RISC-V® Everywhere

Mark Himelstein
CTO, RISC-V International

SC 2023
November 2023

Overview

- RISC-V & HPC Overview
- Why People Use RISC-V
- Open Standard
- Portability, Profiles, & Platforms
- SW ecosystem

The definition of open computing is RISC-V

RISC-V

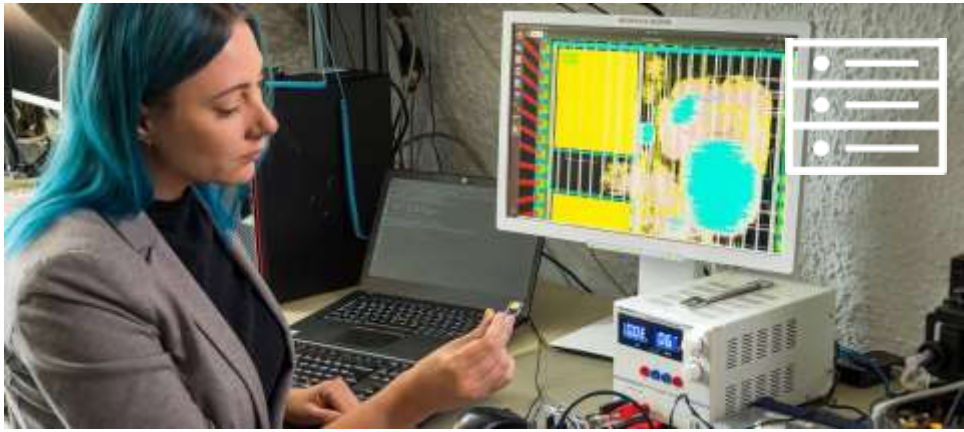
RISC-V is the most prolific and open Instruction Set Architecture in history

- RISC-V is inevitable
- RISC-V enables the best processors
- RISC-V is rapidly building the strongest ecosystem

10s of Billions of RISC-V cores deployed for profit!

High Performance Computing

- **E4** Monte Cimone Cluster along with DEI-UNIBO contributing to architecture, software, and integration.
- **European Processor Initiative** RISC-V accelerator with first chip Sep 2021
- **Technical University of Munich** (TUM) quantum cryptography chip for quantum computing security demands
- **Tactical Computing Labs** HPC-centric software test suite for GCC and LLVM
- **Cortus** is developing a high-performance RISC-V Out-of-Order processor core for the European eProcessor project.
- **De-RISC** HW-SW platform for multi-core RISC-V SoC for safety critical aerospace



Johanna Baehr of TUM heads a team that has hidden four hardware Trojans on this chip - malicious functions that are integrated directly into the circuits.

Image: Astrid Eckert / TUM

RISC-V®

Building an Open HPC Ecosystem with RISC-V

- HPC requires customized hardware solutions for many HPC domains
- HPC is tackling grand challenges that benefit from the Global Technology ecosystem
- HPC is on the technology leading edge
- An Open ISA complements Open Source Software and combines to create an open Ecosystem
- RISC-V accelerates innovation both in Research and Industry
- The RISC-V community and ecosystem are rapidly growing

RISC-V is an Open Standard

Open Source HW or Open Standard?

- We are officially an Open Standard HW ISA Architecture
- We are an Open Standard that works heavily with Open Source upstream projects (LINUX, GCC, LLVM, etc.)

- We don’t do reference implementations
- Our work product are specifications with support from Golden Models and Basic Tests

Unlike Open Source Software …

- Proprietary Custom Extensions are Encouraged and Welcome
- Products don’t just include the ISA, they can do custom implementations/extensions of/to the ISA
- Copyleft is a non-sequitur
- No restrictions on how the specification can be used
  - Only restrictions on branding

Why RISC-V?

- Flexibility/Open Access
- Cost
- Ecosystem
- Our position in History
- EDA Renaissance
- Pride of Ownership

Why Open Source Hardware?

Software: Leverage a large ecosystem compatible across implementations

Security: A fully auditable collection of IPs: processors, accelerators, etc.

