RISC-V

User-Level ISA, Privilege-Level ISA, Implementations, Software Support

Emantor

14. Februar 2018

Stratum 0

Overview

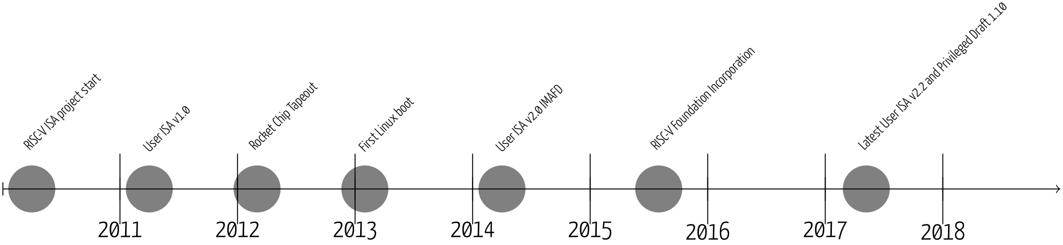
- **RISC-V ISAs**
  - RISC-V Background
- **Chisel**
  - Chisel HDL
- **RISC-V Implementations**
  - Spike
  - Rocket
  - Berkley Out Of Order Machine (BOOM)
  - Sodor
- **Linux Software Support**
- **Conclusion and Questions**

Why?

- Deficiencies in several other instruction sets, e.g.:
  - x86 being in constant register pressure
  - ARMv8 being a fixed width ISA
  - often ISAs are bound to a specific implementation (e.g. ALPHA)
- None of the mentioned instruction sets are free!
- Build on top of previous designs: DLX and openRISC
- Provide an ISA not an implementation

RISC-V History

- User ISA v2.2 from 7.5.2017
- Privileged ISA Draft v1.10 from 7.5.2017

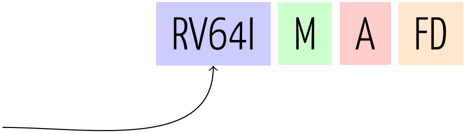


RISC-V Foundation

- Standardizes and promotes RISC-V
- Different membership levels
- Some members:

RISC-V User Level ISA

- Current: Version 2.2
- Describes a base ISA + Extensions
- Example:

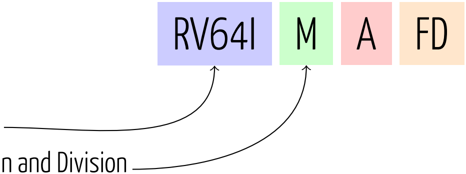


RV64I M A FD

- RISC-V 64bit Integer
- Integer Multiplication and Division
- Atomic Multiprocessor Synchronization
- Single/Double Precision Floating Point

RISC-V User Level ISA

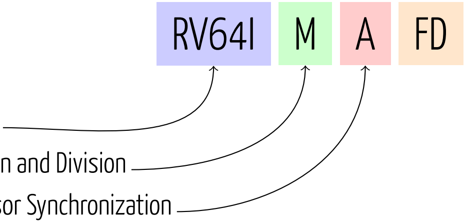
- Current: Version 2.2
- Describes a base ISA + Extensions
- Example:



- RISC-V 64bit Integer
- Integer Multiplication and Division
- Atomic Multiprocessor Synchronization
- Single/Double Precision Floating Point

RISC-V User Level ISA

- Current: Version 2.2
- Describes a base ISA + Extensions
- Example:

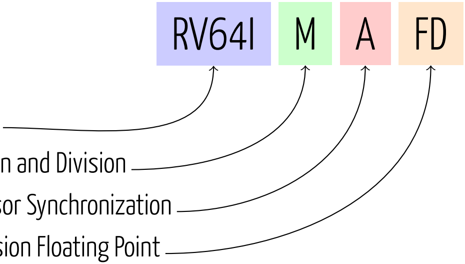


RV64I M A FD

- RISC-V 64bit Integer
- Integer Multiplication and Division
- Atomic Multiprocessor Synchronization
- Single/Double Precision Floating Point

RISC-V User Level ISA

- Current: Version 2.2
- Describes a base ISA + Extensions
- Example:



- RISC-V 64bit Integer
- Integer Multiplication and Division
- Atomic Multiprocessor Synchronization
- Single/Double Precision Floating Point

RISC-V Privilige Level ISA

- Current: Version 1.10
- Describes a base ISA + Extensions
- Contains:
  - CSR (Control and Status Register) Definitions
  - Privilege seperation levels (Machine, Supervisor, User)

Overview

RISC-V ISAs
- RISC-V Background

Chisel
- Chisel HDL

RISC-V Implementations
- Spike
- Rocket
- Berkley Out Of Order Machine (BOOM)
- Sodor

Linux Software Support

Conclusion and Questions

Chisel HDL

- Extension of scala to support hardware description
- Synthesises to verilog or C++
- Allows generation of modules or module generators
- Full flexibility of the scala language provided to the designer

Chisel by example: Diplomacy

- Problem: interconnects between master and slave devices
- Inputs: Protocol specification and nodes with interconnections
- Solution:
  - build a graph via diplomacy
  - build hardware from the graph using the protcol specification

Overview

RISC-V ISAs

RISC-V Background

Chisel

Chisel HDL

RISC-V Implementations

Spike

Rocket

Berkley Out Of Order Machine (BOOM)

Sodor

Linux Software Support

Conclusion and Questions

Spike - ISA Simulator

- Functional simulator
- Real hardware emulation:
  - Connection via Open On-Chip-Debugger (openOCD)
  - Connect GNU Debugger to openOCD interface
- BSD Clause-3

Rocket (Chip Generator)

- Emits rocket cores
- Sample RISC-V implementation maintained by SiFive and UCB-BAR
- implemented on silicon for different prcoessors:
  - Raven Vector Processor with DC-DC On-Chip convertors
  - 45nm 1.3Ghz processor with Vector Accelerator
- In continuous development by UC-BAR
- Dual License: BSD Clause-3 & Apache 2.0

BOOM v2

- Reuses the rocket chip generator
- Fully Out Of Order Execution Core
- For general purpose computer use
- Uses a Branch Predictor, Branch Target Buffer and Register Renaming
- In continuous development by UC-BAR
- License: BSD Clause-3

Sodor

- In-Order educational design
- Multiple stages:
  - 1-stage: essentially like Spike, ISA simulator
  - 2-stage: demonstrates pipelining
  - 3-stage: Princeton and Havard versions with sequential memory
  - 5-stage: toggle between fully interlocked and fully bypassed

Overview

RISC-V ISAs

- RISC-V Background

Chisel

- Chisel HDL

RISC-V Implementations

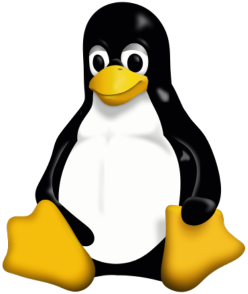
- Spike
- Rocket
- Berkley Out Of Order Machine (BOOM)
- Sodor

Linux Software Support

Conclusion and Questions

Linux Ecosystem - Kernel

- Linux Support merged in v4.15, not released yet
- But:
  - No device drivers for busses or other peripherals
  - Debugging allows a successful Linux boot verification
- Additional support already queued for v4.16
- Maintainership done by SiFive



Linux Ecosystem - GCC

- GCC 7.1 has a RISC-V backend
- Supporting:
  - Different combinations of instruction sets
  - All base instruction sets (RV32I, RV32E,...)
- Maintainership done by SiFive

Linux Ecosystem - glibc

- Not upstream yet
- Patchset v2 posted on 19.12.2017
- also supports:
  - All base instruction sets (RV32I, RV32E,...) for optimizations
- Patches proposed by SiFive

Overview

RISC-V ISAs

- RISC-V Background

Chisel

- Chisel HDL

RISC-V Implementations

- Spike
- Rocket
- Berkley Out Of Order Machine (BOOM)
- Sodor

Linux Software Support

Conclusion and Questions

Conclusion

- RISC-V is still in very active development
- Linux support is coming along fast
- Multiple corporations are in the process of replacing own ISAs with Risc-V

Questions

Questions?