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
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

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

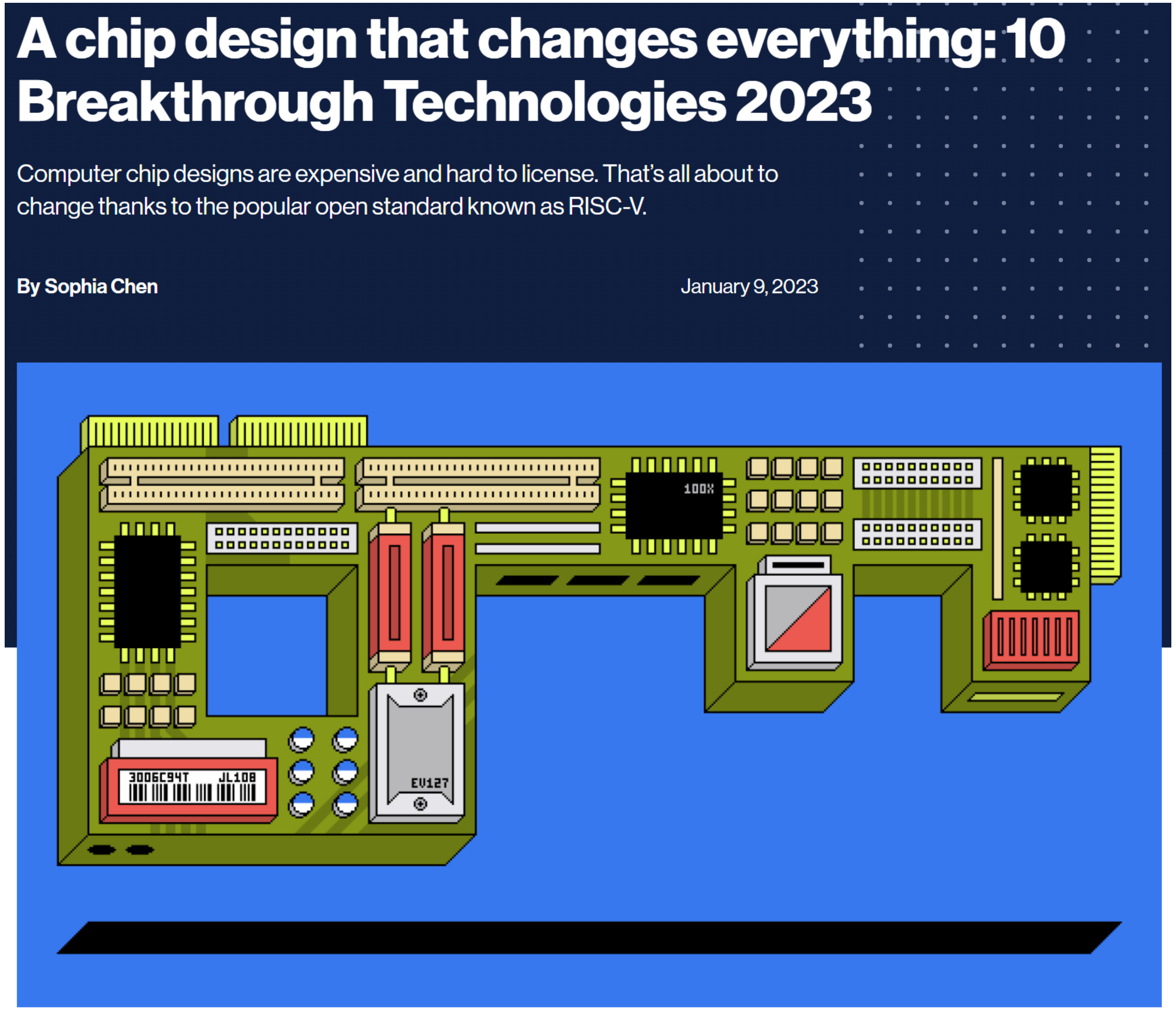
A chip design that changes everything

- **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.

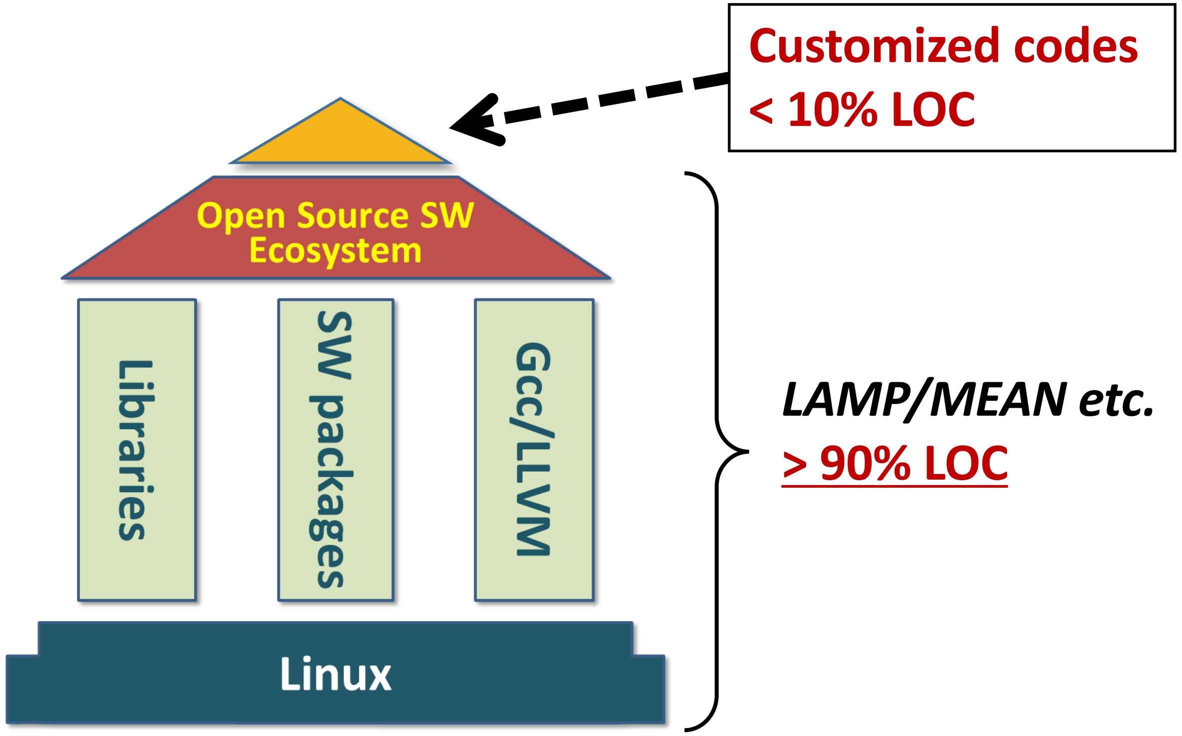
Now an open standard known as RISC-V (pronounced "risk five") could change how companies create computer chips.