Safety: No black-boxes

SWaP & Customization: SW/HW co-design for exact feature match

Performance: State-of-the-art implementations

No vendor lock-in: Ecosystem to enable custom develop from SME to large enterprise

Sovereignty: Freedom of access and implementation from design to production

Open Collaboration: Faster time to market, community, leverage existing open source

Profiles & Platforms

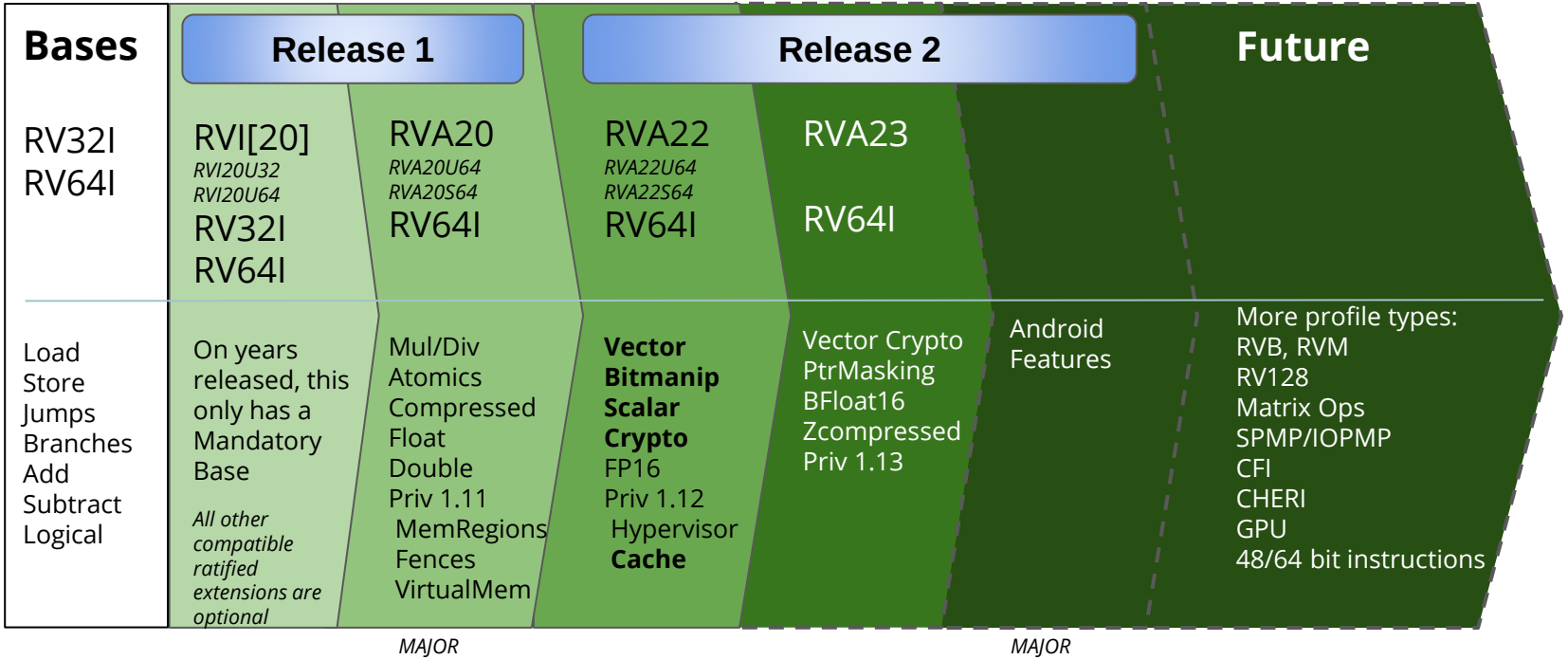
Profiles

- Generational groups of instructions that work together and present a unified target for the software ecosystem
- Always include a base (a base are a group of state, instructions and behaviors fundamental to being RISC-V)
- Optionally include one or more extensions (like bases, extensions are a group of state, instructions, and behaviors). Extensions may be mandatory, optional, or non-profile options (n/a)
- Major and minor releases. We use Major releases to be targets for the SW ecosystem. Minor releases are checkpoints
- Initial implementations of profiles are likely available from members 12-48 months after a profile is ratified
- Current profiles (RVA) all targeted at RICHOS, General Purpose Multi-User computing. More to come.
- Profiles can live and be used forever. The implementers decide on adoption and lifetime

