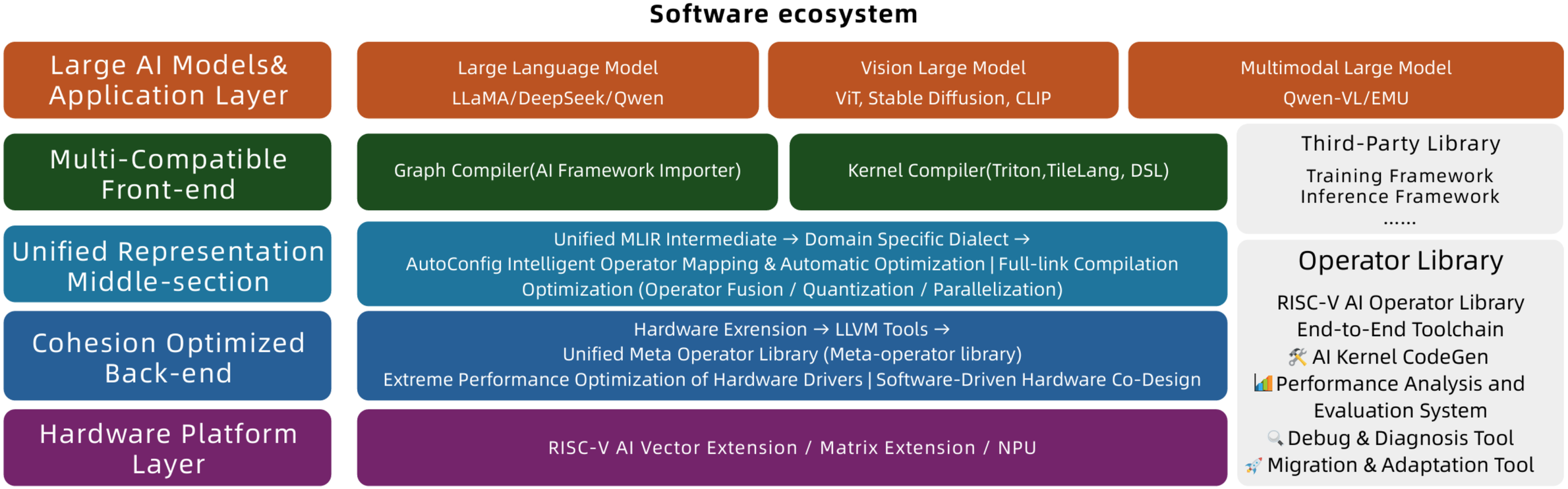
  Single-core vector compute width: configurable 256-4096 bits

  Full matrix data precision support: FP4, MXFP4, FP8, MXFP8, BF16, FP16



Open Source AI IP for High Throughput Applications

- An **end-to-end software stack** for AI chips, comprising large model & application layers, front-end compilers, MLIR, and high-performance operator libraries, leverages a full toolchain to enable end-to-end support from model deployment to performance tuning.



Open Source AI CPU for Low Latency Applications

- **Unified General-Purpose and AI-Inference CPU (XSAI)**
- XSAI integrates a dedicated AI matrix unit (AMU) and a high-performance Xiangshan CPU core within a single AI CPU core. The AMU handles intensive matrix computations, while the high-performance Xiangshan CPU core manages general-purpose computing, with both sharing data through a high-bandwidth L2 Cache. This design eliminates data movement bottlenecks, enabling seamless collaboration between general-purpose computing and AI inference.

ISA: **RVA23 + RISC-V AME**

Manufacturing Process: **7nm**

AI Performance: **8TOPS@INT8**

Matrix Precision

Supporting INT8 / FP16 / TF32 / BF16

Core Configuration

Supporting 1/2/4/8/16AI CPU Core(s)

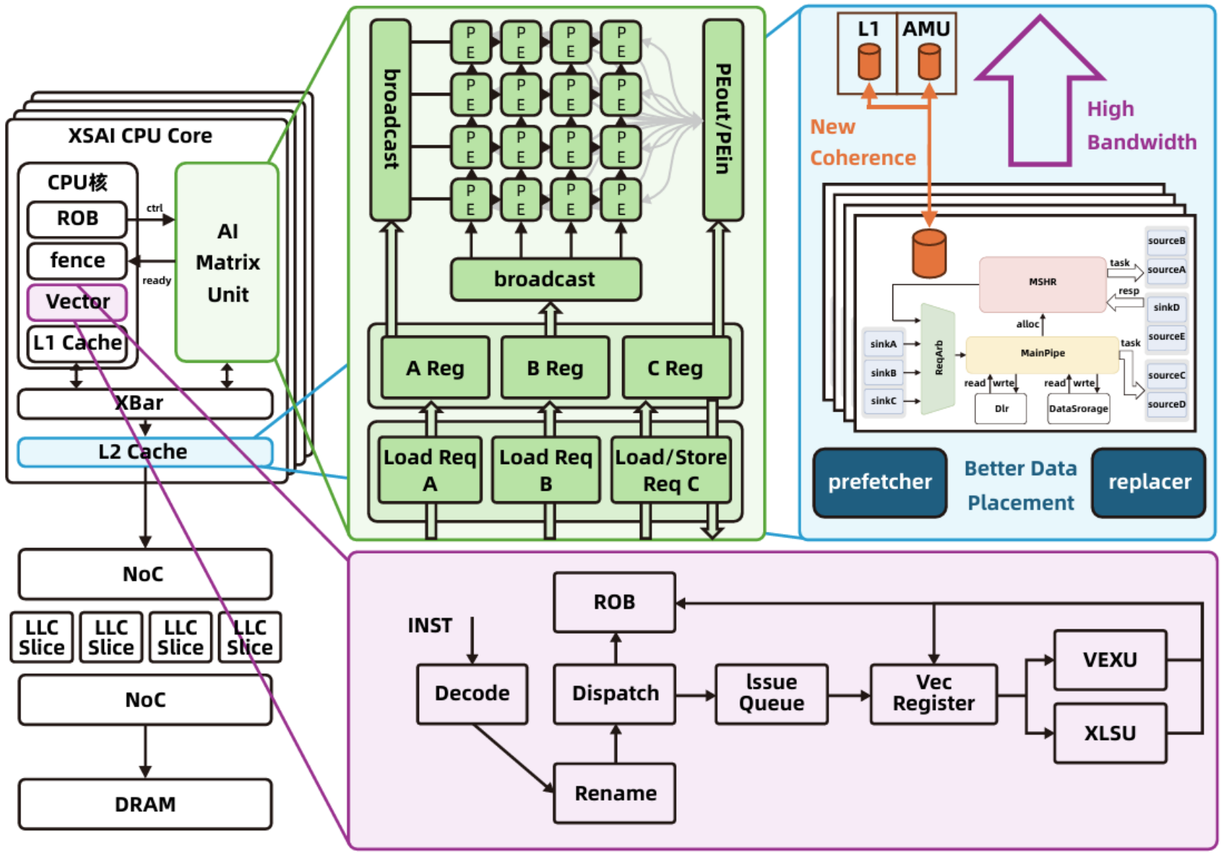
Suitable for various scenario:
from edge devices to Cloud Servers

Area and Power

about 2mm² / 5W

Extreme balance between high performance and low power

Open Source AI CPU for Low Latency Applications



- **CPU Core with AI Capability**

  CPU core does general-purpose computing, while AI matrix unit runs AI inference logic. These two modules share data through high-bandwidth and low latency L2 Cache.