--- MIT Technology Review

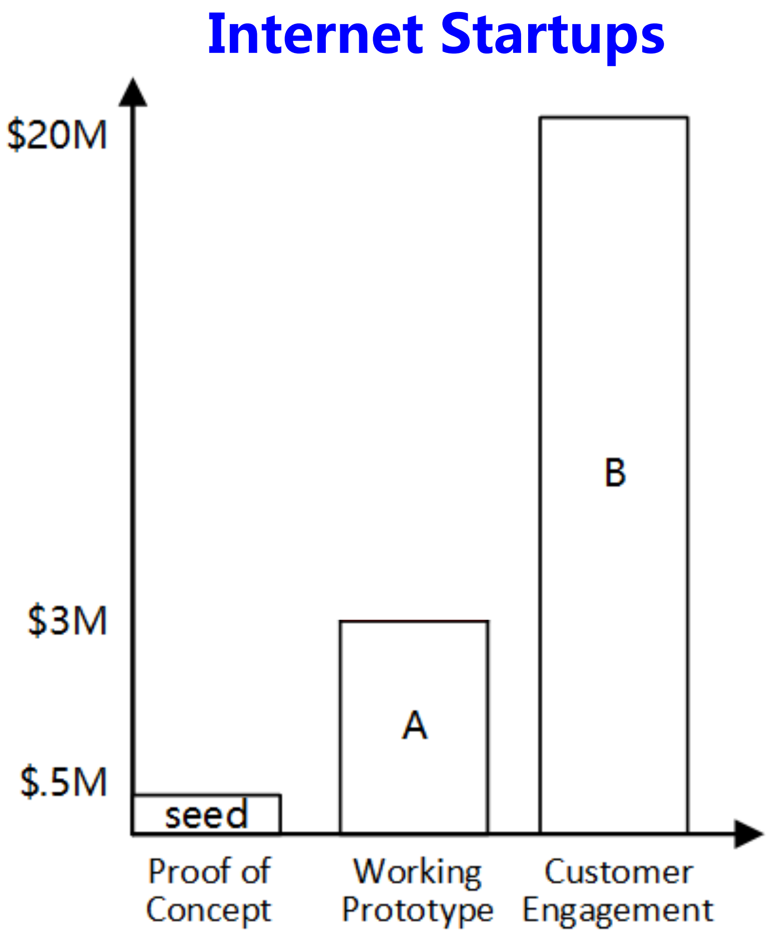


Open-Source Software Ecosystem

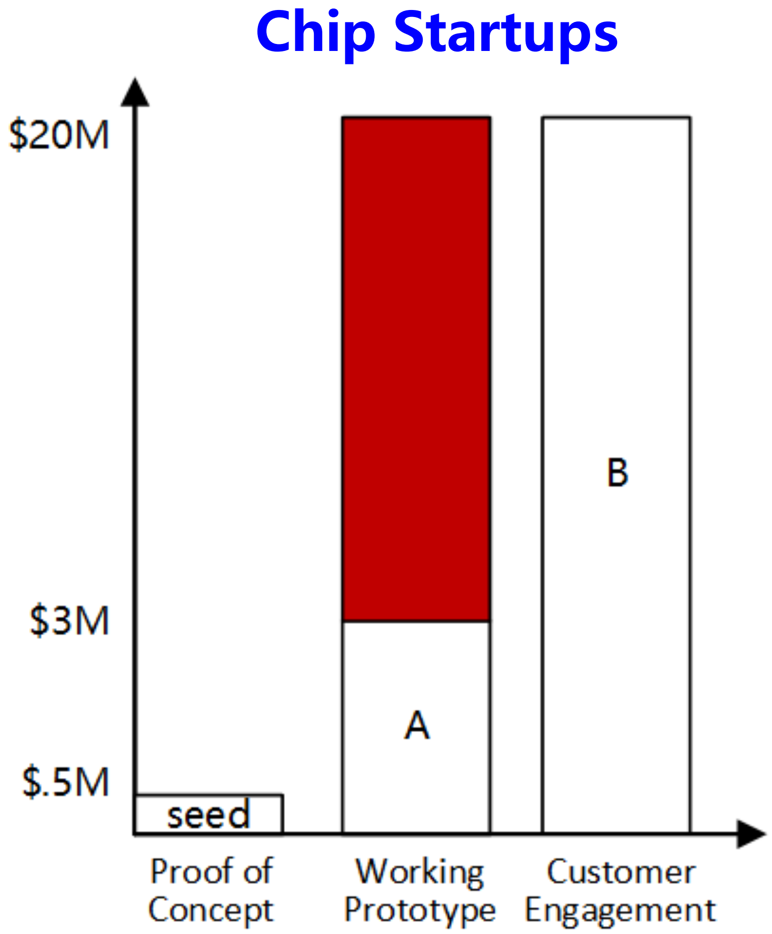
- **Lower the barrier of innovation**
  - E.g., developing an App by **3-5 engineers** within **3-5 months**



High Barrier of the Chip Industry



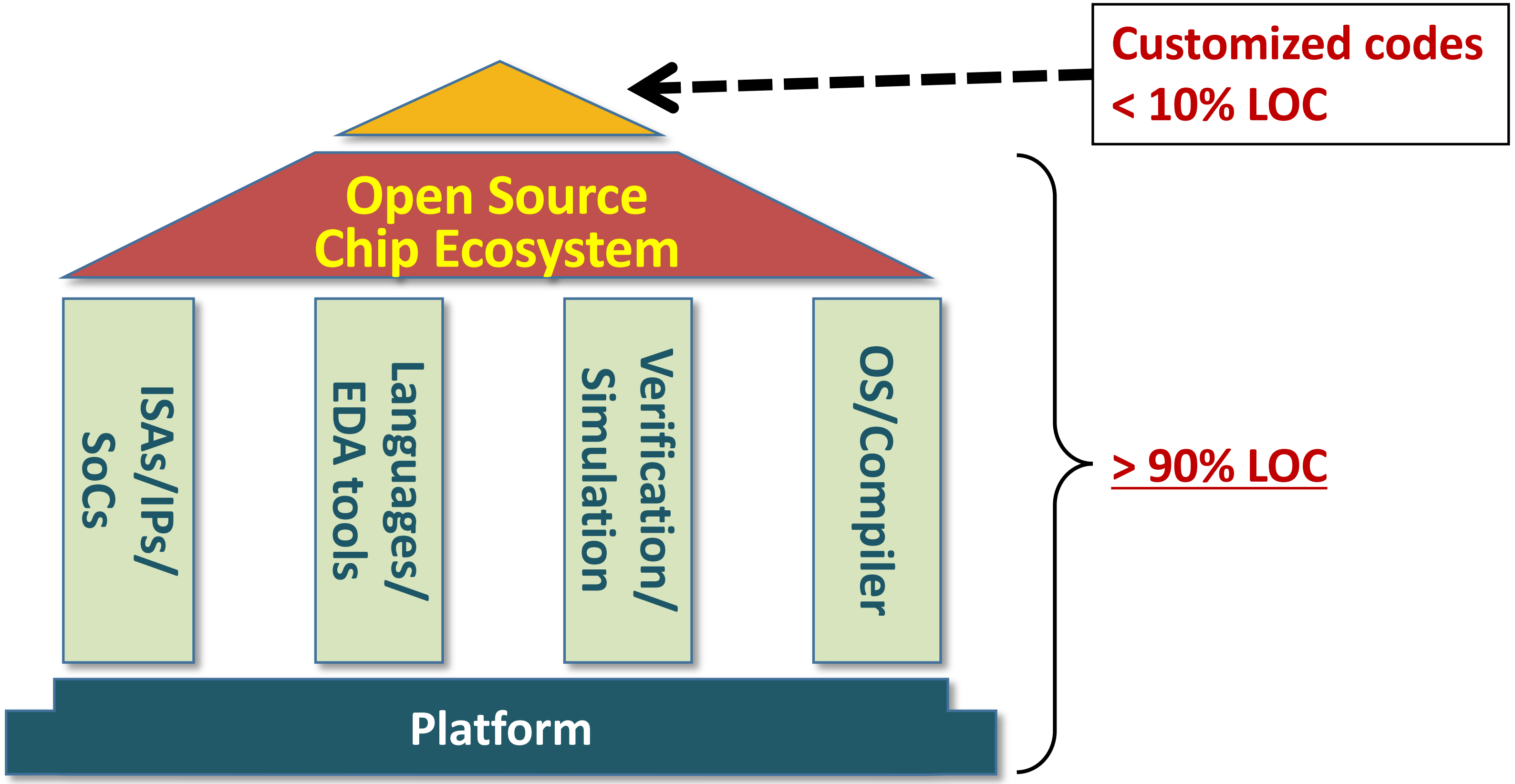
V.S.