Portability

- **Unified common standard and a robust Software Ecosystem**
- **A robust economy around systems and software is reliant on portability**
- Application & Runtime Software
  - Profiles
  - ABI
  - API (e.g. POSIX)
- Operating Systems
  - Platforms
    - Profiles, Boot to OS, Firmware Interfaces, SOC minimum HW, minimum security

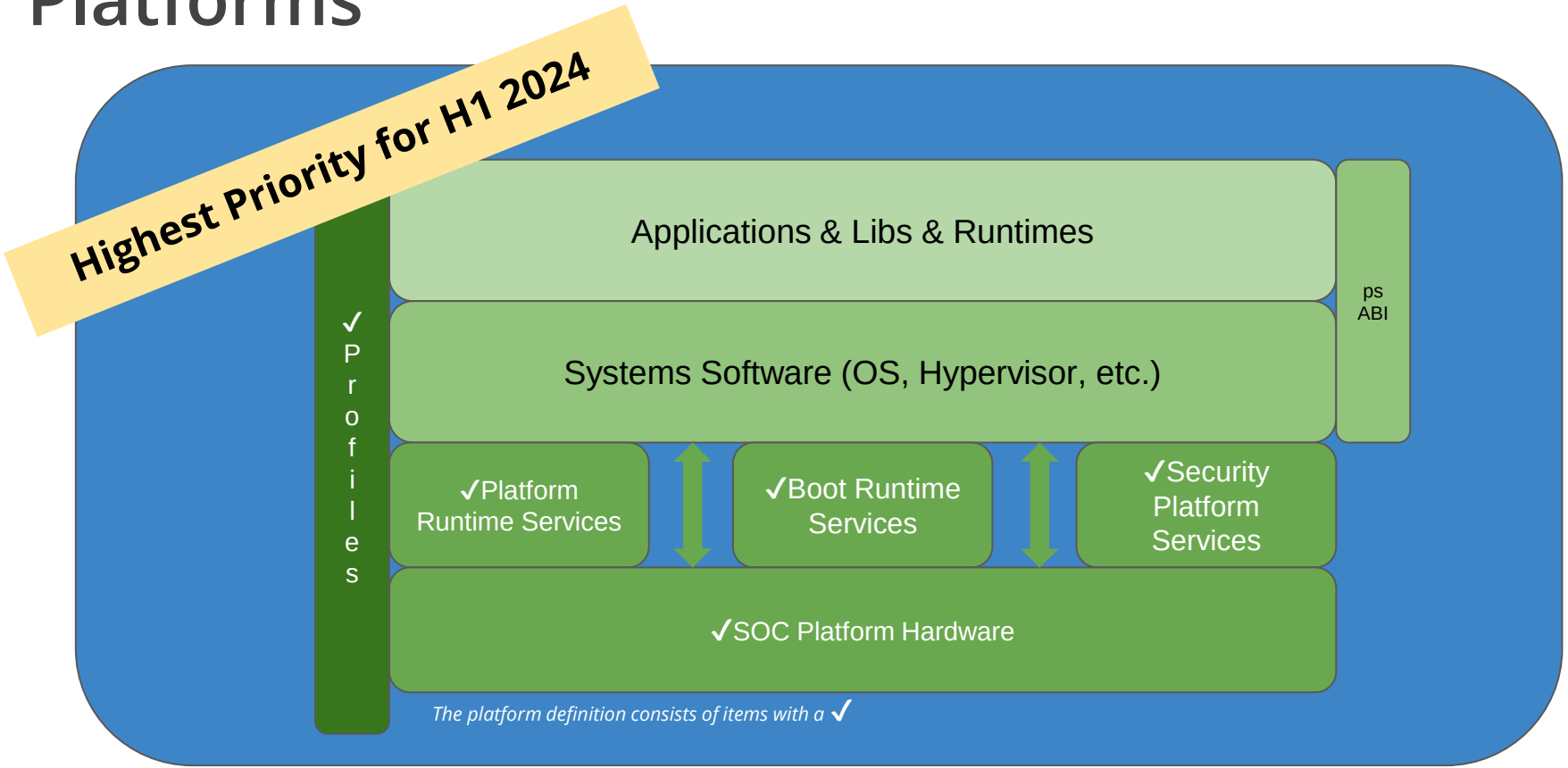
Profiles



- *Only a subset of extensions are listed above and it is not an exhaustive list*
- *Some extensions may be optional or non-profile in one profile and be mandatory in another*