- **Unified Memory Access**

  CPU core and AI matrix unit shares uniformed memory space, eliminating frequent data movement costs under heterogenous architecture, and reducing end-to-end AI inference latency.

- **High-efficient Communication Mechanism**

  Built-in high-performance NoC, supporting large-scale data synchronization of parallel computing, making it suitable for LLM inference .

Llama2-15M running on XSAI@FPGA

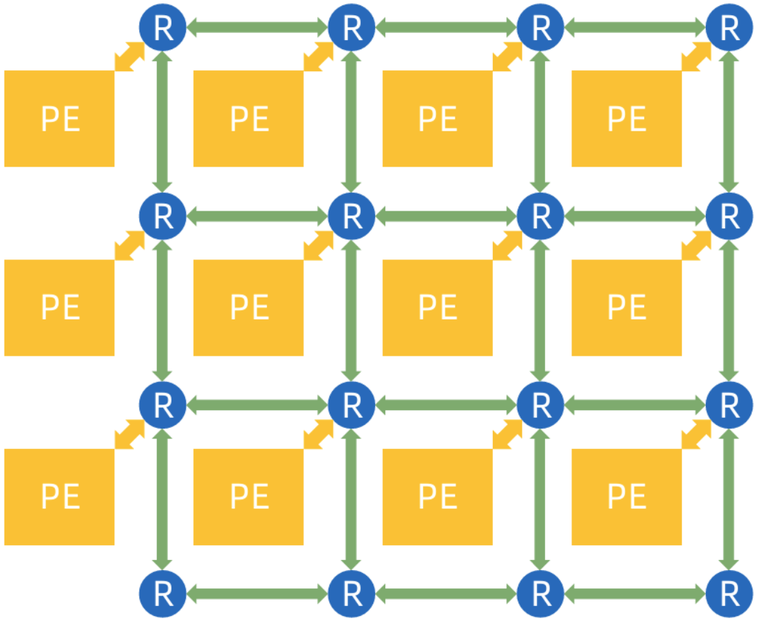
```
<s> once upon a time, in a forest where the trees never lost their leaves, a young woodcutter discovered a door hidden beneath an ancient oak. the door was no taller than his knee and sealed wit
h a lock shaped like a crescent moon. he had passed this way a hundred times but never noticed it before. that morning, something made him kneel in the soft earth and press his ear against the w
eathered wood. a voice from inside whispered his name. he reached into his coat pocket and found, to his astonishment, a silver key he had never seen before. with trembling
--- prefill: processing 128 prompt tokens ---
icy thorns, he quickly grabbed the key and opened the door. Inside, he found a magical world full of wonderful things. He expl

======= XSAI Inference Report  (CPU @ 2 GHz assumed) =======
  model load        :     885132934 cycles     442.566 ms

  prompt tokens     : 128
  prefill cycles    :     745026565            372.513 ms
  prefill IPC       : 0.876
  TTFT              :                          372.513 ms
  prompt throughput :     343.61 tok/s

  decode tokens     : 32
  decode cycles     :    1766140587            883.070 ms
  decode IPC        : 0.476
  decode throughput :      36.24 tok/s
  TPOT avg          :      55191893 cycles      27.596 ms/tok
  TPOT min          :      56668658 cycles      28.334 ms/tok
  TPOT max          :      57489946 cycles      28.745 ms/tok

  total inference   :    2511167152 cycles    1255.584 ms
============================================================


======= xsai_alloc statistics ================================
  pool capacity     : 100 MiB  (104857600 bytes)
  total alloc calls : 4324
  live allocations  : 9
  live bytes        : 43.148 MiB  (45243648 bytes)
  peak usage        : 43.227 MiB  (45326784 bytes)
  peak / capacity   : 43.2%
  free blocks       : 2  (56.852 MiB free)
============================================================


~llama context:        CPU compute buffer size is  16.5190 MiB, matches expectation of  16.5190 MiB
```

(4) XiangShan • NoC

- **XiangShan·NoC** is an advanced Network-on-Chip (NoC) solution purpose-built for modern data centers and AI computing.

Flexible In-Die Specifications

- **Arbitrary Network Topologies:**
  Supports CHI-E protocol, 4/8/16/32/64 cores, multiple channel configurations, and flexible integration to form high-performance compute dies.
- **High-Performance Router Nodes:**
  Built on 7nm advanced process, operating at up to 2.6GHz, offering both low latency and high throughput.
- **Multi-Dimensional Optimization:**
  Supports flexible 1/2/4 channel configurations to match memory access requirements; provides AXI / CHI-DDR interfaces; up to 16 channels per die; supports efficient multicast of KB-sized data blocks.



High Interconnect Frequency	Peak Routing Bandwidth	Configurable Compute Topology	Router Point-to-Point Latency	Network Throughput
1.8 GHz @12nm tt0p8v	**166 GB/s** @256bit	**M × N** configuration	**< 2 ns**	**> 2.6TB/s** @8x8

