

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

---

*Kaifan Wang, Yinan Xu, Kan Shi, Yungang Bao*

*The XiangShan Team*

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

ASPLOS'23@Vancouver, Canada  
March 25, 2023



# Tutorial@ASPLOS'23 Schedule

Time	Topic
9:00-9:25	Introduction of the XiangShan Project
9:25-9:35	Tutorial Overview and Highlights
9:35-10:20	Microarchitecture Design and Implementation
	Coffee Break
10:40-12:00	Hands-on Development

## 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.**

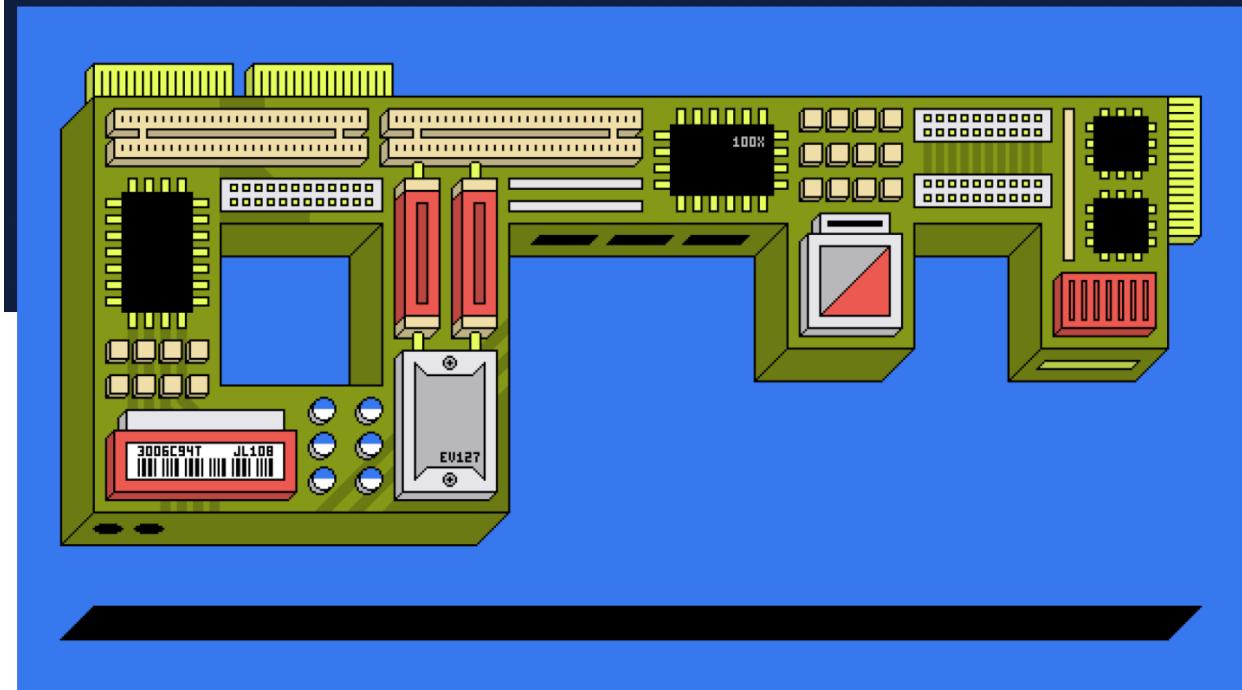
--- MIT Technology Review

## 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

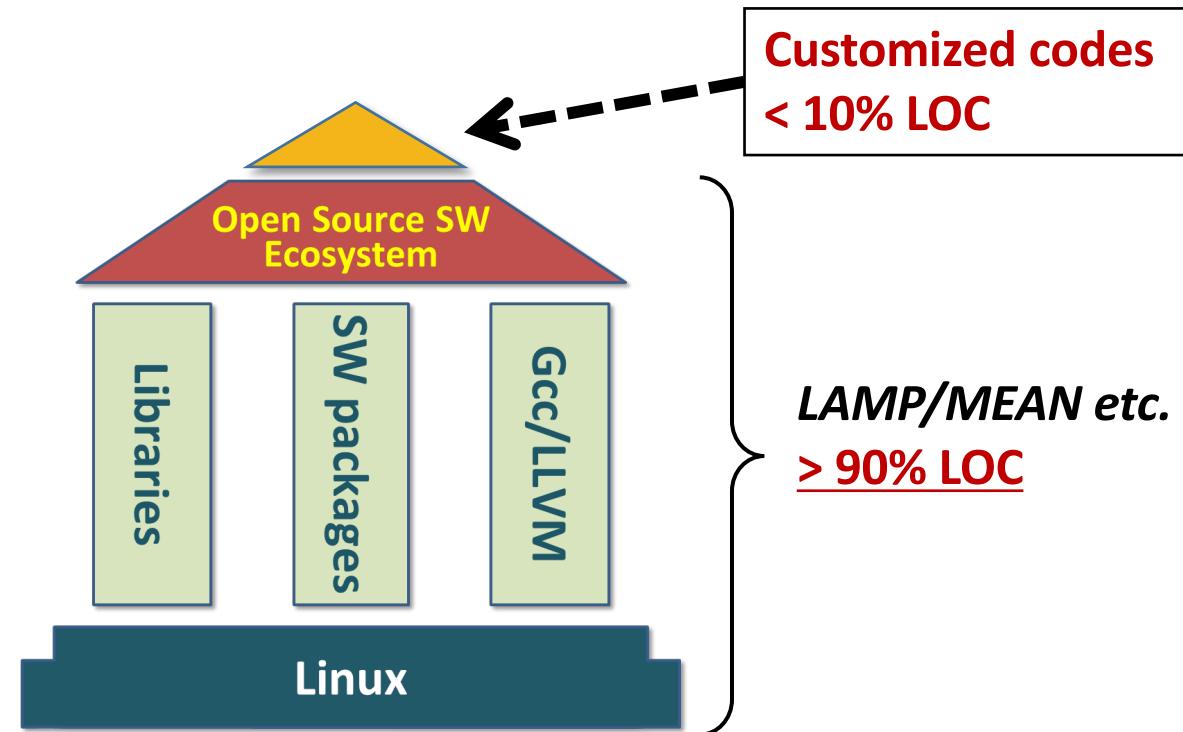
January 9, 2023





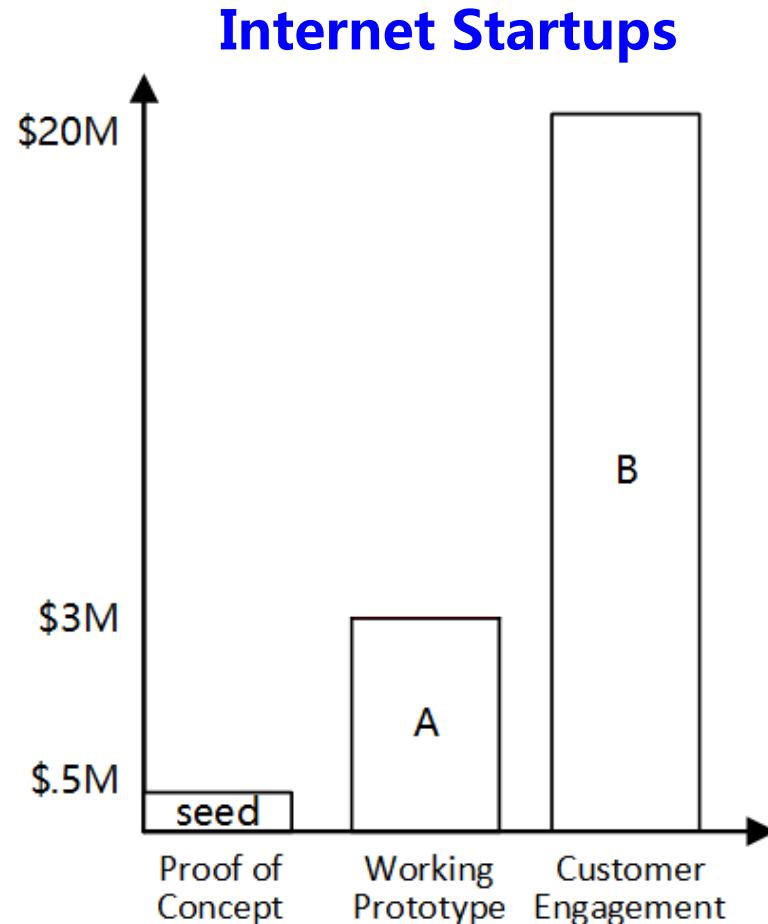
# Open-Source Software Ecosystem

- Lower the barrier of innovation
  - E.g., developing an App by **3-5 engineers** within **3-5 months**