Serge Leef, DARPA, Automatic Implementation of Secure Silicon, SIGARCH Visioning Workshop, 2019

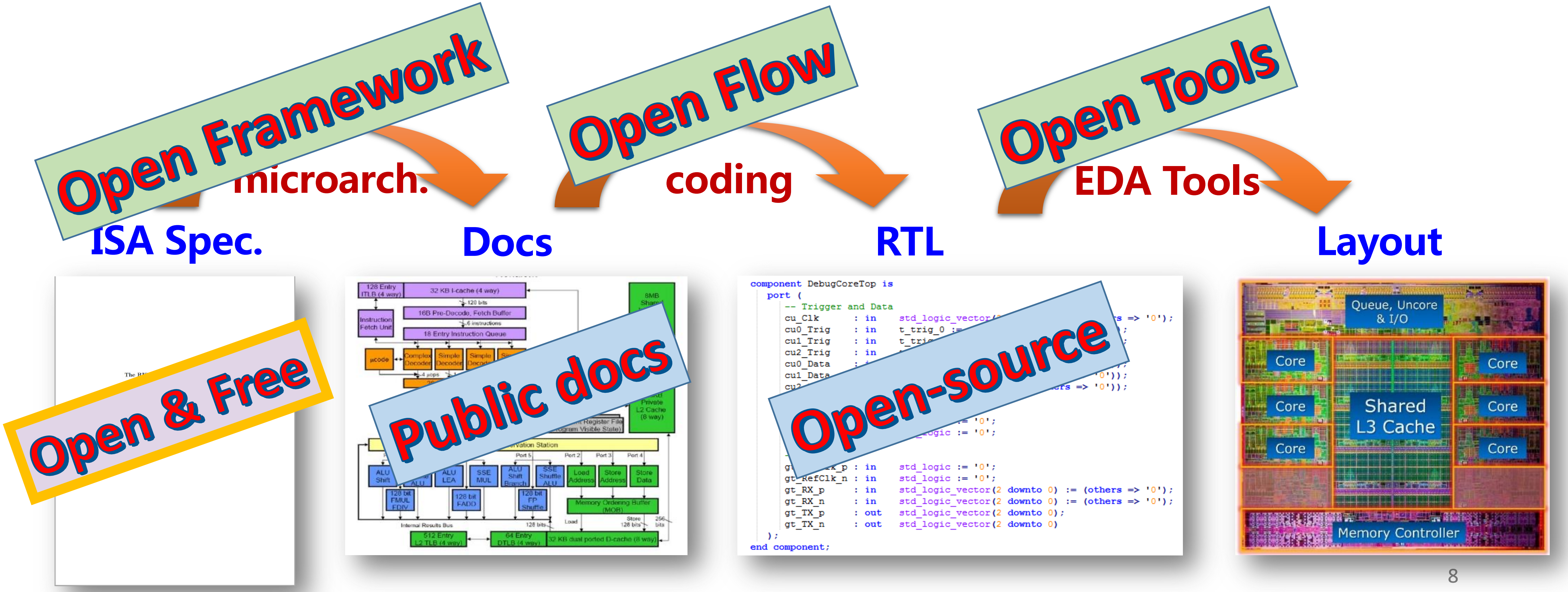
Open-source Chip Ecosystem

- 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

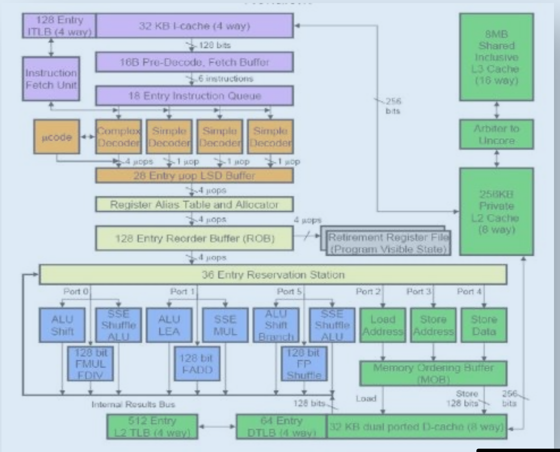
ISA Spec.

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

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

1 Open ISA

Docs



RTL

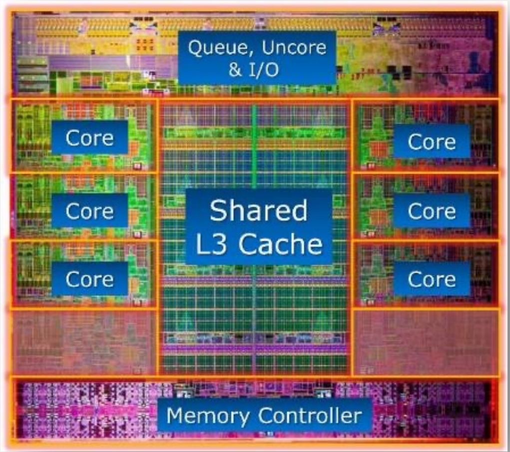
```vhdl
component DebugCoreTop is
  port (
    -- Trigger and Data
    cu_Clk       : in    std_logic_vector(2 downto 0) := (others => '0');
    cu0_Trig     : in    t_trig_0 := (others => (others => '0'));
    cu1_Trig     : in    t_trig_1 := (others => (others => '0'));
    cu2_Trig     : in    t_trig_2 := (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;
```

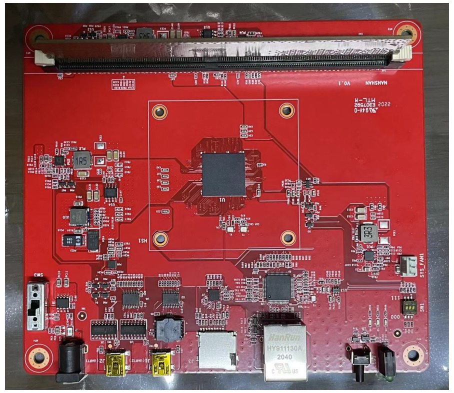
2 Open Design/Implt

Layout



Part 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 Licensed or Closed
Closed Spec				Based on Closed Designs

