RISC-V ISA & Foundation Overview

Rick O’Connor
rickoco@riscv.org

http://www.riscv.org
Outline

- RISC-V ISA Overview
- RISC-V Foundation Overview & Growth
- RISC-V Use Case Examples
  - NVIDIA and Western Digital
- Summary
In 2010, after many years and many projects using MIPS, SPARC, and x86 as basis of research, it was time for the Computer Science team at UC Berkeley to look at what ISAs to use for their next set of projects.

- Obvious choices: x86 and ARM
  - x86 impossible – too complex, IP issues
  - ARM mostly impossible – complex, IP issues

- So UC Berkeley started “3-month project” during the summer of 2010 to develop their own clean-slate ISA
Four years later, in May of 2014, UC Berkeley released frozen base user spec
- many tapeouts and several research publications along the way

The name RISC-V (pronounced risk-five), was chosen to represent the fifth major RISC ISA design effort at UC Berkeley
- RISC-I, RISC-II, SOAR, and SPUR were the first four projects with the original RISC-I publications dating back to 1981

In August 2015, articles of incorporation were filed to create a non-profit RISC-V Foundation to govern the ISA
Why Instruction Set Architectures matter

- Why are 99%+ of laptops/desktops/servers based on AMD x86-64 ISA (over 95%+ built by Intel)?
- Why are 99%+ of mobile phones + tablets based on ARM v7/v8 ISA?
- Why can’t Intel sell mobile chips?
- Why can’t ARM vendors sell servers?
- How can IBM still be selling mainframes?
- ISA is most important interface in a computer system – Where software meets hardware
Why are there no successful free & open ISA standards and free & open implementations, like other fields?

<table>
<thead>
<tr>
<th>Field</th>
<th>Standard</th>
<th>Free, Open Impl.</th>
<th>Proprietary Impl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Networking</td>
<td>Ethernet, TCP/IP</td>
<td>Many</td>
<td>Many</td>
</tr>
<tr>
<td>OS</td>
<td>Posix</td>
<td>Linux, FreeBSD</td>
<td>M/S Windows</td>
</tr>
<tr>
<td>Compilers</td>
<td>C</td>
<td>gcc, LLVM</td>
<td>Intel icc, ARMcc</td>
</tr>
<tr>
<td>Databases</td>
<td>SQL</td>
<td>MySQL, PostgreSQL</td>
<td>Oracle 12C, M/S DB2</td>
</tr>
<tr>
<td>Graphics</td>
<td>OpenGL</td>
<td>Mesa3D</td>
<td>M/S DirectX</td>
</tr>
<tr>
<td>ISA</td>
<td>???????</td>
<td>--</td>
<td>x86, ARM</td>
</tr>
</tbody>
</table>
Most CPU chips are SoCs with many ISAs

- Applications processor (usually ARM)
- Graphics processors
- Image processors
- Radio DSPs
- Audio DSPs
- Security processors
- Power-management processor
- ....

- Apps processor ISA too large for base accelerator ISA
- IP bought from different places, each proprietary ISA
- Home-grown ISA cores
- Over a dozen ISAs on some SoCs – each with unique software stack

NVIDIA Tegra SoC
Why so Many ISAs?

Do we need all these different ISAs?

Must they be proprietary?

*What if there was one free and open ISA everyone could use for everything?*
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**
  - Small standard base ISA
  - Multiple standard extensions

- Designed for **extensibility/specialization**
  - Variable-length instruction encoding
  - Vast opcode space available for instruction-set extensions

- **Stable**
  - Base and standard extensions are frozen
  - Additions via optional extensions, not new versions
RISC-V Base Plus Standard Extensions

- Four base integer ISAs
  - RV32E, RV32I, RV64I, RV128I
  - Only <50 hardware instructions needed for base
- Standard extensions
  - M: Integer multiply/divide
  - A: Atomic memory operations (AMOs + LR/SC)
  - F: Single-precision floating-point
  - D: Double-precision floating-point
  - G = IMAFD, “General-purpose” ISA
  - Q: Quad-precision floating-point
  - C: compressed 16b encodings for 32b instructions
- All the above are a fairly standard RISC encoding in a fixed 32-bit instruction format
### Base Integer Instructions (32|64|128)