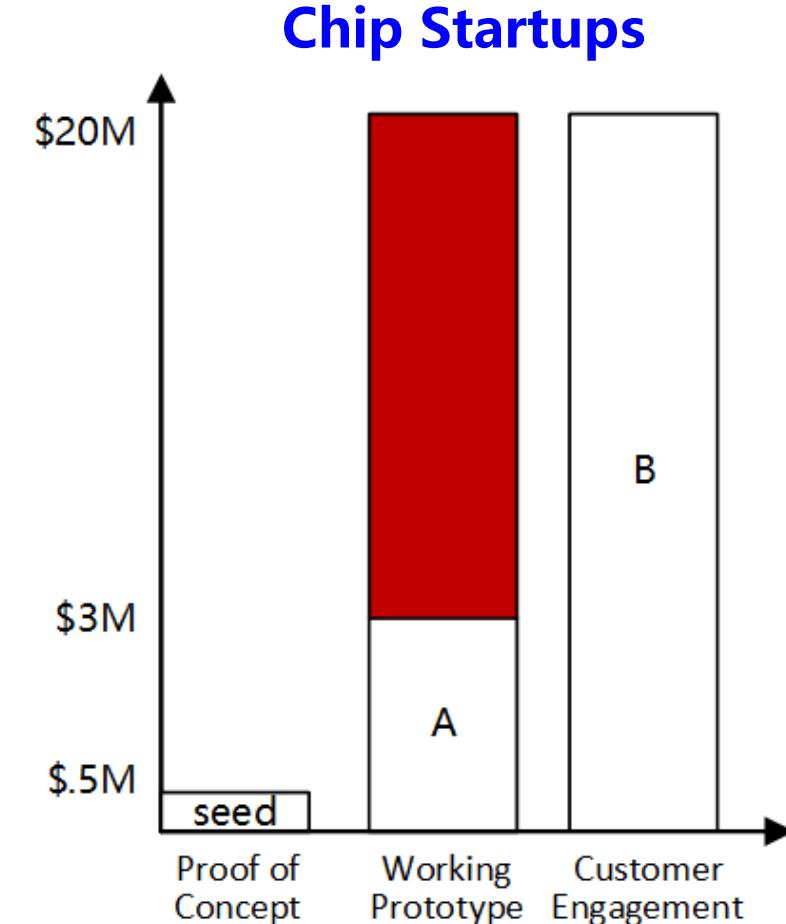




# High Barrier of the Chip Industry



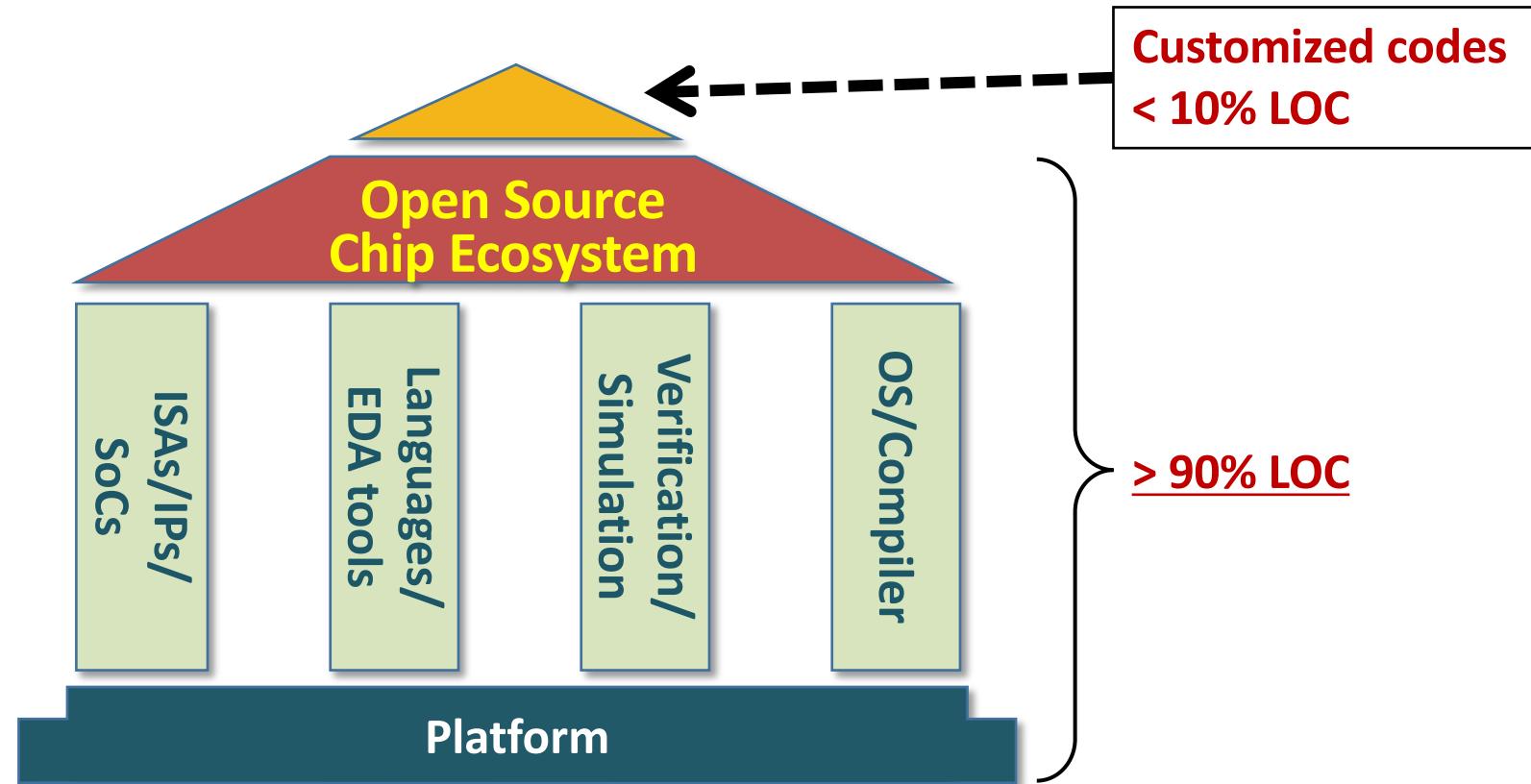
V.S.





