

# Compliance Task Group Call – Minutes

Thur, 14Jan2021 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.

# 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

# 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 specifications for
    - Architectural versions, standard extensions, and implementation options.
  - Tools that generate tests.
  - Tools for measuring test quality
  - Framework(s) that
    - Specify configuration schema describing architecture, extensions and options supported by an implementation
    - Apply and configure tests appropriate to an implementation and reference model based on that configuration,
    - Verify that an implementation is architecturally compatible by comparing signatures from a golden model
    - Print test results that meet the acceptance criteria of the ISA Infrastructure Horizontal Committee
- Evolve, maintain and update the test format specification
- 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

# 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. Current Tests tagged v2.1
  2. Resources: IIT team started work on FP tests, Rios Labs team looking into Sail support, CAS/PLCT looking to help with QEMU
  3. First step: Spike targets moved to riscv-isa-sim repo & removed from riscv-compliance repo
- II. **Next steps and Ongoing maintenance**
  1. Charter Updates and transitioning to a SIG
  2. 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?
  3. Defining/documenting/communicating the policy for certification of physical chips, RTL model generators
  4. Changing group and repo names to arch-test (short for architectural compatibility test)
  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 (riscvf). 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, ninescleaning, N people have run it, testing all the “fixed in riscvf” issues

# Discussion

## CTO Passdowns:

- Proposed simulator charter SIG: In a similar vein, tech-compliance charter should change because other groups are doing the actual tests; its about infrastructure steering.
- Suggestion: no longer a task group, but rather a SIG. Overhead is reduced (eg - approval and vote from TSC). *AI - CHAIR: rewrite charter and send to TG members for review*
- Krste is insistent that Horizontal Committee approve anything goes out for public review or goes out for a vote. Check for consistency amongst the various TGs.
- Industry/academic groups who are doing work for RVI. IIT-Madras (FP Tests), RIOS (Sail), CAS/PLCT(Qemu). TGs (&SIGS, etc) spec architecture. These groups implement.
- Top priority for tests: Vector, Crypto Scalar, BitManip, VM

**CHAIR:** A certification procedure needs to be defined&documents so users can self-certify "compatibility". Ex: along with report test pass/fail, report versions of SAIL, tool chain, test repo version used, YAML file that specifies config, vendor/impl ID.

*AI - CHAIR: talk to toolchain/runtime And CI group (Continuous Integration).*

**CTO:** Concerned about making sure to stay consistent with toolchain upgrades/changes other groups (like the toolchain group).

**CHAIR:** What do we do when someone wants to add a new test?

**CTO:** Need to \*reduce\* the work that tech-compliance has to do. However, at the start, there will have to be manual audits. Perhaps evolve into scripting, etc.

**CHAIR:** How do we handle waivers? A user runs tests, gets a failure, wants a waiver.

**CTO:** It's very clear. It's not compatible.

**INCORE:** What about things we can't (or haven't) test? Deficiency in our testing.

**CHAIR:** Example: mis-aligned addressing. These will (and currently do) fail

**CTO:** *AI-CHAIR: send me email describing this.*

**Inspire:** Errata is what happens after silicon.

**INCORE:** Can validate RTL and then real silicon?

**CTO:** Branding needs to be on a real product, whether soft or hard.

**INCORE:** This is a problem. Pass/Fail is based on signatures in memory.

No guarantee it can b extracted with requiring that users supply a debugger.

**CHAIR:** Another: really tiny version of chip with only small SRAM

We test max offset jumps larger than available memory, or even load the tests.

**CTO:** That's ok. And may be irrelevant. But we MUST be able to prove that apps can be run.

**CoChair:** That's a huge addition to the goals of this group. It's never been a part of the charter of this group. Can't overstate this.

**CTO:** If not a real "platform", then run on RTL and test it. That could (would) be good enough.

**CHAIR /Inspire:** There are problems with this.

**CTO:** Compare this to X86 machine. It seems like you're saying that Windows needs to run, as an example. I don't see that happening. I don't see RISC-V desktops. There are platforms. Beagle-board and Freedom Unleashed

We need to make changes to the branding policy in order to explain this.

And this may (will?) need to be run by the Board.

Inspire gave a good description about why (and how) the arch tests can (and should) be run in an RTL environment. So, I'm retracting what I said earlier.

We will need to come up with an RTL vs. Silicon waiver policy and bring it to chairs mtg. next week.?

**CHAIR:** Maybe a list of permitted reasons?:

e.g. mem space, debug, trusted environment limitations

**CTO:** We can't increase number of trademarks/brand

*AI Chair: bring slide for strategy platform, vs non platform, waiver vs errata,*

- where can we certify rva20?, does it depend on end product?

- If not complying either: deliberate or errata.

Other: how do we deal with new tests that expose bugs?)

RVA20 is dependent on testsuite V3.x and Sail model - part of profile? Or platform?



# Email Announcement (sent)

- 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
- 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 riscov) 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 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>

### Old

**QC** will ask Krste about dealing with emulation <new>

**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 <removed, replaced with code from TGChair>

**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](mailto:cto@riscv.org)

# Sample target README file (Spike)

## Using the Spike Simulator as an Architectural test model

This is a reference for running Spike as a target for the RISC-V Architectural Test framework.

### Getting Spike

The Spike repository should be cloned from [here](#), preferably at the same directory level as the riscv-compliance repository.

### Building Spike

The [README.md](#) at the top level of the riscv-isa-sim directory gives details on building an executable spike model.

### Integrating Spike into the Architectural Test framework

Also at the top level spike repo is an `arch_test_target` directory.

The **ROOTDIR** environment variable must point to the riscv-compliance repo, and the **RISCV\_TARGET** environment variable must be set using: `export RISCV-TARGET ?=spike`

There are two ways to do the model integration:

1. Copy the `arch_test_target` directory into the `riscv-compliance` directory, e.g. from the directory that contains both `riscv-compliance` and `riscv-isa-sim` repo directories, run: 

```
cp -r ./riscv-isa-sim/arch_test_target
./riscv-compliance/riscv-target/spike
```

 and then set the **RISCV-TARGET** environment variable: 

```
export TARGETDIR ?= $(ROOTDIR)/riscv-target
```
2. Point the **TARGETDIR** environment variable directly to the spike repo: 

```
export TARGETDIR ?= $ROOTDIR/./riscv-isa-sim/arch_test_target
```

Other environment variables are detailed in the compliance repo `doc/README.md`:

**RISCV\_DEVICE** *Only all or single extension can be tested*

**RISCV\_TARGET\_FLAGS**

**RISCV\_ASSERT**

**JOBS**

Files in the target directory may need modification to customize what will be tested.

This includes

**model\_test.h**: contains any model specific setup and required macro definitions

**link.ld**: defines where tests are loaded & where input and output results are located

**Makefile.include**: for any special parameters to make  
(inside each device/<arch>/<extension> subdirectory)

# 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 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)

# 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