



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



中国开放指令生态 (RISC-V) 联盟
China RISC-V Alliance

XiangShan: An Open-Source High-Performance RISC-V Processor and Infrastructure for Architecture Research

The XiangShan Team

ASPLOS'24@San Diego, USA

April 27, 2024

Tutorial@ASPLOS'24 Schedule

Time (PM)	Topic
1:30 - 2:00	Introduction of the XiangShan Project
2:05 - 3:00	Microarchitecture Design and Implementation
	Afternoon Break
3:30 – 5:00	Hands-on Development & Discussions

Together for a shared future

- HPCA'24 @ Edinburgh, Scotland
- MICRO'23 @ Toronto, Canada
- ASPLOS'23 @ Vancouver, Canada
- Our official document is continuously updating
- Close connections with open-source community
- Feel free to contact us through email or file issues on GitHub!
 - xiangshan-all@ict.ac.cn
 - <https://github.com/OpenXiangShan/XiangShan>



XiangShan Document

Part I

The Era of Open-Source Chip





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.

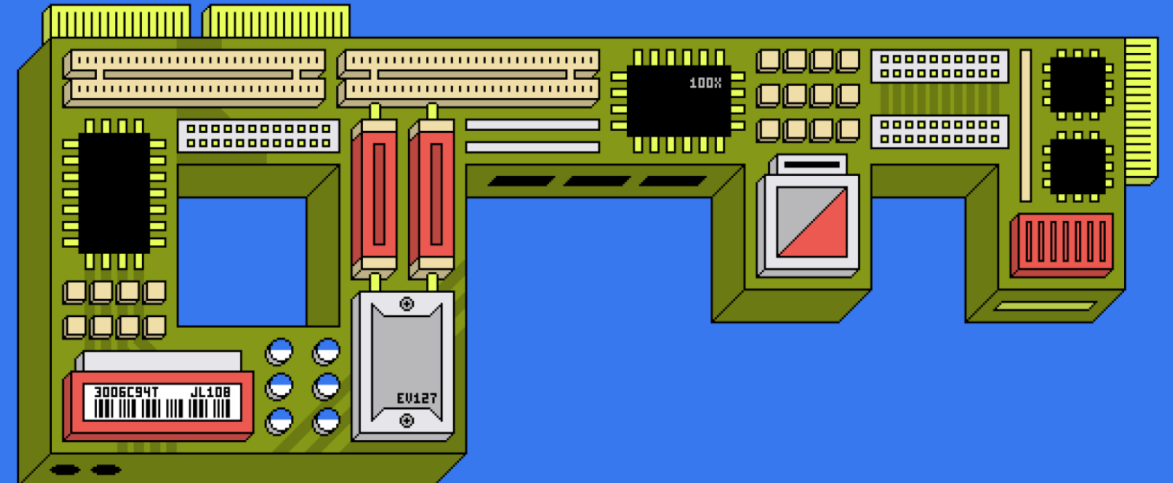
--- MIT Technology Review

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.

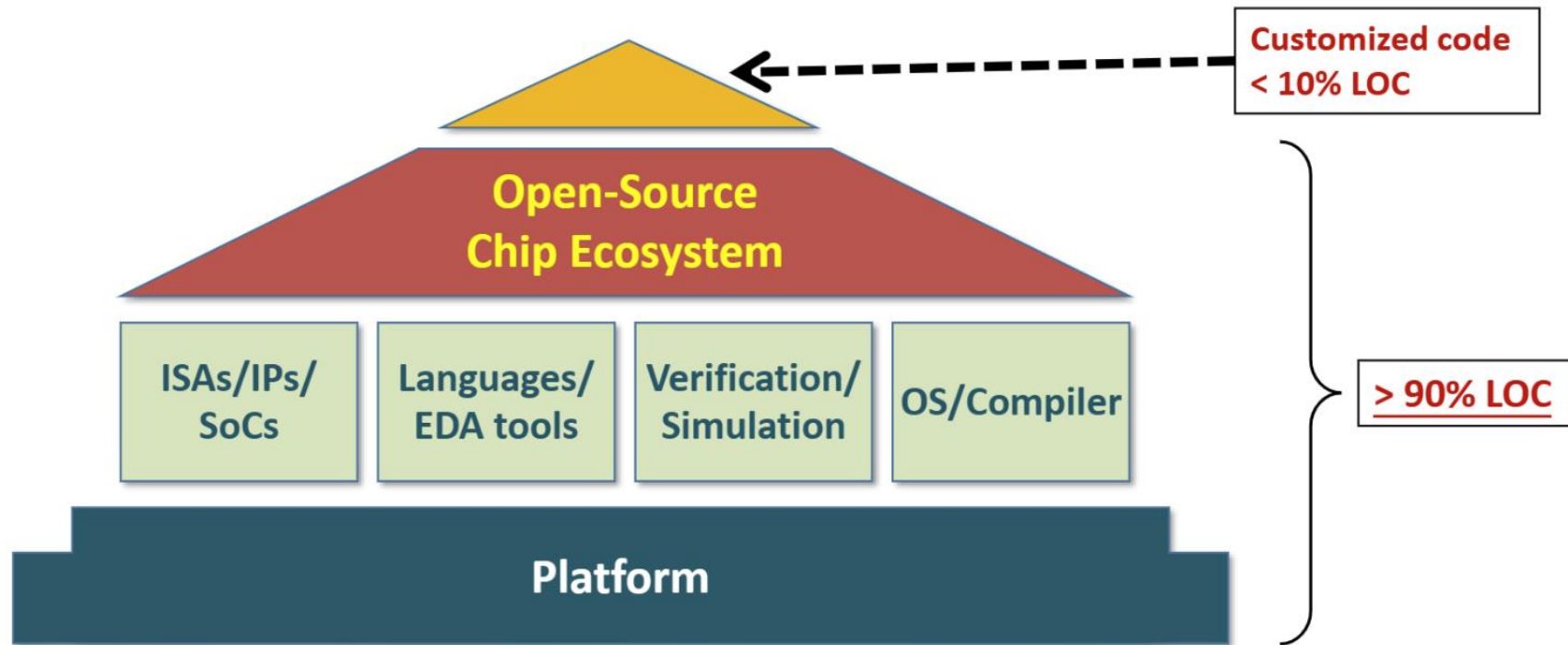
By Sophia Chen

January 9, 2023



Open-Source Chip Ecosystem

- Goal: mirror the success of the open-source software ecosystem



To Lower the barrier of chip development

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



L2: OPEN Design/Implementation

L3: OPEN Framework/Tools



Open Framework
microarch.



microarch.

ISA Spec.

Docs



Open Flow

coding

RTL

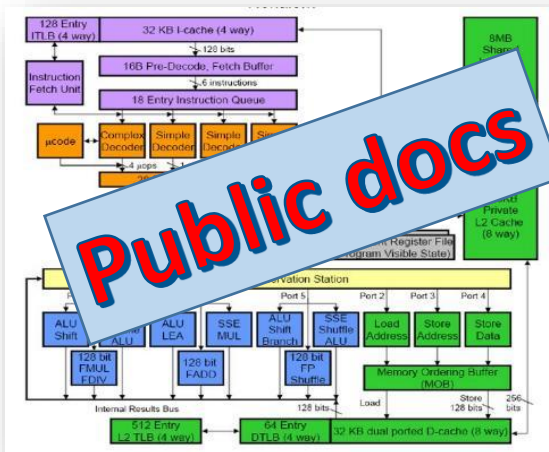
Open Tools

EDA Tools

Layout

The RISC-V Instruction Set Architecture

Open & Free



Public docs

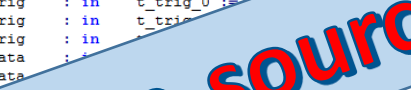
```

component DebugCoreTop is
port (
    -- Trigger and Data
    cu_Clk      : in    std_logic_vector(31 downto 0);
    cu0_Trig    : in    t_trig_0 := (others => '0');
    cu1_Trig    : in    t_trig_1 := (others => '0');
    cu2_Trig    : in    t_trig_2 := (others => '0');
    cu0_Data    : in    std_logic_vector(31 downto 0);
    cu1_Data    : in    std_logic_vector(31 downto 0);
    cu2_Data    : in    std_logic_vector(31 downto 0);

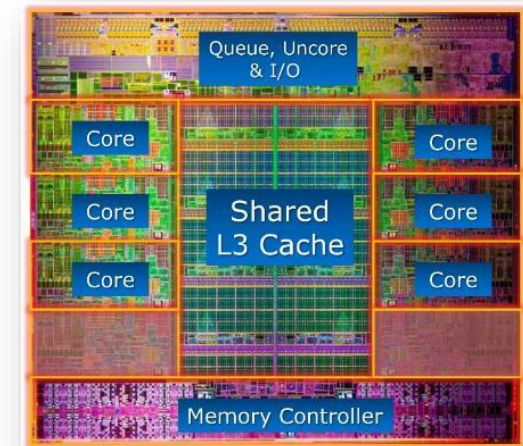
    gt_Trig     : in    std_logic := '0';
    gt_Data     : in    std_logic := '0';

    gt_RX_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;

```



Open-source





Three Level of Open-Source Chip

L1: OPEN ISA

L2: OPEN Design/Implementation

L3: OPEN Framework/Tools

3

Open Framework/Tools

microarch.

coding

EDA Tools

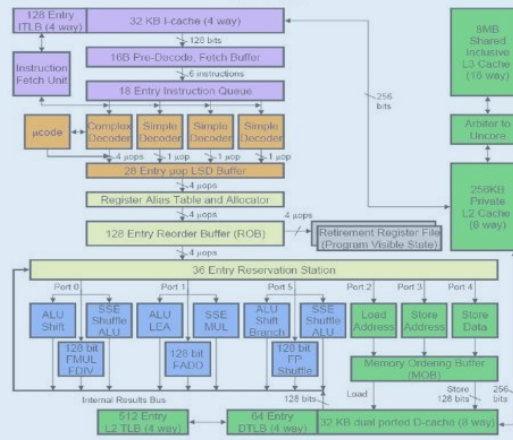
ISA Spec.