# Open-source Chip Ecosystem

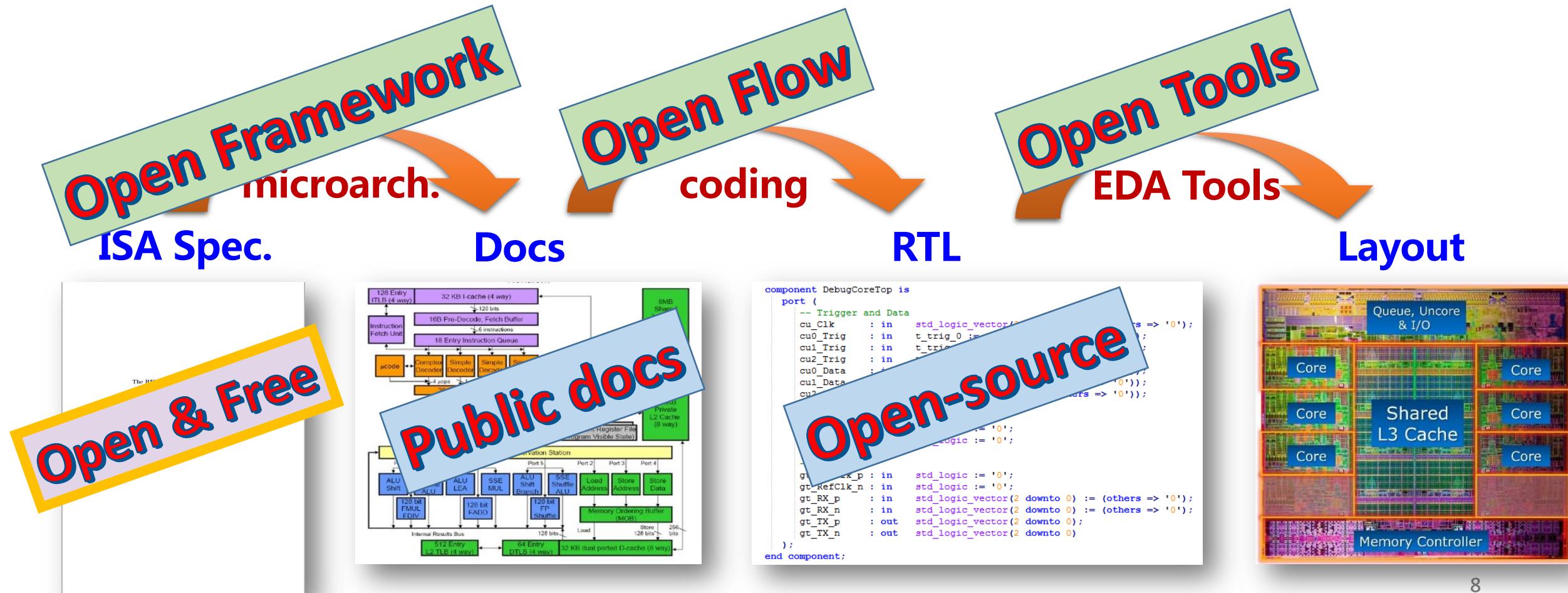
- Lower the barrier of chip development by saving **time-to-market** and the **cost** of IPs, EDA tools, facilities and engineers etc.





# Three Level of Open-Source Chip

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





# 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.

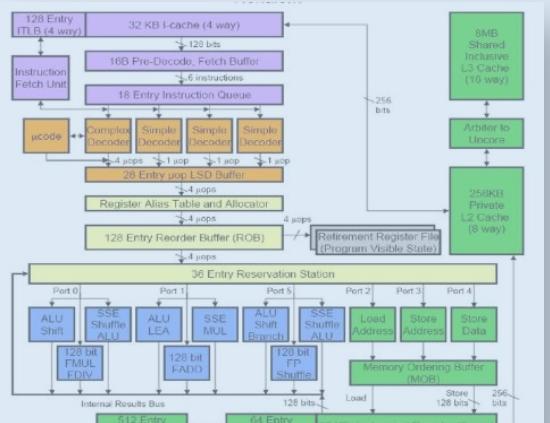
The RISC-V Instruction Set Manual  
Volume I: User-Level ISA  
Document Version 2.2

Editors: Andrew Waterman\*, Kritee Arora\*,<sup>1,2</sup>  
\*SRI International,  
<sup>1</sup>CNS Division, UC Berkeley, Berkeley, California, USA  
kritee@cs.berkeley.edu, kritee@berkeley.edu  
May 7, 2017

1

Open ISA

Docs



2

Open Design/Implt

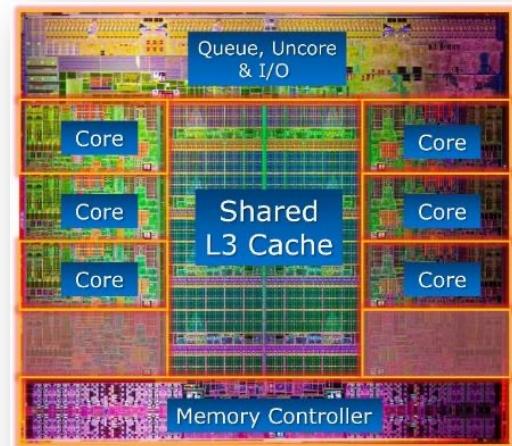
RTL

```
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'));
    cui_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'));
    cui_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';

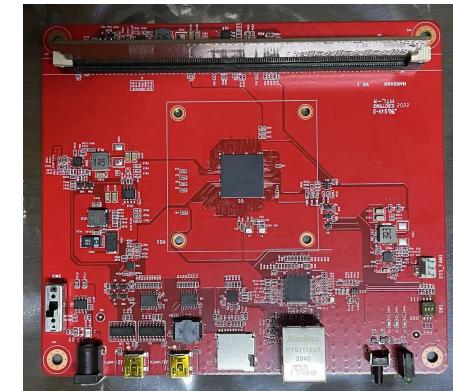
    -- Upstream
    gt_RefClk_p : in std_logic := '0';
    gt_RefClk_n : in std_logic := '0';
    gt_RX_p : in std_logic_vector(2 downto 0) := (others => '0');
    gt_RX_n : in std_logic_vector(2 downto 0) := (others => '0');
    gt_TX_p : out std_logic_vector(2 downto 0);
    gt_TX_n : out std_logic_vector(2 downto 0)
);
```

Layout



## Part II

# XiangShan: Open-Source High Performance RISC-V Processor





# 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/AI, but both academic and industrial community need high-performance RISC-V processors
- Why open-source: An open and innovative hardware platform, “hardware version of Linux”
- Build a leading platform with agile development flows and tools

Specifications	Designs (“Source”)				Products
	Designs Specifications	Free & Open Designs	Licensable Designs	Closed Designs	
Free & Open Spec	“Open Source”				Based on Free & Open Licensed Closed
Licensable Spec					Based on Licensable or Closed
Closed Spec					Based on Closed Designs

Source: David Patterson,  
Keynote @ CRVF 2019,  
<https://crvf2019.github.io/pdf/keynote1.pdf>

# XiangShan: Open-Source High Performance Processors



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



Fragrant Hills in Beijing

A screenshot of the XiangShan GitHub repository page. The repository is named "OpenXiangShan / XiangShan" and is described as a "Public" repository. It has 75 issues, 3 pull requests, and 409 forks. The repository is a "Code" repository with 237 branches and 2 tags. The master branch is active. The repository is described as an "Open-source high-performance RISC-V processor" and includes tags for "chisel3", "risc-v", and "microarchitecture". The repository has 3.3k stars and 75 watching. The GitHub interface shows recent commits from "happy-lx", "fudian", and "huancun".

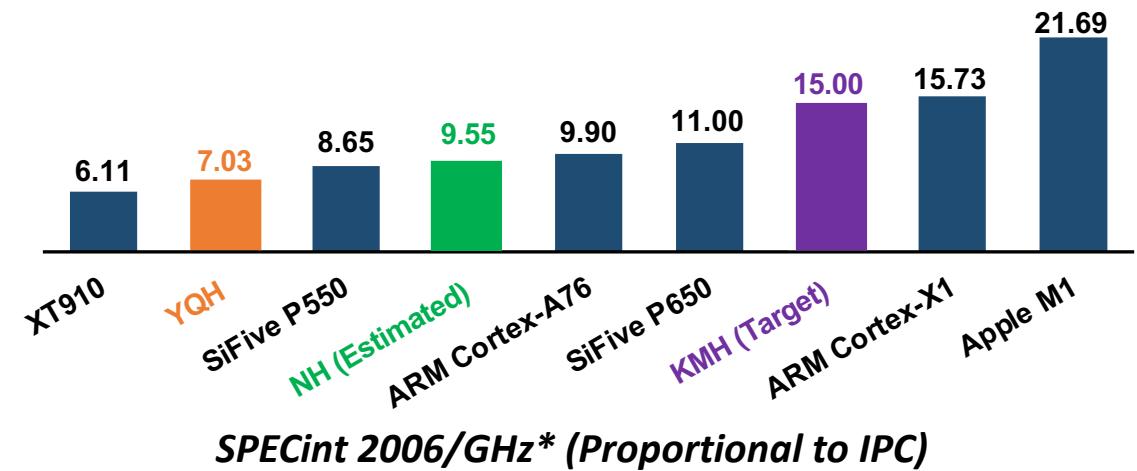
Commit	Message	Date
happy-lx	Fix replay logic in unified load queue (#1966)	yesterday
.github	ci: use checkout@v3 instead of v2 (#1942)	3 weeks ago
debug	bump difftest & mkdir for wave/perf for local-ci script's run-mode (#...)	last month
diffest @ f630d03	bump difftest, track master branch (#1967)	5 days ago
fudian @ 43474be	Switch to asynchronous reset for all modules (#1867)	2 months ago
huancun @ 9a729b9	util: change ElaborationArtefacts to FileRegisters (#1973)	yesterday

