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

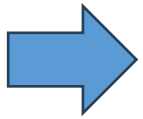
XiangShan KMH:

An Open Source RISC-V Core with >15/GHz for SPECCPU2006

Yungang Bao

May 14th, 2025

Outline



- **Part I:** The XiangShan Project
- **Part II:** Agile Chip Verification

RISC-V Brings New Opportunities

Prof. David Patterson
UC Berkeley

Need Free & Open Specification
To Have Free & Open Designs

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

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

XIANGSHAN

The XiangShan Project

- **Goal**: building an **open-source high-performance** RISC-V core, which is widely adopted by both the industry and the research community

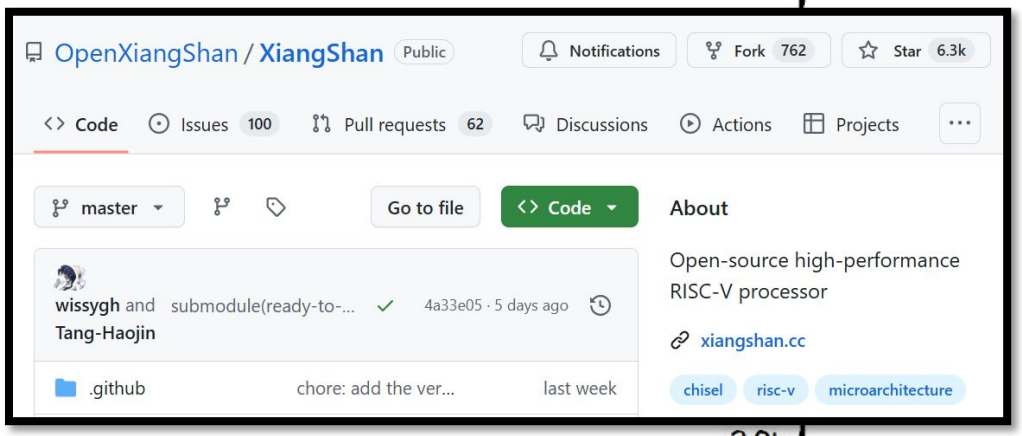
Linux

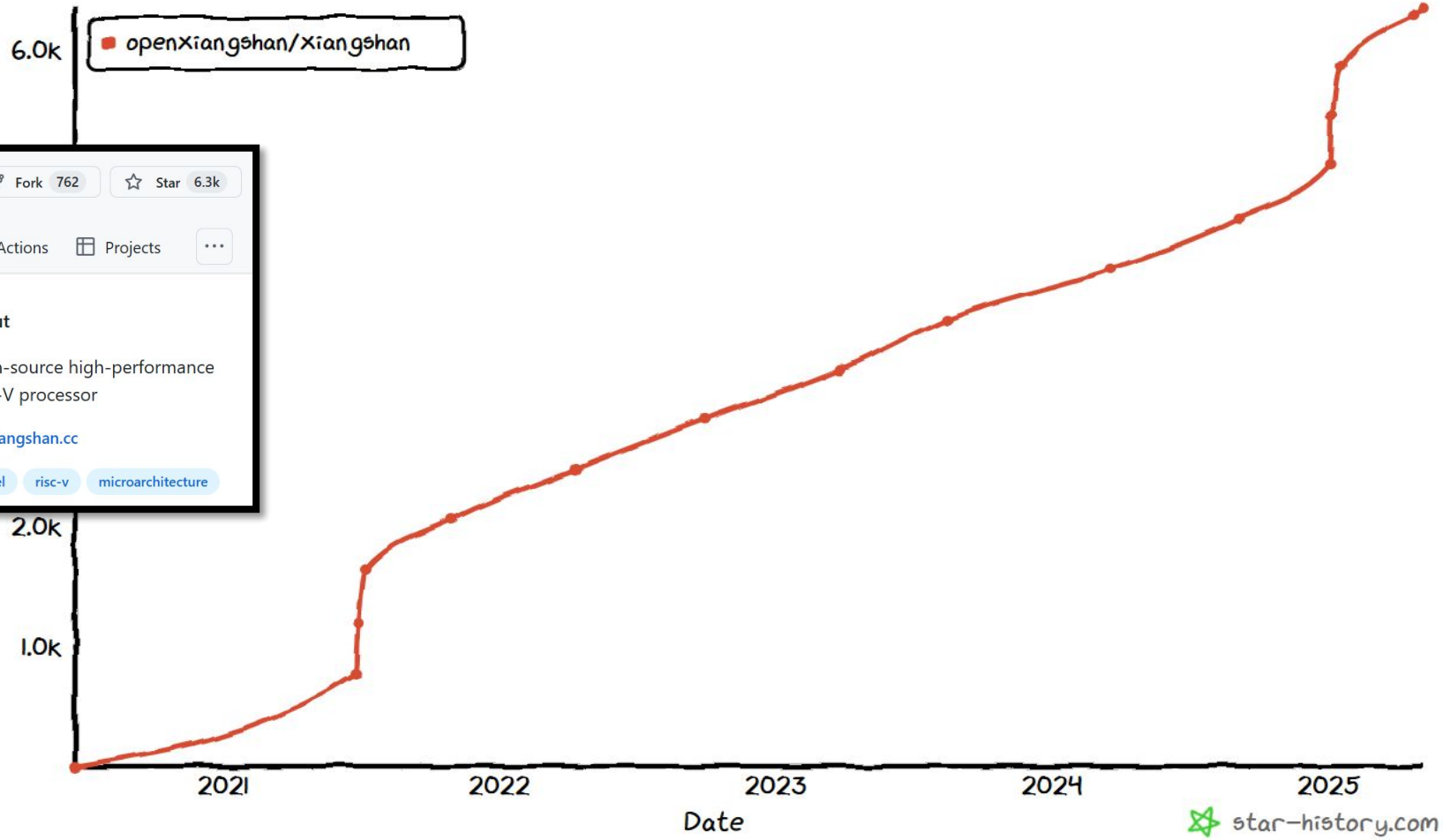
v.

s.

XIANGSHAN

One of the most active open-source chip projects





Data collected on May 6, 2025

Roadmap of Two-tier Cores

Kunminghu (KMH) Architecture

- High performance
- Targeting server/data-center segment

- RVA-23 profile
- Leading RISC-V feature sets (H/V ext.)