The RISC-V Instruction Set Manual
Volume I: Core-Level ISA
Document Version 1.2
Editors: Andrew Waterman¹, Krste Asanović²
¹SRI for Intel, ²UCSD Division, EECS Department, University of California, Berkeley
andrew@eecs.berkeley.edu, krste@eecs.berkeley.edu
May 7, 2017

1

Open ISA

Docs



2

Open Design/Implt

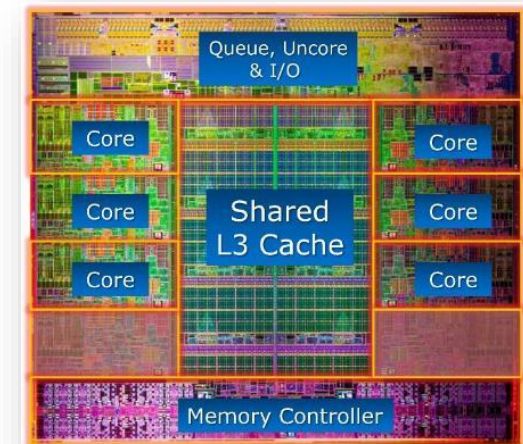
RTL

```
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;
```

Layout



Why Open-Source High-perf. RISC-V Processor?

- **Why RISC-V:** Free and open ISA
- **Why high-perf :** Most RISC-V processors are for IoT/edge, but both academic and industrial community need high-performance RISC-V processors
- **Why open-source:** An open and innovative hardware platform
- Build a leading platform with end-to-end agile development flows and tools



Linux

V.S.

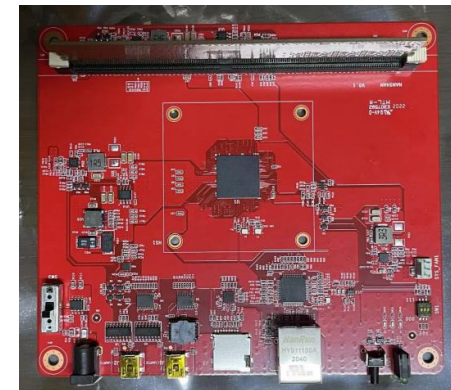


XIANGSHAN

**Envision:
“Hardware Version of Linux”**

Part II

XiangShan: Open-Source High Performance RISC-V Processor





XiangShan: Open-Source High Performance Processors



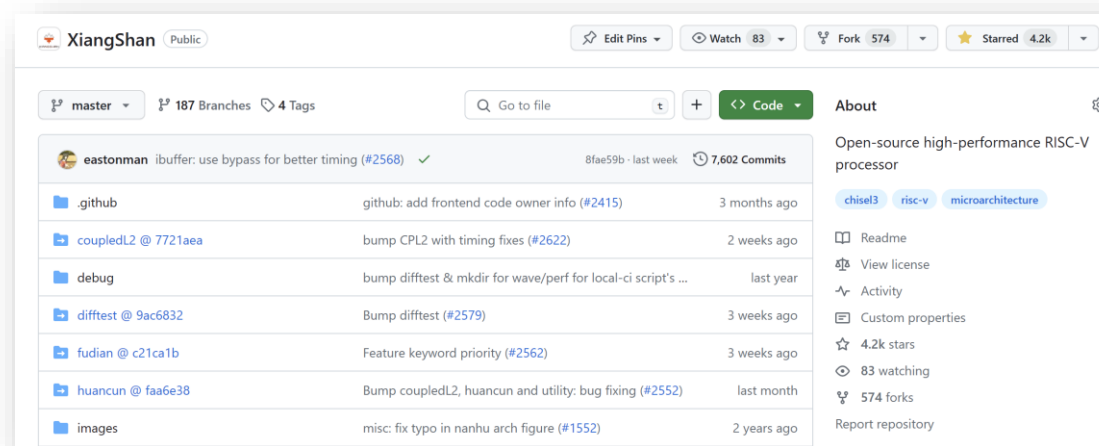
L1: OPEN ISA

L2: OPEN Design/Implementation

L3: OPEN Framework/Tools



Fragrant Hill in Beijing



> 4.3K stars, > 590 forks on GitHub



XiangShan: Open-Source High Performance Processors

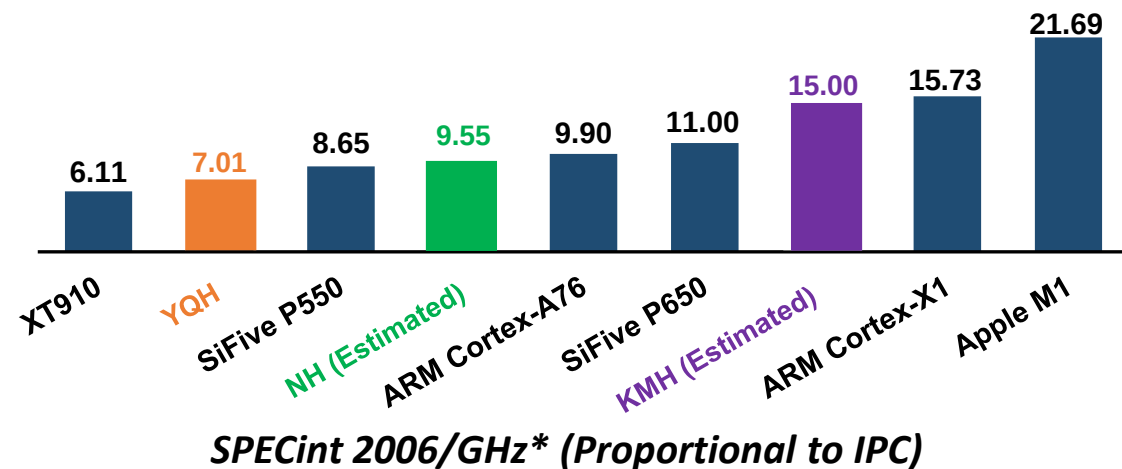
• 1st generation: YQH

- RV64GC, single-core, superscalar OoO
- 28nm tape-out, 1.3GHz, July 2021
- SPEC CPU2006 7.01@1GHz, DDR4-1600



• 2nd generation: NH

- RV64GCBK, dual-core, superscalar OoO
- 14nm GDSII delivery, 2GHz, 2023 Q3
- Estimated** SPECint 2006 19.10@2GHz



• 3rd generation: KMH

- RV64GCBKHV, quad-core, superscalar OoO
- Advanced-node, 3GHz, 1.5x IPC of NH
- Close collaboration with industrial partners



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

** Updated January 5, 2023



XiangShan V1 (Yanqihu)

- **Test chip developed almost entirely by students**
 - RV64GC, 11-stage, superscalar, out-of-order
 - 5.3 CoreMark/MHz (gcc-9.3.0 -O2)
 - **Real chip: SPEC CPU2006 7@1GHz with DDR4-1600 (DDR not fully optimized)**
- **Tape-out:** single XiangShan core with 1MB L2 Cache



Yanqi Lake in Beijing

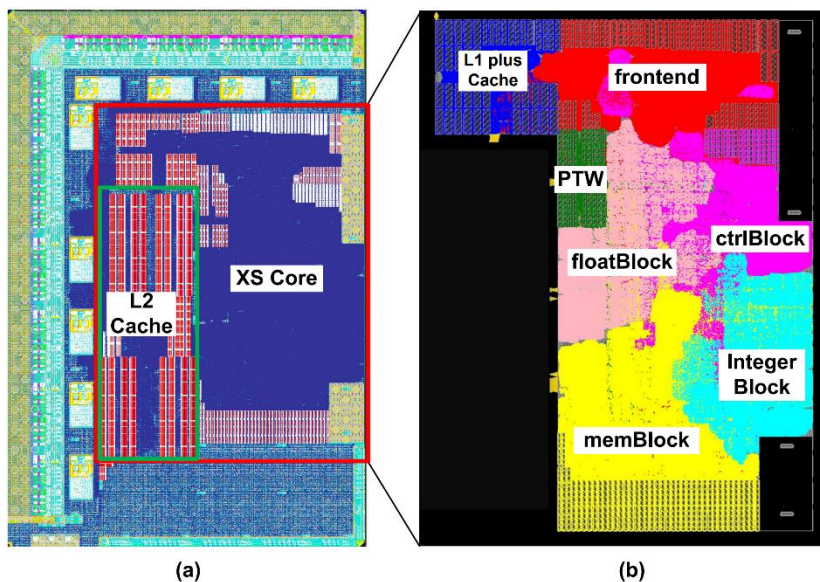


Figure. Layout of (a) the entire chip; (b) the core

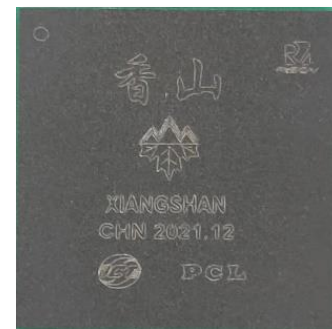


Tape-out information for the processor core

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

Real Chip of XiangShan V1

- The chip was back in January 2022
 - SoC: CPU, SPI Flash, UART, SD card, Ethernet, DIMM
 - Correctly running Debian with SD card and ethernet
- Performance: SPEC CPU2006 7.01@1GHz



SPECint 2006 @ 1GHz		SPECfp 2006 @ 1GHz	
400.perlbench	6.14	410.bwaves	9.28
401.bzip2	4.37	416.gamess	6.59
403.gcc	6.71	433.milc	8.41
429.mcf	6.83	434.zeusmp	7.65
445.gobmk	7.92	435.gromacs	4.99
456.hmmer	5.24	436.cactusADM	3.97
458.sjeng	6.85	437.leslie3d	6.93
462.libquantum	17.71	444.namd	8.00
464.h264ref	10.91	447.dealII	10.17
471.omnetpp	5.65	450.soplex	7.03
473.astar	5.16	453.povray	7.14
483.xalancbmk	7.35	454.Calculix	2.86
		459.GemsFDTD	8.35
		465.tonto	6.42
		470.lbm	10.39
		481.wrf	7.26
		482.sphinx3	9.07

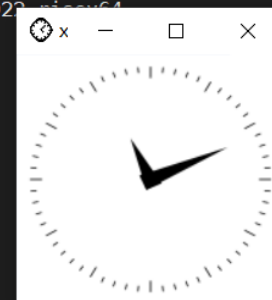
SPECint 2006: 7.03@1GHz

SPECfp 2006: 7.00@1GHz

```
wanghuizhe@open02:~$ ssh -X xs@172.28.2.246
xs@172.28.2.246's password:
Linux open02 4.20.0-44668-ge9c195ab0c63-dirty #109 Thu Feb 17 17:41:13 CST 2022; root@open02

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have no mail.
Last login: Thu Feb 17 11:10:31 2022 from 172.28.9.102
xs@open02:~$ xclock
Warning: locale not supported by C library, locale unchanged
```



**SSH into the Debian on XiangShan, and
run a GUI program via X11 forwarding**



XiangShan V2 (Nanhu)

- **Target: 2GHz@14nm, SPEC CPU2006 ~20 marks**
- **New frontend:** decoupled BP and instruction fetch
- **Improved backend:** better scheduler, instruction fusions, move elimination, and more
- **New L2/L3 cache:** designed for high frequency and high performance with hybrid prefetchers
- **Verified under dual-core config** (RV64GCBK), more devices support (PCIe, USB, ...)