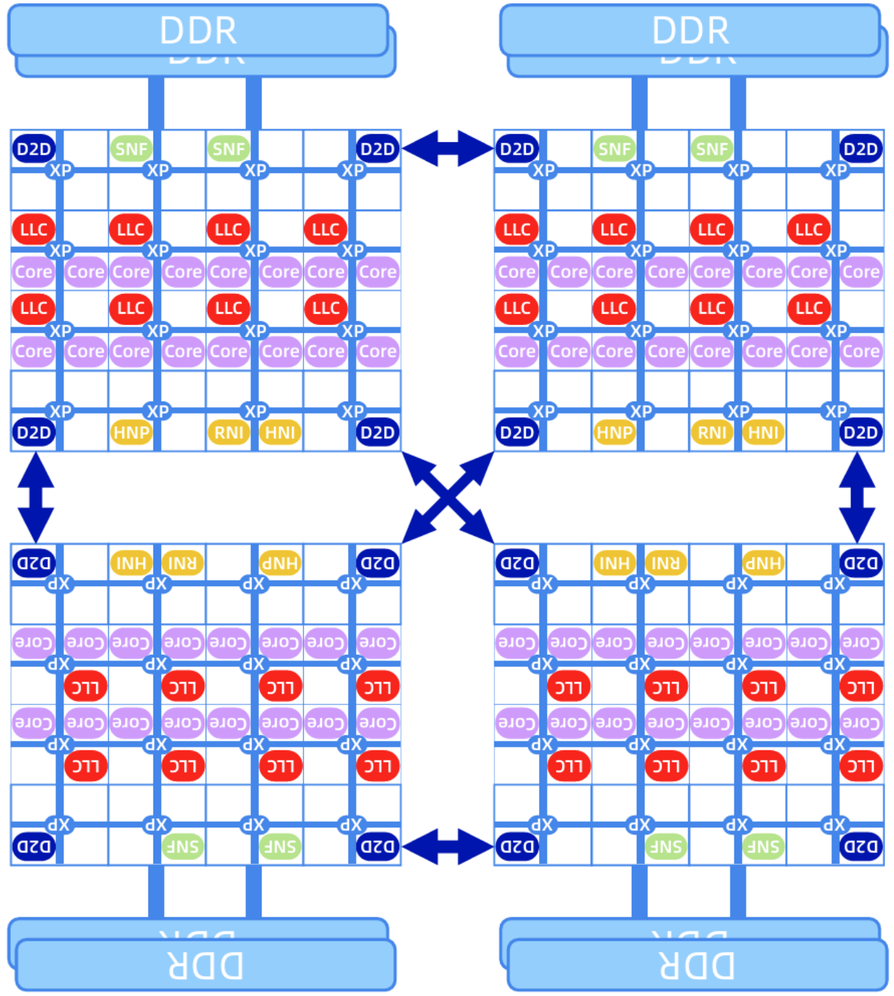
XiangShan • NoC

Advanced Multi-Die Interconnect

- **UCIe Advanced Packaging:**
  Fully designed based on the UCIe 2.0 standard, supporting 2-die or 4-die advanced packaging solutions to enable seamless chiplet-to-chiplet connectivity.

- **Ultra-High Bandwidth Interconnect:**
  Inter-die UCIe bandwidth supports flexible configurations of 16/32/64/128 lanes, providing unprecedented communication capability for large-scale systems-on-chip.

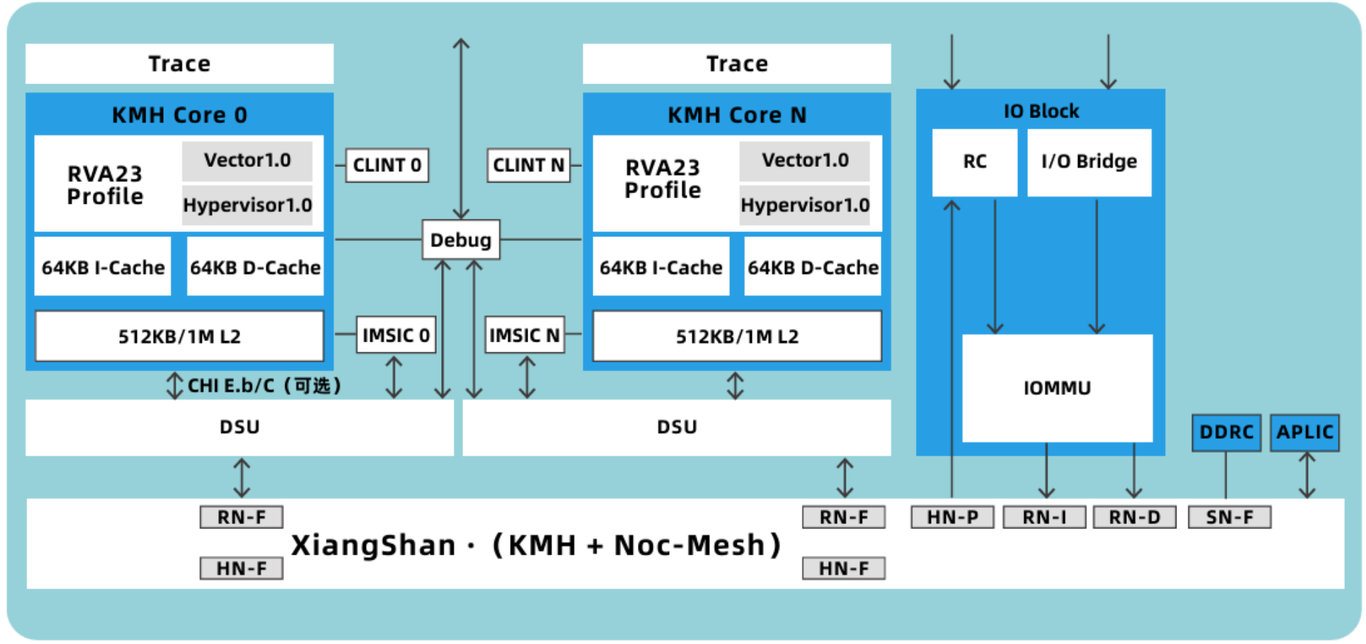
- **Hardware-Software Co-Design Control:**
  Provides automated boot and configuration control for multi-die systems. Combining hardware acceleration with software scheduling, it ensures stable and efficient system operation.



(5) XiangShan Open Source Computing Subsystems

XiangShan · (KMH + Noc-Mesh)