Versus ARM Neoverse N2

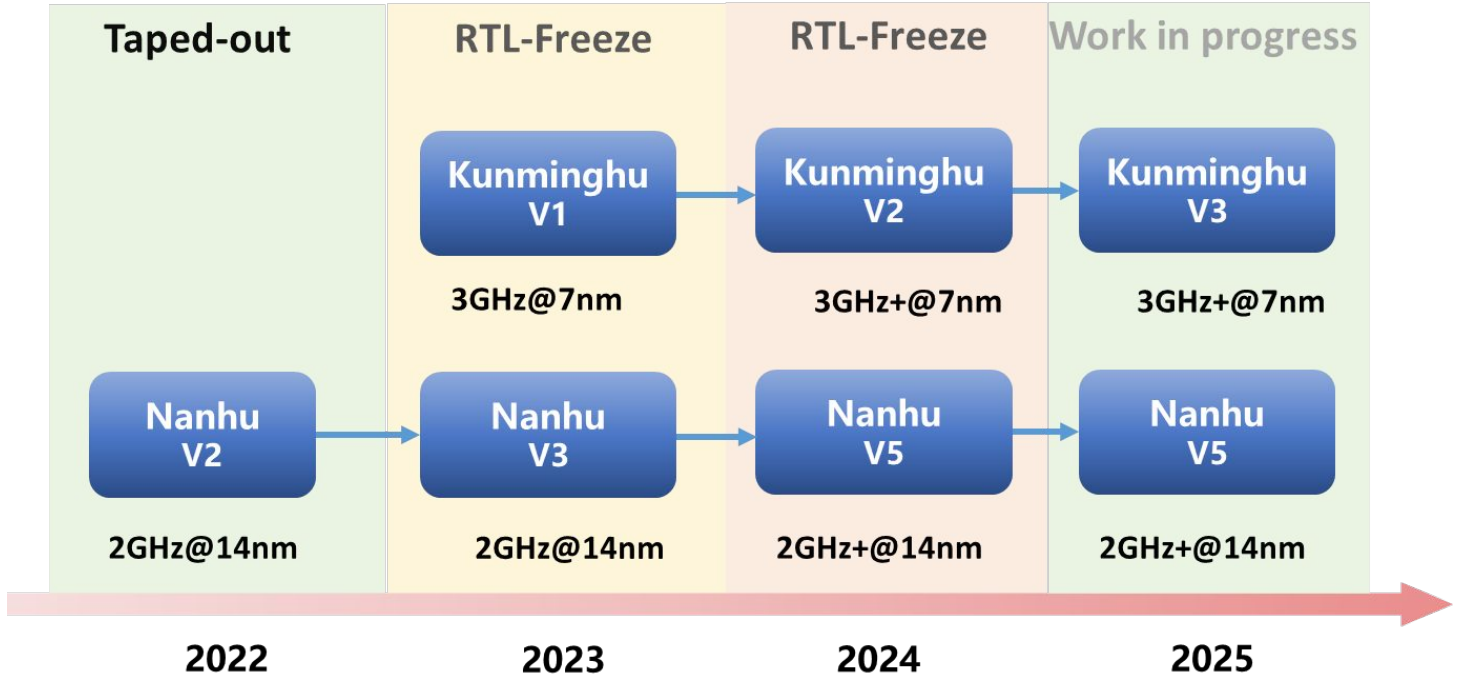
Nanhu (NH) Architecture

- Power/area efficiency
- Targeting industry-control segment

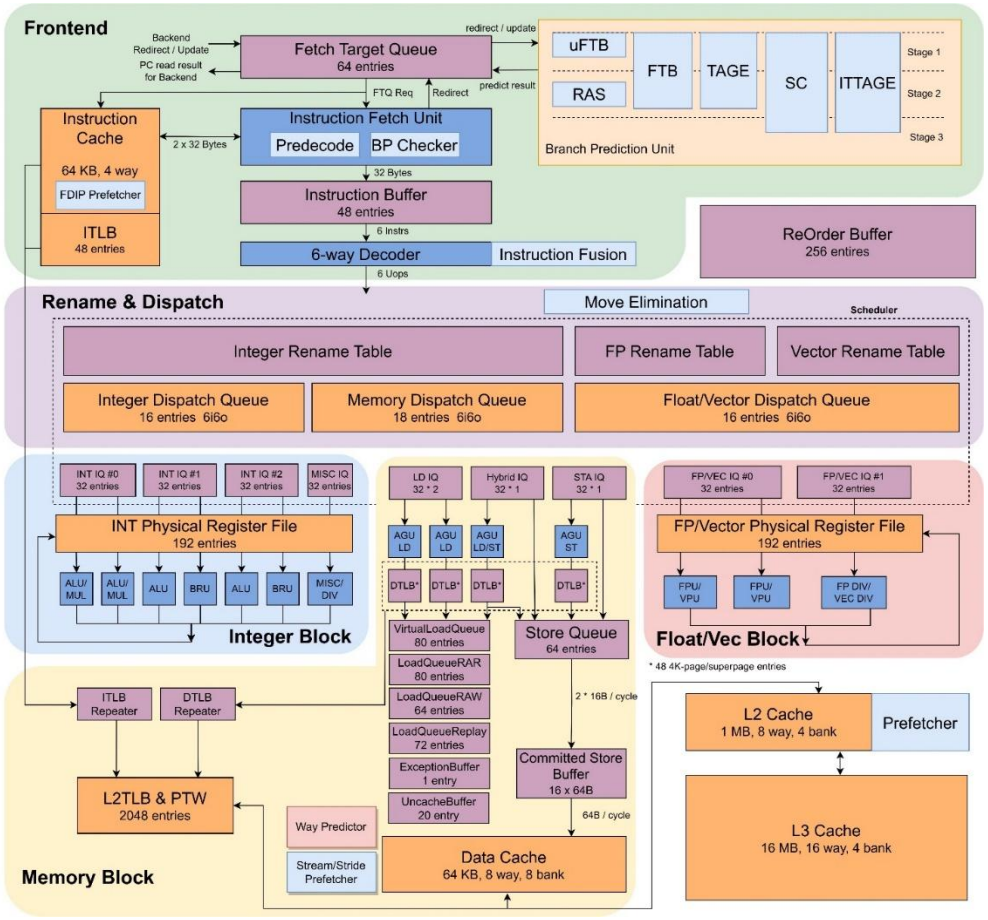
- RVA-20 profile
- Taped out and tested

Versus ARM Cortex A76

- **<u>Industrial-grade</u> µarch design and development workflow**
- **<u>Highly configurable</u> with agile development methodology**
  - Written in Chisel, delivered in both Chisel and Verilog



µArch Overview of KMH Architecture



[HotChips'24]

