

Compliance Task Group Call – Agenda

Thur, 17Dec2020 8am Pacific → Daylight ← Time

See slide 6 for agenda

Antitrust Policy Notice



RISC-V International meetings involve participation by industry competitors, and it is the intention of RISC-V International to conduct all its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

Examples of types of actions that are prohibited at RISC-V International meetings and in connection with RISC-V International activities are described in the RISC-V International Regulations Article 7 available here: <https://riscv.org/regulations/>

If you have questions about these matters, please contact your company counsel.

RISC-V International Code of Conduct



RISC-V is a free and open ISA enabling a new era of processor innovation through open standard collaboration. Born in academia and research, RISC-V ISA delivers a new level of free, extensible software and hardware freedom on architecture, paving the way for the next 50 years of computing design and innovation.

We are a transparent, collaborative community where all are welcomed, and all members are encouraged to participate.

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone.

<https://riscv.org/risc-v-international-community-code-of-conduct/>

Charter

The Compliance Task Group will

- Develop compliance tests for RISC-V implementations, taking into account approved specifications for:
 - Architectural versions (e.g. RV32I, RV32E, RV64I, RV128I)
 - Standard Extensions (H,**S,U,A,B,C,D,F,J,K,M,N,P,Q,T,V,N**), Priv Mode **<red is ratified >**
 - All spec'ed implementation options
 - (incl. MHSU modes, optional CSRs, optional CSR bits)
- Develop a method for selecting and configuring appropriate tests for a RISC-V implementation, taking into account:
 - Platform profile and Execution Environment (EE)
 - Implemented architecture, extensions, and options
- Develop a framework to apply the appropriate tests to an implementation and verify that it meets the standard
 - test result signature stored in memory will be compared to a golden model result signature

Administrative Pointers

- Chair – Allen Baum allen.baum@esperantotech.com
- Co-chair – Bill McSpadden bill.mcspadden@seagate.com
- TG Email tech-compliance@lists.riscv.org
 - Notetakers: please send emails to allen.baum@esperantotech.com
- Meetings -Bi-monthly at 8am Pacific time on 2nd/4th Wednesdays.
 - See https://docs.google.com/spreadsheets/d/1L15_gHI5b2ApkcHVtpZyl4s_A7sgSrNN zoom link
- Documents, calendar, roster, etc. in
 - <https://lists.riscv.org/tech-compliance/> see /documents & /calendars subdirectories ← will be changing to new git repo
 - <https://drive.google.com/drive/folders/1DemKMAD3D0Ka1MeESRoVCJipSrwiUIEs>
(lifecycle in “policies/supporting docs” folder, gaps in “planning” folder, compliance specific in “compliance folder”)
- Git repositories

	← docs riscv	→ tools
• https://github.com/riscv/riscv-compliance/tree/master/docs/		https://github.com/riscv/riscv-compliance/
• https://riscv.readthedocs.io/en/latest/index.html riscv		https://gitlab.com/incoresemi/riscv/
• https://riscv-isac.readthedocs.io/ ISA coverage		https://gitlab.com/incoresemi/riscv-compliance/riscv_isac
• https://riscv-ctg.readthedocs.io/ Compat. Test Gen.		https://gitlab.com/incoresemi/riscv-compliance/riscv_ctg
• https://github.com/riscv/riscv-config/tree/master/docs YAML, WARL config		https://github.com/riscv/riscv-config/
• https://github.com/rem-s-project/sail-riscv/tree/master/doc Sail formal model		https://github.com/rem-s-project/sail-riscv
- JIRA: <https://jira.riscv.org/projects/CSC/issues/CSC-1?filter=alopenissues>
- Sail annotated ISA spec: in <https://github.com/rem-s-project/riscv-isa-manual/blob/sail/>

	how to annotate	release/riscv-spec-sail-draft.pdf	
• README.SAIL			annotated unpriv spec
• release/riscv-spec-sail-draft.pdf	annotated source	release/riscv-privileged-sail-draft.pdf	annotated priv spec