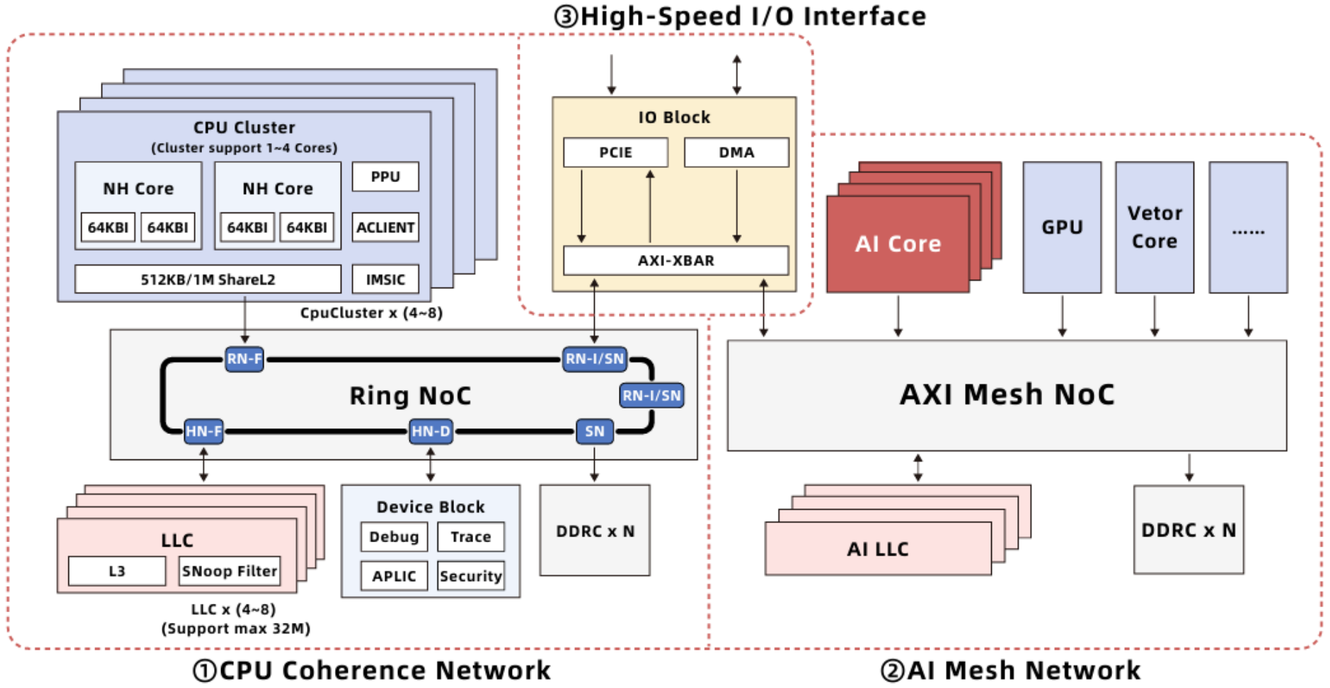
The XiangShan · (KMH + Noc-Mesh) Interconnect server computing subsystem has been performance-optimized for multi-core scenarios ranging from 64 to 256 cores, as well as multi-channel DDR scenarios with 8 to 12 channels. It also supports interfacing with AI inference engines. In addition, co-debugging with more than one high-performance commercial IP has been completed.



XiangShan Open Source Computing Subsystems

XiangShan ·（NH + NoC-Ring）

The XiangShan ·（NH + NoC-Ring） Interconnect computing subsystem strikes a balance between compute performance and PPA specifically for balanced AI inference. It also ensures better support for high-bandwidth AXI IPs, simplifying the integration complexity for members working with non-coherent interconnects.



XiangShan Family Open-Source Projects

XiangShan·KMH

XiangShan·NH

XiangShan·AI

XiangShan·NoC

XIANGSHAN

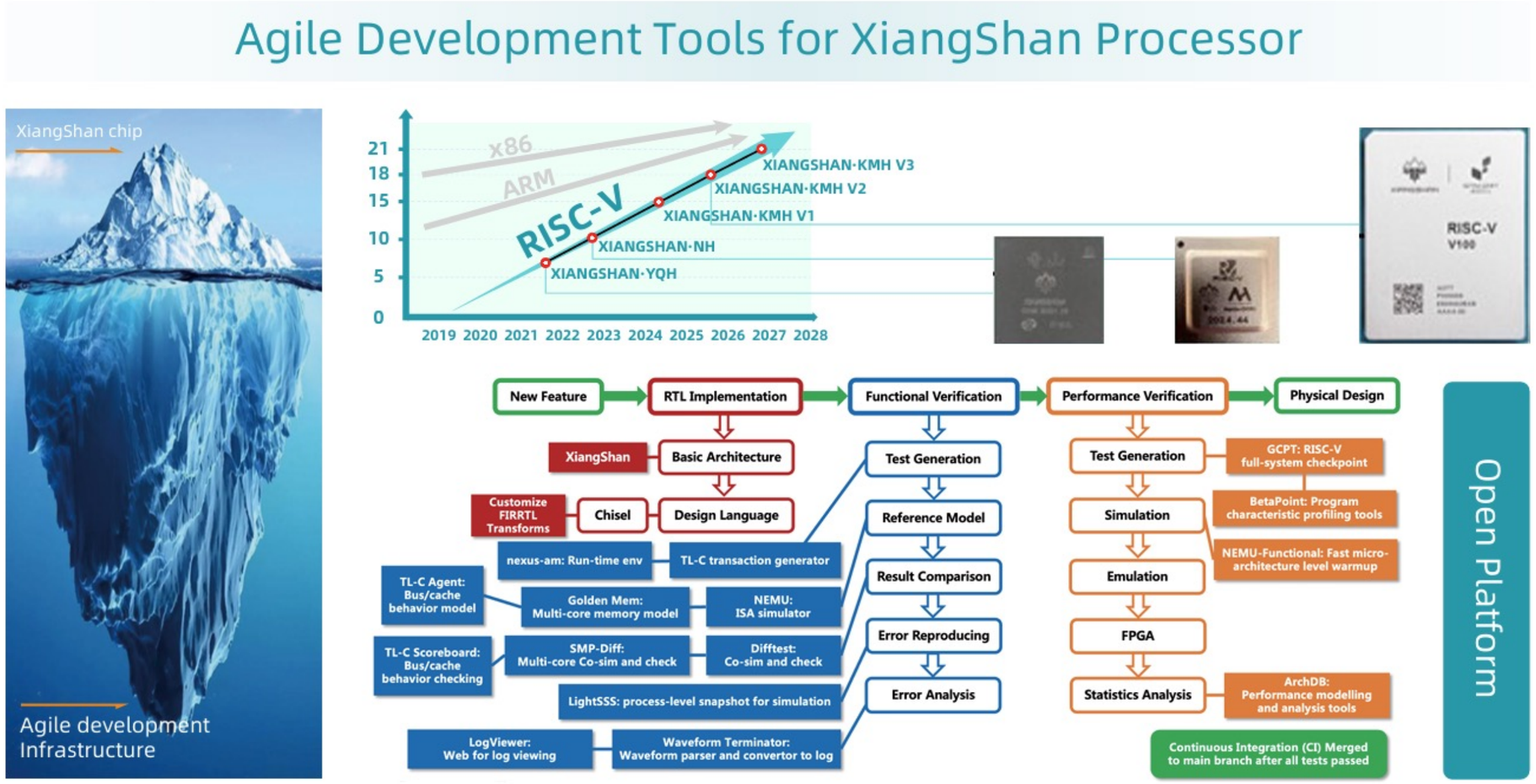
openxiangshan.cc

XiangShan
(KMH + NoC-Mesh)

XiangShan
(NH + NoC-Ring)