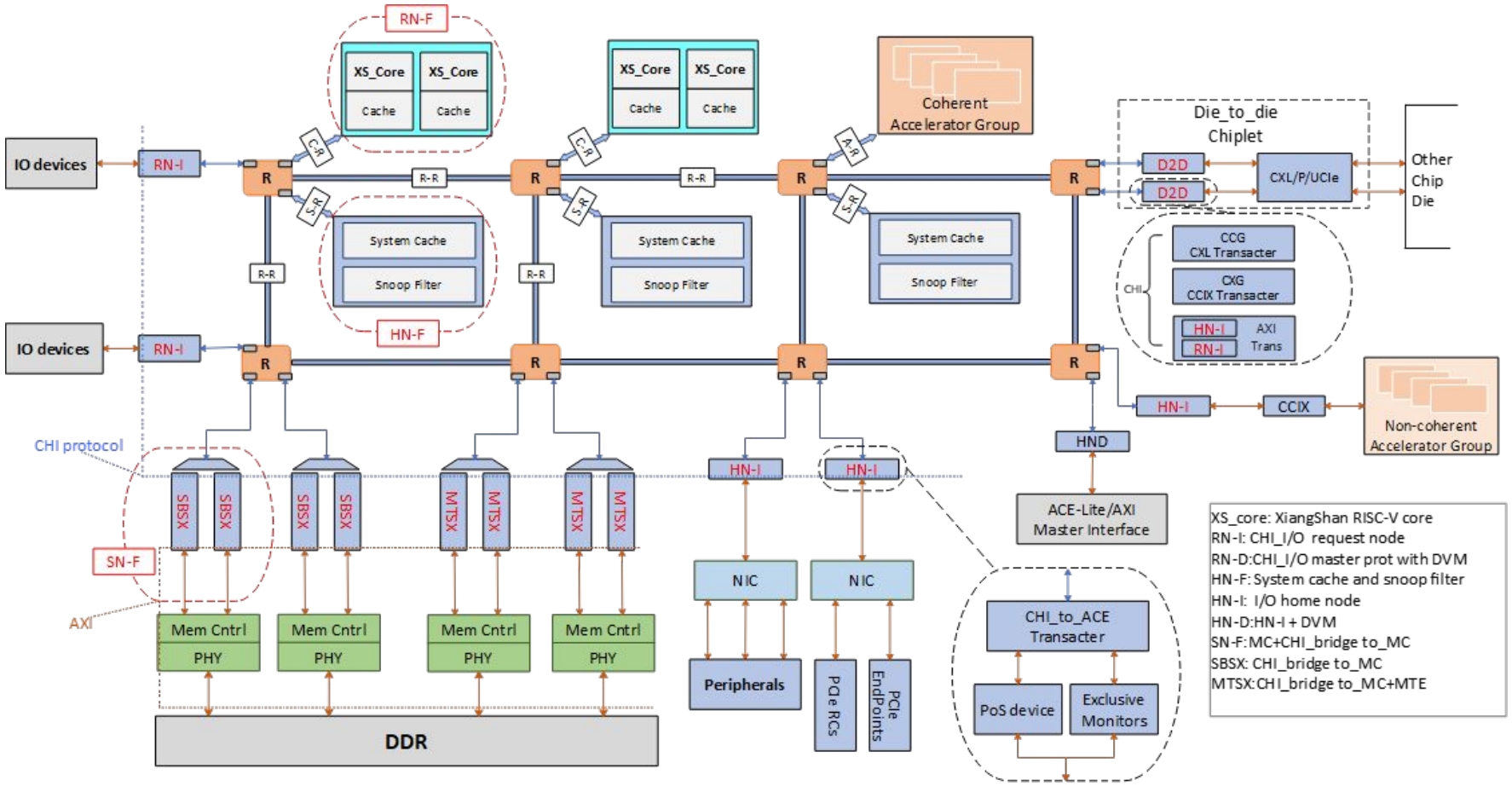
Comparison between Kunminghu & Nanhu

Feature	Kunminghu	Neoverse N2	Nanhu	Cortex A76
Pipeline depth	13	10	13	13
Rename width	6	5	4	4
Rename checkpoint	Y	Y	N	N
ROB size	160 (x6)	160+	192	128
ALUs	4	4	4	3
L1 instruction cache	64KB	64KB	64KB	64KB
L1 data cache	64KB	64KB	64KB	64KB
L2 cache	1024KB	512/1024KB	256KB	256/512KB
L3 cache	Up to 16MB	4MB per slice	Up to 4MB	Up to 4MB
NoC support	Y	Y	N	N
L2 outstanding txns	64	64	32	46
ITLB	48	48	32	48
DTLB	48	44	128 direct mapped	48
L2 TLB	2048	1280	2048	1280
Vector	Y	Y	Y	Y
Virtualization	Y	Y	N	Y
ECC support	Y	Y	Y	Y
PMA/PMP support	Y	Y	Y	Y
Debug support	Y	Y	Y	Y
External interface	AXI4/TL/CHI	AXI4/CHI	AXI4/TL	AXI4/CHI

Competitive

Need optimization

WenYuHe (WYH): NoC IP for RISC-V server and AI Chips



Items	Params
Mesh Frequency (P&R)	2.6 GHz @7nm_SS
Maximum Bidirection Bandwith	3.99TB/s @512b
Mesh Scale	up to 12x12
Router Delay	<2ns
Router Area	0.041 mm^2
Router Power	27.8 mW
Router Type	Input buffered Output Mux
Router Port	4 Direction+ 4 Device
Routing Strategy	Pseudo one-off
Package/Flit	1~N
Maximum Wire Number	1440
Routing Algorithm	XY
Topology	mesh

- KMHv2+WYHv2 SoC: 2/4/8 cores emu/FPGA test (passed), 16/32 cores (WIP)
- NoC mesh already supports multi-flit routing and large data package multicasting
- WIP: D2D devices

XiangShan KMHv2 Performance w/ NoC

- SimPoint-Based Program Sampling Evaluation
  - Compiler：GCC 12 -O3，RV64GCB，jemalloc
  - Cache：64KB I/D + 1MB L2
  - 3200 DDR4 *2
  - 3x5 Mesh + 2MB LLC * 8

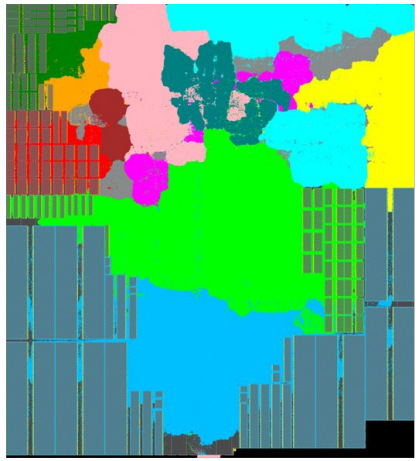
Base Score	SPECint 2006	SPECfp 2006
KMHv2 @3GHz	44.1	43.86

