

# Architectural Test Task Group Call – Minutes

Thur, 25Mar2021 8am Pacific → Daylight ← Time

See slide 7 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/>

# 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)
- SIG Email [sig-arch-test@lists.riscv.org](mailto:sig-arch-test@lists.riscv.org)
  - 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://sites.google.com/a/riscv.org/riscv-staff/home/tech-groups-cal>  
<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
<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-arch-test/">https://github.com/riscv/riscv-arch-test/</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://github.com/riscv_isac">https://github.com/riscv_isac</a>
<a href="https://riscv-ctg.readthedocs.io/">https://riscv-ctg.readthedocs.io/</a>	Test Gen.	<a href="https://github.com/riscv_ctg">https://github.com/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/remns-project/sail-riscv/tree/master/doc">https://github.com/remns-project/sail-riscv/tree/master/doc</a>	Sail formal model	<a href="https://github.com/remns-project/sail-riscv/">https://github.com/remns-project/sail-riscv/</a>
- JIRA: <https://jira.riscv.org/projects/CSC/issues/CSC-1?filter=allopenissues>
- Sail annotated ISA spec: in <https://github.com/remns-project/riscv-isa-manual/blob/sail/>

<a href="#">README.SAIL</a> ← how to annotate	annotated unpriv spec → <a href="#">release/riscv-spec-sail-draft.pdf</a>
<a href="#">release/riscv-spec-sail-draft.pdf</a> ← annotated source	annotated priv spec → <a href="#">release/riscv-privileged-sail-draft.pdf</a>
- <https://us02web.zoom.us/rec/share/-XIYazzhIBbQoiZdarCfebdjxjDWiVhf-LxnuVrliN4Bc30yf17ztKkKDU4Og54b.fArPPqnuR-NiXpQU> Tutorial  
 Access Passcode: tHAR#5\$V

# SIG Charter

The Architectural Compatibility Test SIG is an umbrella group that will provide guidance, strategy and oversight for the development of tests used to help find incompatibilities with the RISC-V Architecture as a step in the Architectural Compatibility self-certification process

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 help identify architectural incompatibilities in implementations
- 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

# TGs under the SIG

- IF you're creating work product, you should be a TG
- If changing requirements, plans ABIs, etc
  - Test plan==SOW
- The Architectural Testing Task Group will define and maintain specifications for
  - test formats
  - test-benches and frameworks needed for
    - privilege testing privilege testing,
    - Concurrency/ Memory model testing
    - Asynchronous event testing (interrupts)
    - Nondeterministic tests
  - ISA test coverage goals
  - test tools (e.g. coverage, generators)
- The Architectural Testing Task Group will maintain the appropriate GitHub:
  - tests for the individual ISA extensions
  - issues related to the tests
  - the operation and issues related to the framework
- The Architectural Testing Task Group will
  - work with the different privilege and un-privilege ISA extension Task Groups
    - to help them write test plans/specs for the ISA tests
    - to help them work with the sub-contractors (IITMadras, RIOS, CAS, etc) to deliver the tests
  - assess quality of delivered tests and be maintainer for the test GitHub

# Meeting Agenda

0. **Looking for more admins, maintainers for riscv-compliance git repo !!**
- I. **Updates, Status, Progress:**
- II. **Next steps and Ongoing maintenance**
  1. Define report standard
  2. Discussion: Migration to Framework v.3.0 (riscov).
    - What steps do we need to complete to cut over to V.3 (blocking items):
      - (e.g. Sail/Spike 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
  3. Close github issues as a result of repo v2.1
  4. 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
  5. Tests for non-deterministic result (see attached discussion in email)
    1. Provide a reference RTL test fixture (as opposed to SW functional model). See. JIRA CSC-6
    2. Define hooks for concurrency tests
  6. Specific Compliance Policy/Process Gaps:
    - a. Identify Tool providers, e.g. coverage model, test generation for new features/extensions
    - b. Flesh out test development order & identify resources (e.g. Priv,FDD or F,Priv,D..., JIRA CSC-3,5

# Discussion

## Passdowns, Status

**Chair** - Charter has been updated with a clarification on architectural tests vs self-certification

**CTO** - We're trying to complete the definitions for the platforms and how extensions, esp as how instructions/features are shared between extensions. The intent is having binaries run from platform to platform.