Setup an open and standardized development workflow



A lake in Jiaying, Zhejiang, China

☐	10 Open	✓ 2,322 Closed
☐	Bump difftest for palladium simulation ✓	#2662 opened 11 hours ago by klin02 • Approved
☐	ICache: fix ICacheMainPipe bug about sfence ✓ bug do not merge	#2660 opened 3 days ago by ssszwic • Review required
☐	ICache: change data SRAM partWayNum from 2 to 4 ✓	#2653 opened 5 days ago by ssszwic • Approved
☐	ITTAGE meta width shrink ✓ do not merge	#2592 opened last month by eastonman • Review required
☐	pf: fix negative stream ✓ bug	#2508 opened on Nov 27, 2023 by happy-lx • Review required
☐	SQ: add sq merge ✓ do not merge enhancement	#2439 opened on Oct 29, 2023 by sfencevma • Review required
☐	rm refillPipe ✓ enhancement	#2426 opened on Oct 25, 2023 by YukunXue • Approved

Pull Request Snapshot

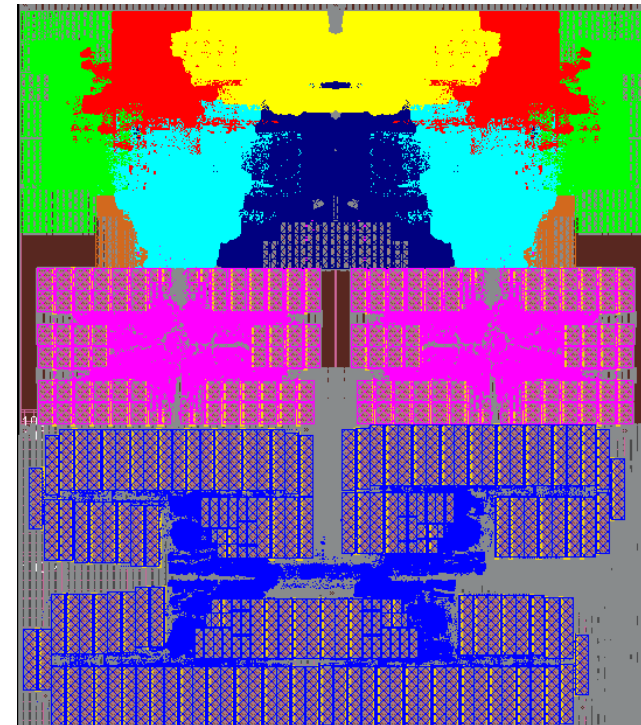


Estimated Performance of XiangShan V2

- Estimated **SPECint 2006 19.10, SPECfp 2006 22.18@2GHz**
 - Compile with GCC 10.2.0, -O2, RV64GCB
 - RTL simulation, DDR4-2400 under DRAMsim3

SPECint 06	
400.perlbench	19.27
401.bzip2	11.36
403.gcc	21.97
429.mcf	20.53
445.gobmk	15.97
456.hmmer	19.22
458.sjeng	17.22
462.libquantum	36.99
464.h264ref	28.54
471.omnetpp	14.02
473.astar	14.19
483.xalancbmk	21.48
SPECint2006@2GHz	19.10

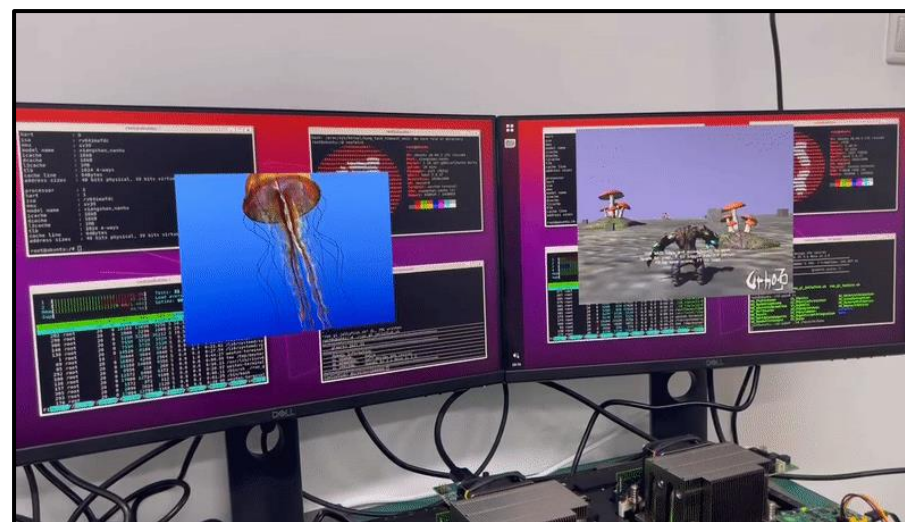
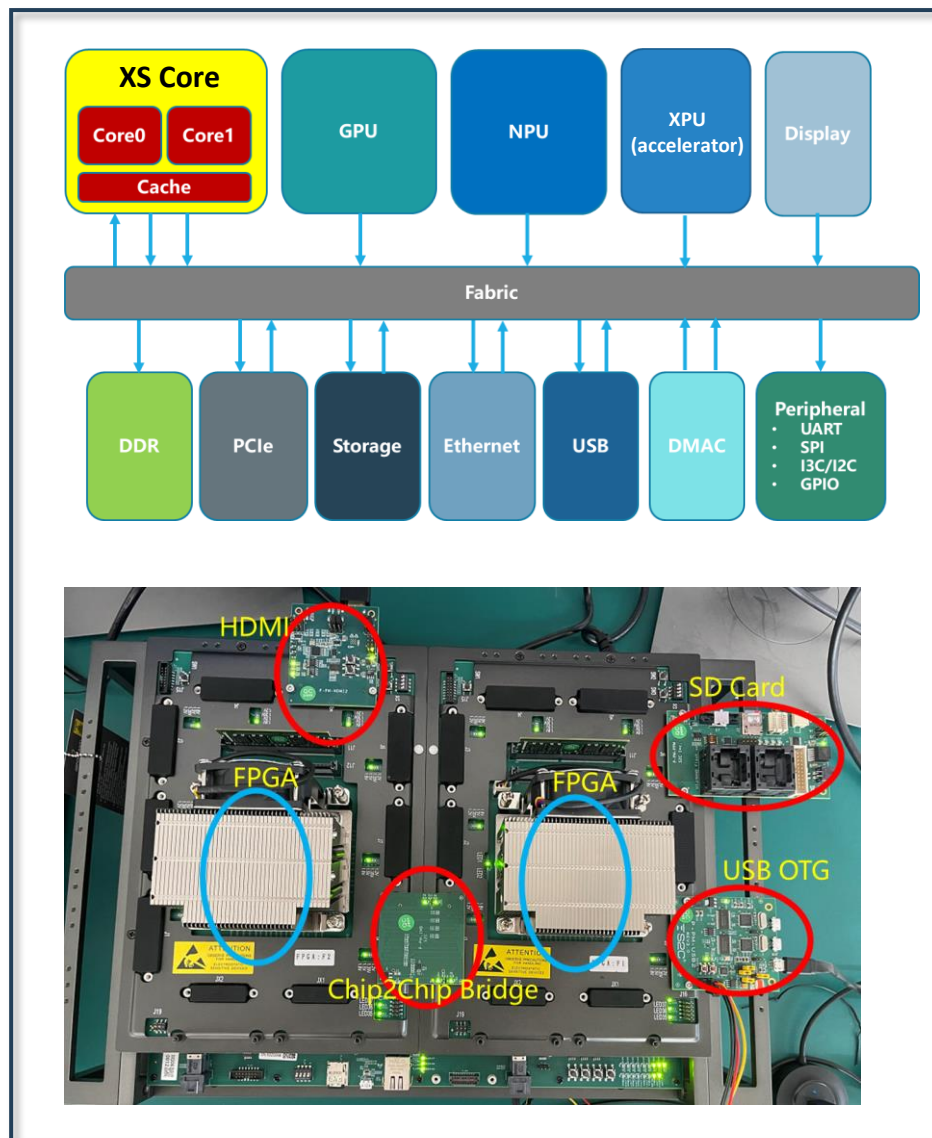
SPECfp 06	
410.bwaves	18.09
416.gamess	23.82
433.milc	18.33
434.zeusmp	28.19
435.gromacs	17.53
436.cactusADM	24.26
437.leslie3d	20.28
444.namd	23.83
447.dealII	33.50
450.soplex	25.61
453.povray	27.06
454.Calculix	9.18
459.GemsFDTD	24.66
465.tonto	17.68
470.lbm	32.04
481.wrf	19.73
482.sphinx3	28.38
SPECfp2006@2GHz	22.18



Feature	V1 YQH	V2 NH
ISA	RV64GC	RV64GCBK
Process Node	28nm	14nm
Core Count	1	2
Die Size	8.6 mm ²	22.13 mm ²
Std Cell Num/Area	5.05M, 4.27 mm ²	11.3M, 4.53 mm ²
Mem Num/Area	261, 1.7 mm ²	692, 8.93 mm ²
Density	66%	35%
Frequency	1.3GHz, TT 0.9V	2GHz, TT 0.9V



DEMO——Prototype in Only Two Weeks



Source: Xinchen Technology

XiangShan V3 (Kunminghu)

- Target ARM Neoverse N2
 - SPECCPU2006: 45@3GHz (15/GHz)
 - Vector/Hypervisor extension supported
- A Joint Dev Team (coordinated by BOSC)
 - About 10 institutions



Kunming Lake in Beijing



Tencent 腾讯

阿里巴巴
Alibaba.com

ThunderSoft

ZTE 中兴

中科创达

ESWIN

算能
SOPHGO



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

.....