SPECint 2006 est. / GHz		SPECfp 2006 est. / GHz	
400.perlbench	11.959	410.bwaves	21.816
401.bzip2	8.369	416.gamess	13.639
403.gcc	15.42	433.milc	13.191
429.mcf	20.23	434.zeusmp	17.101
445.gobmk	10.038	435.gromacs	11.213
456.hmmer	13.85	436.cactusADM	15.231
458.sjeng	10.236	437.leslie3d	15.482
462.libquantum	39.204	444.namd	9.623
464.h264ref	18.823	447.dealII	24.408
471.omnetpp	13.499	450.soplex	17.172
473.astar	9.688	453.povray	17.808
483.xalancbmk	23.78	454.Calculix	5.459
GEOMEAN	**14.7**	459.GemsFDTD	11.647
		465.tonto	12.21
		470.lbm	29.892
		481.wrf	13.601
		482.sphinx3	15.763
		GEOMEAN	**14.62**

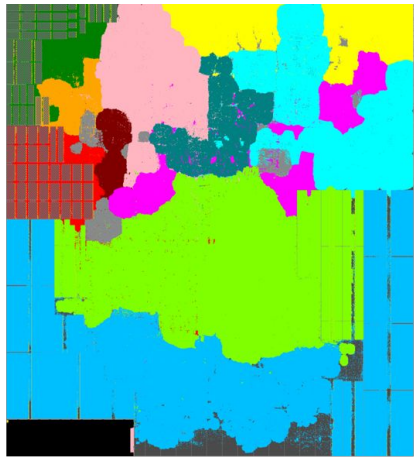
Tape-out Status

- **NHv2 has been taped out**
  - 2.5GHz, SPECCPU 2006 ~10/GHz

- **KMHv2: Ready to tape-out**

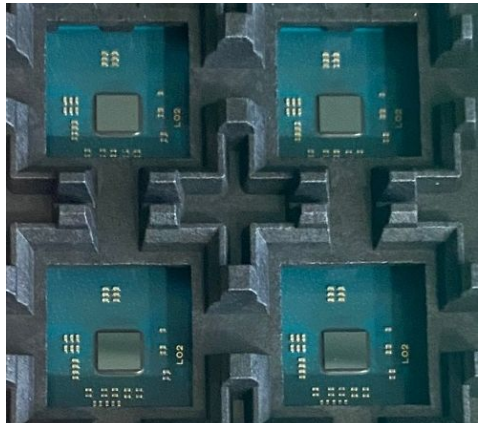


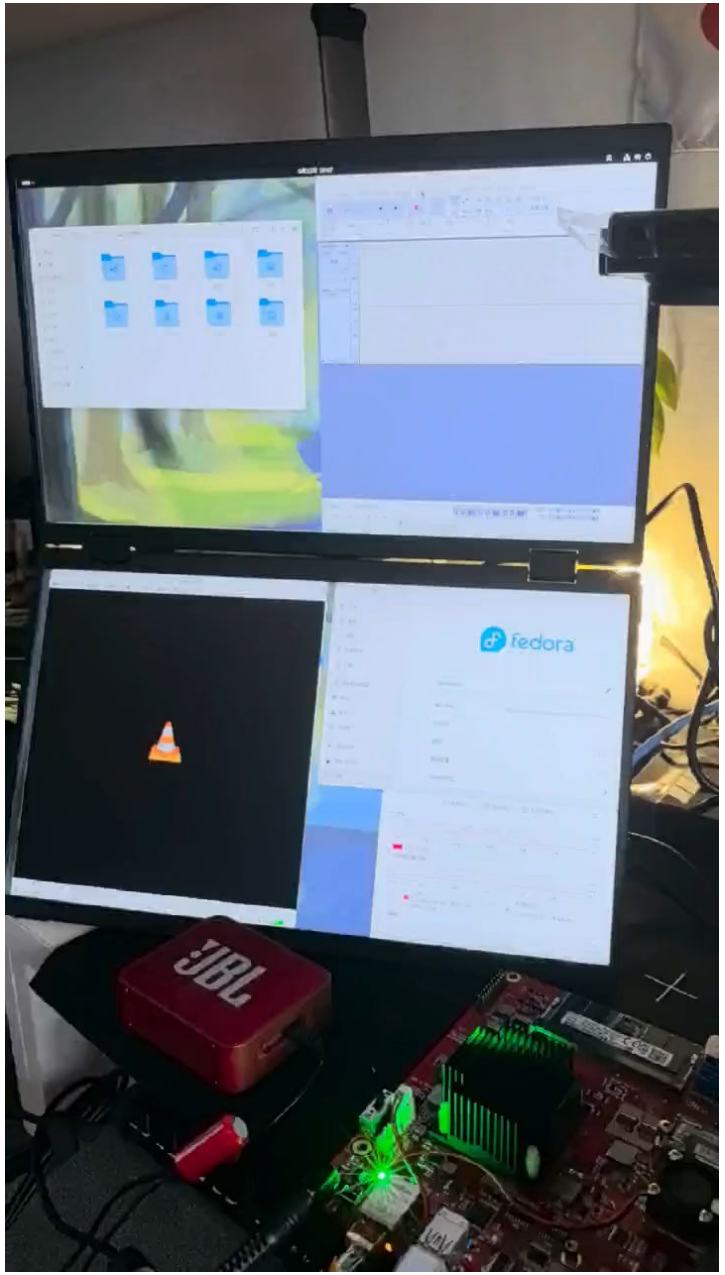
On 7nm node:
1.6-1.9W (core+1M L2)
1.8-2.1mm^2 (core + 1M L2)
Max core Freq 3.0Ghz



On 7nm node:
1.6-1.9W (core+512KB L2)
1.6-1.8mm2 (core+512KB L2)
Max core Freq 3.0Ghz

NHv2 Test Chip Demo





KMHv3 under development (over KMHv2)

- **~30% ↑ for SPEC06 Benchmark**

==========	specInt06(1.0c)		
	KMH_V2 RTL	KMHV3(c9bd2b3_2025.4.25)	perf_gain
perlbench	11.959	14.777	23.56%
bzip2	8.515	10.123	18.88%
gcc	15.555	20.410	31.21%
mcf	19.376	27.646	42.68%
gobmk	10.114	17.378	71.82%
hmmer	13.867	16.854	21.54%
sjeng	10.167	14.141	39.09%
libquantum	40.857	40.674	-0.45%
h264ref	18.886	20.433	8.19%
omnetpp	13.118	22.506	71.57%
astar	9.747	12.888	32.23%
xalancbmk	24.004	33.826	40.92%
score/GHz	14.722	**19.405**	31.81%
total_score@3GHz	**44.165**	**58.215**	31.81%