RISC-V®

Platforms


Highest Priority for H1 2024
✓Profiles
Applications & Libs & Runtimes
Systems Software (OS, Hypervisor, etc.)
ps ABI
✓Platform Runtime Services
✓Boot Runtime Services
✓Security Platform Services
✓SOC Platform Hardware
The platform definition consists of items with a ✓

Past & Future

What We Have Done This Year

- Ratified ISA Specifications
  - Profiles, Code Size Reduction, Advanced Interrupt Architecture, **Vector Crypto**
- Ratified Fast Tracks
  - Counters, Total Store Ordering, RV32E/RV64E, Non Temporal Hints, **Vector FP16**, Scalar FP
- Ratified Non-ISA Specifications
  - IOMMU, PLIC
- Documentation
  - Unpriv in Asciidoc, Asciidoc Priv draft
- Ecosystem Development
  - Labs Policy created and approved by CTO
  - 130 DevBoards to Academic (27), Distro (45), Labs (9), and Individual (49) projects shipped
  - 5 Accepted PRs, 16 Submitted PRs but under review from DevPartners
- New Task Groups & Special Interest Groups (2023)
  - Debug, Trace, and Performance Monitoring (DTPM) TG , Graphics SIG, RISC-V Common Software Interface (RVM-CSI) SIG, Vector (SIMD) SIG, Control Transfer Records TG, **AI/ML SIG**, CHERI SIG, Runtime Integrity SIG

What's Coming Soon?

- Profiles
  - RVA23, RVI23, RVB23, RVM23, BOD committee on comprehensive ACTs & Certification
- Platforms
  - Portability for Systems Software (PRS, BRS, Platform Security, SOC HW, Profiles) -- Server Platform for LINUX and other operating systems
- ISA
  - Pointer Masking, QOS Register Interface, Control Transfer Records, Debug, Fast Interrupts, I/D Synch, Priv 1.13, Shadow Stacks/Landing Pads (includes maybe ops), Memory Tracking Table, Supervisor PMP, **Matrix**
- ISA Fast Tracks
  - CAS, **BFloat16**, Conditional Ops, Counter Mode Filtering, HW PTE A/D, Maybe Ops, Counter Delegation, Byte & Halfword Atomics, Resumable NMI, Load acquire/store release
- Non-ISA
  - Confidential VM extension, IOPMP, Nexus Trace, PRS (SBI, etc.) update, Boot to OS services, SOC HW, Confidential VM (COVE), COVE-IO, Unified Discovery, RAS error records, Vector C Intrinsics,
- Documentation
  - BOD committee on dev experience (content & UX)
  - Glossary, Navigation
  - Release cadence, integrating all ratified specs regularly

Vector & Matrix

V Vector

- V - basic vector
  - 32 vector registers (min length 128 bits) - varying length
  - vector CSRs
  - element width determines elements per vector register
  - grouping determines how many vector registers a vector instruction operates on if the grouping is >= 1 (grouping 2 length 128 width 8 means 32 elements) and how many elements used from a single register if the grouping is fractional (grouping ½ length 128, width 8 means 16 elements)
  - which element to start executing an instruction with
  - rounding mode for fixed point
  - saturation handling (e.g. overflow)
  - masking
  - masking agnostic, tail agnostic support
  - widening, narrowing

V Vector

- V - basic vector
  - 32 vector registers (min length 128 bits), vector CSRs, element width determines elements per vector register, grouping (LMUL) determines how many vector registers a vector instruction operates on if the grouping is >= 1 (grouping 2 length 128b width 8 means 32 elements) and how many elements used from a single register if the grouping is fractional (grouping ½ length 128b, width 8 means 16 elements)
  - Load/Store - index, unit-strided, strided, segments (fields), element width, LMUL, masked, ordered/unordered, fault-first, multi-register
- Zvlsseg - load/store segments (fields)
- Zvamo - atomics - future
- Operand types: vector vector, vector scalar, vector immediate
- Register types: Zve32x - 32 bit integer, Zve32f - 32 bit single precision floating point, Zve64x - 64 bit integer, Zve64f - 64 bit single precision floating point, Zve64d - 64 bit double precision floating point