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

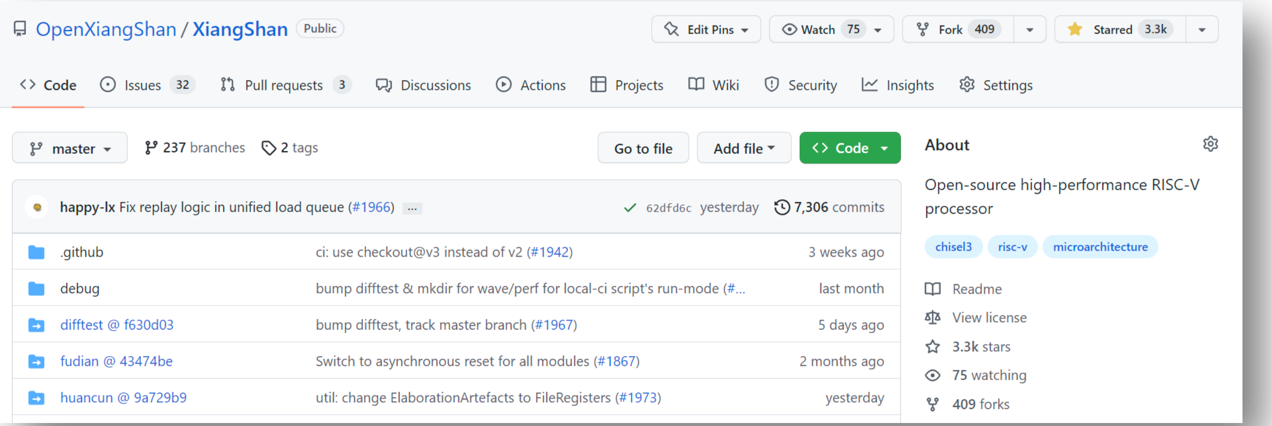
XiangShan: Open-Source High Performance Processors

L1: OPEN ISA

L2: OPEN Design/Implementation

L3: OPEN Framework/Tools

Fragrant Hills in Beijing



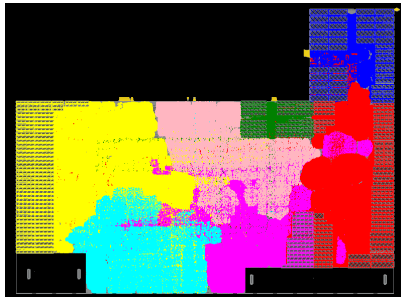
>3.3K stars, >400 forks on GitHub

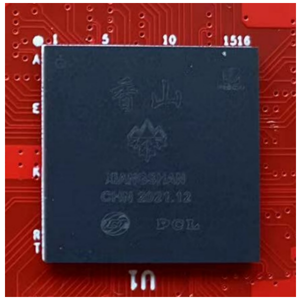
XiangShan: Open-Source High Performance Processors

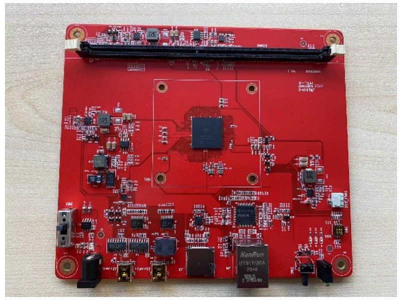
- **1st generation: YQH**
  - RV64GC, single-core, superscalar OoO
  - **28nm tape-out, 1.3GHz, July 2021**
  - **SPEC CPU2006 7.01@1GHz, DDR4-1600**

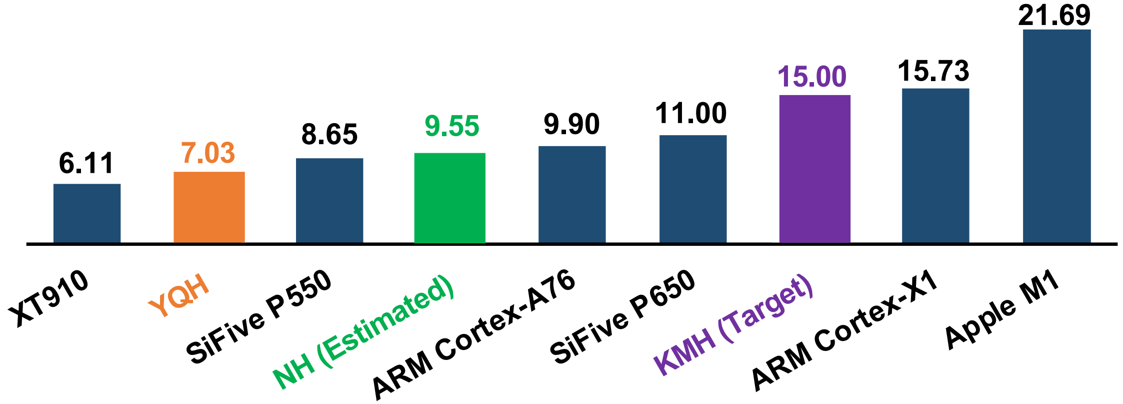
- **2nd generation: NH**
  - RV64GCBK, dual-core, superscalar OoO
  - **Scheduled 2GHz@14nm tape-out, Q1 2023**
  - **Estimated** SPECint 2006 19.10@2GHz**

- **3rd generation: KMH**
  - RV64GCBKHV, quad-core, superscalar OoO
  - **Close collaboration with industrial partners**









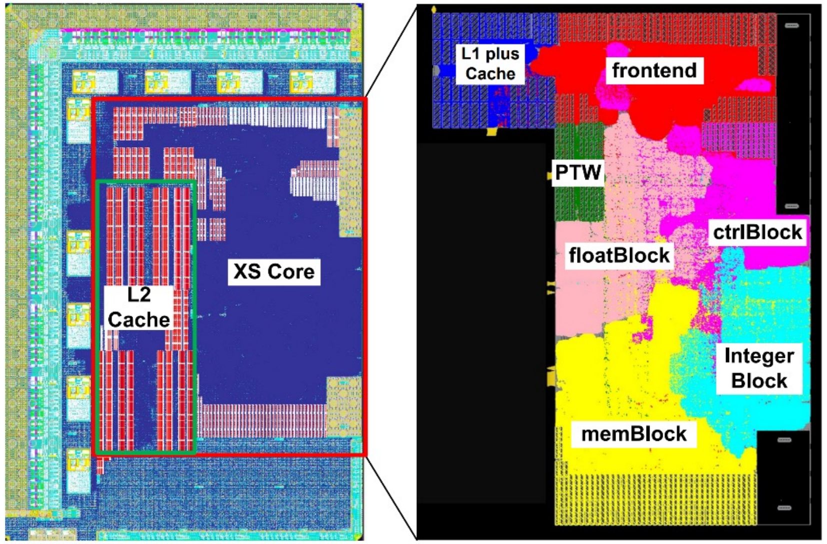
SPECint 2006/GHz (Proportional to IPC)*

* Source: XT910@ISCA'20, SiFive, AnandTech
** Updated January 5, 2023

Yanqihu: 1st generation of XiangShan

Yanqi Lake in Beijing

- **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



(a) (b)