>3.3K stars, >400 forks on GitHub



# XiangShan: Open-Source High Performance Processors

- **1<sup>st</sup> generation: YQH**
  - RV64GC, single-core, superscalar OoO
  - **28nm tape-out, 1.3GHz, July 2021**
  - **SPEC CPU2006 7.01@1GHz, DDR4-1600**
- **2<sup>nd</sup> generation: NH**
  - RV64GCBK, dual-core, superscalar OoO
  - **Scheduled 2GHz@14nm tape-out, Q1 2023**
  - **Estimated\*\* SPECint 2006 19.10@2GHz**
- **3<sup>rd</sup> generation: KMH**
  - RV64GCBKHV, quad-core, superscalar OoO
  - **Close collaboration with industrial partners**



*SPECint 2006/GHz\* (Proportional to IPC)*



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

\*\* Updated January 5, 2023



# Yanqihu: 1<sup>st</sup> generation of XiangShan

- Yanqihu: named after a lake in Beijing, China
  - 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 (commit hash ccbca07) with 1MB L2 Cache



Yanqi Lake in Beijing

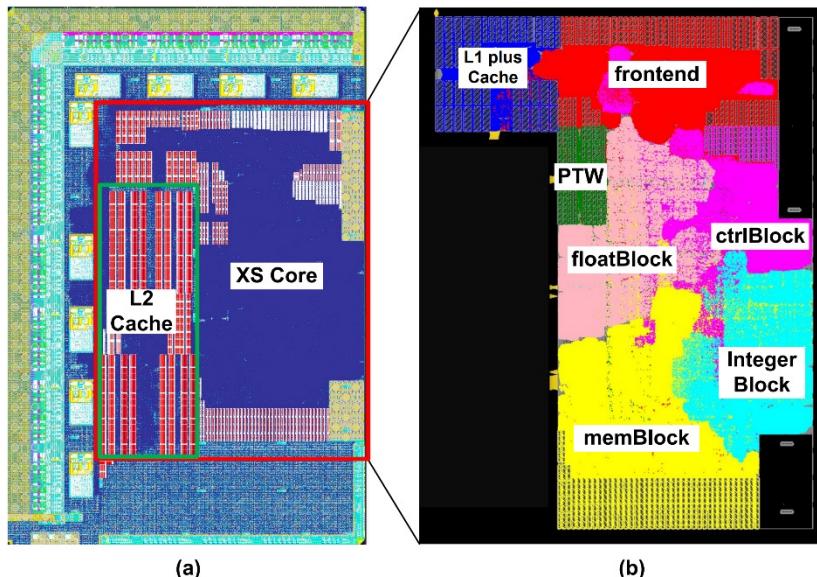
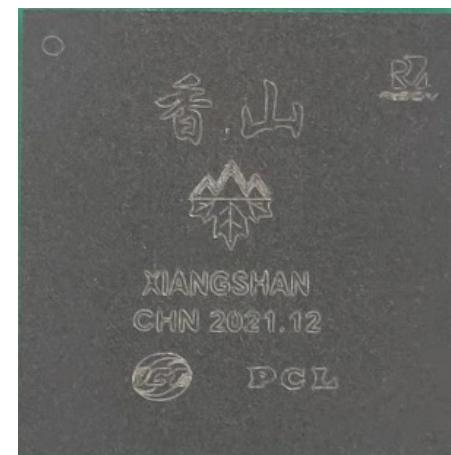


Figure. Layout of (a) the entire chip; (b) the core



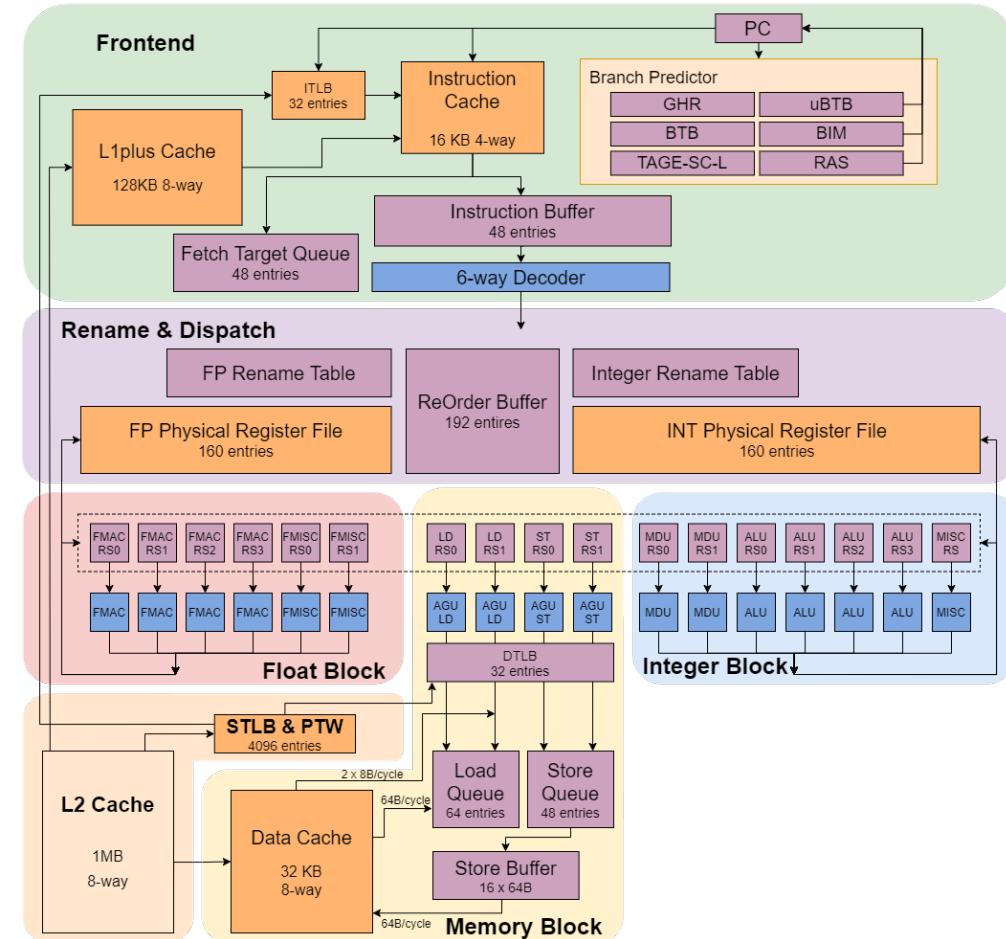
Tape-out information for the processor core	
Process Node	28nm
Die Size	8.6 mm <sup>2</sup>
Std Cell	5.05M, 4.27 mm <sup>2</sup>
Mem	261, 1.7mm <sup>2</sup>
Density	66%
Cell	ULVT 1.04%, LVT 19.32%, SVT 25.19%, HVT 53.67%
Estimated Power	5W
Frequency	1.3GHz, TT85C