V Vector

- Int ops: add, sub, reverse sub, sign/zero extend, add w carry, sub w borrow, bitwise logical, single width (s-w) shift, narrowing right shift, signed/unsigned (s/u) compares, min/max, s-w mul, divide, widening (wid) mult, s-w mul add, wid mul add, merge, move, s-w/wid reductions (red) for sum, min/max, logical
- Fixed point (fxp) ops: s-w saturating (sat)/averaging (avg) add/sub, s-w fractional mul with rounding (rnd) and sat, s-w scaling (scl) shifts, narrowing (nar) s/u fxp clip w sat/rnd/scl
- Floating point (fp) ops: s-w/wid add/sub/fused-mul-add, s-w mult/div, wid mult, square root, reciprocal square-root estimate, min/max, sign injection, compare, classify, merge, move, s-w/wid/nar converts fp & int, red for ordered/unordered sum & min/maxMask ops: logical, pop, find first, set only before/including first, iota (sum of mask bits), element index
- Move ops: scalar, slide up/down, scatter, gather, compress, whole vector reg move

RISC-V Vector versus ARM vs AVX512

	RVV	SVE	AVX512
Approximate number of instructions	100	400	2015
Spec supports multiple vector sizes	Yes	Yes	No
VL register (implementation defined configurable)	Yes	No	No
Masks	Yes (1)	Yes (16)	Yes (8)
Narrowing/Widening	Yes, LMUL	Yes, 2 instructions	Yes, 2 instructions
Mixed Datatype Vectorization	Uses LMUL	Pack/Unpack	Pack/Unpack
Hardware Unrolling (LMUL)	Yes	No	No
Polymorphic Encoding	Yes, vtype	No	No
Strided Memory Access	Yes	No	No
Gather/Scatter	Yes	Yes	Yes
Structured/Segmented Loads	Yes	Yes	No
Forward Progress on gather/scatter	Vstart	Repeat full instruction	Mask state
Fixed Point Support	Yes	Partial	No
Complex Support	No	Yes	Partial
Reductions	Yes	Yes	No
Register Gather	Yes	Yes	No
Tail Element control	Merge, Agnostic	Merge, Zeroing	Merge, Zeroingº
Destructive destination	No	Yes	No

Standardizing Matrix Extensions for RISC-V

Use cases

- HPC: GEMM as in BLAS
- ML: Fully connected layers
- ML: Convolution (direct or Winograd)
- Others: Discrete Fourier Transform

Source data types

- FP8/16/32/64
- bfloat16
- Integer 8/16/32 bits
- Quantized integer 1/2/4/8/16 bits

Accumulator:

- Widening of sources
- 32- and 64-bit accuracy

Demanding requirements, for demanding applications:

- Scalable (build once, reuse across generations)
- High computational intensity (multiply-adds per load)
- High computation rates (TeraOps)

Two complementary flavours for design flexibility

- Integrated Matrix Facility (**IMF**):
  Fully integrated with a standard vector unit
  Driven from Vector SIG

- Attached Matrix Facility (**AMF**):
  Orthogonal with any vector unit.
  Driven from AI/ML SIG

Software Ecosystem

Software Ecosystem

This is our number one priority

RISC-V will have the best ecosystem

- Largest number of players
- All cores in system become RISC-V
- Software wants to run on the best hardware

- Hardware and software have been co-evolving rapidly
- Long-running silicon and core developments bearing fruit now
- As advantages and future become clearer, greater motivation to move to RISC-V
- Positive feedback on software ecosystem growth

HPC Specific (1 of 3)

