

# Architectural Test Task Group Call

—

## Minutes

Thur, 28Jan2021 8am Pacific → **Standard** ← 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.

# Collaborative & Welcoming Community

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 are a continuous improvement organization. If you see something that can be improved, please tell us. [help@riscv.org](mailto:help@riscv.org)

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](mailto:allen.baum@esperantotech.com) Co-chair – Bill McSpadden [bill.mcspadden@seagate.com](mailto:bill.mcspadden@seagate.com)
- TG Email [tech-compliance@lists.riscv.org](mailto:tech-compliance@lists.riscv.org) ← will be changing
  - Notetakers: please send emails to [allen.baum@esperantotech.com](mailto:allen.baum@esperantotech.com)
- Meetings -Bi-monthly at 8am Pacific time on 2<sup>nd</sup>/4<sup>th</sup> Wednesdays.
  - See [https://docs.google.com/spreadsheets/d/1L15\\_gHI5b2ApkcHVtpZyl4s\\_A7sgSrNN](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> (lifecyle in "policies/supporting docs" folder, gaps in "planning" folder, compliance specific in "compliance folder")
- Git repositories
 

← docs	riscv	→ tools
<a href="https://github.com/riscv/riscv-compliance/tree/master/doc/">https://github.com/riscv/riscv-compliance/tree/master/doc/</a>	tests	<a href="https://github.com/riscv/riscv-compliance/">https://github.com/riscv/riscv-compliance/</a>
<a href="https://riscv.readthedocs.io/en/latest/index.html">https://riscv.readthedocs.io/en/latest/index.html</a>	riscv	<a href="https://gitlab.com/incoresemi/riscv/">https://gitlab.com/incoresemi/riscv/</a>
<a href="https://riscv-isac.readthedocs.io/">https://riscv-isac.readthedocs.io/</a>	ISA coverage	<a href="https://gitlab.com/incoresemi/riscv-compliance/riscv_isac">https://gitlab.com/incoresemi/riscv-compliance/riscv_isac</a>
<a href="https://riscv-ctg.readthedocs.io/">https://riscv-ctg.readthedocs.io/</a>	Test Gen.	<a href="https://gitlab.com/incoresemi/riscv-compliance/riscv_ctg">https://gitlab.com/incoresemi/riscv-compliance/riscv_ctg</a>
<a href="https://github.com/riscv/riscv-config/tree/master/docs">https://github.com/riscv/riscv-config/tree/master/docs</a>	YAML, WARL config	<a href="https://github.com/riscv/riscv-config/">https://github.com/riscv/riscv-config/</a>
<a href="https://github.com/rem-s-project/sail-riscv/tree/master/doc">https://github.com/rem-s-project/sail-riscv/tree/master/doc</a>	Sail formal model	<a href="https://github.com/rem-s-project/sail-riscv">https://github.com/rem-s-project/sail-riscv</a>
- JIRA: <https://jira.riscv.org/projects/CSC/issues/CSC-1?filter=allopenissues>
- Sail annotated ISA spec: in <https://github.com/rem-s-project/riscv-isa-manual/blob/sail/>
  - [README.SAIL](#) ← how to annotate annotated unpriv spec → [release/riscv-spec-sail-draft.pdf](https://github.com/rem-s-project/riscv-isa-manual/blob/sail/release/riscv-spec-sail-draft.pdf)
  - [release/riscv-spec-sail-draft.pdf](https://github.com/rem-s-project/riscv-isa-manual/blob/sail/release/riscv-spec-sail-draft.pdf) ← annotated source annotated priv spec → [release/riscv-privileged-sail-draft.pdf](https://github.com/rem-s-project/riscv-isa-manual/blob/sail/release/riscv-privileged-sail-draft.pdf)
  - <https://us02web.zoom.us/rec/share/-XIYazzhIBbQoiZdarCfebdxjDwiVhf-LxnuVrliN4Bc30yf17ztKkKDU4Og54b.fArPPqnuR-NiXpQU> Tutorial

Access Passcode: tHAR#5\$V

# Meeting Agenda

## 0. Looking for more admins, maintainers for riscv-compliance git repo !!

## I. Updates, Status, Progress:

1. Relocating test target directories. First step: Spike targets moved to riscv-isa-sim repo; next is removal from riscv-compliance repo, replaced with README
2. Elections: Want to be a chair of this? Send (self?) nominations Feb 01..15 to ?