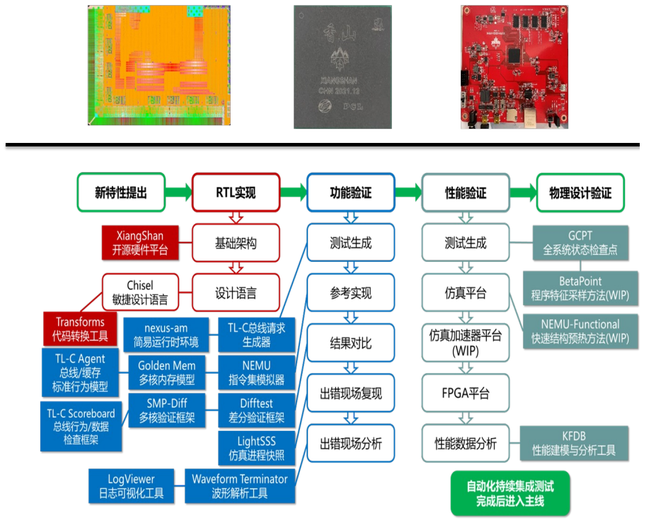
XiangShan will release a new version every year

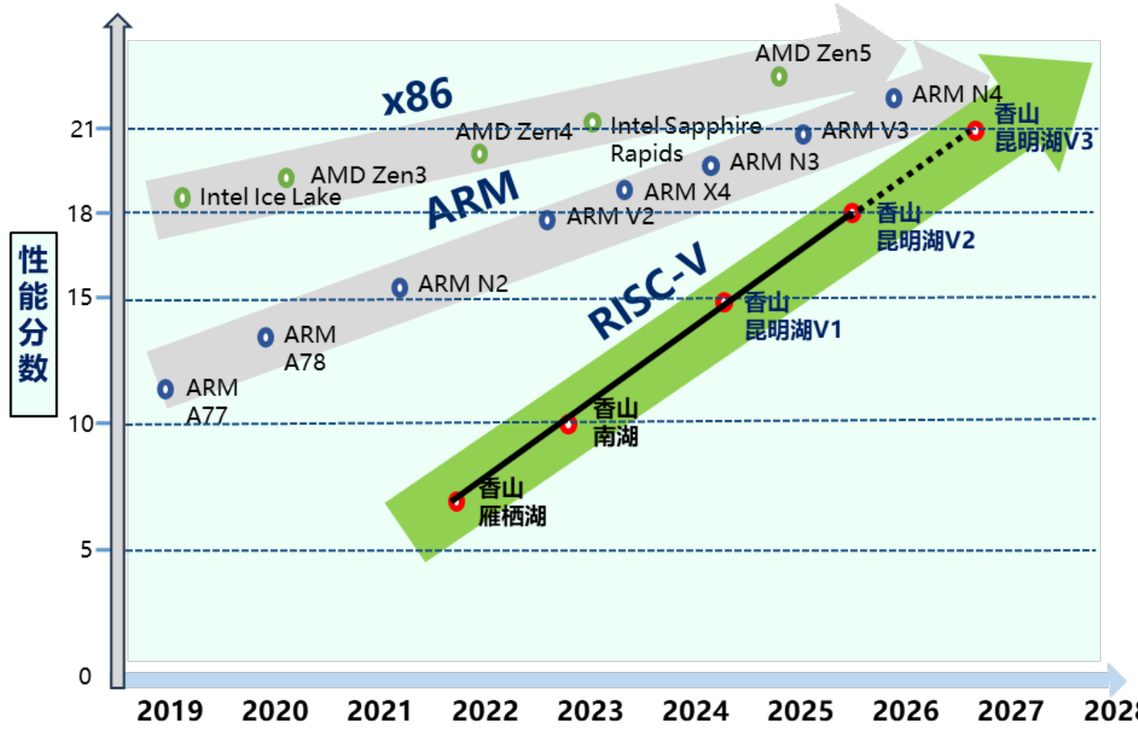
- Through the agile open-source methodology, XiangShan achieves rapid product iteration.



(1) High performance + Fast iteration

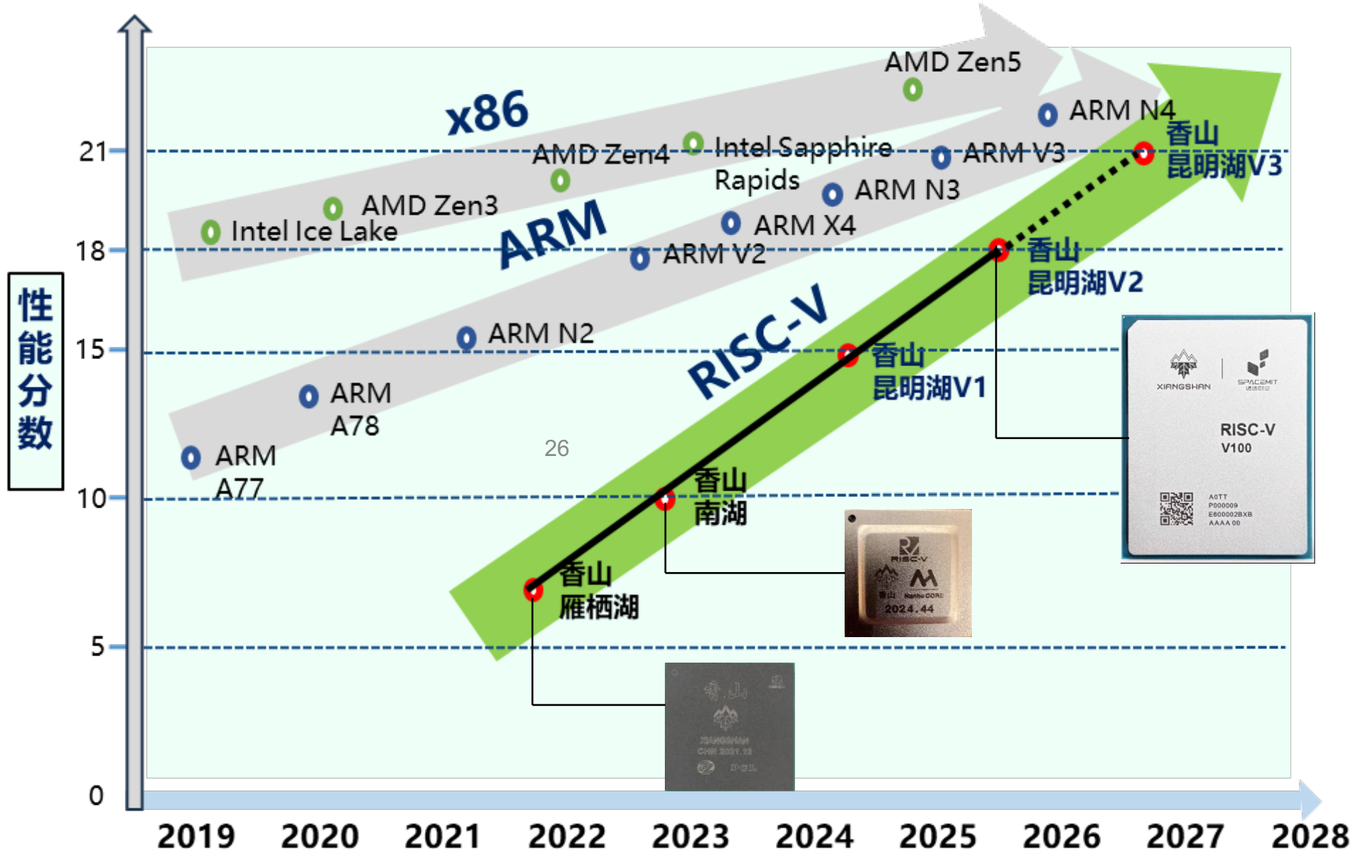
- **High performance**: develop an open-source processor that enters the top international tier of the RISC-V track
- **Fast iteration**: build agile chip-development infrastructure for faster iteration