1. Operating System: Linux.
   - Military prefer Red Hat Enterprise Linux (from John Leidel)
   - Note: BSC MareNostrum runs on SUSE
2. Compilers : GCC and LLVM (moving towards LLVM)
   - Military needs Fortran and LLVM does not have a Fortran Compiler (however there is a project called LFortran which might have some potential.https://lfortran.org/)
3. Application specific :
   - OpenBLAS : supported on RV (OpenBLAS is an optimized BLAS (Basic Linear Algebra Subprograms) library based on GotoBLAS2 1.13 BSD version.) Current upstream support in BLIS/FLAME
   - OpenSHMEM, (TCL did some work for RV but more needs to be done)OpenSHMEM is an effort to create a specification for a standardized API for parallel programming in the Partitioned Global Address Space.
   - OpenUCX, (TCL did some work for RV but more needs to be done) Unified Communication X an open-source, production-grade communication framework for data-centric and high-performance applications

HPC Specific (2 of 3)

2. Application specific (continued from previous slide):
    - RISE projects:
        - simpleperf Profiler
        - PAPI Profiler
        - Libpmf4 Profiler
        - System Tap (Supported on RV) Tracer
        - OpenOCD (Supported on RV) Tracer
        - ASan Checker
        - San Checker
        - LKP Benchmarks
        - Spec CPU Benchmarks
        - DynamoRIO DBI simulator
        - Valgrind (In progress on RV) Dynamic Binary Instrumentation
        - Gprof Profiler

HPC Specific (3 of 3)

2. Application specific (continued from previous slide):
   - GraphBlas**:** (Not on RV) **(/ˈgræf blɑːz/** ⓘ) is an API specification that defines standard building blocks for graph algorithms in the language of linear algebra.
   - BLIS Support is upcoming https://github.com/flame/blis (portable software framework for instantiating high-performance BLAS-like dense linear algebra libraries.) https://github.com/flame/blis/pull/731
   - The GNU Multiple Precision Arithmetic Library. (Status of RV unknown, GNU MP is a portable library written in C for arbitrary precision arithmetic on integers, rational numbers, and floating-point numbers.)
   - OpenMP needs to be optimized for RV (Status or RV unknown and would need to be optimized; OpenMP API for parallel programming)
   - MPI The Message Passing Interface (MPI) is an open library standard for distributed memory parallelization. (Status of RV: Unknown and needs to be optimized for RV)
   - RAJA Portability Suite: Enabling Performance Portable CPU and GPU HPC Application from LLNL. (Status of RV: unknown and needs to be optimized for RV)
   - Kokkos Ecosystem from Sandia and five other DOE National Labs(Status unknown) Providing a vendor independent performance portable programming system for scientific, engineering, and mathematical software applications written in the C++ programming language.
   - FFTW ("Fastest Fourier Transform in the West") (Status non on RV yet) is a C subroutine library for computing the Discrete Fourier Transform (DFT) in one or more dimensions, of both real and complex data, and of arbitrarily large input size. FFTW also efficiently handles multiple, strided transforms.
   - TAU (Tuning and Analysis Utilities) (Status not on RV yet) is a comprehensive ***profiling*** and ***tracing*** toolkit for performance analysis of parallel programs written in Fortran, C, C++, Java, and Python. It is capable of gathering performance information through instrumentation of functions, methods, basic blocks, and statements.
   - LLNL Development Environment Software : full list: https://hpc.llnl.gov/software/development-environment-software

Accelerating the Rich RISC-V Ecosystem

Security | HPC | AI / ML | Cloud - Data Center - Storage - Networking | Consumer - IoT - Automotive | Mobile

Applications

Libraries

Infrastructure

Runtimes

OS

Hypervisor

Boot

Design Tools + Compilers

Ecosystem Standards, upstream / support common tools

ISA Extensions, Profiles, Platforms

Security model, IOMMU, Industry SIGs

SAIL Golden Model

RISC-V ISA

Architecture Tests

Attributes: Debuggable, Secure, Performant, Reliable/Serviceable/Diagnosable

Implementation

RTL | DV

Design & Microarchitecture

Silicon
Training
Research

Soft IP
Academia
Services

Software Ecosystem Resources

- Foundational Software Status for each extensions
- Draft spreadsheet of software on RISC-V status (140+ being tracked)
  - Hired RVI Software Ecosystem Director. First task is a comprehensive one stop clearinghouse of RISC-V commercial and open source software status
- RISC-V Ecosystem Landscape
- RISC-V Exchange (100s)
- A new Linux Foundation Project named RISE to accelerate open source software development on RISC-V

