History & State of the Union

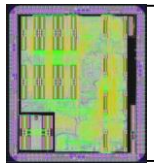
Yunsup Lee
SiFive, RISC-V Foundation
yunsup@sifive.com

SiFive Tech Symposium
June 17, 2019

What is RISC-V?

- A high-quality, license-free, royalty-free RISC ISA specification originally from UC Berkeley
- Standard maintained by non-profit RISC-V Foundation
- Suitable for all types of computing system, microcontrollers to supercomputers
- Numerous proprietary and open-source cores
- Experiencing rapid uptake in industry and academia
- Supported by growing shared software ecosystem
- A work in progress…

RISC-V Timeline



2010: RISC-V ISA project begins

2011: User ISA v1.0, Raven-1 tapeout (28nm), RVC MS thesis

2012: 1st Rocket tapeout, EOS14, 45nm

2013: First Linux Port

2014: User ISA v2.0 IMAFD

2015: Hot Chips 2014

2015: RISC-V Foundation Incorporated

2016: 1st Workshop

2016: Privileged Arch, v1.10

2017: NVIDIA to RISC-V

2017: Commercial Softcores

2018: 1st Commercial SoC

2019: WDC to RISC-V

2019: 1st Commerical Unix SoC

2019: Main ISA specs ratified!
RV32IMFDQC
RV64IMFDQC
Privileged MSU 1.11

2020: 1st Summit

2020: Daejeon Workshop

You are here

Berkeley

World

What's Different about RISC-V?

- ***Simple***
  - Far smaller than other commercial ISAs
- ***Clean-slate design***
  - Clear separation between user and privileged ISA
  - Avoids µarchitecture or technology-dependent features
- A ***modular*** ISA designed for ***extensibility/specialization***
  - Small standard base ISA, with multiple standard extensions
  - Sparse and variable-length instruction encoding for vast opcode space
- ***Stable***
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions
- ***Community designed***
  - With leading industry/academic experts and software developers

RISC-V Ecosystem

Software

Open-source software:
Gcc, binutils, glibc, Linux, BSD, LLVM, QEMU, FreeRTOS, ZephyrOS, LiteOS, SylixOS, …

Commercial software:
Lauterbach, Segger, IAR, Micrium, ExpressLogic, Ashling, AntMicro, Imperas, UltraSoC …

RISC-V Foundation

ISA specification | Golden Model | Compliance

Hardware

Open-source cores:
Rocket, BOOM, RI5CY, Ariane, PicoRV32, Piccolo, SCR1, Shakti, Swerv, Hummingbird, …

Commercial core providers:
Andes, Bluespec, Cloudbear, Codasip, Cortus, C-Sky, InCore, Nuclei, SiFive, Syntacore, …

Inhouse cores:
Nvidia, +others

Why is RISC-V so popular?

- Engineers sometimes "*don't see forest for the trees*"
- The movement is ***not*** happening because some benchmark ran 10% faster, or some implementation was 30% lower power
- The movement ***is*** happening because ***new business model*** changes everything
  - Pick ISA first, then pick vendor or build own core
  - Add your own extension without getting permission
- Implementation features/PPA will follow
  - Whatever is broken/missing in RISC-V will get fixed

Modest RISC-V Project Goal

Become the industry-standard ISA for all computing devices

So, how's it going?

RISC-V at Heart of Open-Source Hardware

- Upswing in interest in open-source hardware IP
  - FOSSi, China Open Instruction Ecosystem (RISC-V) Alliance, CHIPS Alliance, OpenHW, …
- Yes, open-source hardware needs open-source cores, and open-source cores need open spec
- But more, success of RISC-V *commercially* is driving mainstream interest in open-source hardware
  - No longer only academics/hobbyists

RISC-V: An Everyday Design Choice

- For embedded/IoT, RISC-V is strong competitor
- Production ramp starting, expect “millions” of SoCs to ship with RISC-V cores in 2019
- SiFive announced >100 RISC-V IP design wins

- Message: *You won’t get fired for choosing RISC-V!*

ISA Standards and Roadmap

Core RISC-V ISA Ratified!