# XiangShan microarchitecture (Yanqihu)



- **11-stage, 6-wide decode/rename**
- **TAGE-SC-L** branch prediction
- **160** Int PRF + **160** FP PRF
- **192**-entry ROB, **64**-entry LQ, **48**-entry SQ
- **16**-entry RS for each FU
- **16KB** L1 Cache, **128KB** L1plus Cache for instruction
- **32KB** L1 Data Cache
- **32**-entry ITLB/DTLB, **4K**-entry STLB
- **1MB** inclusive L2 Cache





# Real chip of YANQIHU

- 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	
400.perlbench	6.14
401.bzip2	4.37
403.gcc	6.71
429.mcf	6.83
445.gobmk	7.92
456.hmmer	5.24
458.sjeng	6.85
462.libquantum	17.71
464.h264ref	10.91
471.omnetpp	5.65
473.astar	5.16
483.xalancbmk	7.35

SPECint 2006: 7.03@1GHz  
SPECfp 2006: 7.00@1GHz

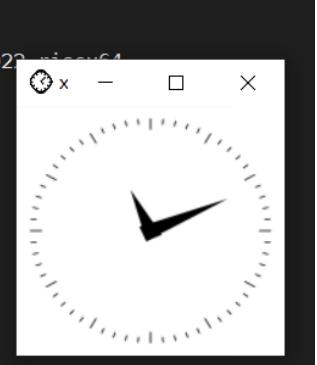
SPECfp 2006 @ 1GHz	
410.bwaves	9.28
416.gamess	6.59
433.milc	8.41
434.zeusmp	7.65
435.gromacs	4.99
436.cactusADM	3.97
437.leslie3d	6.93
444.namd	8.00
447.dealII	10.17
450.soplex	7.03
453.povray	7.14
454.Calculix	2.86
459.GemsFDTD	8.35
465.tonto	6.42
470.lbm	10.39
481.wrf	7.26
482.sphinx3	9.07



```
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 x86_64

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



# NANHU: 2<sup>nd</sup> generation microarchitecture

- 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
- Tape-out with dual cores (RV64GCBK), more devices support (PCIe, USB, ...)
- *To build an open and standardized development workflow*



NANHU Seminar, 2021.6.19

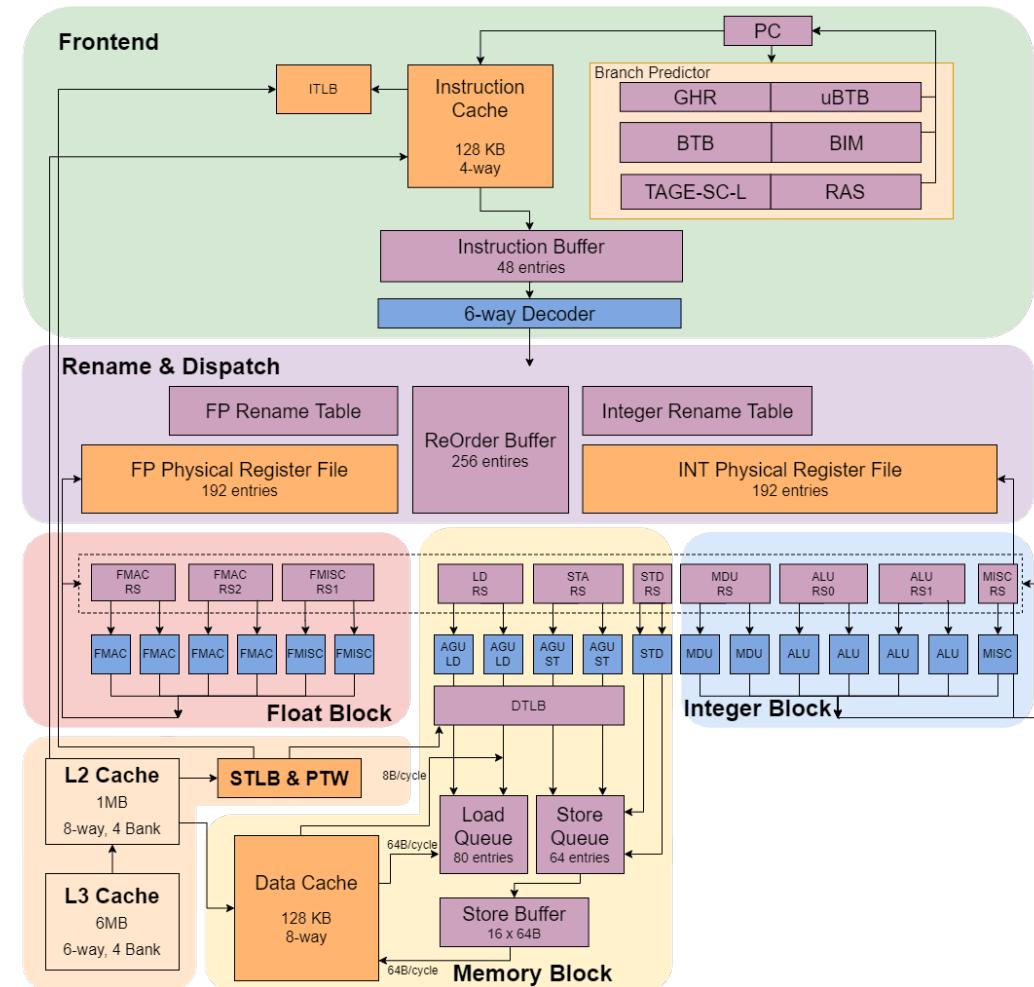


A lake in Jiaxing, Zhejiang, China



# XiangShan microarchitecture (Nanhу)

- **192** Int PRF + **192** FP PRF
- **256**-entry ROB, **80**-entry LQ, **64**-entry SQ
- **16**-entry RS for each FU (32-entry as an RS)
- **64KB** L1 Cache, **64KB** L1 Data Cache
- **136**-entry DTLB, **40**-entry ITLB, **2K**-entry STLB
- **1MB** non-inclusive L2 Cache
- **6MB** non-inclusive L3 Cache (LLC)
- **BOP and SMS** prefetchers at L2



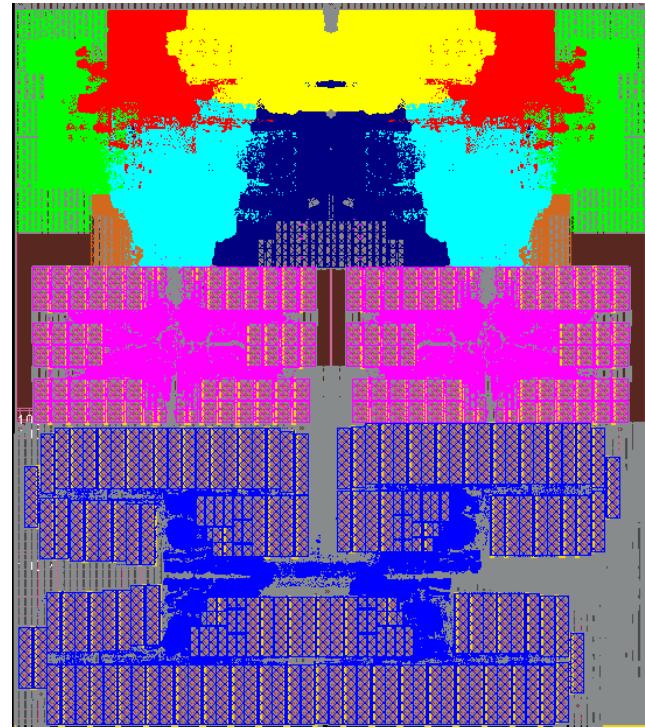


# Estimated Performance of NANHU

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

400.perlbench	19.27
401.bzip2	11.36
403.gcc	21.97
429.mcf	20.53
445.gobmk	15.97
456.hmmr	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
<b>SPECint2006@2GHz</b>	<b>19.10</b>