Profiles with Key Ecosystem Status

extension/base name - best guess	ratification package name	description (what this does, in English)	IC	extensions included (subsets)	implies (and transitives)	incompatible (and transitive)	ratified (y/n)	ratified year (or expected) or future	MAJOR						ACT	SAIL	QEMU	SPIKE	GCC	LLVM	binutils
									RV I20	RVA20 (64 only)		RVA22 (64 only)		RVA23 (64 only)							
									m - mandatory, mH - m if H is implemented, o - optional, n - non-profile options, p - part of an optional extension but not an optional extension itself												
									RVI32 is only applicable to RVM and RVI profiles												
									Mode												
										u	s	u	s	u	s						
A	original	Atomics	unpriv	Zaamo, Zalrsc			y	2019	o	m	m	m	m	m	m	n	n					
C	original	Compressed instructions	priv				y	2019	o	m	m	m	m	m	m	y	y					
D	original	floating point, double-precision (implies F)	unpriv		F	Zdinx	y	2019	o	m	m	m	m	m	m	y	y					
F	original	floating point, single-precision	unpriv		Zicsr	Zfinx	y	2019	o	m	m	m	m	m	m	y	y					
H	H	hypervisor	unpriv				y	2021	n	n	n	n	o	n	o	n	n			n/a	n/a		
M	original	multiply/divide	priv				y	2019	o	m	m	m	m	m	m	y	y					
N	N	user level interrupts	unpriv				n	future	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
Q	original	floating point, quad-precision	priv				y	2019	n	n	n	n	n	n	n	n	n	n	n	n	n		
RV32E	RVE	integer base for RVE32	unpriv				y	2023	n	n	n	n	n	n	n	y	y						n
RV32I	original	integer base for RVI32	unpriv				y	2019	m	n	n	n	n	n	n	n	y						
RV64E	RVE	integer base for RVE64	unpriv				y	2023	n	n	n	n	n	n	n	y	y						n
RV64I	original	integer base for RVI64	unpriv				y	2019	m	m	m	m	m	m	m	n	n						
		only if H: For any hpmcounter that is																					

RISC-V®

Commercial & Open Source SW

Project Name	Source Code Location	mark's target	mark's categories	Project Area	Has it started work on RISC-V?	RISC-V Support Stage (the baseline is the x86_64 support when possible; supported means it has the base support for RISC-V, works, but there is still more work to be done; fully supported means out-of-box suppport for RISC-V)	riscv32	riscv64	oss or commercial	community-driven
LibreOffice	https://git.libreoffice.org/core/	user	app	Office Suite	1	supported	☐	☑	Open Source	☐
MediaWiki	https://github.com/wikimedia/mediawiki	it	application	Collaboration	1	supported	☐	☐	Open Source	☑
WordPress	https://core.trac.wordpress.org/browser		application	Collaboration/Content Management	?	unknown	☐	☐	Open Source	☐
gcc	https://github.com/gcc-mirror/gcc	developer	compiler	Toolchain	1	supported	☐	☐	Open Source	☐
GNU Mes	https://www.gnu.org/software/mes/	developer	compiler	Bootstrap	1	in progress	☐	☑	Open Source	☑
GO	https://github.com/golang/go	developer	compiler	Toolchain	1	supported	☐	☑	Open Source	☐
Green Hills	?	developer	compiler	Operating System	1	supported	☑	☑	Commercial	☐
LLVM	https://github.com/llvm/llvm-project	general development	compiler	Compiler	1	supported	☐	☐	Open Source	☐
Rust	https://github.com/rust-lang/rust		compiler	Language	1	supported	☐	☐	Open Source	☐
MariaDB	https://github.com/MariaDB/server	infrastructure	database	Database	1	supported	☐	☐	Open Source	☑
memcached	https://github.com/memcached/memcached	infrastructure	database	Storage	?	unknown	☐	☐	Open Source	☐
MongoDB	?		database	Database	?	unknown	☐	☐	Open Source	☑
MySQL	https://github.com/mysql		database	Database	1	supported	☑	☑	Open Source	☐
Redis	https://github.com/redis/redis		database	Database	1	supported	☑	☑	Open Source	☐
cryptoapi		security	library	library	?	unknown	☐	☐	Open Source	☐