Highlights in XiangShan V3

- **Functional Enhancement**

- Support RISC-V **Vector/Hypervisor** extension
- Support interconnection based on **CHI protocol**

- **Performance Exploration**

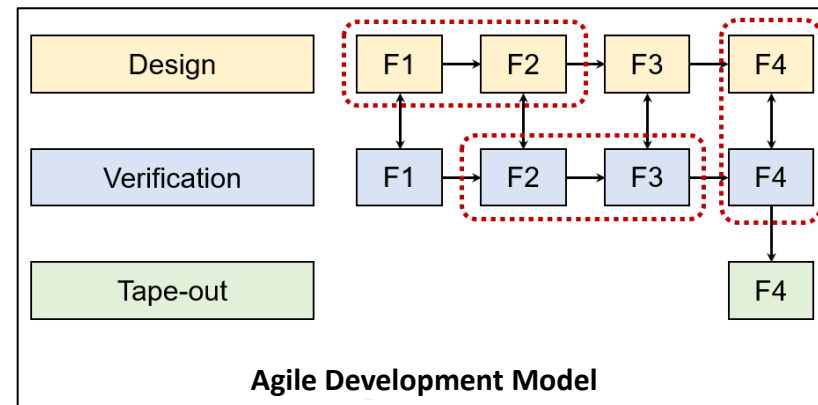
- Performance boost in frontend, backend, load-store unit and cache
- Performance model calibrated with RTL
- Workflow: **DSE on perf model => Impl. & fine tuning on RTL**

- **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



Performance Evaluation of XiangShan V3

- 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	
400.perlbench	36.648
401.bzip2	24.283
403.gcc	47.692
429.mcf	57.852
445.gobmk	31.711
456.hmmer	39.567
458.sjeng	31.500
462.libquantum	125.487
464.h264ref	57.376
471.omnetpp	42.243
473.astar	30.738
483.xalancbmk	75.535
SPECint2006@3GHz	44.977

Estimated SPECint 2006 Base
44.98@3GHz

SPECint 2006	
400.perlbench	42.143
401.bzip2	24.283
403.gcc	47.692
429.mcf	57.772
445.gobmk	31.711
456.hmmer	60.826
458.sjeng	31.500
462.libquantum	222.412
464.h264ref	60.243
471.omnetpp	42.243
473.astar	30.738
483.xalancbmk	81.213
SPECint2006@3GHz	49.964

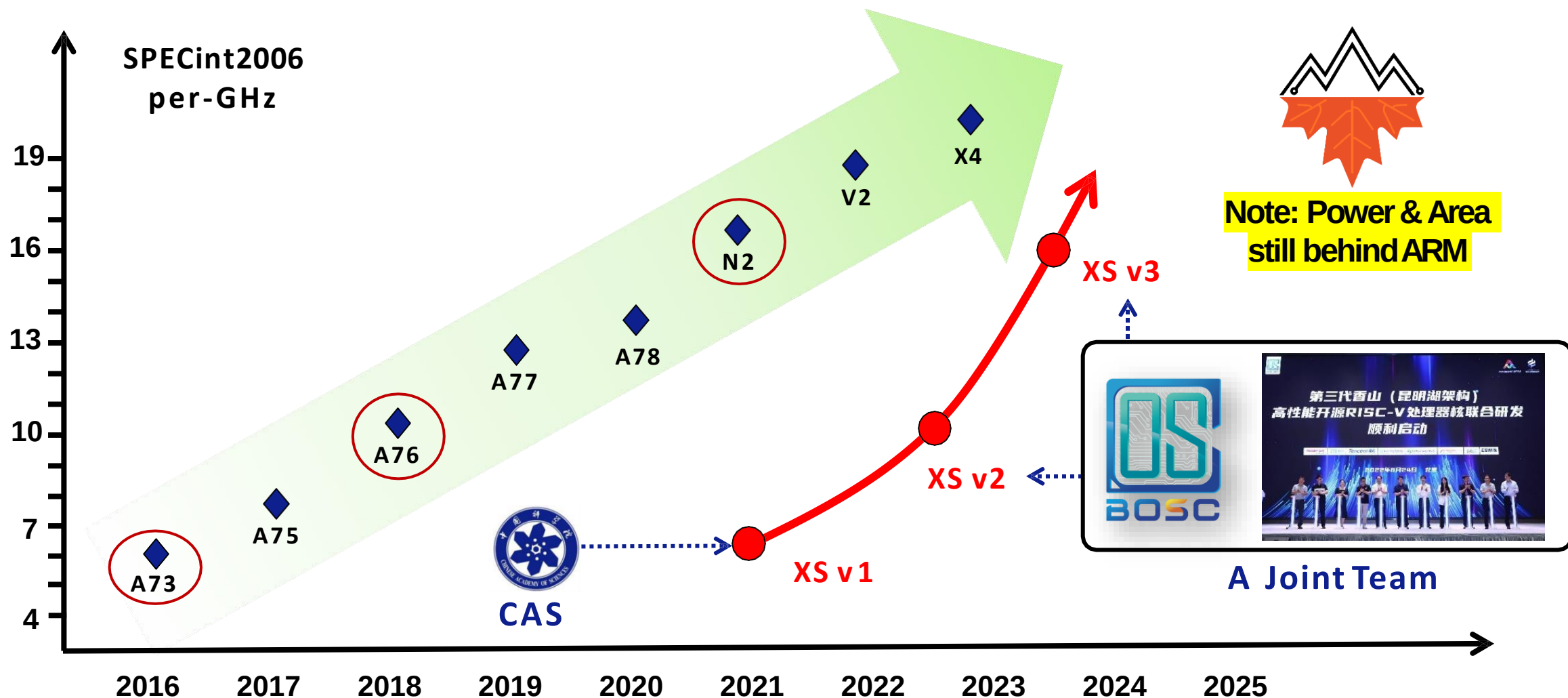
Estimated SPECint 2006 Peak
49.96@3GHz



Floorplan of V3 (single core)



XiangShan: Open-Source High Performance Processors



Prospect: "Dual Core" Roadmap



Based on V3 (KMH)

- **Big Core: Ultimate Performance (v.s. ARM N2)**

Target High-Throughput Advanced Computing Platform

Goal: become mainstream CPU for data centers and computational facilities



Based on V2 (NH)

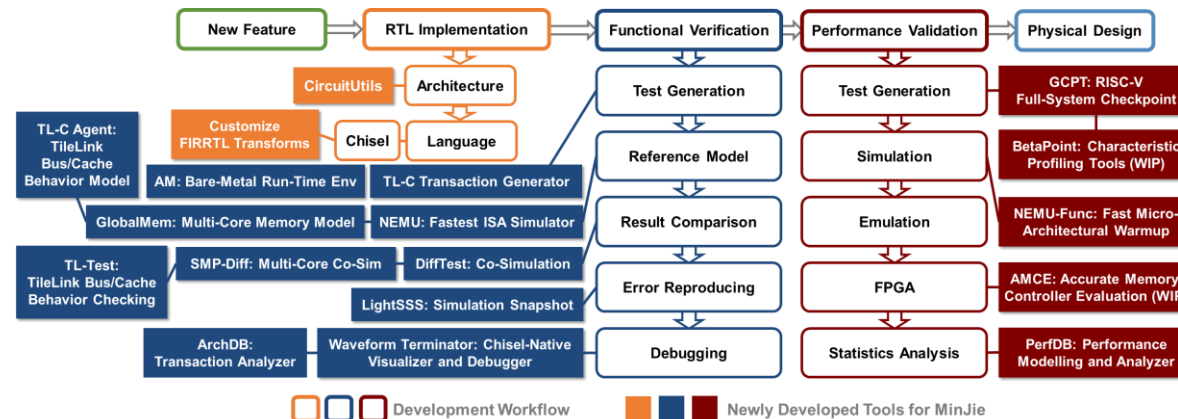
- **Mid Core: Balanced Perf & Efficiency (v.s. ARM A76)**

Target Mid-to-High-End General Industry Domain

Goal: support broad industrial spectrum including industrial control, automotive, communication, aviation and more

Part III

MinJie: Agile Development for High-Performance RISC-V Processors





First Step to Agile Design: Use Chisel

- 2018: quantitative experiments between Chisel and Verilog

• **Task #1:** Design an L2 Cache for RISC-V Rocket-chip core

• **Who:** A 5-year engineer vs. a senior student

	A 5-year Engineer	An Undergraduate
Experience	Familiar w/ OpenSparc T1; Modified Xilinx Cache	Course projects on CPU design; Using Chisel for 9 months
Language	Verilog	Chisel
Time	6 weeks	3 days
LOCs	~1700	~350
Results	Unable to boot Linux	Boot Linux on multicore RISC-V; support NIC w/ DMA mode

• **1st Round results:** Chisel is more productive than Verilog by **14X** with only **1/5** LOC

• **Task #2:** Translate the Verilog codes into Chisel

• Evaluated on FPGA (xc7v2000tfhg1716-1), Vivado 2017.01

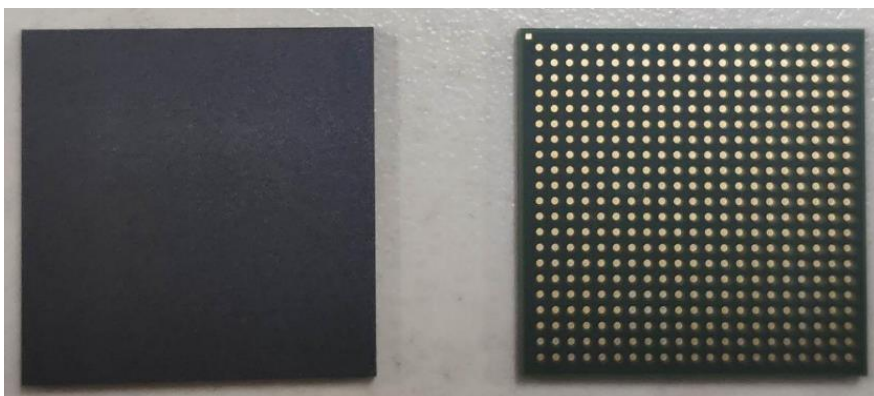
• **Who:** A junior student who never knew Chisel

	Verilog	Chisel (direct translation)	Chisel-opt (adv. features & libs)
Freq./MHz	135.814	136.388 (+0.42%)	154.107 (+13.47%)
Power/W	0.770	0.749 (-2.73%)	0.749 (-2.73%)
LUT Logic	5676	6422 (+13.14%)	2594 (-54.30%)
LUT Storage	1796	1264 (-29.62%)	1492 (-16.93%)
FF	4266	3638 (-14.72%)	747 (-82.49%)
LOCS	618	470 (-23.95%)	155 (-74.92%)

• **2nd Round results:** Chisel can achieve better PPA than verilog

Yu Zihao, Liu Zhigang, Li Yiwei, Huang Bowen, Wang Sa, Sun Ninghui, Bao Yungang. Practice of Chip Agile Development: Labeled RISC-V. Journal of Computer Research and Development, 2019, 56(1): 35-48.

