

Compliance Task Group Call – Minutes

Thur, 30Jul2020 8am Pacific → Daylight ← Time

See slide 8 for discussions and action items

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,H,J,M,N,P,T,V,N), Priv Mode
 - 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://lists.riscv.org/g/tech-compliance/calendar> entry for zoom link
- Documents, calendar, roster, etc. in
 - <https://lists.riscv.org/tech-compliance/> see /documents & /calendars subdirectories
 - <https://riscof.readthedocs.io/en/latest/> riscof
 - <https://riscv-config.readthedocs.io/en/latest/> config: YAML and WARL spec
 - <https://drive.google.com/drive/folders/1DemKMAD3D0Ka1MeESRoVCJipSrwiUIEs>
(lifecycle in "policies/supporting docs" folder, gaps in "planning" folder, compliance specific in "compliance folder")
- Git repositories
 - <https://github.com/riscv/riscv-compliance/>
 - <https://gitlab.com/incoresemi/riscof> (riscof framework)

Meeting Agenda

- Updates, Status, Progress
 - Policy/process updates (specifically Done policy)
- Discussion:
 - Compliance FAQ - any more comments?
 - Specific Compliance Policy/Process Gaps:
 - Criteria for approving merge requests (e.g. coverage, Sail model integration, size, time to run)
 - Certifying compliance: what needs to be in the report? Where does report get sent? (e.g. vendorID/archID)
 - Can we certify actual HW if only its core RTL has passed compliance test?
 - How do we enable configurable & licensed core IP
 - Migration to Framework v.2
 - Review Pipecleaner tests: What do we need to do to exercise capabilities for Priv Mode tests
 - What steps do we need to complete to cut over to V.2
(e.g. Sail model updates, pipecleaning, N people have run it, testing all the “fixed in riscov” issues)
 - Coverage Spreadsheet
 - critique, review (See: Coverage Rules.xlsx in <https://lists.riscv.org/g/tech-compliance/files/Review%20Documents>)
 - Next steps and Ongoing maintenance
 - Who should provide Tools, e.g. coverage model test generation for new features/extensions
 - Need to Map out order in which tests of ratified spec should be developed next & identify resources (e.g. Priv,FDD or F,Priv,D...)
 - TG Reorganization – subgroups?
 - (more discussion if time permits)

Discussion

- New meeting times: 2nd and 4th Thursday, 8am PDT.
- Off cycle meeting next week, Thurs Aug 06, 8am PDT
- Progress (see previous slide)
- CTO: I am working on getting resources. Need compliance tests and SAIL model to go with spec. What does compliance mean? We have a fundamental charter issue. Will there be branding associated with it? Is SAIL adequate by itself? (Krste says "yes") Or do we need something like QEMU [in addition]?
Another criteria: apps will run and Linux will run on priv implementation.
"Profiles" is key: provides the definition of needs for some type(s) of compatibility
CTO is asking for things to be split up into a matrix that specifies <what??>
- SH: QEMU is a red herring.
- CTO: I don't want to overload SAIL <with too many requirements>. It's too hard.
- IMP: Then why do we have SAIL model at all? That's not been the path.
- CTO: I'm giving you permission for additional degrees of freedom
- QC: Use compatibility rather than compliance. (certification?)
- CTO: This is a functional compliance group, not a verification group. You (compliance TG) need to define what *compliance/compatibility/whatever* means.
- Chair: Where do we document?
- CTO: Just write it down, we'll figure out later where to put it. We have same problem with "profiles". Compliance shouldn't worry about profiles. Worry about ISA spec and SAIL model.
- IMP: What date do you want deliverables for action items?
- CTO: I need the list. This is all about getting resources from the board in order to complete "the list". Need to be able to prioritize amongst the various TGs.
- CTO: 4 states of life cycle: Kickoff, freeze, Vote, Complete. Need to define transitions. Eg - What does "done/complete" mean?
- CTO: Use of RISC-V trademark has not been defined yet, esp as it relates to compliance pass. Don't worry about what the name of passing compliance test
We may even want to change the name to something other than compliance
- VT: what is important is passing the tests that match a profile, not passing tests outside the profile, and profiles are a restricted subset of the ISA tests.
- IMP: questions about how RISC-V will use the word compliant. Cannot be subjective by a group of people; must be mechanical.
- CTO: there will always be subjectivity. A review board will grant compliance.
- CTO: <discussion about the profiles TG formation>

Summary:

- the words *compliance/compatible/consistent* will be used for profiles. Profiles TG, TSC and Marketing will decide branding
- **This TG :**
 - is more of an ISA test TG. passing ISA tests will be a component of being profile compliant
 - should identify and communicate what they will ask extensions to test including :, etc. *unit, concurrency, asynch events, complex sequences, variables/parameters*
The TG should also explicitly list what they won't test (that they know).
 - should specify in which milestone of the extension lifecycle (see slide 3 pointers) that tests by the extension TGs be done (can break into parts)
 - is **not** responsible for driving the completion of tests, but should review & agree plans by extension TG for completing their tests
 - should identify tests that must be developed for already ratified extensions. They can ask the appropriate SC for help. This list should be added to the gaps list (see slide 3 pointers)
 - should specify which simulators should be used for which tests. It has the flexibility of picking others besides SAIL. It has no responsibility for specifying what goes into the model or its attributes (like provable or testable with the SAIL simulator). This team will create requirements for SAIL and other simulators they choose so those simulators can be used to run the ISA tests and get golden results.
 - will minimally break the tests into Priv and Unpriv groups but may further subdivide if they wish (e.g. ordering/asynch/concurrency).
 - can approve skipping functional coverage on parameters/edge cases from the extension TGs with some rationale (e.g. not often used variant that would cause an explosion in the number of tests) and guidance on what compliance/test-pass looks like for those members who need it.
 - should advise the TSC and Profiles TG on what they can infer from an implementation passing the tests and what failures mean (e.g. is there a conditional pass with implementor rationale this group approves?).