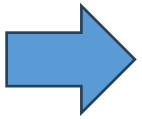
==========	specFp06		
	KMH_V2	KMHV3(c9bd2b3_2025.4.25)	perf_gain
bwaves	22.206	35.311	59.02%
gamess	13.636	17.003	24.69%
milc	15.071	22.282	47.85%
zeusmp	17.319	18.334	5.86%
gromacs	11.204	13.700	22.28%
cactusADM	15.420	20.795	34.86%
leslie3d	15.365	18.016	17.25%
namd	9.636	11.078	14.96%
dealII	24.562	30.306	23.39%
soplex	17.333	25.566	47.50%
povray	17.805	24.736	38.93%
calculix	5.459	7.277	33.30%
GemsFDTD	12.004	21.334	77.72%
tonto	12.247	14.674	19.82%
lbm	30.392	34.380	13.12%
wrf	13.540	15.488	14.39%
sphinx3	16.203	17.185	6.06%
score/GHz	14.839	**19.019**	28.17%
total_score@3GHz	**44.517**	**57.056**	28.17%

- **~33% ↑ for SPEC17 Benchmark**

==========	specInt17		
	KMH_V2	KMHV3(c9bd2b3_2025.4.25)	perf_gain
perlbench	1.483	2.152	45.11%
gcc	1.803	2.899	60.79%
mcf	1.806	2.417	33.83%
omnetpp	1.269	1.888	48.78%
xalancbmk	1.630	2.738	67.98%
x264	1.667	1.919	15.12%
deepsjeng	1.174	1.610	37.14%
leela	1.127	1.425	26.44%
exchange2	3.877	4.488	15.76%
xz	0.975	1.213	24.41%
score/GHz	1.556	**2.124**	36.49%
total_score@3GHz	4.668	**6.371**	36.49%

==========	specFp17		
	KMH_V2	KMHV3(c9bd2b3_2025.4.25)	perf_gain
bwaves	8.680	9.475	9.16%
cactuBSSN	1.441	1.651	14.57%
namd	1.966	2.141	8.90%
parest	1.682	1.893	12.54%
povray	1.796	2.587	44.04%
lbm	2.275	2.485	9.23%
wrf	1.490	1.751	17.52%
blender	1.724	2.352	36.43%
cam4	1.643	2.185	32.99%
imagick	1.011	1.701	68.25%
nab	0.844	2.152	154.98%
fotonik3d	6.313	7.230	14.53%
roms	1.924	2.355	22.40%
score/GHz	1.984	**2.584**	30.22%
total_score@3GHz	**5.952**	**7.751**	30.22%

Outline



- **Part I:** The XiangShan Project
- **Part II:** Agile Chip Verification

Obstacles between XiangShan and industrial deployment

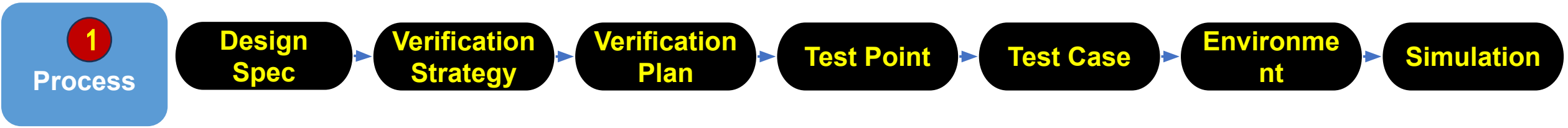
XIANGSHAN

High-Performance CPU Core

Chip Verification

Ready for Companies

Build a rigorous verification framework to ensure quality



UVM EDA

DIFFTEST

SPIKE EMU

Forceriscv

FPGA

Riscv-Dv

F Formal

Business tools

UT

IT

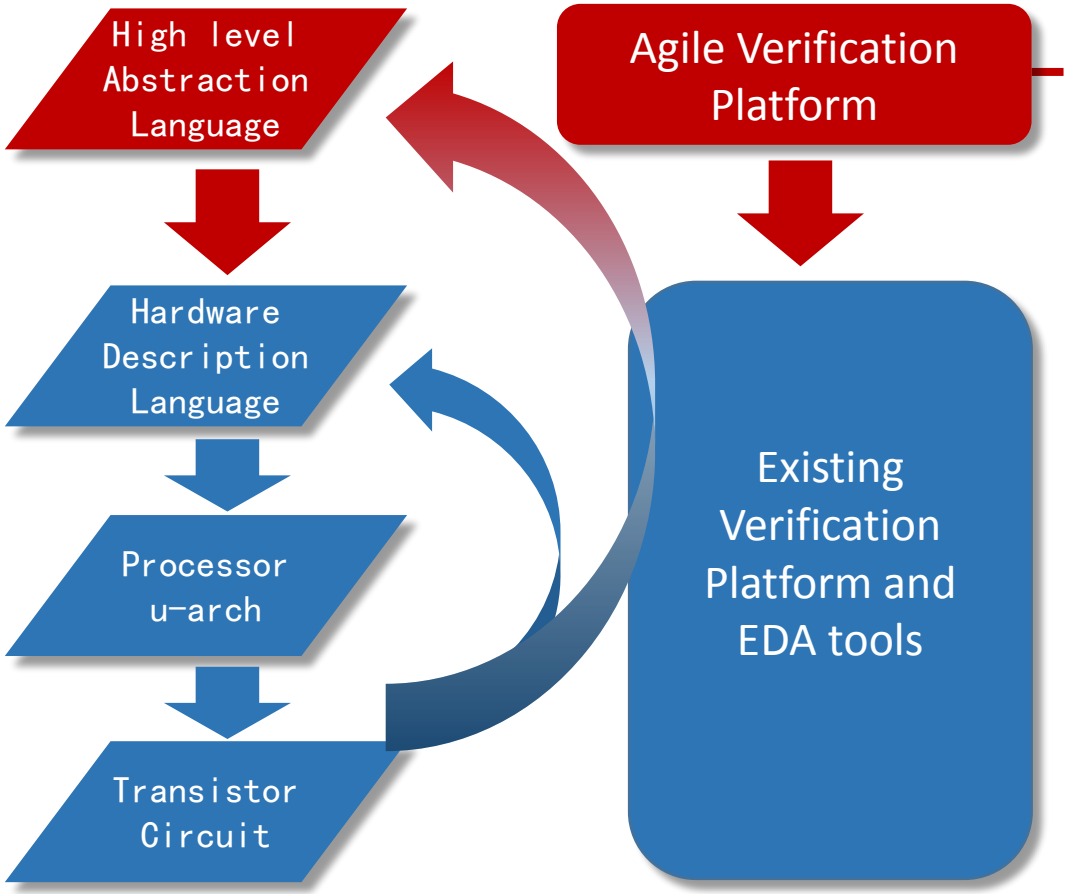
ST

Prototype

FrontEnd	BackEnd		MemBlock	L2C
BPU	CtrlBlock	IntExu (F)	LSU	L2Cache
IFU		FpExu (F)	Dcache	Noc
Icache		VfExu (F)	PTW	
ITLB			prefetch	