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.dealll	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
<b>SPECfp2006@2GHz</b>	<b>22.18</b>



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



# KUNMINGHU: 3<sup>rd</sup> generation microarchitecture

- Work with *Beijing Institute of Open Source Chip (BOSC)*
  - Non-profit organization with 18 founding members from the industry
- RISC-V Vector Extension 1.0 Support
- RISC-V Hypervisor Extension 1.0 Support
- Loop predictor, loop buffer
- Refactored issue queues, execution units, load/store units/queues
- L1D prefetcher, cross-level cache optimizations
- AMBA CHI compatible
- Comprehensive verification plan
- Advanced process node and EDA flows
- GEM5 simulator with aligned microarchitecture



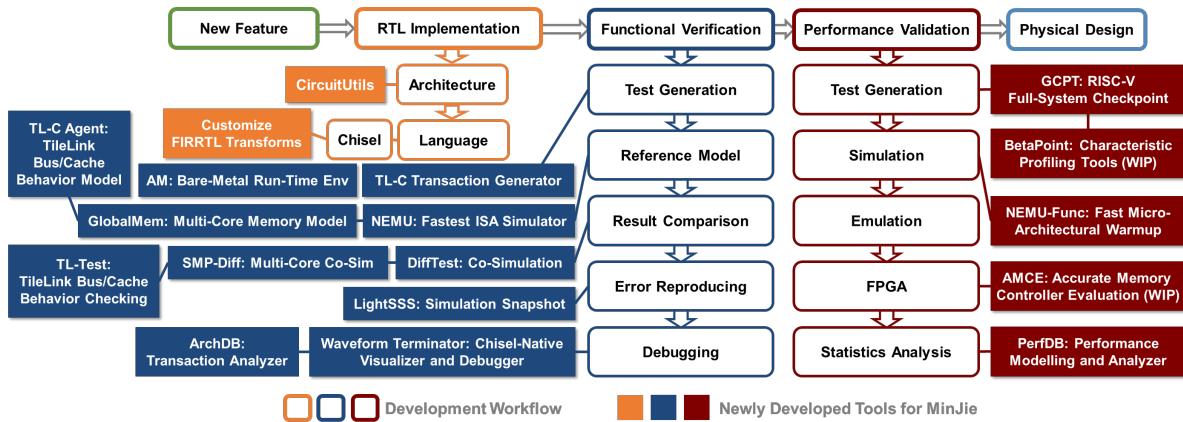
The screenshot shows a GitHub repository's commit history across different branches. The main branch is 'master'. Commits are grouped by date:

- Commits on Mar 19, 2023:
  - Fix replay logic in unified load queue (#1966) - happy-lx committed 2 days ago ✓ Verified 624fdecc
  - util: change ElaborationArtifacts to FileRegisters (#1973) - Maopice-U committed 2 days ago ✓ Verified 87619eb
- Commits on Mar 16, 2023:
  - dcache: optimize the ready signal of missqueue (#1965) - happy-lx committed 5 days ago ✓ Verified 6008857
  - bump diffest, track master branch (#1967) - Lemover committed 5 days ago ✓ Verified ee44c19
- Commits on Mar 15, 2023:
  - MMU: Add sector tlb for larger capacity (#1964) - good-circle committed last week ✓ Verified 6361282
- Commits on Mar 13, 2023:
  - dcache: fix pilru update logic (#1921) - AugusteWillSwing committed last week ✓ Verified fa5ac9b
- Commits on Feb 27, 2023:
  - c: use checkout@v3 instead of v2 (#1942) - Tang-Haijin committed 3 weeks ago ✓ Verified 33d13d4
- Commits on Feb 22, 2023:
  - ftq: revert #1875, #1920 (#1931) - sfencevm and Lyn committed last month ✓ Verified 10a8efc

Active development on GitHub

## Part III

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





# Decision: Use Chisel

- 2018: quantitative experiments between Chisel and Verilog

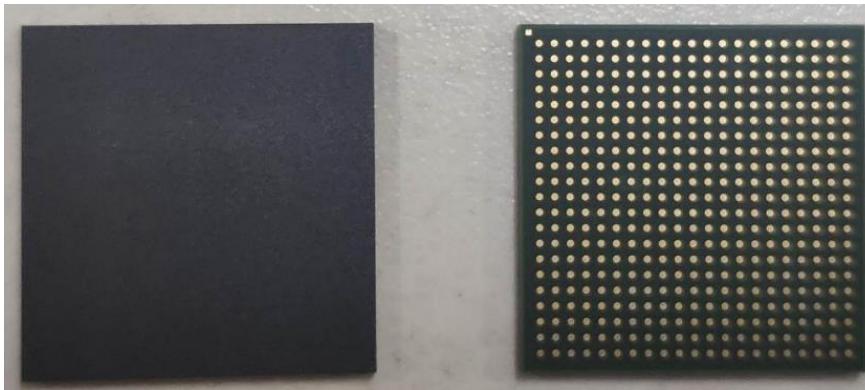
• Task #1: Design an L2 Cache for RISC-V Rocket-chip core	
• Who: A 5-year engineer vs. a senior student	
A 5-year Engineer	An Undergraduate
Experience	Familiar w/ OpenSparc T1; Modified Xilinx Cache