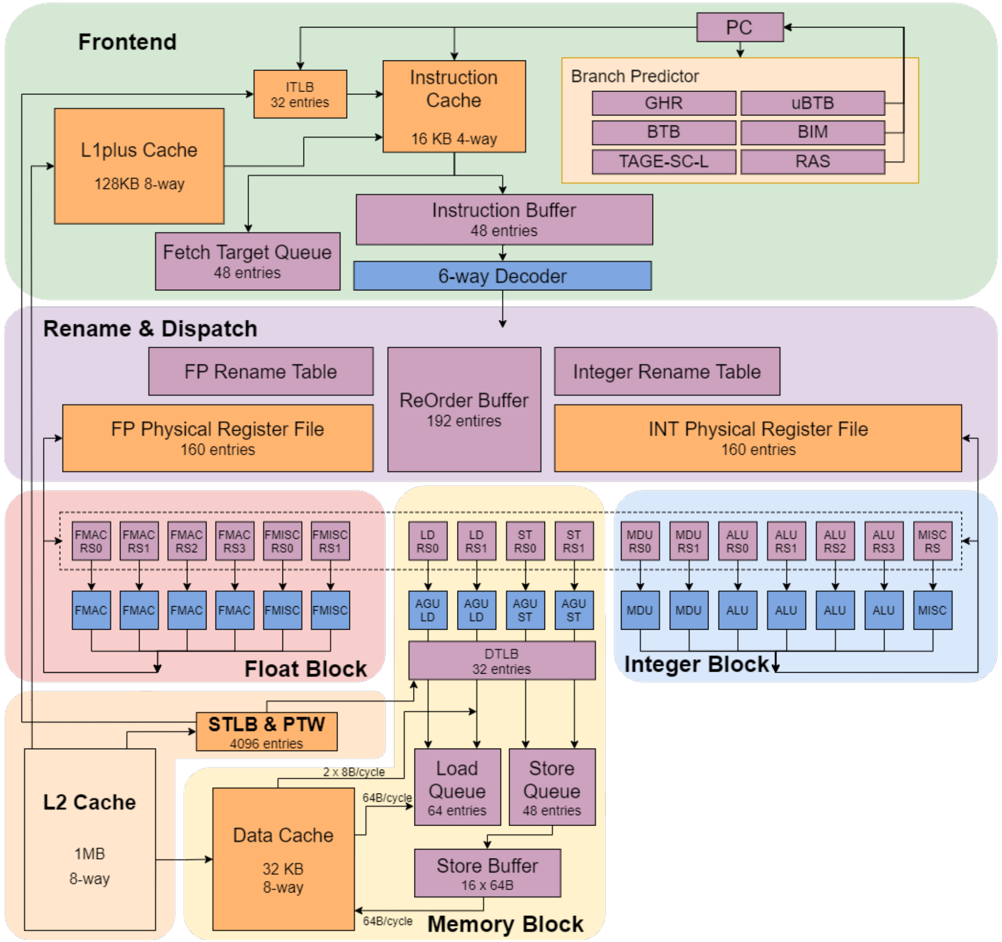
Figure. Layout of (a) the entire chip; (b) the core

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

XiangShan microarchitecture (Yanqihu)

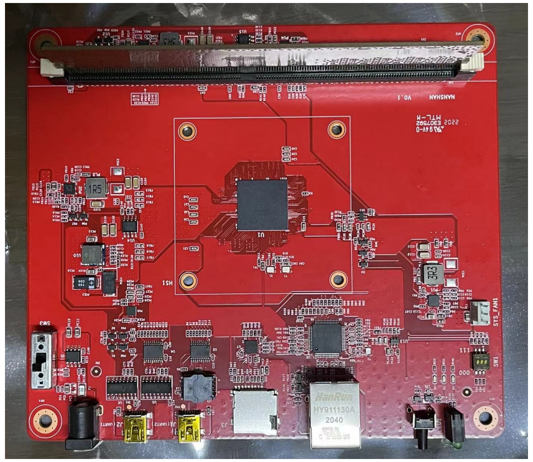
IF1 → IF2 → IF3 → IF4 → DEC → REN → DP → RF → ISS → EXE → CM

- **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

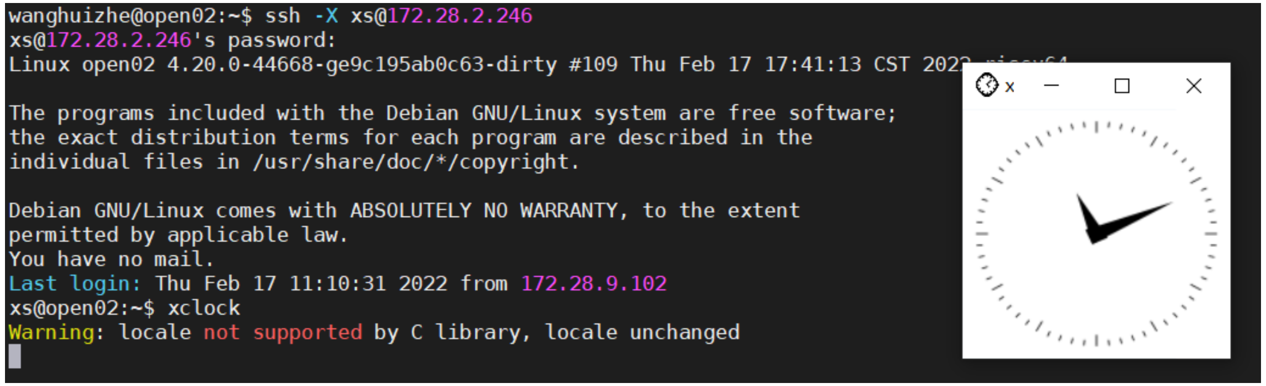
- **The chip was back in January 2022**
  - SoC: CPU, SPI Flash, UART, SD card, Ethernet, DIMM
  - Correctly running Debian with SD card and ethernet
- **Performance: SPEC CPU2006 7.01@1GHz**



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

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

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



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

NANHU: 2nd 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, …)

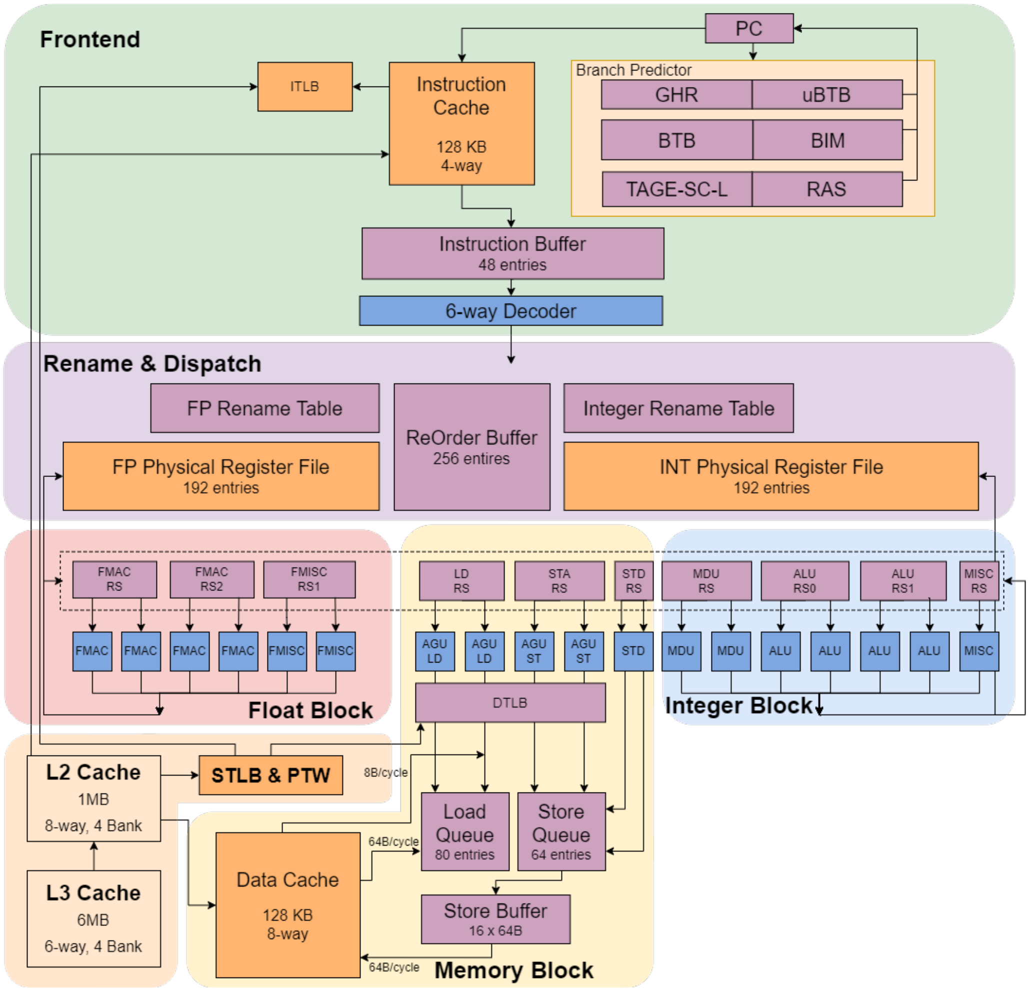
- *To build an open and standardized development workflow*