Object-Oriented Architecture (OOA) Design Methodology

- Decouple a general-purpose processor into **objects**, use a high-level chip-design language (e.g. **Chisel**), and build an object library
- Enabling faster iteration of the XiangShan high-performance RISC-V core



Key Idea of OOA

Idea

Encapsulate circuit primitives and operations as hardware objects; design processor architecture using the MFC paradigm

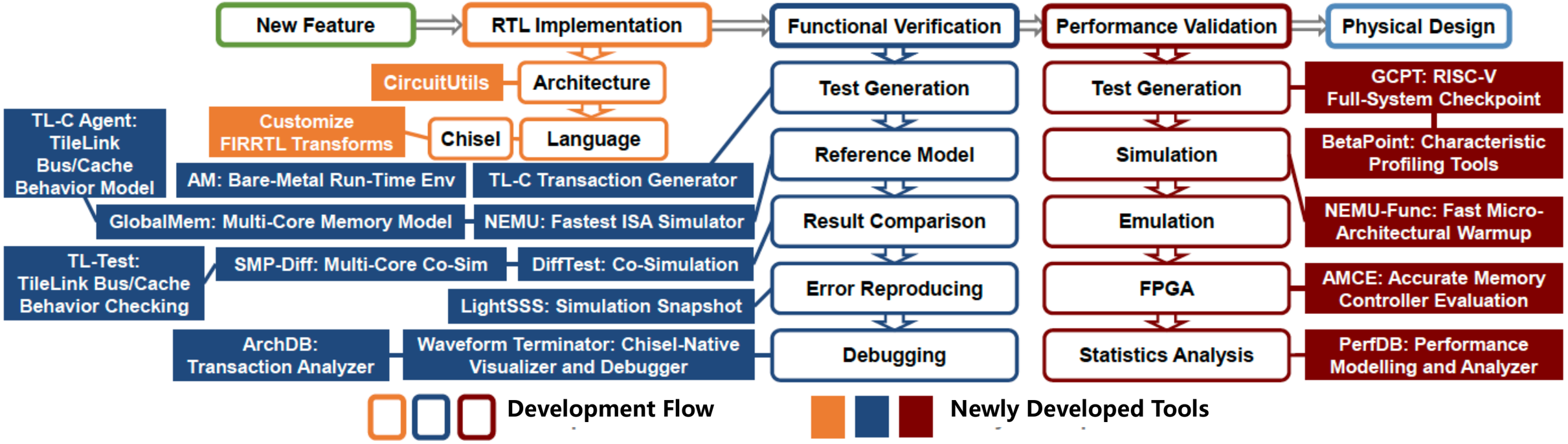
	Software Design	OOA Design
Base Language	C, C++	Chisel (Scala)
Primitives	int, float, char, function, class	**Wire, Reg, Module, etc.**
OOP Support	Encapsulation / inheritance / polymorphism	**Encapsulation / inheritance / polymorphism**
Design Pattern	MVC, MVVM, etc.	**Model-Flow-Controller (MFC)**
Template Library	C++ STL	**Template library for CPU and XPU**

Effect

- Defined **4,175 object types**, composed to build **4** XiangShan microarchitectures, with **>85%** code reuse
- Related papers published at **MICRO'22, DAC'24, ASPLOS'25, HPCA'26**, etc.

An OOA-Based Agile Processor-Development Platform

- 30+ newly developed tools form an agile chip-development platform

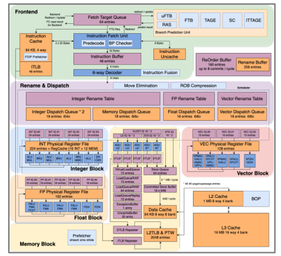


Case Study: Four XiangShan Microarchitectures with >85% Code Reuse

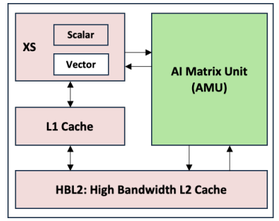
October 2024

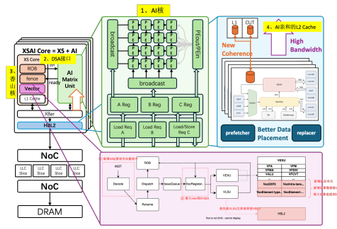
May 2025

XiangShan (KMH V2)
High-performance core

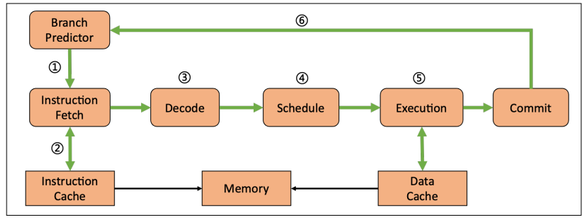


XiangShan (XSAI)
AI accelerator

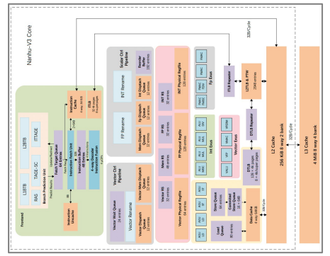




XiangShan (TraceRTL)
Trace-driven performance model



XiangShan (NH V5)
Efficient core



OOA Achieves 85% Code Reuse for XiangShan

- Based on a single set of OOA object descriptions, four XiangShan microarchitectures were designed

Submodule	KMH V2 LOC	Code Reuse Ratio (new lines)		
		NH V5	TraceRTL	XSAI
XiangShan	~134k	81.24%	94.43%	94.57%
YunSuan	~42k	98.25%	100%	100%
Utility	~7k	75.01%	89.90%	100%
CoupledL2	~21k	80.61%	99.28%	94.83%
huancun	~12k	94.17%	99.86%	100%
OpenLLC	~4k	93.61%	100%	100%
New code	—	(27.8k)	(7.5k)	(16.7k)
Total	**~220k**	**85.54%**	**96.29%**	**92.57%**

(2) How to Ensure Open-Source Chip Quality?