Language	Verilog
Time	6 weeks
LOCs	~1700
Results	Unable to boot Linux
• 1 <sup>st</sup> Round results: 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
Power/W	0.770
LUT Logic	5676
LUT Storage	1796
FF	4266
LOCs	618
	136.388 (+0.42%)
	0.749 (-2.73%)
	6422 (+13.14%)
	2594 (-54.30%)
	1264 (-29.62%)
	1492 (-16.93%)
	3638 (-14.72%)
	747 (-82.49%)
	470 (-23.95%)
	155 (-74.92%)

• 2<sup>nd</sup> 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.

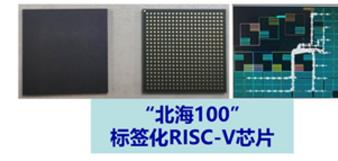
- 2020: 28-nm tape-out of an 8-core labeled RISC-V processor



16节点原型系统机箱



单节点板卡

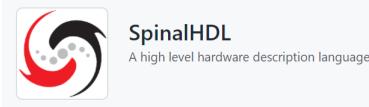
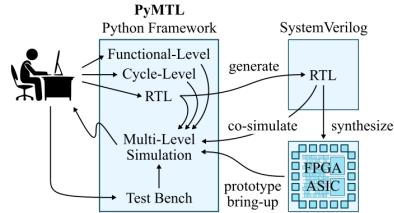


“北海100”  
标签化RISC-V芯片

- 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

# Agile design is easy

CHISEL



bluespec

cucapra/latte21

Languages, Tools, and Techniques for Accelerator Design



Scala

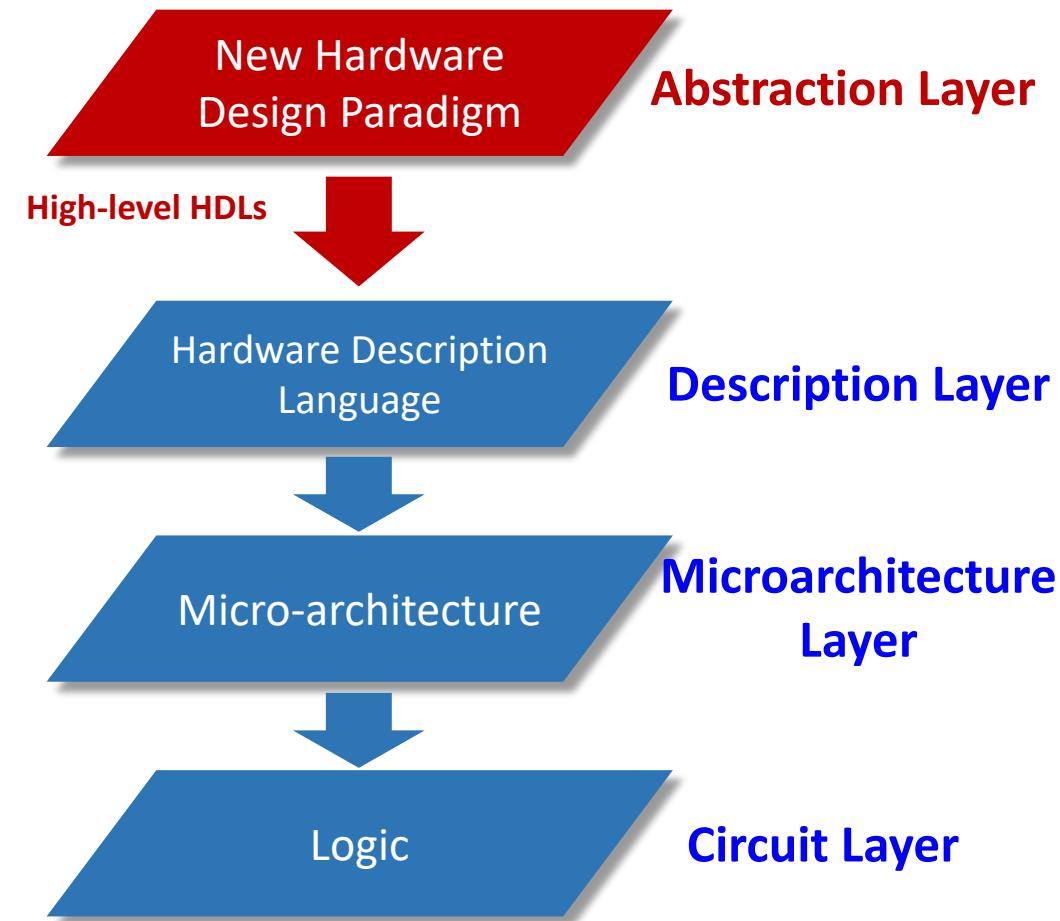
python™

OCaml

Verilog

SystemVerilog

Agile Design Languages



Agile Design Methodology

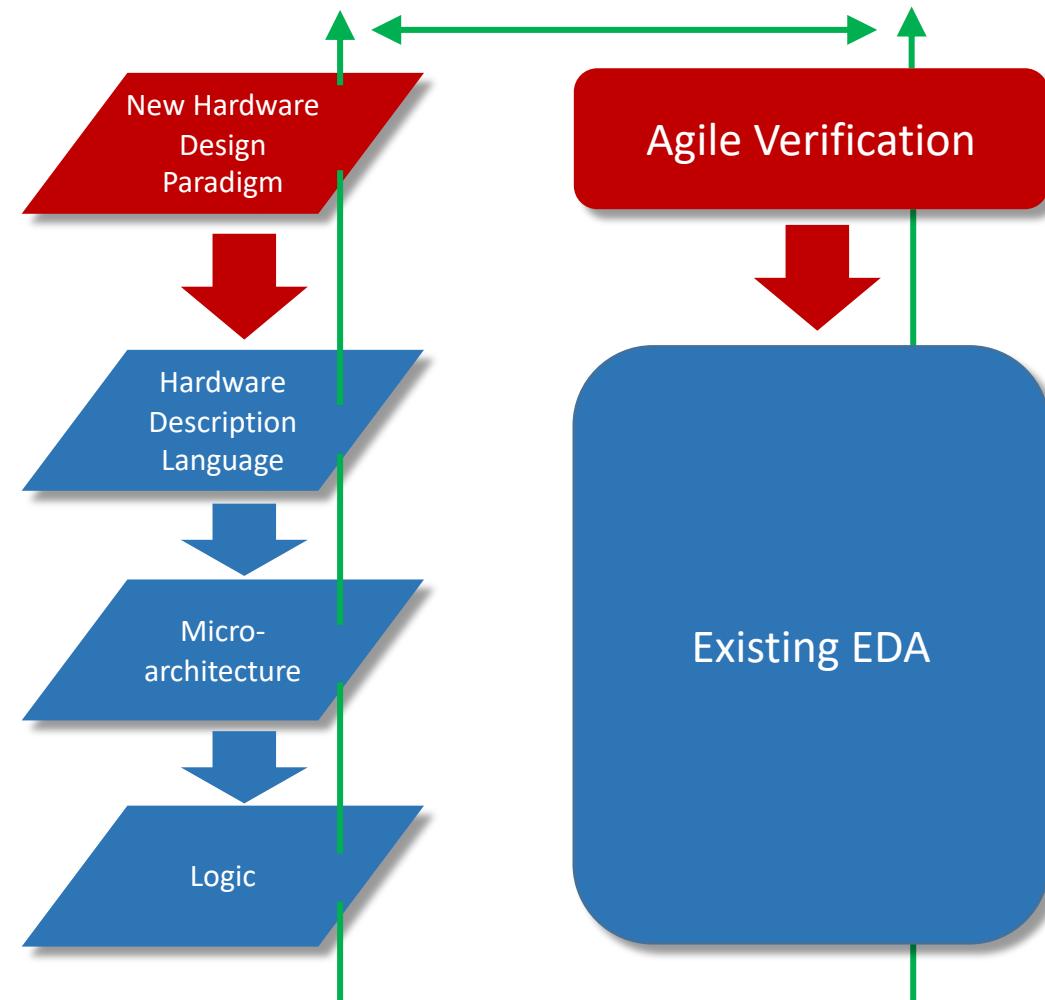
# Agile verification is hard

CHISEL

FIRRTL

Scala

Verilog

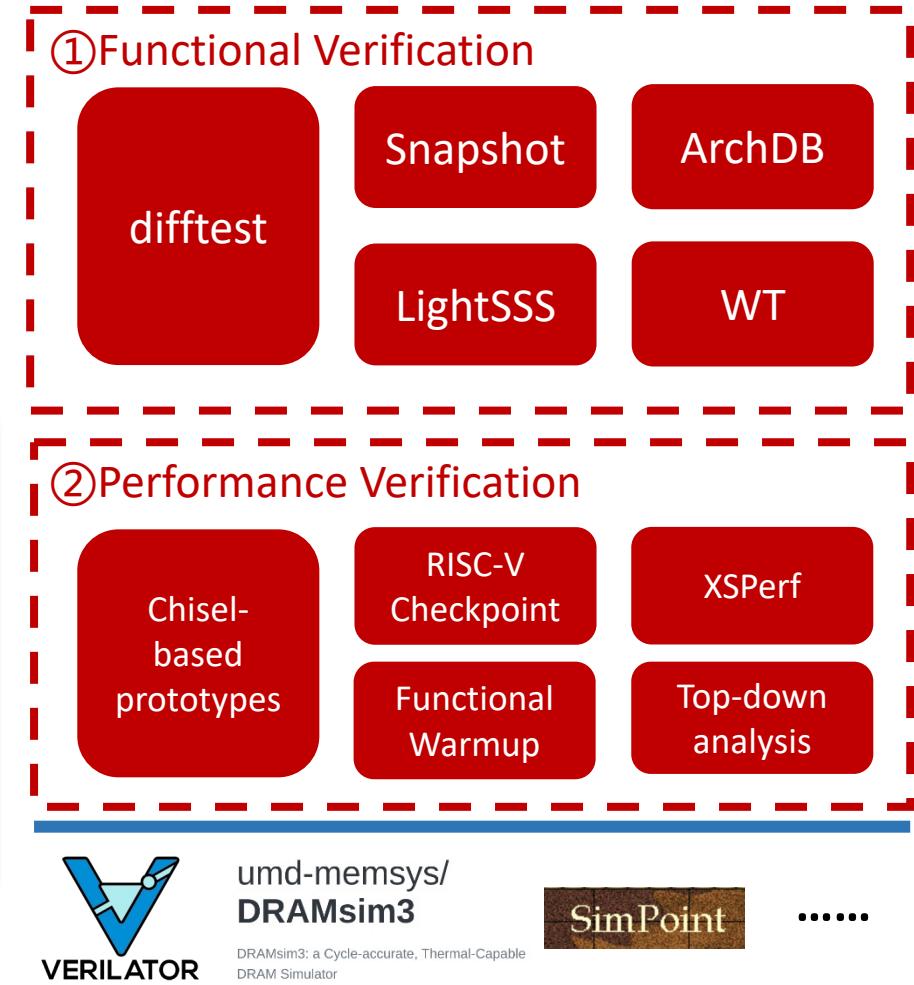


Agile Design Languages

Agile Design Method

Agile Verification Method

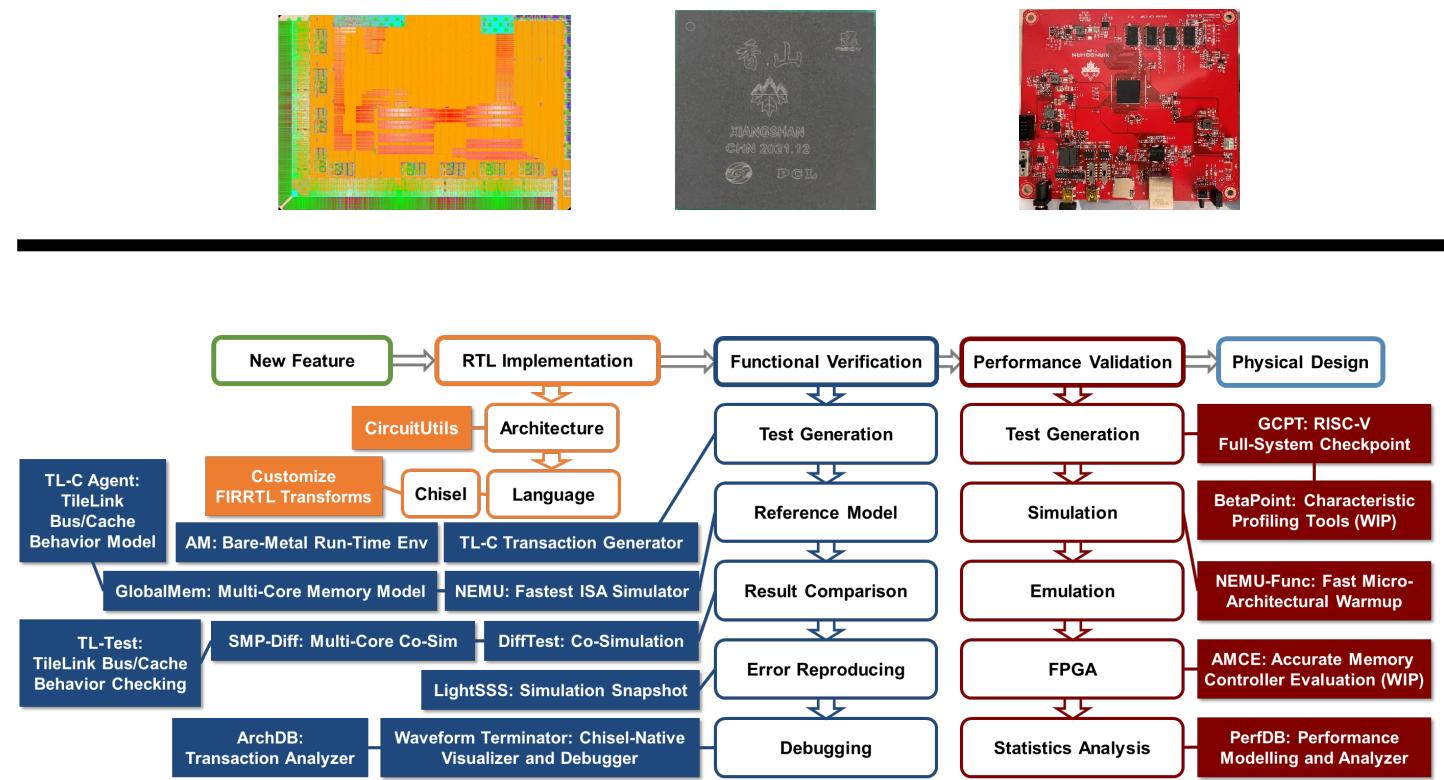
Agile Verification Tools





# Decision: Open & Agile Development Infrastructure

- Key outcome of the XiangShan Project
- Open source to benefit academia and industry





# XiangShan achieves L2.5

L1: OPEN ISA



L2: OPEN Design/Implementation



L3: OPEN Framework/Tools



3

**Open Framework/Tools**

microarch.

coding

EDA Tools

Layout

**ISA Spec.**

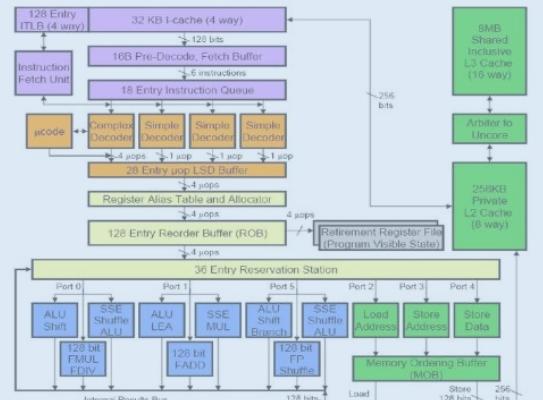
The RISC-V Instruction Set Manual  
Volume I: User-Level ISA  
Document Version 2.2

Editors: Andrew Waterman\*, Kritee Aswani<sup>1,2</sup>  
\*CS Division, EECS Department, University of California, Berkeley  
[andrewwaterma.com](http://andrewwaterma.com), [kritee@berkeley.edu](mailto:kritee@berkeley.edu)  
May 7, 2017

1

**Open ISA**

**Docs**



2

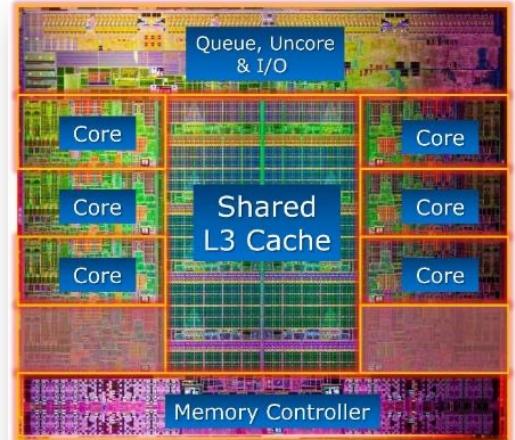
**Open Design/Implt**

**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'));
    cul_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'));
    cul_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;
```