Single-core scenario	Multi-core features	Program slicing	IO testing	Others	
Multi-core scenario	OS	Benchmark	IO testing	Other applications	

Agile Verification Platform (1)



Design Flow

Verification Flow

① Functional verification

difftest

Snapshot

ArchDB

LightSSS

WT

② Performance verification

Chisel-based prototypes

RISC-V Checkpoint

XSPerf

Functional Warmup

Top-down analysis

VERILATOR

umd-memsys/ DRAMsim3

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

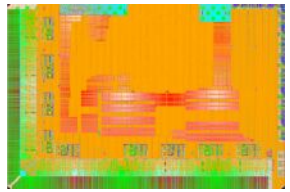
SimPoint

.....

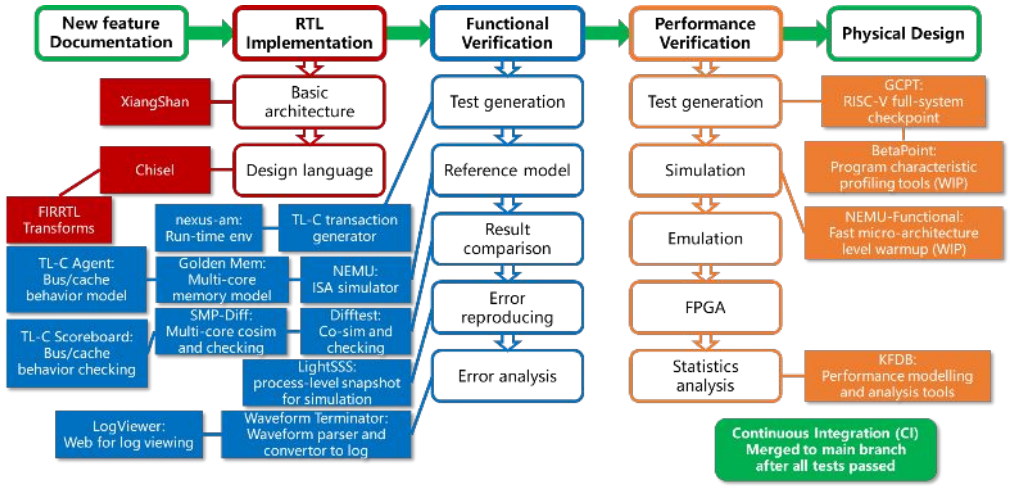
Agile hardware verification tools

Agile Verification Platform (2)

- Consist of **20+** new open-sourced tools (IEEE Micro **Top Picks** 2022)