XIANGSHAN

Kunminghu performance benchmarks against ARM Neoverse N2
7nm@**3GHz**
SPECINT2006: 15 points/GHz

High-Performance

Chip Verification

Verification effort is enormous (>70%)

SOPHGO · Tencent 腾讯 · Alibaba · Baidu 百度 · GigaDevice
地平线 Horizon Robotics · ESWIN · 摩尔线程 MOORE THREADS · 中关村发展 ZGC GROUP · ThunderSoft
ZTE中兴 · 长城汽车 · 中科彼岸 ZHONG KE BI AN · HYSEIM · HX-DA 华思昇科

Enterprises can adopt with confidence

Product-Grade Verification Flow + **Agile Verification Platform** + **AI-Assisted Verification**

Measure #1: Industrial-Grade Verification Flow

Verification Process

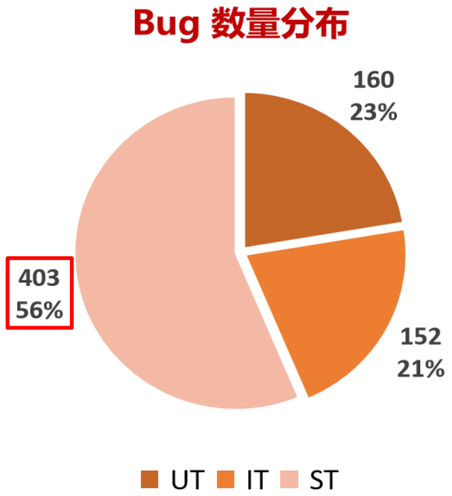
Design Spec → Verification Strategy → Planning → Test-Point Decomposition → Test-Case Writing → Verification Environment → Simulation Regression

Verification Strategy

Unit Test (UT) → Key Samples → ST Online → ST Signoff

UVM | VCS/Verilator | Palladium | FPGA

ST Fail cases → Bug reproduce → UT Analysis → Unit Test (UT)



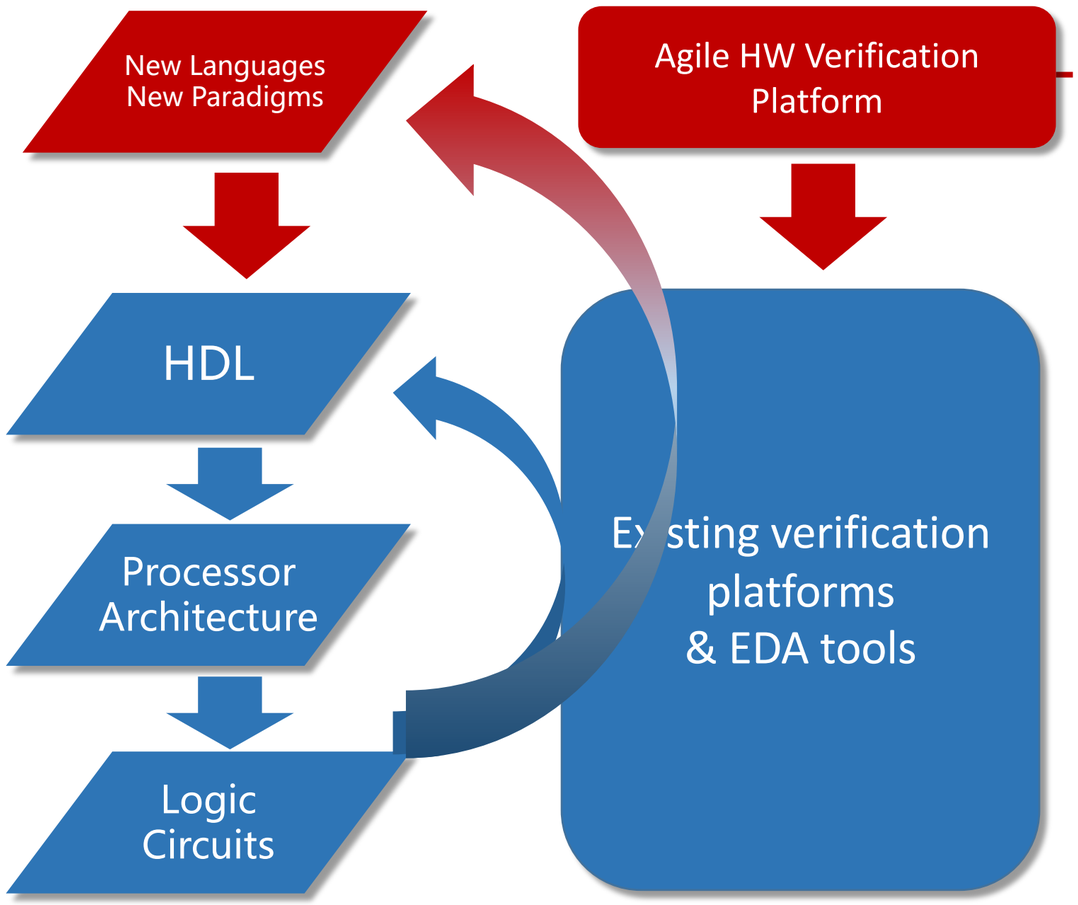
Verification Tools

- Commercial tools: FORCE-RISCV, RISC-V DV, Formal, Sting...
- Self-developed: DiffTest-H, GSIM, TL-test-new...

Open-Source Verification

- Partner companies: 492 bugs submitted — EpicSilicon 370, DiluTec 84, BlueChip 25, SOPHGO 13
- Crowdsourced verification supports multiple languages — 5 undergraduates found 10 bugs in two months

Measure #2: New Agile Verification Flow



① Functional Verification

- DiffTest
- Snapshot
- ArchDB
- LightSSS
- WT

② Performance

- Chisel-based prototypes
- RISC-V Checkpoint
- XSPerf
- Functional Warmup
- Top-down analysis