## Part IV

# Conclusion





# If you are doing research on ...

- **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, compilers, ...*

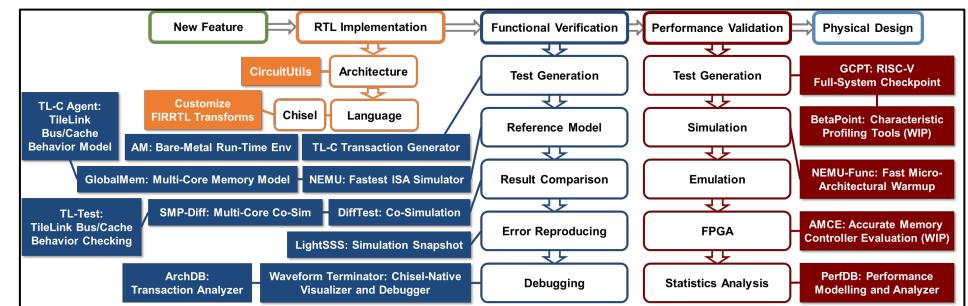


<https://midgard.epfl.ch/>

Imprecise Store Exceptions, ISCA'23 (EPFL)

- **Agile Chip Development**

- XiangShan is a progressive, configurable, complicated, challenging benchmark
  - MinJie provides a good startpoint
- 
- *HCLs, verification, performance, power, area, prototyping, DFT, synthesis, placement, routing, ECO, ...*



MinJie, MICRO'22, selected as IEEE Micro Top Pick



一起向未来  
*Together for a Shared Future*

Thanks!