Open-Source Design

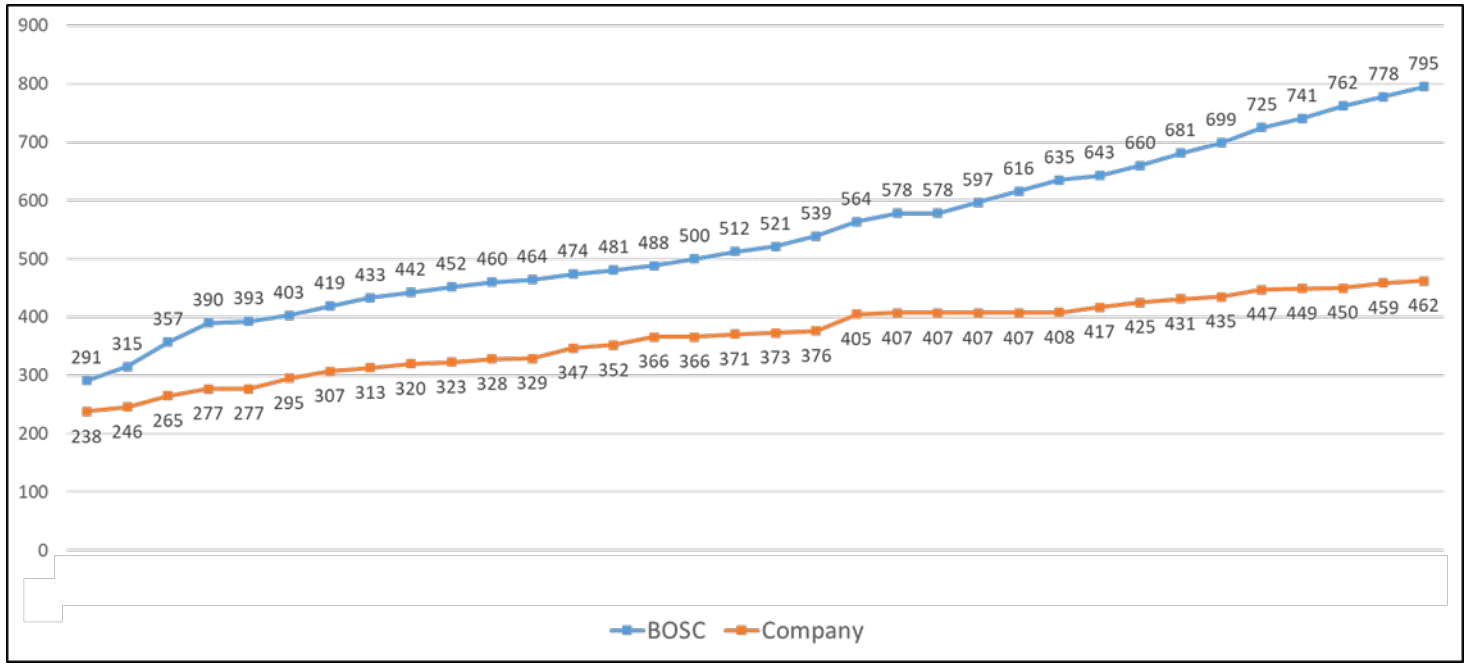


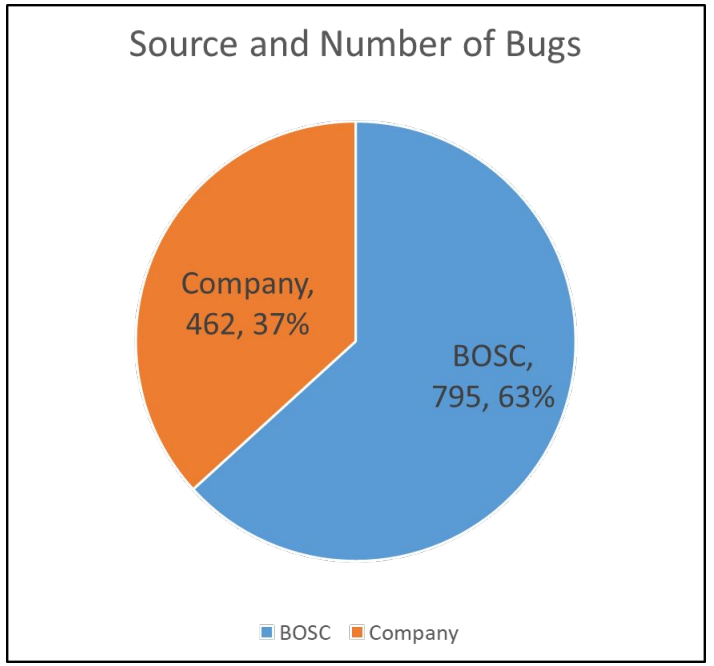
Open-Source Tools

[Micro'22][IEEE Micro Top Picks 2022] [DAC'24][ASPLOS'24]

Leverage Test Cases of Partners via Open Source

- The test cases are the companies' private assets and are not suitable for sharing. But XiangShan is open sourced and can be **deployed and verified** by many companies together
  - E.g., companies contributed **462 bugs (37%)** for KMH's development

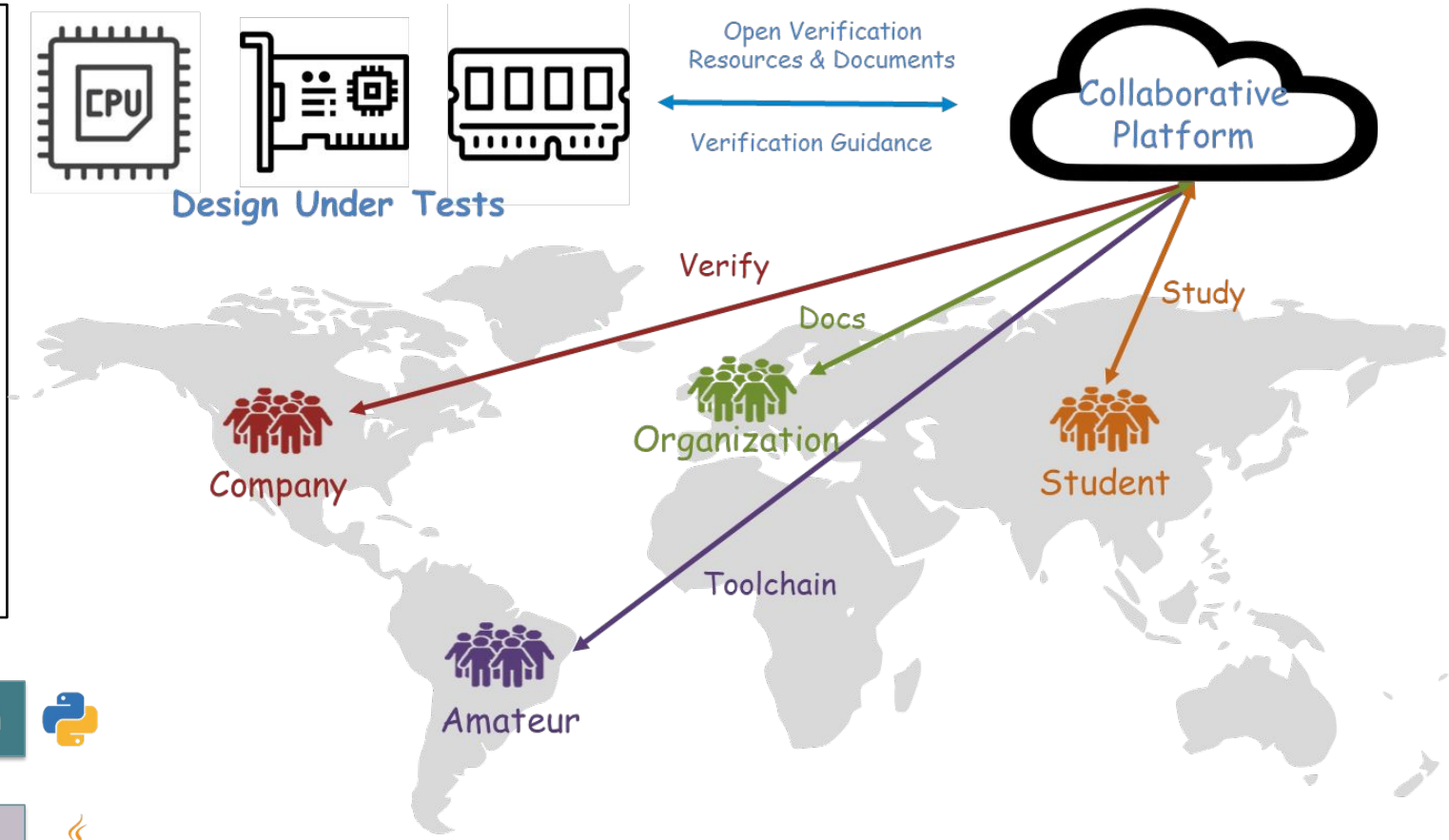


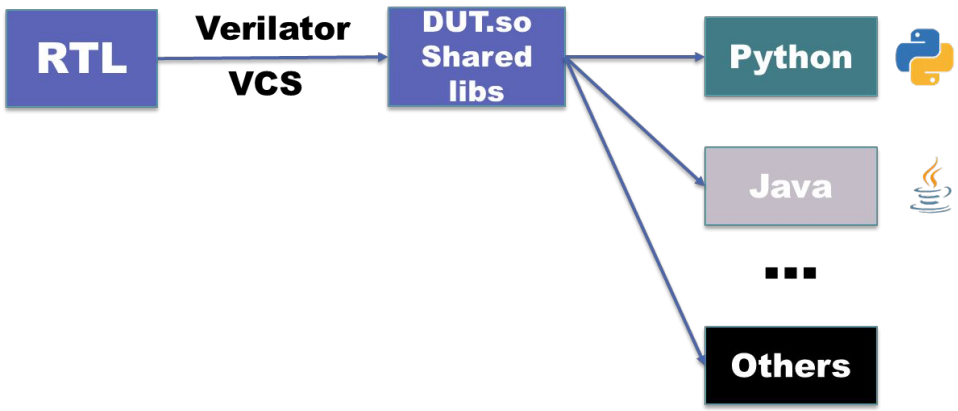


UnityChip : A Croudsourcing Platform for Chip Verification

Features:

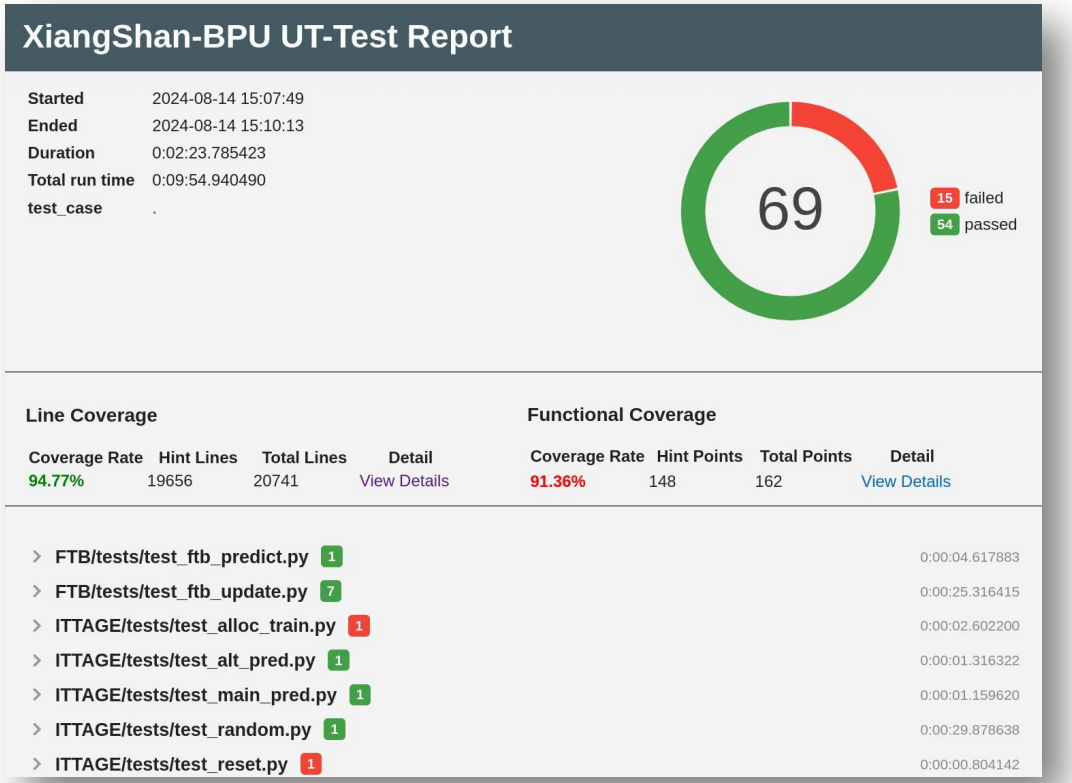
- Support crowdsourcing
- Based on cloud
- Support multi-languages
- Be compatible with UVM
- **Involve software engineers**





Evaluation

- **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 BPU and iCache
  - **285** teams participated in the competition



Acknowledgments

- A list outlines 33 techniques used in the XiangShan RTL codes
  - https://docs.xiangshan.cc/acknowledgments/

Instruction Prefetch

[1] Glenn Reinman, Brad Calder, and Todd Austin. "Fetch directed instruction prefetching." MICRO-32. Proceedings of the 32nd Annual ACM/IEEE International Symposium on Microarchitecture. IEEE, 1999. [RTL Codes]

[2] Yasuo Ishii, Jaekyu Lee, Krishnendra Nathella, and Dam Sunwoo. "Rebasing instruction prefetching: An industry perspective." IEEE Computer Architecture Letters 19.2 (2020): 147-150. [RTL Codes]

[3] Yasuo Ishii, Jaekyu Lee, Krishnendra Nathella, and Dam Sunwoo. "Re-establishing fetch-directed instruction prefetching: An industry perspective." 2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). IEEE, 2021. [RTL Codes]

Branch prediction

[1] Kevin Skadron, P. S. Ahuja, Margaret Martonosi, and D. W. Clark. "Improving prediction for procedure returns with return-address-stack repair mechanisms." Proceedings. 31st Annual ACM/IEEE International Symposium on Microarchitecture. IEEE, 1998. [RTL Codes]

[2] Pierre Michaud. "A PPM-like, tag-based branch predictor." The Journal of Instruction-Level Parallelism 7 (2005): 10. [RTL Codes]

[3] André Seznec, and Pierre Michaud. "A case for (partially) tagged geometric history length branch prediction." The Journal of Instruction-Level Parallelism 8 (2006): 23. [RTL Codes]

[4] André Seznec. "A 256 kbits l-tage branch predictor." Journal of Instruction-Level Parallelism (JILP) Special Issue: The Second Championship Branch Prediction Competition (CBP-2) 9 (2007): 1-6. [RTL Codes]

XiangShan / src / main / scala / xiangshan / frontend / newRAS.scala

eastonman style(frontend): manually wrap some line (#... c3d62b6 · 2 weeks ago History

Code Blame 782 lines (690 loc) · 28.3 KB · Raw

```
/***************************************************************************************
* Copyright (c) 2024 Beijing Institute of Open Source Chip (BOSC)
* Copyright (c) 2020-2024 Institute of Computing Technology, Chinese Academy of Scienc
* 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.
***************************************************************************************/
package xiangshan.frontend

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

class RASEntry()(implicit p: Parameters) extends XSBundle {
  val retAddr = UInt(VAddrBits.W)
  val ctr     = UInt(RasCtrSize.W) // layer of nested call functions
  def =/=(that: RASEntry) = this.retAddr =/= that.retAddr || this.ctr =/= that.ctr
}
```

Open-access Resources

- **Contents**
  - Introduction to the microarchitecture design of XiangShan processor
  - Introduction to the infrastructures for XiangShan development
  - Hands-on development with typical use cases on XiangShan and MinJie

- **Open-access materials**
  - Presentations slides
    - https://github.com/OpenXiangShan/XiangShan-doc/tree/main/tutorial
  - Demonstration environment
    - https://github.com/OpenXiangShan/xs-env/releases


北京开源芯片研究院
BEIJING INSTITUTE OF OPEN SOURCE CHIP
Open Source Chip
Together for a Shared World

Thanks !