VERILATOR

umd-memsys/
DRAMsim3

DRAMsim3: a Cycle-accurate, Thermal-Capable DRAM Simulator

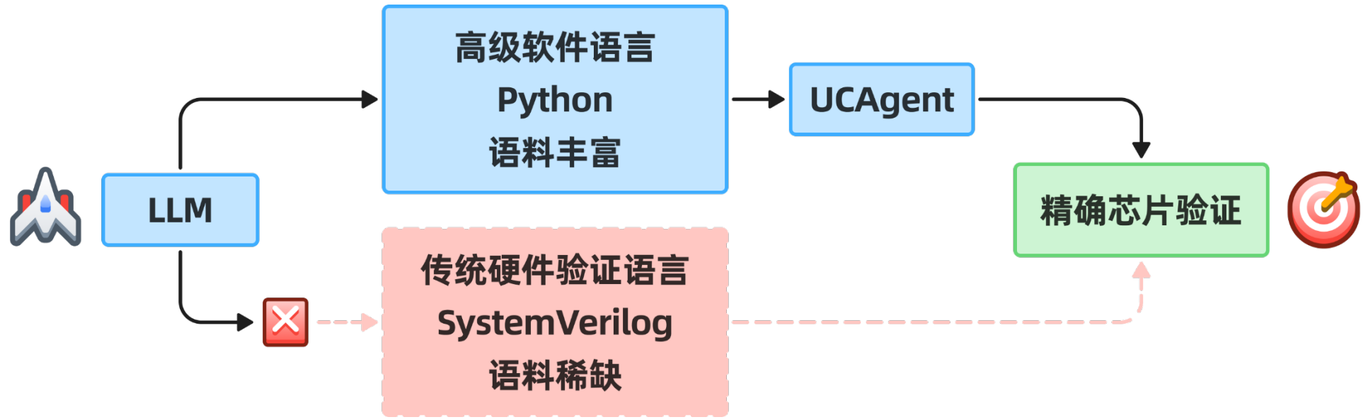
SimPoint ……

Chip Design **Chip Verification**

Measure #3: LLM-Based Chip Verification — UCAgent

Idea

Proposed a method of describing hardware timing behavior using high-level software languages (e.g. Python), enabling LLMs to accurately generate **Python-based** hardware test cases

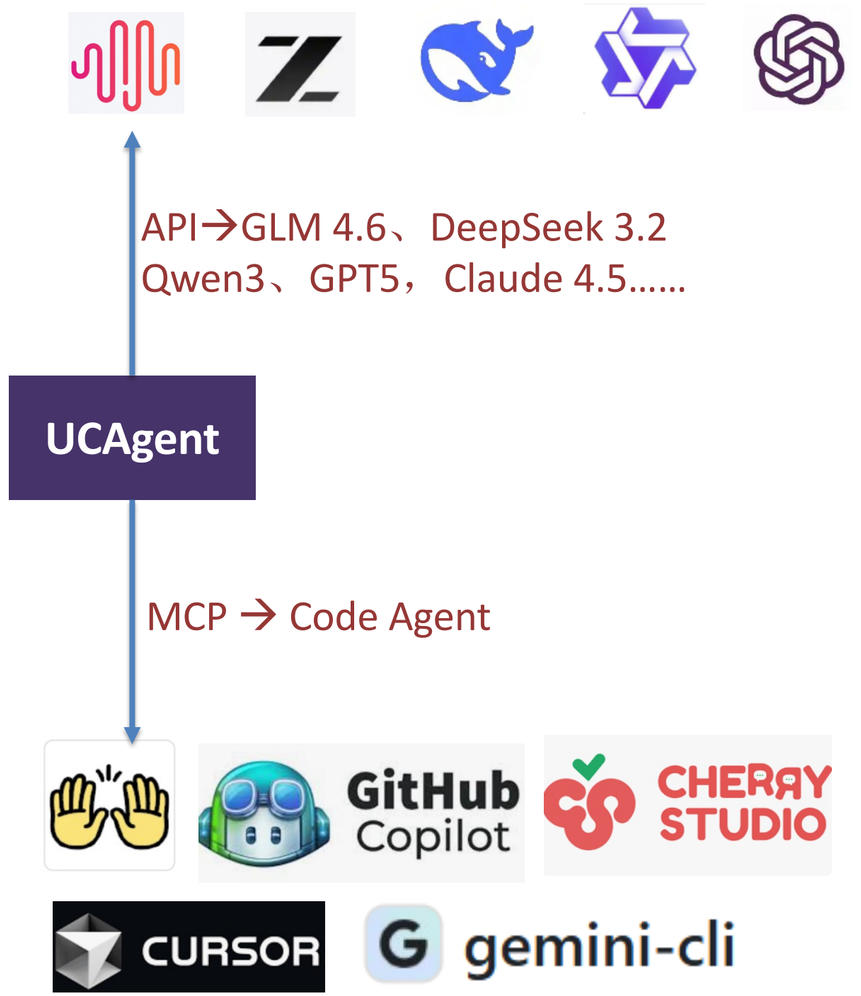


Effect

- UCAgent, the chip-verification intelligent agent, improves verification efficiency >10×
- Supports automated unit testing (UT) verification for XiangShan; paper accepted at ISCA'26
- UCAgent has already been deployed in several chip companies

UCAgent: Unit-Test (UT) Automation

- Fully automated verification for up to **5,000 lines** of RTL — test-case generation, documentation, reports, etc.
- **10×** verification efficiency improvement



One module in an AI vector processor

- **1,283 lines of Verilog**
- **200+ test cases auto-generated**
- **Bugs confirmed: 3**

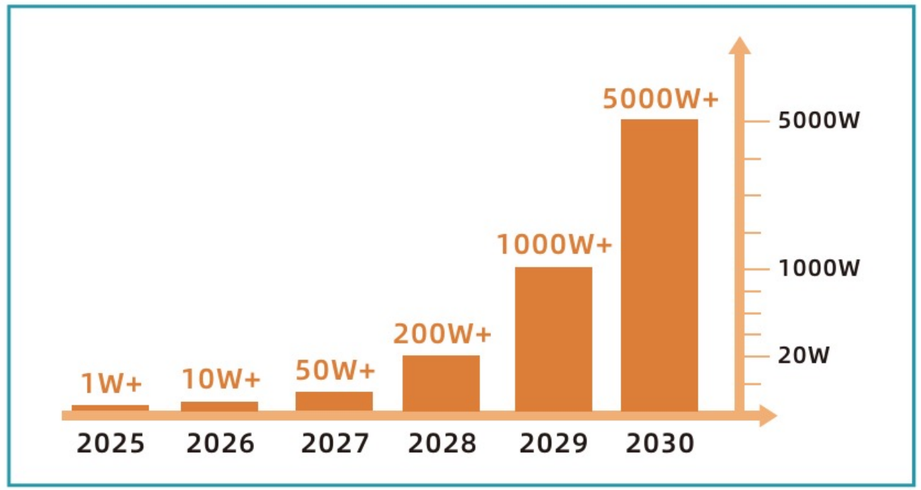
UCAgent vs. Human

5 hours vs. 15+ days

Open-source: https://github.com/XS-MLVP/UCAgent

Agile Verification Enables Mass Deployment

- **14 companies** are actively adopting the platform, and total product shipments have reached **50,000 units**.



2025 - 2030 Shipment Forecast

XiangShan·KMH

LANXIN COMPUTING
LX500 RISC-V+AI CPU

SpacemiT
V100: Server- Grade Chip Platform

XiangShan·NH

Innosilicon
Fantasy 3: Full-Function GPU

Welcome to the

BOSC Booth!

Thanks!