Decisions & Action Items

Decisions

Meetings moving to 8am PDT Thursday instead of Weds.

Off cycle meeting will be scheduled to catch up on agenda items.

We are not bound to use only SAIL to get golden results

We can put limits on what we will test if corner cases are too difficult

Outstanding Action Items

TG: write prose that defines what compliance means,

TG: what things are we are going to do & what we won't do wrt SAIL

TG: make a stab at what "passing" tests vs. "don't pass" tests means

TG: delineate between priv & unpriv compliant, they are 2 separate things

Imperas: make pull request for updated assertion macro

Imperas: measure RV32I, vector max memory footprint, number of instructions executed

Stuart: write up coverage taxonomy

Chair: get architectural hazard pointer – **done, explicit for branches, but only implicit for register ops**

TG: 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

Previous Action Items / Progress Update

- ET Base ISA coverage draft spec is uploaded - done – needs more eyes to review
- SH will add file regarding coverage - no progress....in progress
- Imperas / Incore: ensure headers, macros, dir structure match newest spec, assertions are not inline – waiting for assertion macro update, Imperas pull request
- ET to coordinate with Riscof to determine pipecleaning exercise - to be reviewed in TG
- ET to communicate with TSC about reorganization comments - waiting TSC feedback
- ET/SH to talk w/ SAIL team about transitioning support to Foundation - in progress
- Configuration Structure TG vs. Riscv-Config: discussions underway – see <https://github.com/riscv/configuration-structure/>

Note: initials are company abbreviations

Test Acceptance Criteria – first cut

Tests must:

- conform to current standard of test spec (macros, labels)
- run in framework
- run with assertions on and not fail any
- generate a valid signature (that can be saved and compared with other dut/sim)
- has a clear configuration - i.e. which ISA extension it can be used with
- have a code, data, and signature memory footprint that is small enough*
- have run time less than X instructions
- improve coverage
- use only standard instructions
- use only files that are part of the defined support files in the repository
- must be commented, both in header and inside test cases

** need to define “small”. ** need to define “X”*

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 few 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
- Maybe: have the ability to measure and report coverage
 - Coverage specification is a separate file
 - Could be a separate app

ISA Compliance Standing Committee and TG ReOrg

Because both Compliance and Formal Modeling are ongoing processes, the ISA Compliance Standing Committee has been formed to direct the current Compliance and Formal Modelling TGs

Proposal: reorganize the 2 TGs into:

- ISA Compliance Standing Committee sc-compliance-isa@lists.riscv.org
- Compliance Tests Task Group tech-compliance-test@lists.riscv.org
Charter Statement: Specifying the requirements for the tests (functional coverage), developing the actual test cases, integrating the tests into the framework. (**Deliverable:** Compliance Test Suite)
- Compliance Generators Task Group tech-compliance-generators@lists.riscv.org
Charter Statement: Develop tools which are configured to generate tests and measure functional coverage. Their tests should meet the requirements specified by the compliance tests task group. (**Deliverable:** Test Tools)
- Compliance Framework Task Group tech-compliance-framework@lists.riscv.org
Charter Statement: Develop a framework (workflow) that takes a description of an implementation to select/configure tests, runs them, and compares them to the golden model. Manages the output and logging of the tests being run. (Possibly combine for now with generator group? (**Deliverable:** Framework v2)
- Golden Model Task Group tech-golden-model@lists.riscv.org
Charter Statement: This group will maintain a Formal Specification for the RISC-V ISA. This is a specification of the ISA in a formal language, for precision, unambiguity, consistency and completeness. The spec is readable and understandable as a canonical reference by practising CPU architects and compiler writers. It is executable and machine-manipulable by tools for establishing correctness and transformations in both compilers and CPU designs.

(.....discuss.....)

Pull/Issue Status

Issue#	Date	submitter	title	status	
#04	3-Jul-18	kasanovic	Section 2.3 Target Environment	Fixed in RISCOF	
#22	24-Nov-18	brouhaha	I-MISALIGN_LDST-01 assumes misaligned data access will trap		
#40	4-Feb-19	debs-sifive	Usage of tohost/fromhost should be removed		
#45	12-Feb-19	debs-sifive	Reorganization of test suites for code maintainability		
#63	13-Aug-19	jeremybennett	Global linker script is not appropriate		
#78	26-Jan-20	bobbl	RV_COMPLIANCE_HALT must contain SWSIG		
#90	11-Feb-20	towoe	Report target execution error		
#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
#106	22-Apr-20	jeremybennett	Use of pseudo instructions in compliance tests	under discussion	
#107	22-Apr-20	jeremybennett	Clang/LLVM doesn't support all CSRs used in compliance test suite	under discussion	
#108	22-Apr-20	bluewww	RI5CY's `compliance_io.h` fails to compile with clang	under discussion	
#109	06-May-20	Olofk	Swerv fails because parallel make	under discussion	
pull#113	30-may-20	imphil	Consistently use UNIX line endings	under discussion	
#115	06-jun-20	adchd	How to support on-board execution?	under discussion	
#116	06-jun-20	simon5656	loss of 64bit test infrastucture	under discussion	
#119	17-jun-20	allenjbaum	Missing RV32i/RV64i test: Fence	Test needs to be written	
#125	15-jul-20	ShashankVM	Request to stop hosting closed source code on riscv repo	under discussion	