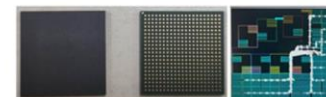
- 2020: 28-nm tape-out of an 8-core labeled RISC-V processor



16节点原型系统机箱



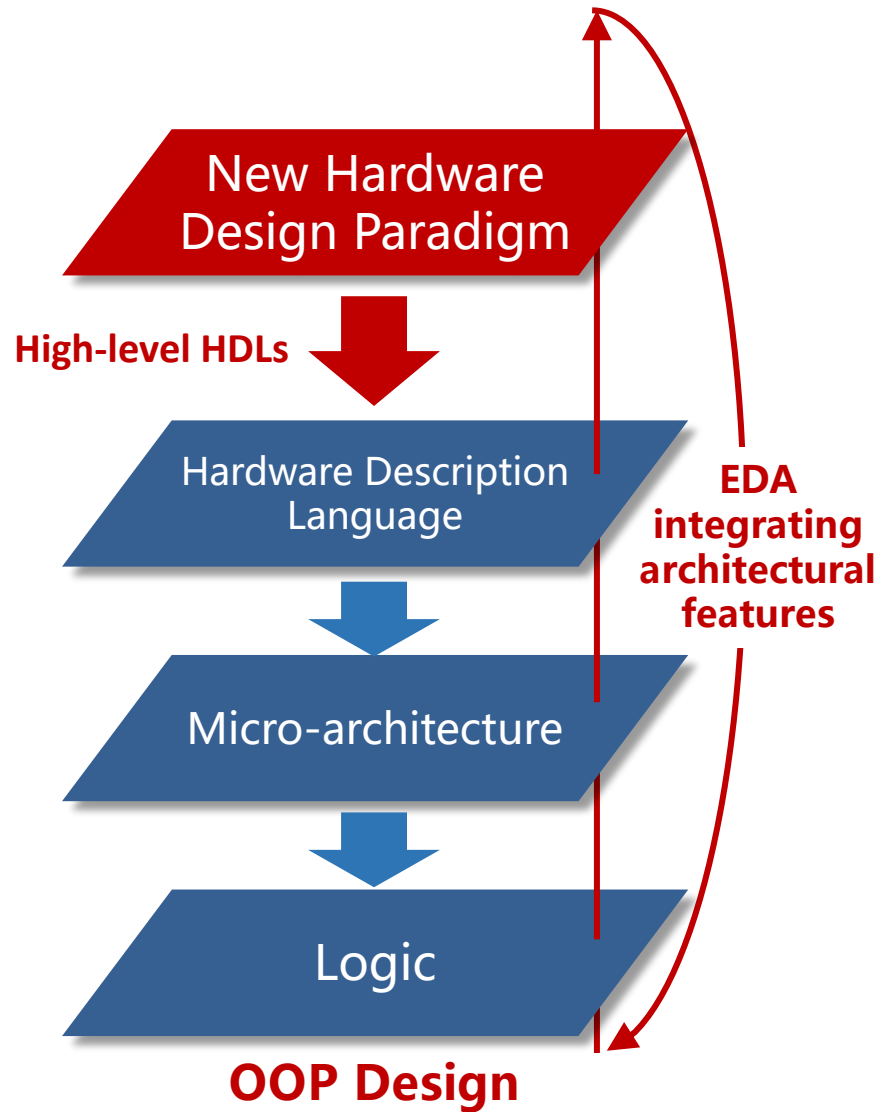
单节点板卡



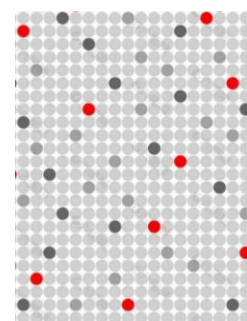
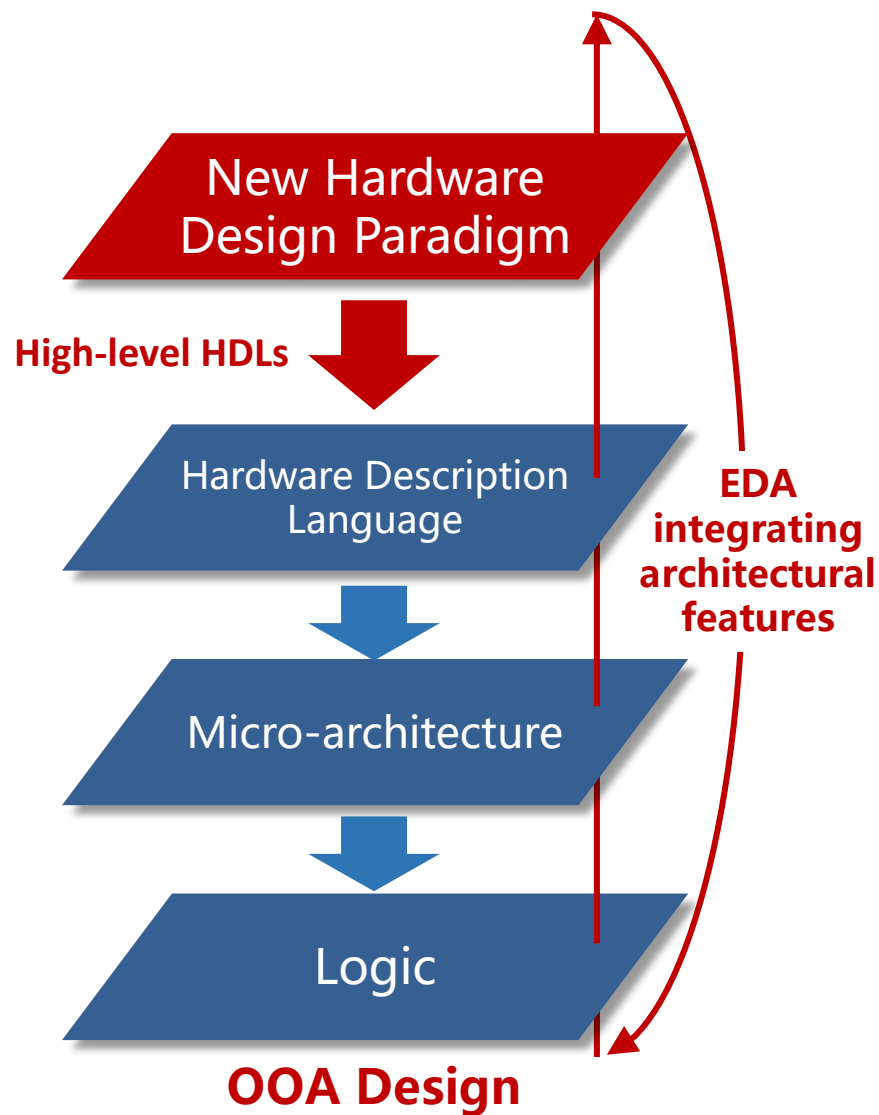
“北海100”
标签化RISC-V芯片

- RV64GC指令集
- 单发射顺序9级流水线
- 内置标签化冯诺依曼结构技术
- 8核/2MB L2 Cache
- ChipLink前端总线
- **1.2GHz@ 28nm**
- Wafer out/WB BGA封装
- 最大支持32GB DDR4内存
- 2*千兆以太网
- 1*PCIe3.0 RC x4

New HDL → New Design Paradigm



New HDL → New Design Paradigm

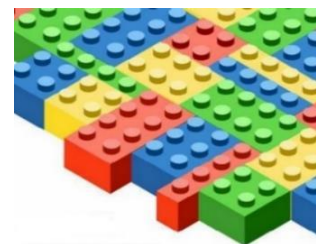


Pixel
(gate-level)

Circuit-oriented
Processor Design



Processor



Object
(Reusable Modules)

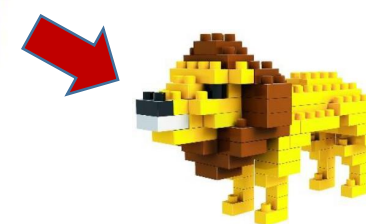
Object-oriented
Processor Design



Processor



Configurable





What's Missing in Agile Hardware Design? Verification!

What's Missing in Agile Hardware Design? Verification!

Babak Falsafi, *Fellow, ACM, IEEE*

Parallel Systems Architecture Laboratory, Institute of Computer and Communication Sciences, School of Computer and Communication Sciences, Ecole polytechnique fédérale de Lausanne CH-1015 Lausanne

E-mail: babak.falsafi@epfl.ch

Agile hardware design is an approach to developing hardware systems that draws inspiration from the principles and practices of agile software development. It emphasizes collaboration, flexibility, iterative development, and quick adaptation to changing requirements. In agile hardware design, the focus is on delivering functional hardware systems in shorter development cycles while maintaining high-quality and customer satisfaction.

In particular, agile hardware design is of great interest in the open-source hardware community. Open-source hardware development —such as RISC-V— is at the forefront of initiatives to democratize hardware and drive innovation in chip design forward. Agile design is instrumental for the RISC-V community because it supports rapid iteration, accommodates the evolving RISC-V standard and the addition of custom extensions, improves community collaboration and time-to-market, and addresses the design challenges associated with complex architectural features.