< discussion about what a sub-extensions are (currently an undefined term) >

**CTO** - the bare-bones micro guys want to be able to minimize so that they choose only what they need – they don't even want to trap if DIV is encountered. .elf file headers define which extensions are required, and won't run if core doesn't support them.

**Chair** – proposing multiply extension. M ext. will only multiply and not divide.

<done – issue is identifying a “sponsor” to establish need, else this is useless>

**Imperas** - crypto test update: updated PR has been filed

*IA: review and integrate PR*

?? - Updates to SPIKE for them are in place.

Some makefile changes are required because std toolchain doesn't have support yet

**Incore** –approved by us

<<. merged and renamed K\_unratified.

*riscv-target/<yourmodel/device/<arch>/K\_unratified/ directory must be created to run them. >>*

## Test Report Requirements

**CTO** – Architecture Test policy needs to be complete. The urgency here, is that marketing won't brand until there is an arch test policy in place. Doesn't have to be perfect. AI: Need an action item to get this complete.

**Chair** – policy draft written; the remaining work is the reporting standard (next agenda item)

*IA: update and send out for review, see: [for riscv members > policies > being written and pre-review](#)*

< refer to slide 12 of Agenda >

**Incore** - ISA string, vendor & implementationIDs are part of the YAML, so inserted by framework

**Chair** – Vender/ImplementationID are allowed to be 0, which means we can't differentiate.cores.

Proposal: report directory name will have venderName, File will have InstanceID-datetime

**Inspire** - need to detail the steps used to build and run the tests. toolchains (version), simulator (version). need to be clear about toolchain that supports the tests; may need use a branch of the toolchain. so that we can reproduce. E.g. downloaded from <blah>, using version <blah>, etc. this needs to be in a file in the same place as the test report.

< refer to proposed policy about re-testing: once a year >

**Incore** - neither SAIL nor SPIKE have version numbers

**Chair** – can we list a github tag instead for now?

< notes are being added by Chair to “Test Report Requirements in minutes, slide 12 >

**Imperas** -if you going to specify tool version, are you going to specify the version Linux? toolchains won't run on all forms of Linux

**CTO** - we haven't specified the version of the infrastructure?

**Inspire** – No, in embedded world, they specify "we ran this on Ubuntu version x, etc."

**Imperas** - this is self-certification. why do we need to specify required toolchains, etc?

*Decision: ACT will document tool versions of framework etc used to test the tests only*

**CTO** – IF someone fails using a different version, don't ask for a waiver until you have a root caused why there was a failure

**Inspire** - we need to have them supply their baseline in any case

*IA- DISCUSS DETAILS: Self cert needs to submit a separate text file listing tools versions used alongside the .html test report. It should contain the same information as the ACT README section that list the tools that were used to compile tests and framework.*

**CTO** - profiles & ACT need consistency. Can't self-certify w/ different toolchain versions.

ACT should consider development branches for Pre-upstream tools

This group decides which versions are used of tools for specific profile needs

*AI: Talk to kristoff, Evandro Menezes to figure out how this should work*

## V3 adoption

**Chair**: blocker is currently that V3 needs more support for yaml in Spike, Sail

**Imperas**: Need a schedule for sail changes

**Incore**: There is a Sail branch with outdated YAML support.

Adding more is under active development, should have a status update this weekend.

Pipcleaning tests will show priv level capabilities, see:

<https://gitlab.com/incoresemi/riscov/-/tree/pipcleaning/riscov/suite/wip/pipelinecleaning>

Riscov is currently in a separate gitlab repo;; should we:

move to a new riscv repo? Add to arch tests as a submodule ? as subdirectory?

*Decision: add to a arch-test branch as a new directory, then merge after testing complete*

*Inspire: AI: confirm using Inspire and Seagate cores that rename and target directory moves work first.*



# Decisions & Action Items

## Decisions

Modify wording of SIG charter to clarify self-certification rationale and relationship to arch tests <done>

Clarified that self-certification with implementation selected configuration variables in vendor supplied IP is not considered an RTL change. <done, to be written up as part of the ACT policy – to be reviewed>

Remove RVTEST\_IO\_CHECK macro from test format spec

Vector tests will not test unordered vector reduce, but only architectural defined ordered cases.