NANHU Seminar, 2021.6.19

A lake in Jiaxing, Zhejiang, China

XiangShan microarchitecture (Nanhu)

- **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

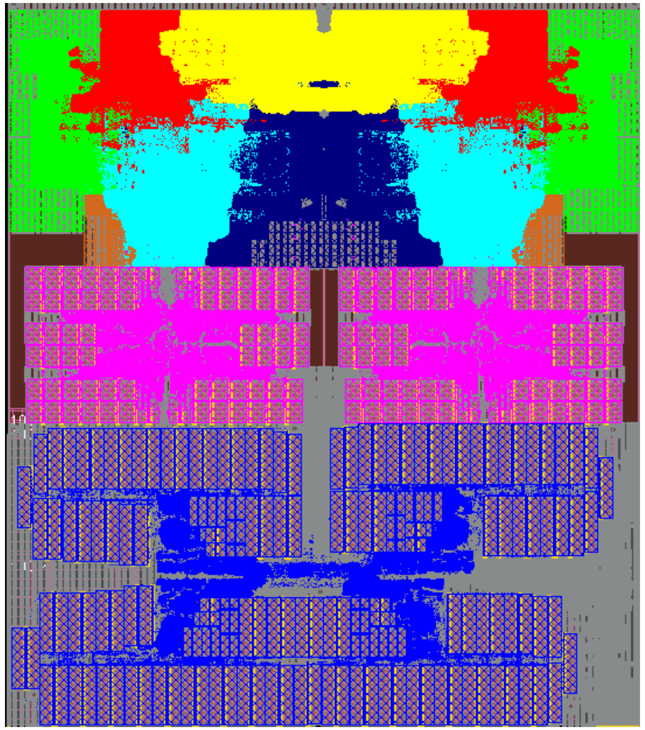


Estimated Performance of NANHU

- **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.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***

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	YQH	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

*** Updated January 5, 2023**

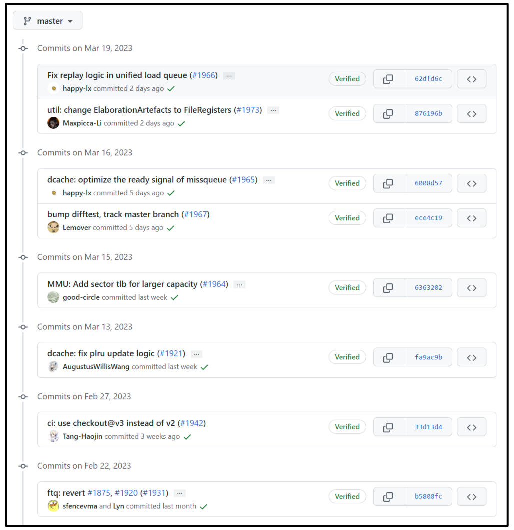
KUNMINGHU: 3rd generation microarchitecture

- Work with *Beijing Institute of Open Source Chip (BOSC)*
  - Non-profit organization with 18 founding members from the industry

BOSC

- 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

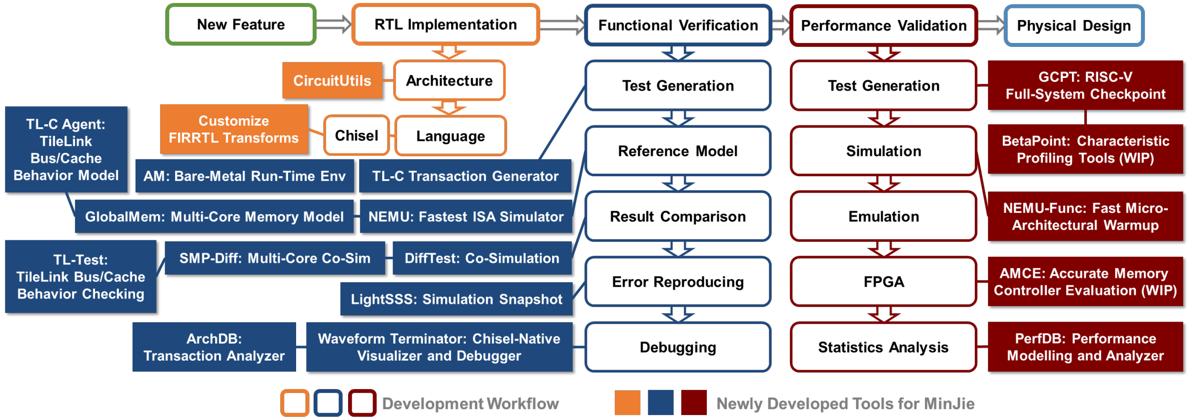
- Comprehensive verification plan
- Advanced process node and EDA flows
- GEM5 simulator with aligned microarchitecture



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	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*

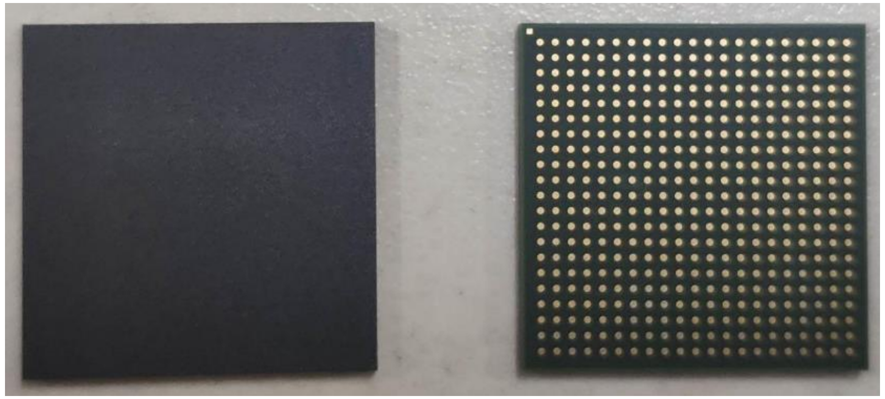
- **Task #2**: *Translate the Verilog codes into Chisel*
  - Evaluated on FPGA (xc7v2000tfhg1716-1), Vivado 2017.01
- **Who**: *A junior student who never knew Chisel*