Among significant innovations based on agile hardware design is the recently announced XIANGSHAN RISC-V core which is currently the highest performing RISC-V out-of-order microprocessor core with single-thread performance exceeding both existing RISC-V cores and a state-of-the-art ARM core, Cortex-A76. The creators of this platform have published their agile design methodology in a flagship computer architecture venue, MICRO, with a paper that has been selected through peer review to be among the best dozen papers in all of computer architecture in one year for publication in IEEE Micro Top Picks.

A key contributor to this breakthrough has been integrating hardware verification into the agile methodology. Hardware verification is crucial in designing digital platforms, as it ensures that semiconductor chips operate correctly and reliably according to the architecture specifications. Verification guarantees compliance with standards, and helps detect and rectify design errors, validate system-level functionality, optimize performance and power consumption, and enhance hardware reliability and safety. It plays a fundamental role in creating robust and dependable CPUs that meet the requirements of various applications and workloads.

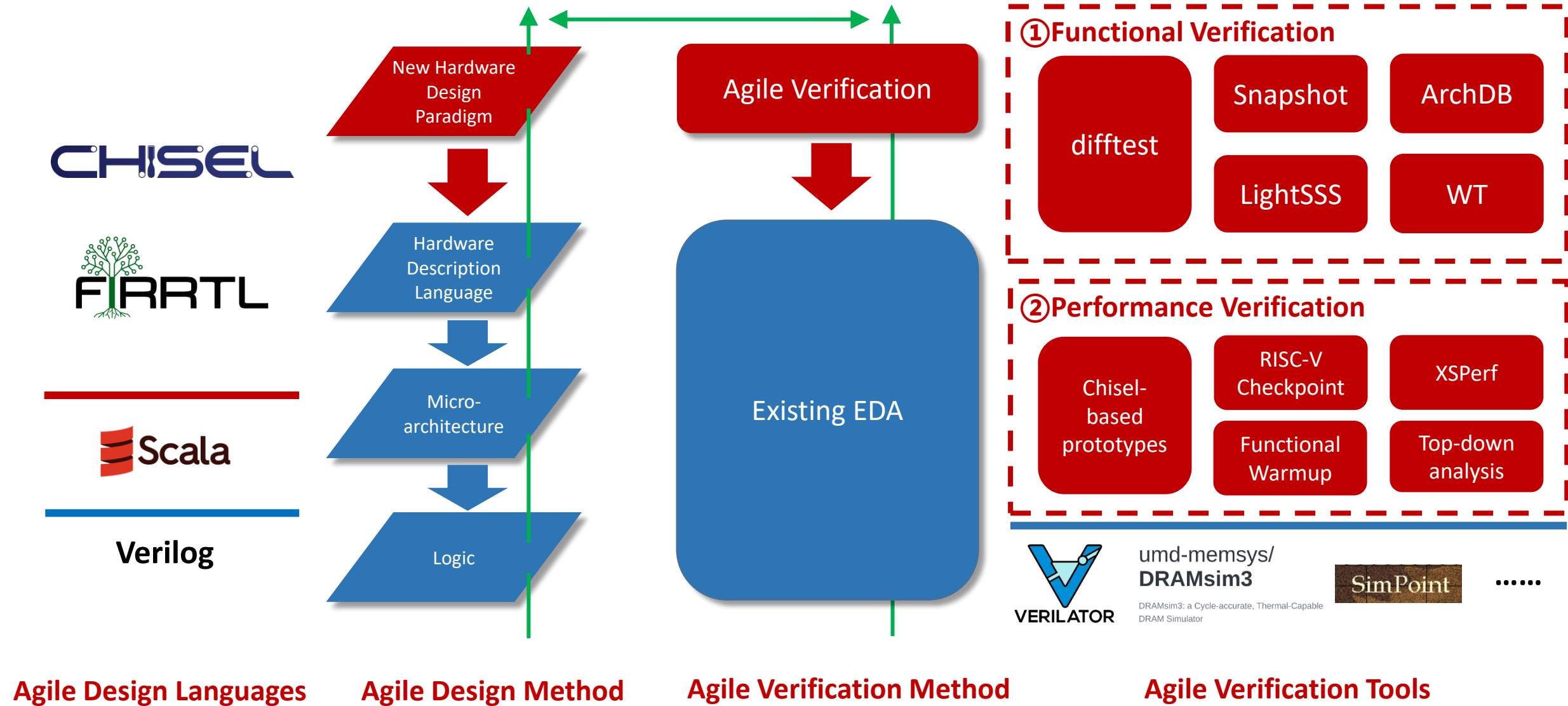


Babak Falsafi

Professor, EPFL

ACM/IEEE Fellow

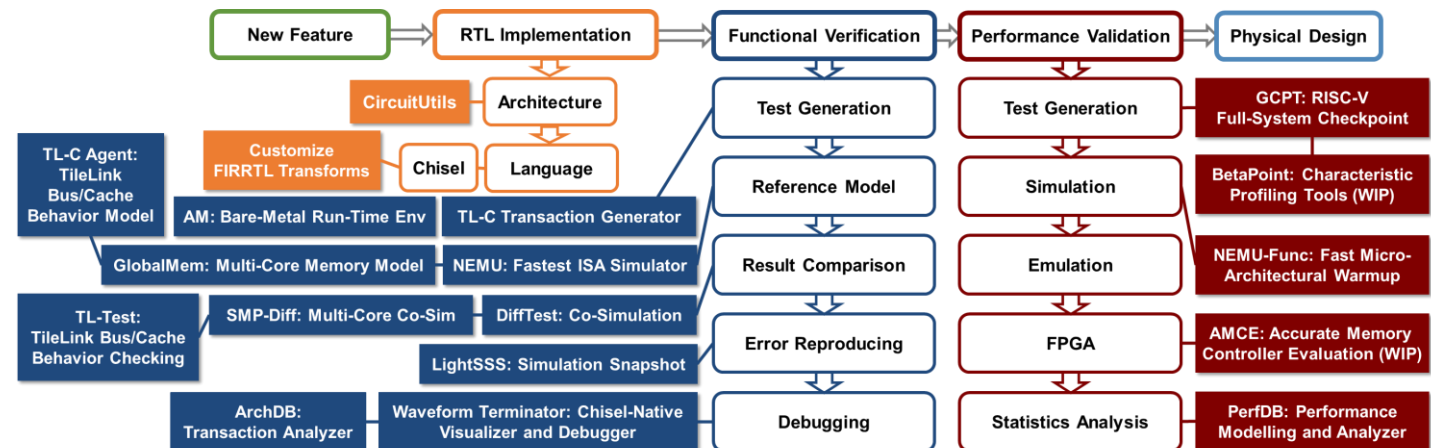
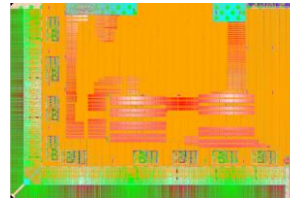
Agile Verification is Challenging



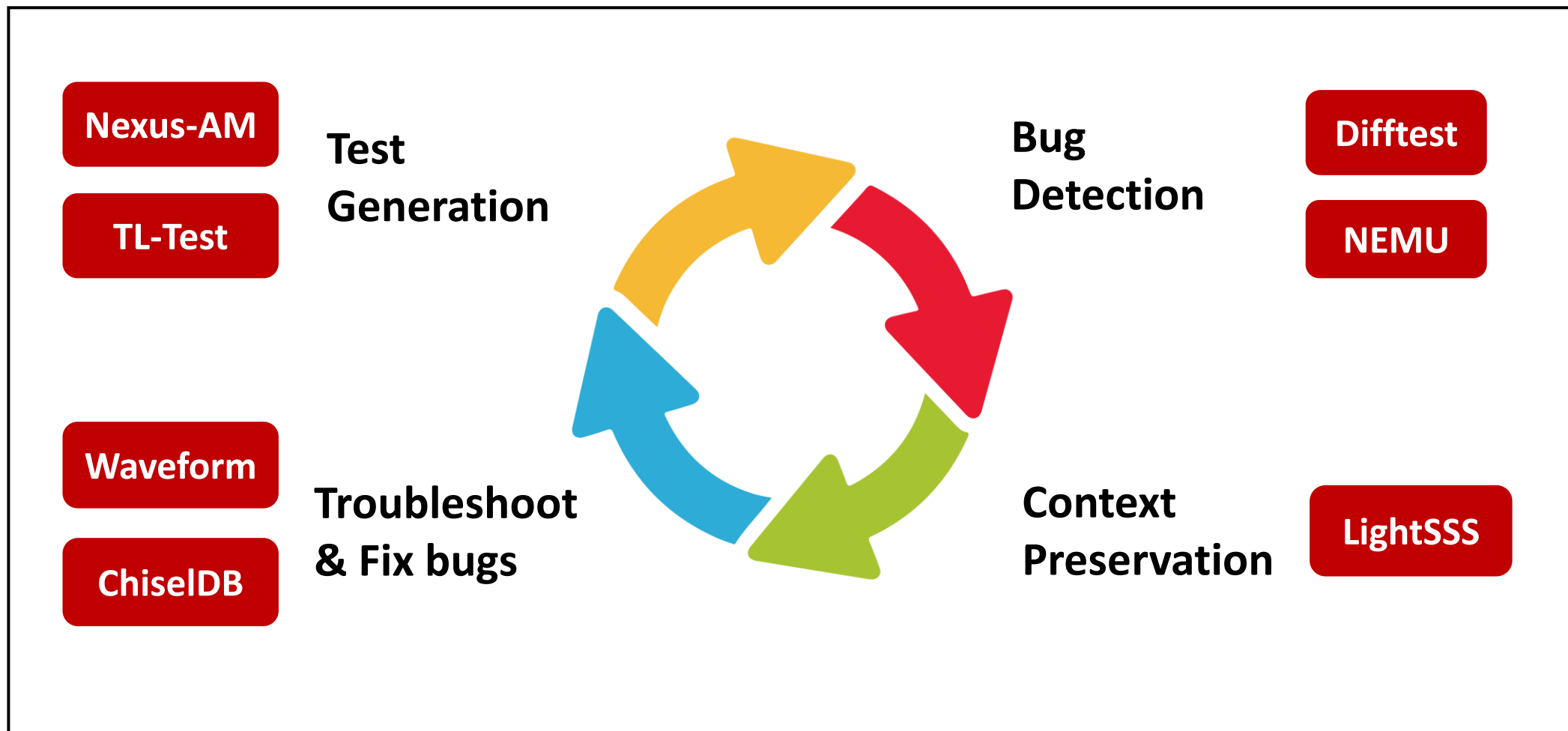


Minjie: Open & Agile Verification Toolchain

- Infrastructure is the key outcome of the XiangShan Project
- Open source to benefit both academia and industry