Meeting Agenda

- 0. **Looking for more admins, maintainers for riscv-compliance git repo !!**
- I. Updates, Status, Progress
 - I. Current Tests tagged v1.0; RFQ tests merged and Tagged v2.0 (draft announcement to mailing lists in slide 8)
 - II. Most github issues will be closed as a result
- II. Next steps and Ongoing maintenance
 - 1. corner case assertion fixes (v2.1)
 - 2. more permanent assertion fixes using Sail model results as the correct value
(additional assertion/sigupd macros needed for FP, vector regs)
 - 3. Hosted Target directories
 - 4. Restructuring assertion macro to enable more out-of-line tests (reduce test size when assertions enabled)
 - 5. Tests for non-deterministic result (see attached discussion in email)
 - 6. Migration to Framework v.3.0 (riscov). video: <https://youtu.be/VIW1or1Oubo>,
slides: <https://lists.riscv.org/g/tech-compliance/files/Presentations/TestFormatSpec.pdf>
 - What steps do we need to complete to cut over to V.2 (see slide 13)
 - (e.g. ~~Sail model updates~~, pipecleaning, N people have run it, testing all the “fixed in riscov” issues
 - Review Pipecleaner tests: What do we need to do to exercise capabilities for Priv Mode tests
 - 7. Specific Compliance Policy/Process Gaps:
 - 1. Develop SAIL maintenance plan
 - 2. Certifying passing architecture tests: what needs to be in the report? Where does report get sent? (e.g. vendorID/archID)
 - 3. Can we certify actual HW if only its core RTL has passed architecture tests?
 - 4. How do we enable configurable & licensed core IP
 - 5. Identify Tool providers, e.g. coverage model, test generation for new features/extensions
 - 6. Flesh out test development order & identify resources (e.g. Priv,FDD or F,Priv,D..., JIRA CSC-3,5

Discussion

Updates:

CHAIR: New tests merged onto trunk as V2.0. Email announcement is forthcoming.

Most github issues will be closed. Still need to do: assertion clean up for correctvals

INCORE: Working on it. Issues w/ Python's various division methods (rounding modes, etc.)

Next Steps

CHAIR: Next version (2.1) correctval cleanup. Further planned changes

- SAIL-generated correctvals for assertions.
- additional assertion macros for FP, vector regs

Hosted Target directories

CHAIR: Currently have ~8 directories for target-specific files. Proposal to remove them from our repo into the repos of the target themselves. This includes Spike and Sail. Comments?

<none>

CHAIR: hearing no objections, they will be removed after they have been merged into their respective repos..

Testing for Non-deterministic results

< need to refer to the pdf attached in email: "Testing for non-determinism" >

CHAIR: How do we test things that are non-deterministic? We already have cases like this. For example, mis-aligned access. Some implementations will handle these with exception traps. Sometimes, the same core will handle mis-aligns with both a trap and without a trap. Our infrastructure can't handle this. Other examples:

- writing to PTE entry followed by Ld/St/Br to an address covered by that entry without intervening Fence

- memory ordering can give many (but bounded) different answers
- unordered vector reduce can give arbitrary results within some bounded range
- FP Nan results may not be result in a specific Nan value

CHAIR: The architecture allows case with 2 or more legal possibilities. How do we handle this?

INSPIRE: If there is a fence instr, then the answer is deterministic.

But now this is a software issue, not an ISA issue. We should test only correct code.

CHAIR: Yes, in specific case we test it, only with the intervening Fence.

That solution is not possible with misaligneds, because effectively all misaligned are "incorrect" This a long known problem ; the question is how or how much we want to test.

For coverage, running tests many times & measuring statistics of possible answers may work.

NVIDIA: <referring to **Sail** email> Tools have been built to stress re-ordering changes. SAIL may not support various interleavings/timings. Another tool is used.

CHAIR: I thought SAIL could be modified to enable external timings. Is the tool standalone or run in conjunction with Sail?

NVIDIA: Don't know

Memory order issues

CHAIR: Imperas uses a (DV) industry approach that requires target models to implement aa timing ABI. Done for ARM. May not be practical for the number of implementations RISC-V will have to deal with.

NVIDIA: I addressed this in my email. Putting in the hooks will be a lot of work.

And still can't guarantee to produce all legal outcomes.

CHAIR: One thing we could do: users must supply configurations.

IncCore2 - Could we use negative testing to handle this?

ie - check that the answer is not a *wrong* answer.

CHAIR: Yes – that is the idea; There are a limited number/range of correct answers.

How many do we need to specify? Hard to produce more than one.

How many do we need to test for compatibility coverage?

Framework must allow a set of possible answers –

How do we measure coverage if not all are possible to generate?

CoChair: Several problems being addressed: SAIL generation of multiple possible outputs; misaligned accesses, & vector reduce are a completely different cases.

CHAIR: For misalign (and perhaps others), SAIL could be run multiple times to get all possible answers.

INSPIRE: How can a system, running at bare metal, support both misaligns?