- Updated version of original Berkeley unprivileged ISA (RV32IMFDQC/RV64IMFDQC) now ratified
  - ”A” extension has one issue to resolve (LR/SC progress)
- Memory model (RVWMO) ratified
- Privileged architecture v1.11 M/S/U modes ratified

- The foundation of the Foundation!

Prioritizing Future ISA Standards

- Will implementers build it?
- Will this increase RISC-V design wins?
- Does this fill a hole in RISC-V capability?
- Does this standardize something being done incompatibly today?
- Does more than one member care about it?
- Are there capable volunteers to work on it?

Fragmentation versus Diversity

Fragmentation:
Same thing done different ways

Diversity:
Solving different problems

Recap: ISA Standards Priorities for 2019

- Develop implementations and software stacks for CLIC, vectors, hypervisor
- Unix platform standard
  - Server platform, SBI, UEFI/ACPI?, security features, …
- Embedded ABI
  - Reduce library code size, reduce interrupt latency
- Zfinx
  - Embedded systems with floating-point out of X registers
- Improve code compression
  - Compilers, libraries, additional compressed instructions

Call for Standards Participation

Design is easy & fun, standards are hard work

- Debating all the fine technical details
- Separating facts from opinions
- Evaluating hardware and software implications
- Understanding commercial realities
- Backwards compatibility
- Pruning unnecessary options (saying “No!”)
- Reaching consensus

- ***We need more skilled participation!***

Embedded ABI

- Original ABI developed for Unix platforms
- Embedded ABI goals
  - reduce interrupt latency by reducing argument registers, caller-saved registers
  - reduce code size by dropping 128b FP support
  - Provide compatible interface are RV32E and RV32I
- Proposal in https://github.com/riscv/riscv-eabi-spec
  - ***Need significant help running design-space exploration with different compiler frameworks and benchmarks***

Zfinx “Float in X registers”

- Intended for embedded platforms
- Removes f registers from ISA
- Floating-point operations take values from x registers
- Supported for both E and I base ISAs

- Reduces system size, interrupt latency, simplifies ABI
- Frees up floating-point load/store opcode space in compressed extension

Improving Embedded Code Size

- Better documentation/ white papers on existing techniques for RISC-V
- Improved library code for code size reduction, including new embedded ABI
- Make compilers aware of compressed instructions
- Further compiler optimizations
- Repurpose FP load/store encodings for Zfinx
- Other new instructions
- **Start new task group soon! Will need compiler work!**

RISC-V Security

- RISC-V natural focus of security research, but how to use ***now*** in practice?
- RISC-V designed to provide composable security primitives not pre-packaged point solutions
  - Users need help in configuring standard design points
- Need to define secure platform profiles for each software ecosystem
- ***Kick off at last summit, but more work needed***

Software Development for RISC-V

- Great progress in many areas in open-source community
- Increased activity in commercial software and tools

- A lot of development/porting work still needed, requiring investment from companies and/or govts
- Contact Foundation if have ideas/proposals

RISC-V Predictions for 2020

- In 2020, see significant investment in high-end RISC-V implementations (server, mobile, auto)
- Need to begin laying the ISA and software foundations to enable these
  - Hypervisor
  - Vectors
  - Security
  - Power-management
  - Server platform profile
  - Functional Safety

RISC-V Predictions for 2020

- Not just general-purpose cores, start to take all the other core sockets on an SoC
  - DSP
  - AI
  - Graphics?
  - Network processing

Summary RISC-V 2019

- Core standards ratified
- Widespread acceptance and adoption
- Community growing and growing
- Software ecosystem filling out rapidly
- Standards efforts filling in gaps
- Lots of work ahead…