## Outstanding Action Items

### NEW

**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>

**Chair**: need to write a non-determinism policy that documents how a new non-deterministic feature testing fits into the lifecycle of an extension. At what point does the testing methodology need to be in-place in order to approve it? <done – added to D.O.D policy and checklist: plan stage must describe methodology>

**Chair**: more brainstorming on handling nondeterminism, concurrency <discussion started on retrofitting riscv for concurrency>

**Chair**: need clarity on tool source/version report.

### Old

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

**Chair**: get SAIL repo moved into a riscv repo <done, but for viewing only>

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

**Incore**: Try YAML version of SAIL to see if it works <ongoing>

**Imperas**: make pull request for updated assertion macro <removed, replaced with code from TGChair>

**SH**: write up coverage taxonomy

# 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 microarchitectural state, speculation, or timing between concurrent threads (e.g. modifying page table entry without fencing)

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

- Modify (if necessary) & configure reference model to generate each legal result, run it with each config, & accept either result from the DUT (e.g. misalign or un-fenced PTE modification)
- 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
- Ultimately: develop new frameworks that can handle concurrency along with reference models that can generate all legal outcomes
- It is the responsibility of the TG that develops an extension to develop the strategy for testing features and extensions that can have nondeterministic results

BACKUP

# Test Report Requirements

- Architecture test version policy (mechanism TBD by ACT group)
  - Architectural test reports are generated by the framework and include:
    - The YAML description of the features and options implemented (for v3 of the framework)
      - This will include the ISA string, and options that claim to be implemented
      - The vendor and implementation IDs that the part will report
    - Test pass/fail reports
    - Version numbers of
      - Toolchain used to compile tests,
      - reference model used,
      - Architecture Compatibility Test (ACT) suite, and riscv-config tool (for v3 of the framework)
- Vendors who self-certify generate pull request into arch-test-reports github repo
  - Report is stored in subdirectory <vendor\_name> with filename <instanceID>-<date>.html
  - Additional file that describes the buildsteps used with filename <instanceID>-<date>.buildsteps
    - List source and version of tools, simulator, etc
      - same items as listed in ACT README that describes tools and versions ACTs were tested with
    - Failures found using different versions is not a valid reason for waiver unless reproducible with standard
- Each release version of ACT will document, in the repository doc/VERSION.md file, the versions of toolchain utilities used for the version of the tests in that repository
  - detail the steps to run, source of tools, and there version numbers

# 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 – draft3

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 DUT/sim)
- Report that test results propagate to signature
- have a clear configuration - i.e. which ISA extension it can be used with
- improve coverage (compared to existing tests) as measured and reported by ISAC
- use only standard instructions (fixed size per architecture LI, LA allowed)
- be commented in test\_case header

Tests that are otherwise accepted, but depend on tools or simulators that have not be upstreamed must be put into a <Ext-Name\_beta>/ directory instead of <Ext-Name>/

# 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	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	
pull#128	29-jul-20	nmeum	grift: update for new directory structure	Correction made	Reviewed by Marc, needs correction
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#172	27-feb-21	imperas	Crypto Scalar Tests (also issue#173)	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
#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
#170	18-feb-21	Panda1628	Privilege Mode test	Question answered	Can be closed
#175	18-mar-21	AllenJbaum	README.md needs a title change	In process	Simple fix



# 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	done	
<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	done	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

# Conventions



- We don't solve problems or detailed topics in most meetings unless specified in the agenda because we don't often have enough time to do so and it is more efficient to do so offline and/or in email. We identify items and send folks off to do the work and come back with solutions or proposals.
- If some policy, org, extension, etc. can be doing things in a better way, help us make it better. Do not change or not abide by the item unilaterally. Instead let's work together to make it better.
- Please conduct meetings that accommodates the virtual and broad geographical nature of our teams. This includes meeting times, repeating questions before you answer, at appropriate times polling attendees, guide people to interact in a way that has attendees taking turns speaking, ...