**Category** | **Name** | **Format** | **RV32I** | **RV64I** | **RV128I**
---|---|---|---|---|---
**Loads** | Load Byte | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Load HalfWord | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Load Word | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Load Byte Unaligned | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Load Half Unaligned | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
**Stores** | Store Byte | S | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Store HalfWord | S | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Store Word | S | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
**Shifts** | Shift Left | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| Shift Left Immediate | I | rd,rs1,shamt | rd,rs1,shamt | rd,rs1,shamt | rd,rs1,shamt |
| Shift Right | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| Shift Right Immediate | I | rd,rs1,shamt | rd,rs1,shamt | rd,rs1,shamt | rd,rs1,shamt |
| Shift Right Arithmetic | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| Shift Right Arith Imm | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
**Arithmetic** | ADD | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| ADD Immediate | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Subtract | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| Load Upper Imm | U | rd,imm | rd,imm | rd,imm | rd,imm |
| Add Upper Imm to PC | U | rd,imm | rd,imm | rd,imm | rd,imm |
**Logical** | XOR | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| XOR Immediate | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| OR | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| OR Immediate | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| AND | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| AND Immediate | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
**Compare** | Set > | R | rd,rs1,rs2 | rd,rs1,rs2 | rd,rs1,rs2 |
| Set < Immediate | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Set < Imm | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
**Branches** | Branch = | SB | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Branch > | SB | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Branch >= | SB | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Branch < | SB | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
| Branch <= | SB | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm | rs1,rd,imm |
**Jump & Link** | AL | U | rd,imm | rd,imm | rd,imm |
**Synch** | System CALL | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| System BREAK | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
**Counters** | Read CYCLE | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Read CYCLE upper Half | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Read TIME | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Read TIME upper Half | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Read INSTR Retired | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |
| Read INSTR upper Half | I | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm | rd,rs1,imm |

#### RV32I
- **Privileged**: +14
- **M**: +8
- **F, D, Q**: +34
- **C**: +46
- **A**: +11

---

**32-bit Instruction Formats**

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>32-bit Instruction Format</td>
</tr>
<tr>
<td>I</td>
<td>Immediate</td>
</tr>
<tr>
<td>S</td>
<td>Source Register</td>
</tr>
<tr>
<td>B</td>
<td>Branch Target</td>
</tr>
<tr>
<td>U</td>
<td>Unaligned Address</td>
</tr>
<tr>
<td>SB</td>
<td>Short Branch</td>
</tr>
<tr>
<td>U2</td>
<td>Upper Halfword</td>
</tr>
</tbody>
</table>

---

**Additional Instructions**
- **Mov**: Move
- **Add**: Add
- **Sub**: Subtract
- **And**: And
- **Or**: Or
- **Xor**: Xor
- **Shifts**: Shift Left, Shift Right
- **Comparisons**: Set >, Set <
- **Branches**: Branch =, Branch >, Branch >=, Branch <, Branch <=
- **Jumps & Links**: AL
- **Synchronizations**: System CALL, System BREAK
- **Counters**: Read CYCLE, Read CYCLE upper Half, Read TIME, Read TIME upper Half, Read INSTR Retired, Read INSTR upper Half
RISC-V in Education, new books!
Outline

- RISC-V ISA Overview
- RISC-V Foundation Overview & Growth
- RISC-V Use Case Examples
  - NVIDIA and Western Digital
- Summary
RISC-V Foundation Overview

- Incorporated August, 2015 as a 501c6 non-profit Foundation
- Membership Agreement & Bylaws ratified December 2016
- The RISC-V ISA and related standards shall remain open and license-free to all parties
  - RISC-V ISA specifications shall always be publicly available as an online download
- The compliance test suites shall always be publicly available as a source code download
- To protect the standard, only members (with commercial RISC-V products) of the Foundation in good standing can use “RISC-V” and associated trademarks, and only for devices that pass the tests in the open-source compliance suites maintained by the Foundation
RISC-V Foundation Growth History
August 2015 to April 2018
RISC-V Members in 27 Countries Around the World!!
Representing 57% of the global population!!
Foundation Organization

- The Board of Directors consists of seven+ members, whose replacements are elected by the membership.
- The Board can amend the By-Laws of the RISC-V foundation via a two-thirds affirmative vote.
- The Board appoints chairs of ad-hoc committees to address issues concerning RISC-V, and has the final vote of approval of the recommendation of the ad-hoc committees.
  - Technical Committee Chair – Yunsup Lee, SiFive
  - Security Standing Committer Chair - Helena Handschuh, Rambus
  - Marketing Committee Chair – Ted Marena, Microsemi
- All members of committees must be members of the RISC-V Foundation.
RISC-V Foundation Board of Directors