CHAIR: < examples> <discussion about what is bare metal, what we're required to test> We need to be able to test everything listed in the priv & unpriv specs. E.g. the priv spec has PTE definitions, how to walk PT, etc. That must be tested, exercising corner cases.

EMAIL announcement:

CHAIR: Here's the email announcement <see slide 8> for 2.0 . Any issues?

INCORE: Remove version numbers, miscompares are assertion errors, Not signature failures, right?

CHAIR: Right. Wording is ambiguous.

<wordsmithing>

Email Announcement

- To tech, isa-dev mailing lists:

The Architectural Compatibility (néé Compliance) TG is pleased to announce v2.0 of the Architectural Tests and framework

- new tests with much improved coverage for both RV32I and RV64I + M,C-extensions
- now c- as part of this version, two new tools have been open-sourced for test writers:
 - Conforms to the architectural test format specification
 - Compatibility Test Generator docs here: <https://riscv-ctg.readthedocs.io/>
 - Coverage Tool docs here: <https://riscv-isac.readthedocs.io/>
- Improved documentation with example files
- Directory reorganization for future extensions, maintainability, and ease of adding new tests,
- A standard trap handler has been added to enable easy testing of privileged modes, exceptions, and interrupts
- Macro definition and organization changed to enable easy custom model interfacing
- Temporary aliases and symlinks have been included for backwards compatibility with existing models
- Caveats:
 - there are a few known false assertion failures which will be fixed shortly
 - all tests assume that HW misaligned load/store is not implemented.
 - misalign tests assume that C-extension is enabled

Framework V3.0 (currently named riscof) is intended to fix these by configuring the tests to match the implementation, and comparing it to an identically configured golden model – **except** if an implementation supports some misaligned accesses in HW, but not others (even to the same address)

Decisions & Action Items

Decisions

Target files will be removed from riscv-compliance/ riscv-target directory into repos for the models themselves and replace with a README documentation file that points to the mode, the arch-test target environment files, and any build instructions deemed useful

Outstanding Action Items

NEW

Chair: document remedial work needed & send to CTO

Chair : document target process for removing target environment files from riscv-compliance repo into a target repo and contact all model maintainers to inform them of the process and timeline.

Chair: send out wordsmithed email announcement to isa-dev, chairs (tsc?)

Old

QC will ask Krste about dealing with emulation <new>

Inspire: Put dummy link script/model inclusion script into repo <done>

Everyone: report if RVTEST_IO_CHECK macro is used

Inspire: add support for QEMU target <?>

Chair: get SAIL repo moved into a riscv repo <ongoing>

QC: extract bits of FAQ as guidelines for test writing <?>

Incore: Try YAML version of SAIL to see if it works <not done>

[everybody]: comment on base ISA cover points:

<https://github.com/incoresemi/riscv-compliance/tree/dev/coverage>
(this is needed to complete the TG's responsibilities for the RFQ)

Imperas: make pull request for updated assertion macro

SH: write up coverage taxonomy

Everybody: read policy docs, send gaps in compliance (e.g. formal model support, possible mismatch between config TG and riscv-config) and priority to cto@riscv.org

Architectural Test Rationale – Intent and Limits

RISC-V Architectural Tests are an evolving set of tests that are created to help ensure that SW written for a given RISC-V Profile will run on all implementations that comply with that profile.

These tests also help ensure that the implementer has both understood and implemented the specification.

The RISC-V Architectural Tests test suite is a minimal filter. Passing the tests and having the results approved by RISC-V International is a prerequisite to licensing the RISC-V trademarks in connection with the design.

Passing the RISC-V Architectural Tests does **not** mean that the design complies with the RISC-V Architecture. These are only a basic set of tests.

The RISC-V Architectural Tests are **not** a substitute for rigorous design verification; it is the responsibility of the implementer to deploy extensive testing.

To be added to the `riscv/riscv-compliance/doc/` directory as “RISC-V Architectural Test Rationale”

Test Acceptance Criteria – second cut

Tests must:

- conform to current standard of test spec (macros, labels, directory structure)
- use only files that are part of the defined support files in the repository
- run in framework
- run in SAIL and not fail any tests
- generate a valid signature using SAIL (that can be saved & compared with another DUT/sim)
- Report that test results propagate to signature
- has a clear configuration - i.e. which ISA extension it can be used with
- improves coverage
- use only standard instructions (fixed size per architecture LI, LA allowed)
- must be commented in test_case header

Framework Requirements – first cut

The framework must:

- Use the TestFormat spec and macros described therein
 - (which must work - including assertions)
- Choose test cases according to equations that reference the YAML configuration
- Define macro variables that can be used inside tests based on the YAML configuration
- Include the compliance trap handler, & handle its (separate) signature area
- Load, initialize, and run selected tests between two selected models, extract the signatures, compare results, and write out a report file
- Exist in a riscv github repo, with a more than one maintainer.
- Be easy to get running, e.g.:
 - run under a variety of OSES with the minimum number of distro specific tools.
 - Not require sudo privileges
- Have the ability to measure and report coverage
 - Coverage specification is a separate file
 - Could be a separate app

Pull/Issue Status

Issue#	Date	submitter	title	status	comments
#22	24-Nov-18	brouhaha	I-MISALIGN_LDST-01 assumes misaligned data access will trap	^	HW misalign support not configurable
#40	4-Feb-19	debs-sifive	Usage of tohost/fromhost should be removed		
#90	11-Feb-20	towoe	Report target execution error		
#106	22-Apr-20	jeremybennett	Use of pseudo instructions in compliance tests	fixed in RFQ tests	Will be closed in 2.1 or 2.2
#142	17-Nov-20	subhajt26	Not able to run compliance test for rv32E device and RV32E ISA	RV32E only	Not RV32EC or RV32EM
#145-9	01-Dec-20	imperas	Test I EBREAK,ECALL, MISALIGN_JMP/LDST, OpenHW	v	HW misalign support not configurable
#107	22-Apr-20	jeremybennett	Clang/LLVM doesn't support all CSRs used in compliance test suite	under discussion	-can we add an alias?
#109	06-May-20	Olofk	Swerv fails because parallel make	under discussion	May be fixed?
#115	06-jun-20	adchd	How to support on-board execution?	under discussion	
#157	15-Dec-20	stnolting	Memory requirement for new test framework	Unfixable?	
pull#128	29-jul-20	nmeum	grift: update for new directory structure	Correction made	Review by Marc, needs corectiono
pull#129	31-jul-20	nmeum	sail-riscv-ocaml: Disable RVC extension on all devices not using it	in process	Who can review this?
#45	12-Feb-19	debs-sifive	Reorganization of test suites for code maintainability	deferred	fixed in v2
#63	13-Aug-19	jeremybennett	Global linker script is not appropriate	fixed	Needs target provided linker scripts
#72	26-Oct-19	vogelpi	Allow for non-word aligned `mtvec`	deferred	needs v.2
#78	26-Jan-20	bobbl	RV_COMPLIANCE_HALT must contain SWSIG	Fixed	
#105	22-Apr-20	jeremybennett	Non-standard assembler usage	under discussion	Simple fix
#108	22-Apr-20	bluewww	RI5CY's `compliance_io.h` fails to compile with clang	Pull #152 fixes it	close after merge
#116	06-jun-20	simon5656	loss of 64bit test infrastucture	under discussion	Will be fixed by RFQ tests
#119	17-jun-20	allenjbaum	Missing RV32i/RV64i test: Fence	Test has been written	Close when RFQ test is merged
#132	15-aug-20	davidmlw	Why not just use mepc for mret?	Answered - close	Should be resolved
#135	04-sep-20	MikeOpenHWGroup	Request for a Tag on this Repo	assigned	Req. for non-hash tag; needs process
#155	03-Dec-20	bluewww	RI5CY: add minimum clang version#	Fixes issue #108	Merge after review
#156	08-Dec-20	panda1628	PMP/PMA Tests	Question answered	Can be closed

JIRA Status

Issue#	Date	submitter	title	status	comments
IT-1	27Aug/20	Allen Baum	Need to modify the description of compliance in https://riscv.org/technical/specifications/	done	
IT-4	01/Sep/20	Allen Baum	Add Jira link to TG home pages	In prog	
CSC-1	20/Aug/20	Ken Dockser	Come up with names for the tests suites that we are creating		1 st step done
CSC-2	20/Aug/20	Ken Dockser	Produce concise text to explain the Architecture Tests intent and Limits		Written, needs pull req
CSC-3	20/Aug/20	Ken Dockser	Come up with an internal goal for what we wish to accomplish with the Architectural Tests		Not written
CSC-4	20/Aug/20	Ken Dockser	Develop a roadmap for all the different categories of test suites that will need to be created		Not written
CSC-5	20/Aug/20	Ken Dockser	Develop a roadmap for releases of single-instruction Architecture Tests		Not written
CSC-6	20/Aug/20	Ken Dockser	Develop a reference RTL test fixture that can stimulate and check the CPU under test		Needs more discussion