## II. Next steps and Ongoing maintenance

1. Charter Updates and transitioning to a SIG
2. Changing group and repo names to arch-test (short for architectural compatibility test)
3. Defining/documenting/communicating procedure for certifying the passing of architecture tests:
  - a. what needs to be in the report besides list of pass/failed tests? (e.g. vendorID/archID, test, reference, and DUT repo tags, (eventually config YAML))
  - b. Where does report get sent? Who reviews it? Who votes on it? What is the waiver process?
4. Defining/documenting/communicating the policy for certification of physical chips, RTL model generators

## 5. Close github issues as a result of repo v2.1

## 6. Maintenance updates to V2 to enable future tests

- a) update RVTEST\_SIGUPD to keep automatically adjust base/hidden offset when offset>2K,
- b) Enable use Sail model results as the assertion value
- c) add assertion macros for FP, DP, Vreg to arch\_test.h and test\_format spec
- d) add trap handlers for S, VS modes

## 7. Rename of TG and Repo to “arch-test”

## 8. Update of charter (TG no longer developing tests, but guiding test development)

## 9. 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

# Draft Proposed Charter Revision

The Architectural Compatibility Test SIG is an umbrella group that will provide guidance, strategy and oversight for the development of tests used to certify that an implementation meets RISC-V ISA architectural compatibility requirements.

The group will:

- Guide Development of:
  - Architectural tests for RISC-V implementations covering ratified and in-flight specifications for
    - Architectural versions, standard extensions, and implementation options.
  - Tools and infrastructure to verify that an implementation is architecturally compatible
- Work with LSM and Chairs for resources to get the above work done.
- Mentor or arrange for mentoring for the resources to get the above work done

# Repository, Group Name Change

Procedure for changing names:

- Clone content from groups.io site
  - There is probably a 1-2 week turnaround time
- Send a note to the existing tech-compliance group announcing and detailing the changes (incl. mailing list name change to sig-arch-test)
  - Make sure that any documentation, scripts, makefiles, etc. that point to riscv/riscv-compliance is changed (e.g. in spike arch-test/README)
  - Should also be changed on <https://riscv.org/technical/specifications/>

After the groups.io site is cloned:

- change group's name, description, and alias the following <day? week?>
- change the name of the github repo
- create a new, empty github repo with the old name and a README file pointing them at the new repo
  - provide instructions on how to switch/add a new origin to local cloned github repo

Does this work for everyone?



# Test Report Requirements

- Architecture test version policy (mechanism TBD by ACT group)
  - Architectural test reports must include:
    - The ISA string that describes the ISA and extensions that claim to be implemented
    - The vendor and implementation IDs that the part will report
    - Test pass/fail reports
    - The YAML description of the features and options implemented (for v3 of the framework)
  - Architecture test reports must include version numbers of \*
    - Toolchain,
    - reference model,
    - Architecture Compatibility Test (ACT) suite , riscv-config (for v3 of the framework)
- Each release version of ACT will document the minimum version of the toolchain utilities required to support the instructions used for that version of the tests in a repository README file (*other info as well, where to find ISA string, etc*)
- Test Reports are submitted by pull request to arch-test-report repo named as implementor/implementationID
- Waivers are submitted by pull request to arch-test-report repo named as implementor/implementationID-waiver

\* *Need to ensure these get updated automatically*

# Who is required run tests? On which model?

- Basically, the last organization that changes any RTL must certify w/ Compatibility tests
  - must be run with latest version of tests, reference model and required versions of tools
  - further details in RISC-V Architecture Test & Certification policy.
- Branding based on this must follow the RISC-V Branding policy
  - This includes flow-through branding based on certification done by suppliers
  - *Failures require a waiver that must be approved by TSC (?)*
- If certification fails on later test-suite versions, an ERRATA report must be generated.

Ideally, tests are run on physical chips, but usually an RTL SW sim or FPGA

Platform definition must provide enough resources to run tests

A physical chip is impractical when it :

- has memory limitations that prevent (some) tests from being loaded or run
- lacks an off-chip interface that enables loading/running tests & extracting signature
- has authentication requirements prohibiting unsigned binaries to be loaded or run
- ... and probably some others

# Discussion

## Progress

**CHAIR:** first test target directory moved to sim dependent area: spike target support moved to arch-test-target subdirectory of riscv-isa-sim/ git repo.

**CHAIR:** elections for chair/vice-chair positions of this TG starting soon. Nominations open Feb 01 and end Feb 15. Email announcing details will be sent.

### 1. Charter revision (see slide 7)

**CHAIR:** we are proposing changing from a TG to a SIG along with new charter

**SAIL:** what is the difference?

**CTO:** TG has work products. SIG only oversees but just reports to HC.

**SAIL:** who delivers tests for this group?

**CTO:** group contributors. 3 groups currently engaged.

**INSPIRE:** need to say something about working with or driving development of in-flight extensions and specifications.

**CHAIR:** added to draft charter

**INSPIRE:** can we vote on it?

**CHAIR:** It needs time for people to review - will vote next meeting.

### 2. Group name change (see slide 8)

**CHAIR:** need to deprecate the term "compliance", so repo & group change to arch-tests

**CTO:** might be easier to just create a new group rather than "changing the name"

**Vice-CHAIR:** concerned about old sandboxes. can point to new repo?

**INSPIRE:** yes. change git origin. There will be a README

### 3. Test report requirements (see slide 9)

**CTO:** The SIG/TG needs a roll up summary for all extension names, tools, pointer to the tests, etc. This is now distributed; you'll have to go through many repos. (AI: Chair)

**CHAIR:** where do the test results get sent?

**CTO:** this is self-certification based on honor system - nobody reviews the report, waivers have to go to TSC. If there's a complaint, we point them at the report  
The TG/SIG should not be in the middle.