	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**

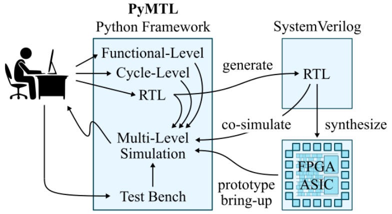


- RV64GC指令集
- 单发射顺序9级流水线
- 内置标签化冯诺依曼结构技术
- 8核/2MB L2 Cache
- ChipLink前端总线
- **1.2GHz@ 28nm**
- Wafer out/WB BGA封装
- 最大支持32GB DDR4内存
- 2*千兆以太网
- 1*PCIe3.0 RC x4

Agile design is easy

CHISEL

PYRTL
register-transfer-level hardware design and simulation



SpinalHDL
A high level hardware description language

bluespec

cucapra/**latte21**
Languages, Tools, and Techniques for Accelerator Design

Scala

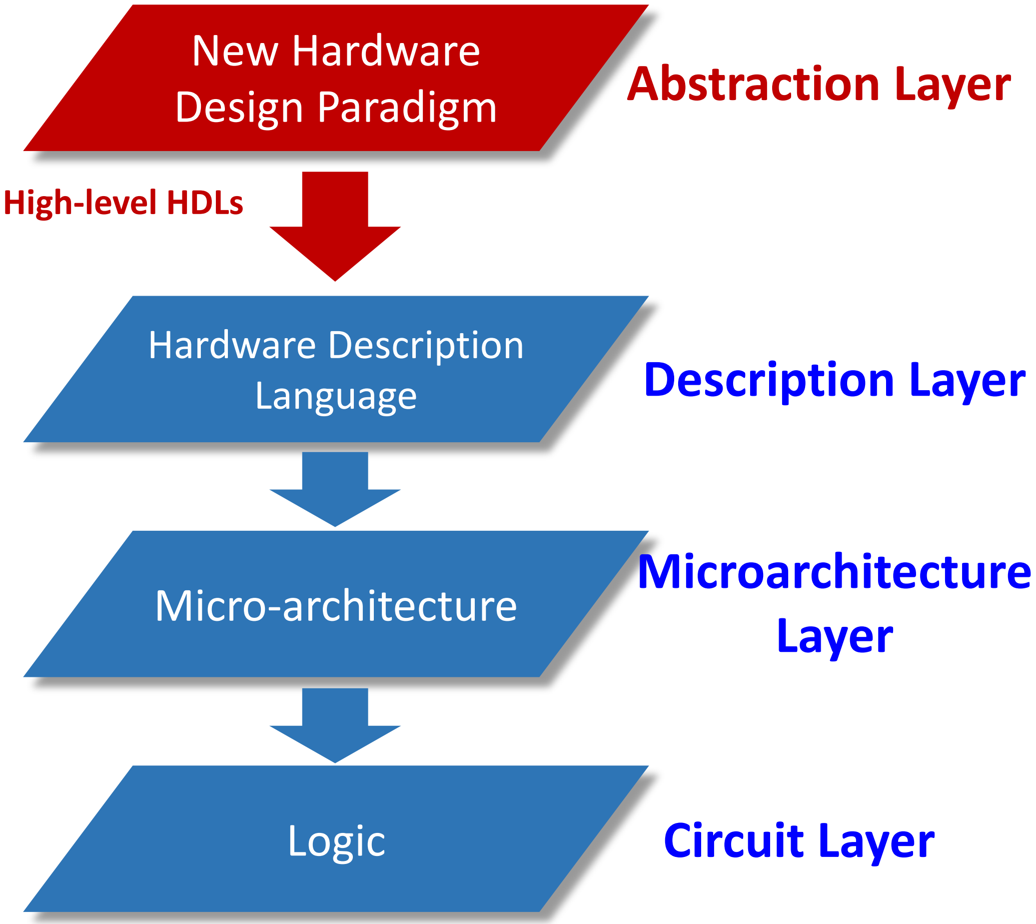
python

OCaml

Verilog

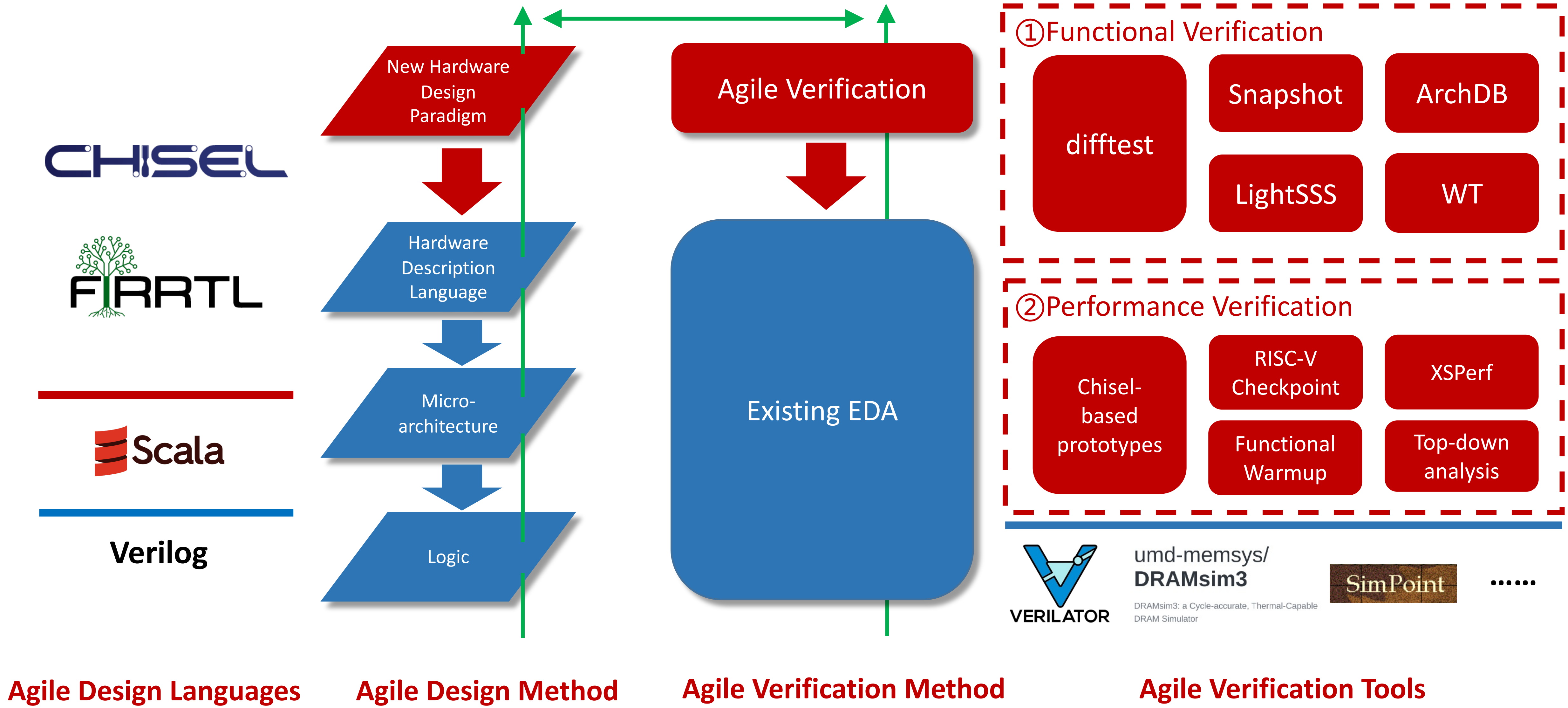
SystemVerilog

Agile Design Languages



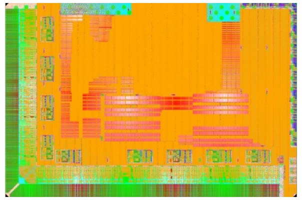
Agile Design Methodology

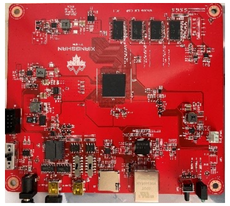
Agile verification is hard

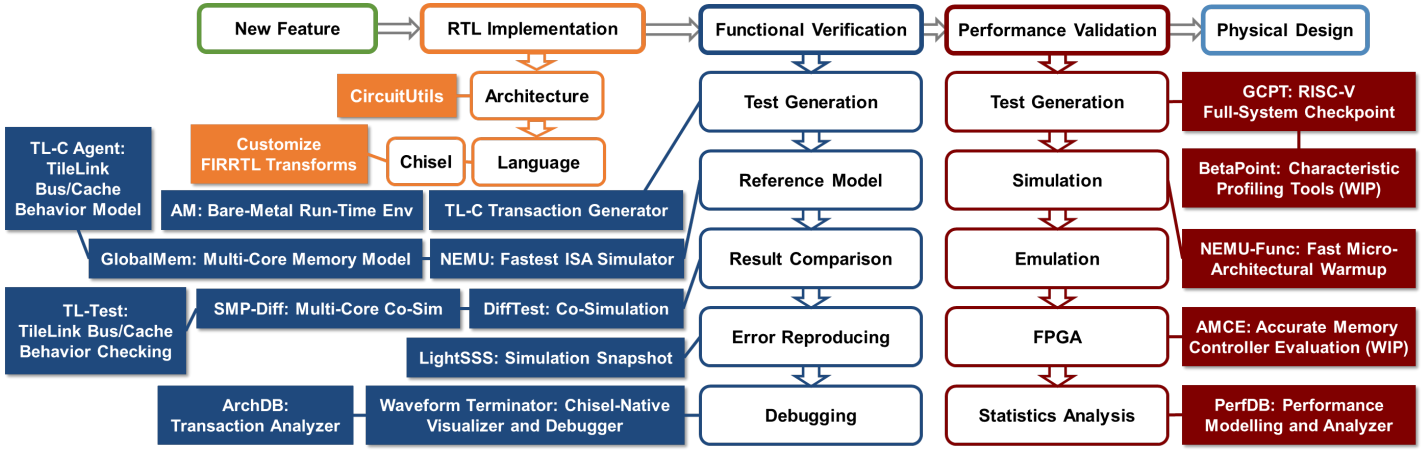


Decision: Open & Agile Development Infrastructure

- 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

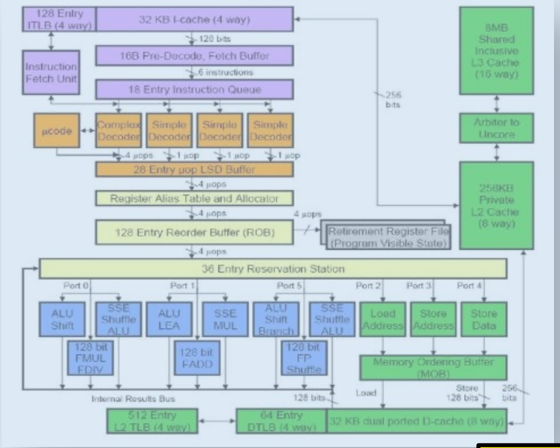
ISA Spec.


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

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

1 Open ISA

Docs



RTL

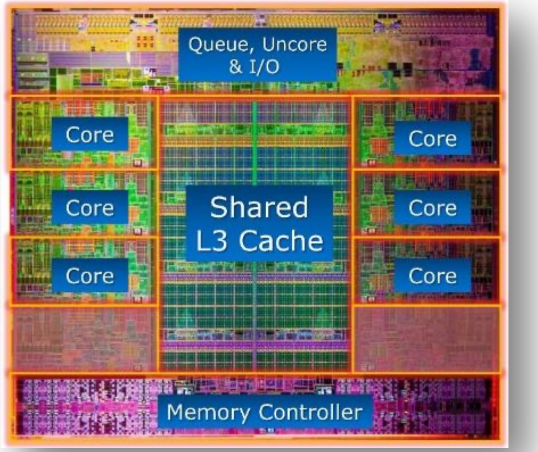
```
component DebugCoreTop is
  port (
    -- Trigger and Data
    cu_Clk       : in    std_logic_vector(2 downto 0) := (others => '0');