Functional Verification Toolchain

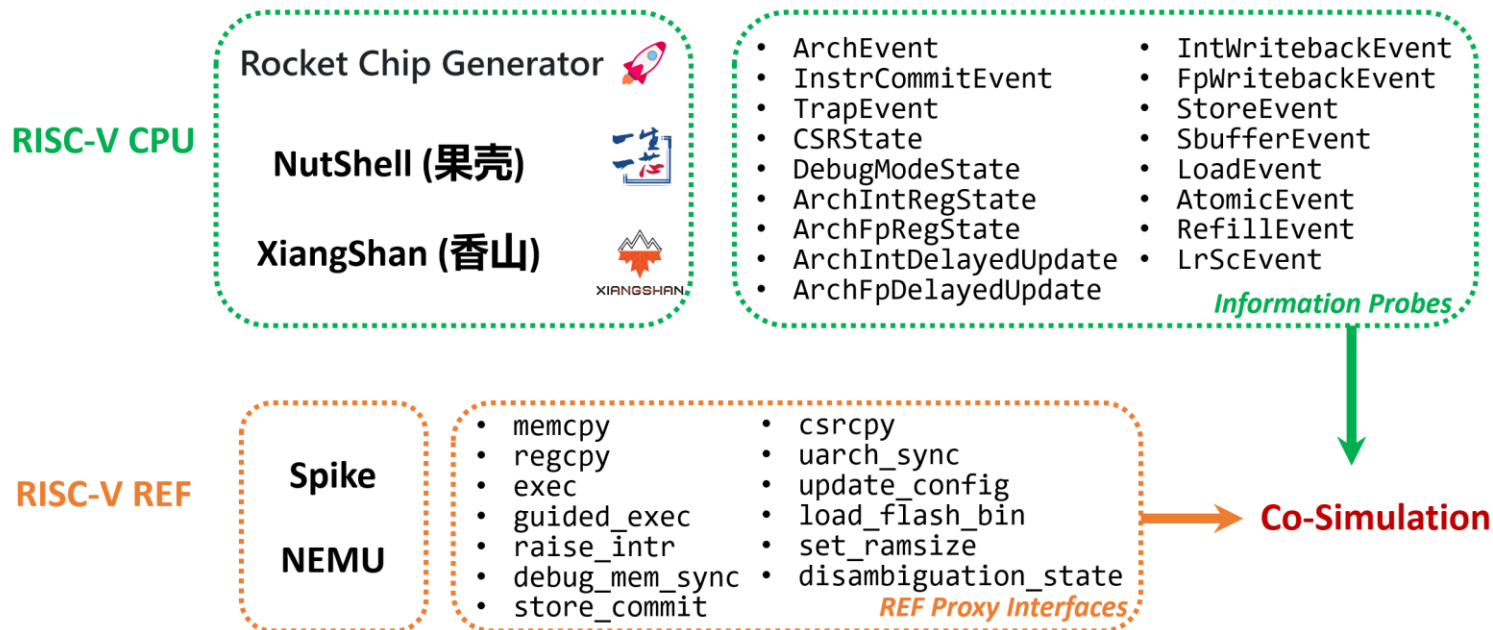
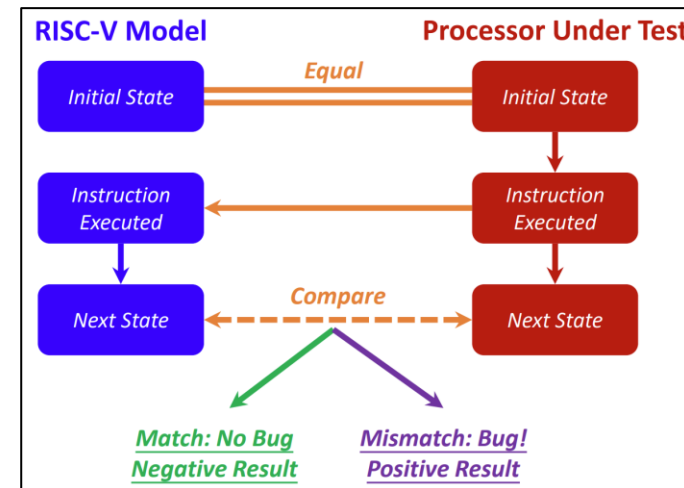




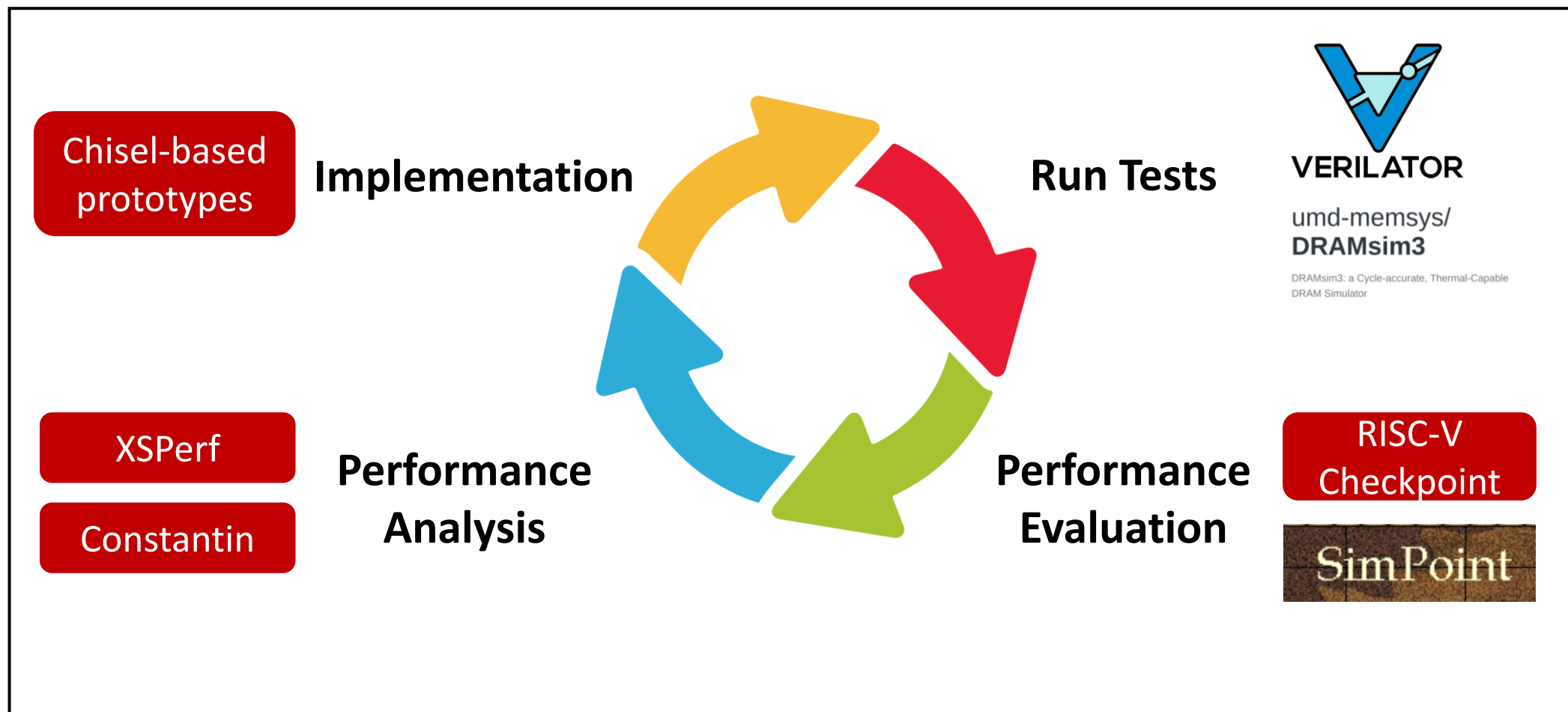
DiffTest: ISA Co-Simulation Framework

• Basic flows

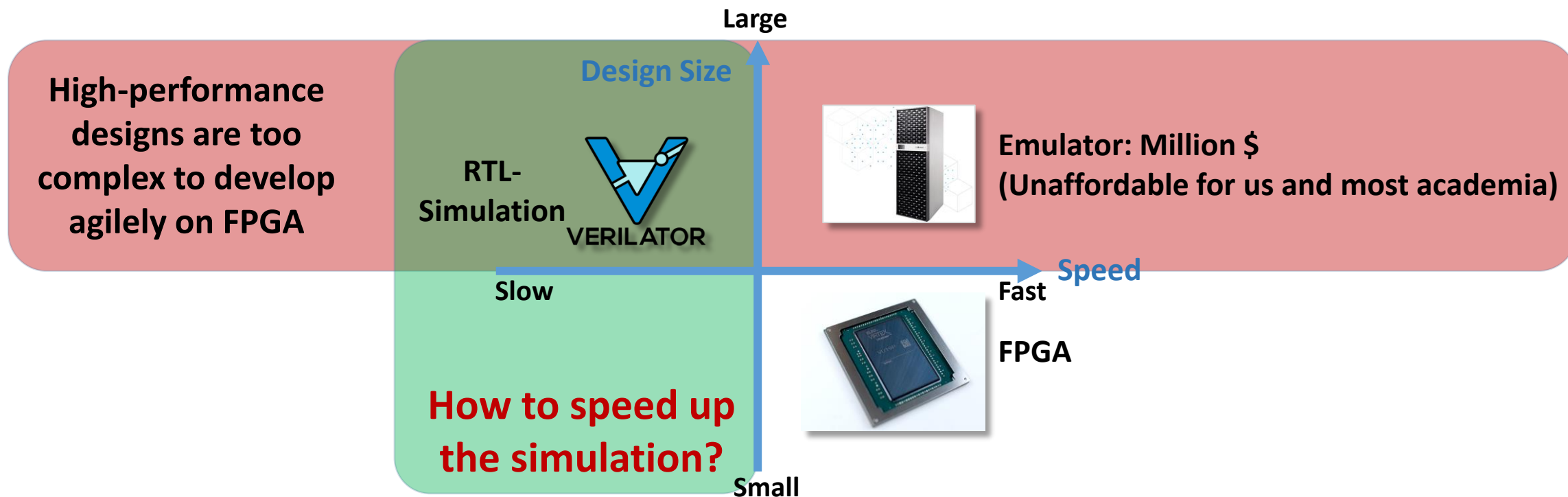
- Instructions commit/other states update
- The simulator executes the same instructions
- Compare the architectural states based on **Diff-rules**
- Abort or continue