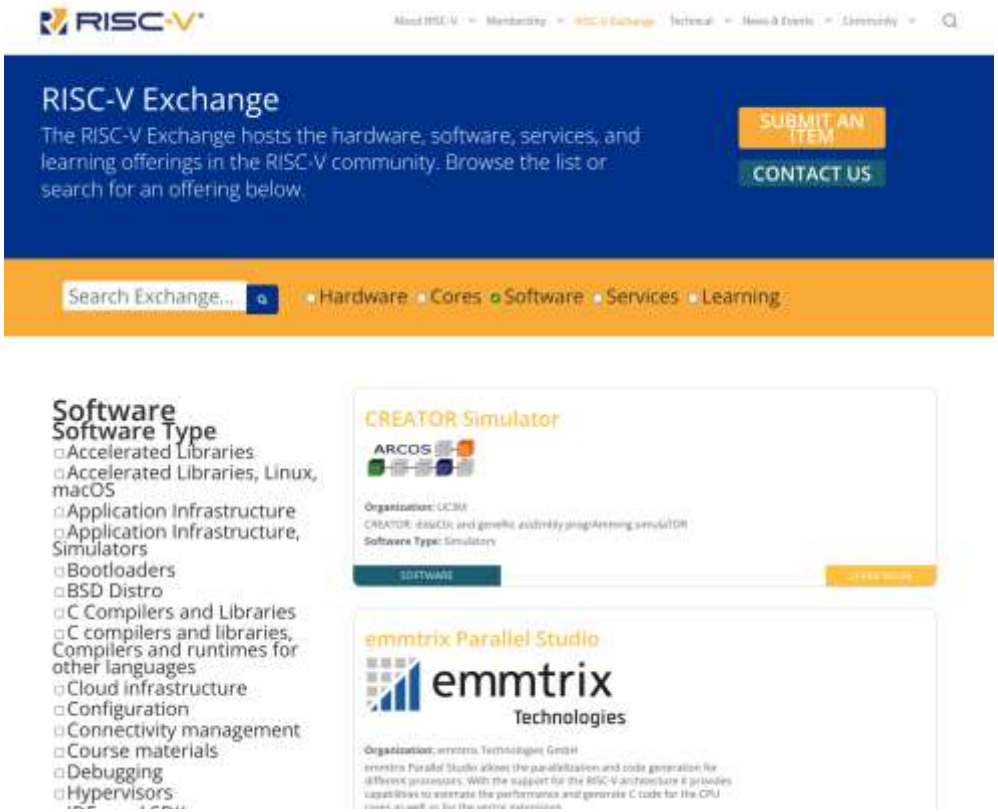
RISC-V®

Accelerating and building the strongest ecosystem

Source: https://landscape.riscv.org/
RISC-V software ecosystem status: https://sites.google.com/riscv.org/software-ecosystem-status

Updated August 2023

RISC-V Exchange



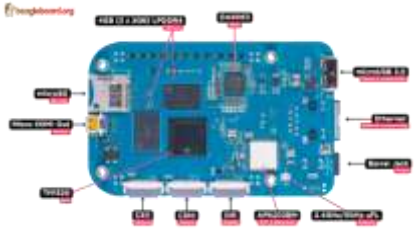
RISC-V®

Verification

- DV is done by implementers and DV providers
  - See risc.org/exchange for a list of providers
- Implementations can be wildly different
- Open source implementations include full RTL and DV
- Compatibility is determined by passing basic architecture tests (supplied by RISC-V and depends on SAIL formal model for golden results) and self-attestation for OSs, DV, etc.

Programs

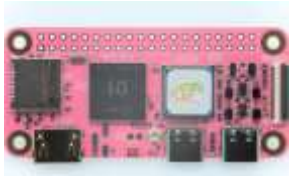
Programs: DevBoard Seed



Beagle

lichee 5

- Supply Chain issues dissipating.
- RVI distributing more boards in 2023
- Lots of new SBCs



Mango PI

Development Partners

- 7 Dev Partners
- 4 projects completed in 2023
- 22 projects underway
- Vector SAIL work still very close to final acceptance. Hopefully performing the final reviews.
- Intel working on RVV instructions for Valgrind as part of on-boarding process and will do future projects there.

Join RISC-V

Change the World!

Questions