We should have a git repo with reports, and need to define the es waiver request mechanism (AI: chair, work with Jefro)

**CTO:** tech-compliance (or arch-tests...) must define report format.

Besides test pass/fail, also need company name, implementation name, tool/test/ref model versions, point-of-contact \*2, etc. AI: group)

**CTO:** There is no license to use the brand. Only the member type they need to be.

**INSPIRE:** branding can be passed down the line, right?

**CTO:** yes (see branding policy)

**INSPIRE:** companies may want to hide waiver requests.

**CTO:** sorry. we don't hide that information.

**INSPIRE:** I like the idea of putting reports into a git repo via a PR (pull request)

I suggest that waivers (also) be submitted via a github issue.

(**CHAIR:** need a new repo for test reports, waivers, etc) Process is a repo README

**QC** - PR to TG/SIG, then to HC, then to TSC for final approval.

### 4. review of slide "Who is required run tests? On which model?" (slide 10)

**CTO:** Don't need to be a member to use brand if "non-commercial+==non-profit

**QC** - what's common in the industry, is that the tests are run on the RTL

**CTO:** this needs to be run up to the TSC. there needs to be buy-in.

**QC** - if it's a burden to run on the chip. have to get the chip back from fab first  
**INSPIRE:** that's not what we're saying. Some companies can't run the full chip in simulation - too big.

??: Only need to test the core in simulation, not entire SOC

??: Physical Core may not have support enough address bits -

??: RTL must have address lines, but tie them off in final chip and/or ask for waiver.

??: How does a part with limited memory capability get tested/advertised as such?

??: Should we break up tests by address width?

AI: Chair: bring up to chairs mtg

**CTO:** May be irrelevant. But we MUST be able to prove that apps can be run.

# Decisions & Action Items

## Decisions

A git repo will be set up for submitting test reports and waivers.

- Test reports will have a defined format (see slide 9)
- Test report repo will have a defined structure
- Test reports will be in subdirectories named for implementor
- Report & waiver submittal will be via pull request to appropriate repo subdirectory named by implementorID.

## Outstanding Action Items

### NEW

**Everybody:** read and review proposed new charter & transition to SIG.

This will be sent to TSC as a ISA Infrastructure Horizontal Decision.

**Chair:** Get ISA Infrastructure HC approval (who is also this **TG Chair**....)

**Chair:** bring question up to chairs meeting about how to test limited memory parts, TEE parts that can't run tests

**Chair:** make slide of chairs meeting on what exactly gets tested, and Silicon vs. RTL waiver policy

**Chair:** write a draft spec for test report format and waiver procedures

**Chair:** get test report Repo set up.

**Chair:** talk to toolchain/runtime And CI group (Continuous Integration) about automatically updating version#s for test reports

**Chair:** gather list of extensions and DOD status.

### Old

**Chair:** send CTO email describing non-determinism < done>

**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. <ongoing>

**Inspire:** add support for QEMU target <?>

**Chair:** get SAIL, CTG, ISA 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>

**SH:** write up coverage taxonomy

# Backup

# Non-determinism in Architectural Tests

The RV architecture defines optional and model/ $\mu$ arch defined behavior.

This implication: there are tests that have multiple correct answers. E.g.:

- Misaligned accesses: can be handled in HW, by "invisible" traps w/ either misaligned or illegal access causes, and do it differently for the same op accessing the same address at different times (e.g. if the 2nd half was in the TLB or not)
- Unordered Vector Reduce ops: (different results depending on ordering & cancellation)
- Tests involving concurrency will have different results depending on speculation or timing between concurrent threads (e.g. modifying page table entry without fencing)

From the point of view of ACTs, there are 2 (and sometimes more) legal answers. Usually, the golden model only generates one. Possible mechanisms to test include:

- Modify (if necessary) & configure reference model to generate either result, run it with each and accept either result from the DUT.
- Provide specific handlers for optional traps
- Use self-testing tests(compare with list or range of allowed outcomes from litmus tests)
- Avoid tests that can generate non-deterministic results

# 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” satisfying Jira CSC-2

## 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 now
#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	subhajit26	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?
pull#163	11-jan-21	snolting	Makefile improvements	In process	Needs review
#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	RISCY'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	RISCY: add minimum clang version#	Fixes issue #108	Merge after review
#156	08-Dec-20	panda1628	PMP/PMA Tests	Question answered	Can be closed
#157	15-dec-20	Stnolting	Memory requirement for new test framework	Question answered	Can be closed
#158/164	23-dec-20	Stnolting	Add white space in verify report [absolutely uncritical]	Non-critical	Should be accepted (Pull #164)
#165	12-jan-21	Towoe	Version numbering	Non-critical	Close?, will use semantic vers from now on
#169	22-jan-21	Towoe	RISCOF redefine of TEST_CASE_1	Question answered	Can be closed

# JIRA Status

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