Performance Verification Toolchain



Agile Performance Evaluation



Time for performance modelling of single-core XiangShan on SPEC CPU2006

	RTL-simulation	FPGA	RTL-Simulation w/ Checkpoint
Compile	20 minutes	5 hours	20 minutes
Simulation	958 years@2KHz (2K CPS)	7 days@100MHz	5.5 hours with enough x86 servers
Our Approach			



Four Steps

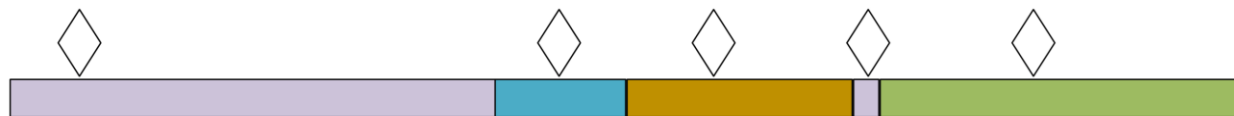
Step 1

Inst. Slices
@Fast Simulators



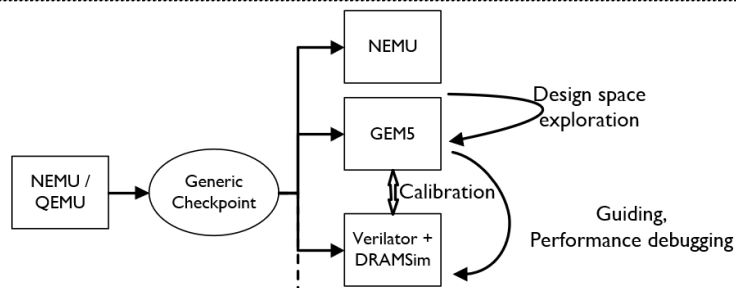
Step 2

Representative Slices
@Clustering alg.



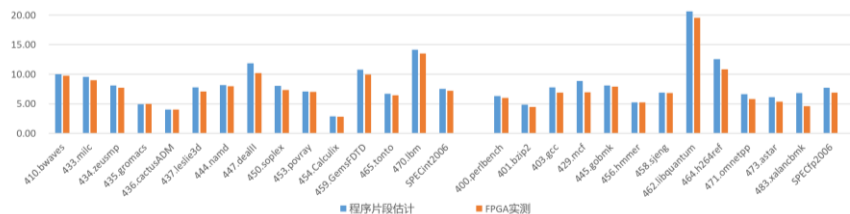
Step 3

Run Slices
@RTL Simulator



Step 4

Reconstruction
@Alg.



Deviation: < 10%



Advanced Agile Chip Design Methodology

- New method and toolchain for agile chip design
- Developed more than 17 new tools to solve agile verification problems
- **MICRO 2022 → IEEE Micro Top Picks**

2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)



Towards Developing High Performance RISC-V Processors Using Agile Methodology

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

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

[†]University of Chinese Academy of Sciences, China

[‡]Beijing Institute of Open Source Chip, China

[§]Peng Cheng Laboratory, China

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

^{||}Shenzhen University, China

^{**}Dalian University of Technology, China

THEME ARTICLE: TOP PICKS FROM THE 2022 COMPUTER ARCHITECTURE CONFERENCES

Toward Developing High-Performance RISC-V Processors Using Agile Methodology

Yinan Xu^{IC} and Zihao Yu^{IC}, State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China

Dan Tang^{IC}, Beijing Institute of Open Source Chip, Beijing, 100080, China

Ye Cai^{IC}, Shenzhen University, Shenzhen, 518060, China

Dandan Huan^{IC}, Beijing VCore Technology, Beijing, 100190, China

Wei He^{IC}, Peng Cheng Laboratory, Shenzhen, 518060, China

Ninghui Sun^{IC} and Yungang Bao^{IC}, State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100190, China





microarch.

coding

EDA Tools

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

Editors: Andrew Waterman¹, Krste Asanovic^{1,2}
¹Sifive Inc.,
²Division, EECS Department, University of California, Berkeley
andrew@sifive.com, krste@berkeley.edu
May 7, 2017

1 Open ISA

[illegible]

```

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'));

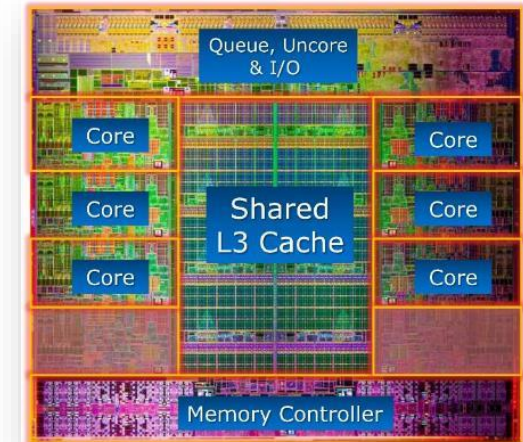
    -- Downstream I2C
    SCL         : in    std_logic := '0';
    SDA         : inout std_logic := '0';

    -- 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



Part IV

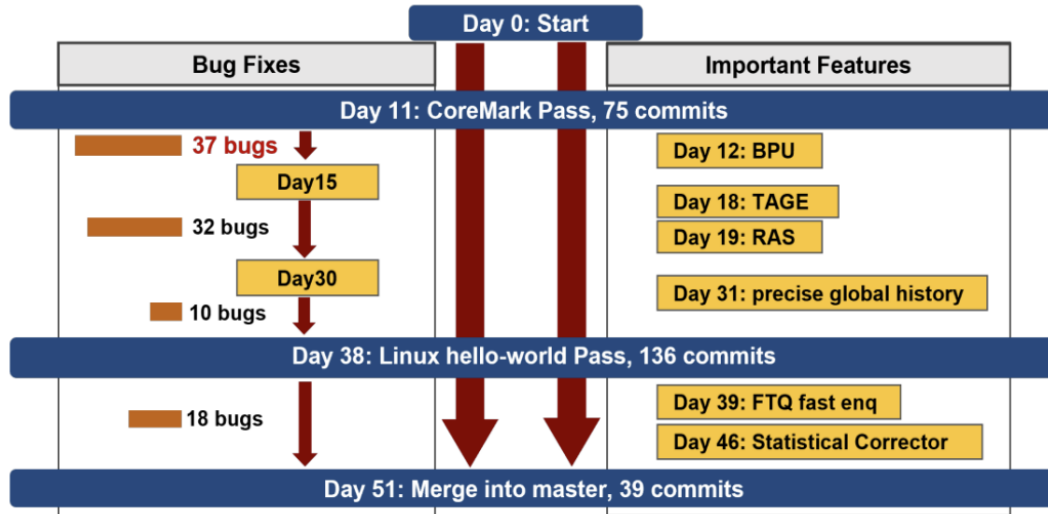
An Effective Infrastructure for Research





An Effective Infrastructure for Research

① Micro-architecture Optimization



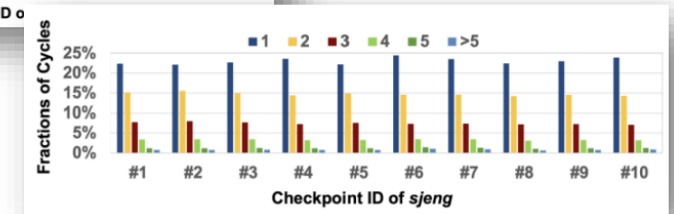
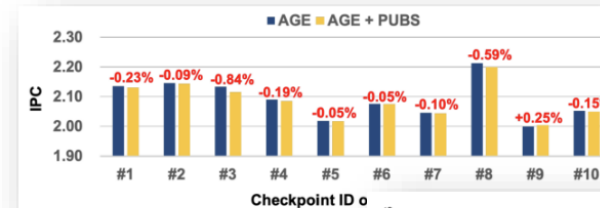
- 3 graduate students
- 11 days for a functionally correct prototype
- 37 bugs in 5 days
- 38 days to boot Linux with BPU
- 51 days for the overall frontend architecture

② Paper Reproduction

2018 51st Annual IEEE/ACM International Symposium on Microarchitecture

Performance Improvement by Prioritizing the Issue of the Instructions in Unconfident Branch Slices

One third-year PhD student, 200 minutes on XiangShan



An Effective Infrastructure for Research

- **Topic: Computer Architecture**

- XiangShan: a realistic 6-wide 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

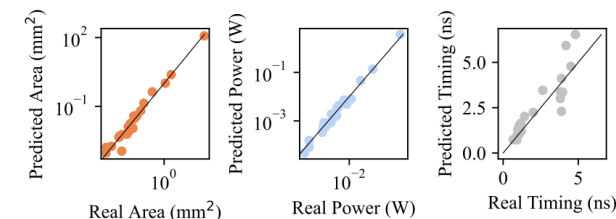
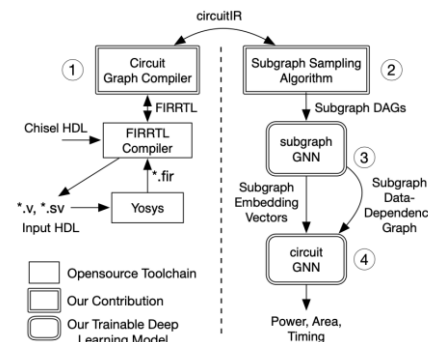
EPFL

<https://midgard.epfl.ch/>

Imprecise Store Exceptions, ISCA'23 (EPFL)

- **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)

Summary

- **Era of open-source chip is coming**
XiangShan fills the gap in high performance open source processors.
- **Three generation, dual-core roadmap**
XiangShan meet the needs from both academia and industry.
- **An effective platform for research on microarchitecture and agile hardware design.**



Thanks!