    cu0_Trig     : in    t_trig_0 := (others => (others => '0'));
    cu1_Trig     : in    t_trig_1 := (others => (others => '0'));
    cu2_Trig     : in    t_trig_2 := (others => (others => '0'));
    cu0_Data     : in    t_data_0 := (others => (others => '0'));
    cu1_Data     : in    t_data_1 := (others => (others => '0'));
    cu2_Data     : in    t_data_2 := (others => (others => '0'));

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

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

2 Open Design/Implt

Layout



Part IV

Conclusion

If you are doing research on …

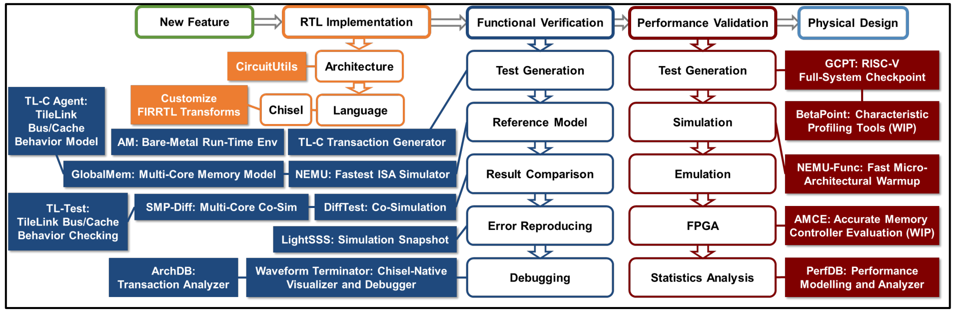
- ***Computer Architecture***
  - XiangShan: a realistic 6-wide out-of-order RISC-V implementation with industry-competitive performance and an active open-source community
  - MinJie provides the toolchains
- *Microarchitecture, accelerators, novel architectures, profiling, systems, benchmarking, compilers, …*

midgard EPFL

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

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

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

BEIJING 2022 BEIJING 2022

Thanks!

This tutorial is organized by Kaifan Wang, Yinan Xu, Guokai Chen, Xi Chen, Haoyuan Feng, Ziyue Zhang, Kan Shi, and Yungang Bao.