- Krste Asanović, Chairman
  - Professor in the EECS Department at UC Berkeley
- David Patterson, Vice-Chairman
  - Google Architect, Retired Professor Computer Science UC Berkeley
- Zvonimir Bandić
  - Senior Director of Next Generation Platform Technologies at Western Digital Corporation
- Charlie Hauck
  - CEO of Bluespec Inc.
- Rob Oshana
  - Director Global SW Development at NXP
- Frans Sijstermans
  - Vice President Engineering at NVIDIA
- Ted Speers
  - Technical Fellow, Head of Product Architecture for Microsemi’s SoC Group
Outline

- RISC-V ISA Overview
- RISC-V Foundation Overview & Growth
- RISC-V Use Case Examples
  - NVIDIA and Western Digital
- Summary
Following slides excerpt from Frans Sijstermans, VP Engineering, NVIDIA Keynote Address - Full presentation is here
Falcon’s history

- Embedded in 15+ designs
- Taped out in ~50 chips
- Shipped ~3 billion times
- No stop-ship bugs

<table>
<thead>
<tr>
<th>Falcons shipped estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>dGPU Volume /year</strong></td>
</tr>
<tr>
<td><strong>Years Falcon shipping</strong></td>
</tr>
<tr>
<td><strong>Avg. #Falcons / GPU</strong></td>
</tr>
<tr>
<td><strong>Avg. NVIDIA market share</strong></td>
</tr>
<tr>
<td><strong>Total shipped</strong></td>
</tr>
</tbody>
</table>

Selecting the next architecture

Technical criteria
- >2x performance of Falcon
- <2x area cost of Falcon
- Support for caches as well tightly coupled memories
- 64-bit addresses
- Suitable for modern OS

Considered architectures
- ARM
- Imagination Technologies MIPS
- Synopsys ARC
- Cadence Tensilica
- RISC-V
Why RISC-V for Falcon Next

RISC-V is the only architecture that meets all our criteria


<table>
<thead>
<tr>
<th>Item</th>
<th>Requirement</th>
<th>ARM A53</th>
<th>ARM A9</th>
<th>ARM R5</th>
<th>RISC-V Rocket</th>
<th>NV RISC-V</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core perf</td>
<td>&gt;2x falcon</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Area (16ff)</td>
<td>&lt;0.1mm^2</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Security</td>
<td>Yes</td>
<td>TZ</td>
<td>TZ</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>TCM</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>L1 I/D $</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Addressing</td>
<td>64bit</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Extensible ISA</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Safety (ECC/Parity)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Functional Simulation model</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

RISC-V is the only architecture that meets all our criteria.
Following slides excerpt from Martin Fink, CTO, Western Digital Keynote Address
Full presentation is here
RISC-V Meets the Needs of Big Data and Fast Data

Big Data
- Genomics
- Predictive Analytics

Fast Data
- Autonomous Machines
- Safety & Security
- Private Exchange
- Machine Learning

©2017 Western Digital Corporation or its affiliates. All rights reserved.
#LetDataThrive
RISC-V Enables Purpose-Built Environments for Big Data and Fast Data Applications

- **Big Data**
  - Storage-centric architecture
  - Capacity-centric scale
  - HDD, SSD
  - Storage Semantic Data Flow
  - Storage SOC

- **Memory**
  - Memory-centric architecture
  - Performance-centric scale
  - NVM, DRAM, SRAM
  - Memory Semantic Data Flow
  - General Purpose CPU, GPU, FPGA, ASIC

- **Compute**
  - Purpose-built data-centric architectures
  - Interconnect
  - RISC-V

©2017 Western Digital Corporation or its affiliates. All rights reserved.
#LetDataThrive
Driving Momentum

Western Digital ships in excess of 1 Billion cores per year...and we expect to double that.
Accelerating the RISC-V Ecosystem

Western Digital to contribute one billion cores annually to fuel RISC-V

1. Support development of open source IP building blocks for the community

2. Actively partner and invest in the ecosystem

3. Accelerate development of purpose-built processors for a broad range of Big Data and Fast Data environments

4. Multi-year transition of Western Digital devices, platforms and systems to RISC-V purpose-built architectures
RISC-V ISA & Foundation Summary

- The free and open RISC-V ISA is enabling a new innovation frontier for all computing devices
- Strong Industry Support
  - 150+ members; Broad commercial and academic interest
- RISC-V Twitter [http://twitter.com/risc_v](http://twitter.com/risc_v) @risc_v
- RISC-V LinkedIn Page
  - [http://www.linkedin.com/company/risc-v-foundation](http://www.linkedin.com/company/risc-v-foundation)
- RISC-V mail lists / groups
  - [https://riscv.org/mailing-lists/](https://riscv.org/mailing-lists/)