Compliance Task Group Call – Minutes

Thur, 17Sep2020 8am Pacific →Daylight← Time

See slide 4 for agenda

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

Adminstrative 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_gHl5b2ApkcHVtpZyl4s_A7sgSrNN 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/1DemKMAD3D0Ka1MeESRoVCJipSrwiUlEs (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)
  - https://github.com/riscv/riscv-config/
- JIRA: https://jira.riscv.org/projects/CSC/issues/CSC-1?filter=allopenissues

Meeting Agenda

0. **The profile TG is looking for a co-chair from the Compliiance TG. Please email me if you have any interest or want to nominate someone**
1. Updates, Status, Progress
    1. RFQ Dashboard
        1. Coverage: Rules are now available here: https://github.com/incoresemi/riscv-compliance/blob/dev/coverage/coverpoint.yaml
        2. Criteria for approving merge requests (e.g. coverage, Sail model integration, size, time to run)(see slide 9)
2. Discussion:
    1. Compliance FAQ - any more comments on
        1. Changing references of "Compliance Tests" to Architecture Tests" in Charter, Test Format spec, lists.riscv.org files, riscv home pages
        2. Wording of concise definition of what Architectural Testing means (page 8)
    2. Next steps and Ongoing maintenance
        1. Migration to Framework v.2. video: https://youtu.be/VIW1or1Oubo, slides: https://lists.riscv.org/g/tech-compliance/files/Presentations/TestFormatSpec.pdf
            1. Review Pipecleaner tests:What do we need to do to exercise capabilities for Priv Mode tests
            2. What steps do we need to complete to cut over to V.2 (see slide 10)
            3. (e.g. Sail model updates, pipecleaning, N people have run it, testing all the "fixed in riscof" issues
        2. Develop SAIL maintenance plan
        3. Identify Tool providers, e.g. coverage model, test generation for new features/extensions
        4. Flesh out test development order & identify resources (e.g. Priv,FDD or F,Priv,D..., JIRA CSC-3,5
        5. Provide a reference RTL test fixture (as opposed to SW functional model). See. JIRA CSC-6
    3. Specific Compliance Policy/Process Gaps:
        1. Certifying passing architecture tests: what needs to be in the report? Where does report get sent? (e.g. vendorID/archID)
        2. Can we certify actual HW if only its core RTL has passed architecture tests?
        3. How do we enable configurable & licensed core IP

Discussion

1. RFQ Dashboard Review

Incore sharing the tracking dashboards.
202 tests, 3ea handwrtitten for RV64,RV32, ~31,000 coverpoints; 100% covered.
Coverpoints here→https://github.com/incoresemi/riscv-compliance/tree/dev/coverage
Tests here→ https://github.com/incoresemi/riscv-compliance/tree/dev/riscv-test-suite
New coverpoint functions can be defined with python
SAIL and Spike signatures match

1.2 Acceptance Criteria discussion - assertions

Chair – Not all signature stores can have assertions: if store is in a loop, or in an ISR, at a branch convergence point, or stores, WARL field. So assertions cannot be universal on all signature elements
QC - Why are assertions not in the test?
Chair - They are. These are signature assertions, not RTL assertions.
QC - How can we check signatures?
Chair - Run tests once with assertions off (using the reference model). That golden signature is used back-annotate the assert macro.
Assertions are primarily for debugging.
IMP - Assertions show tests are correct by construction. Can't use signature for this.
Discussion about assertions, signatures, models.
Disagreement about purpose/intent of assertions. AR: Discussion will move to email.

Examination of coverage report. In the github repo (see above).
Examination of coverpoint.yaml
QC - How are tests constructed?
Incore - See cgf.yaml. Python script uses .yaml and generate tests.

1.2 Acceptance Criteria discussion - validation

IMP - Need to add SAIL signature must be validated
(it was done, but Clarification added to Test Acceptance Criteria)

1.2 Acceptance Criteria discussion – pseudo-instructions

Incore - Need to use LI and LA. Are there issues with using these pseudo instructions?
IMP - We weren't going to use these, right?
Chair - These are needed to handle different sizes. These are clear what they do.
Incore2 - We will not be running the exact same binary on both SAIL and on DUT.
Chair - Whatever is compiled for DUT should run on SAIL/
IMP - Not true, because preamble may not (cannot ???) be the same.
Incore - Issue#106 brought up the issue with 'la'
Pseudo-instructions are described in asm manual (no longer in priv spec). See….
https://github.com/riscv/riscv-asm-manual/blob/master/riscv-asm.md
Incore2 - This only works for gcc. Supports %pcrel, etc..e.g

```
sw x23, %pcrel(signature_x4) compiles to:
        auipc x23,  %pcrel_hi(signature_x4_0)
        addi   x23, x23,    %lo(signature_x4_0)
        sw     x23,56(x4)
```

Chair - This is primarily rv32 specific; 64 bit address generation is different
IMP - Do current tests use la and li?
Incore - Yes..
IMP - Then stick with it.
QC - Then do we re-word the criteria to only allow la and li?
Chair - People write with pseudo-ops, not with specific instructions.
IMP - We shouldn't be using (other) pseudo-ops
Chair - We should certainly use standard pseudo-ops that have fixed encodings. More readable.
lots of discussion about how and when different code would be generated depending on toolchain and model
AR: Cover Pseudo -ops over email,

Decisions & Action Items

Decisions

- We can use pseudo-ops that are guaranteed to generate the same ops in all circumstances

- We don't need assertions for every signature store: specifically can't be used whenever a result is implementation dependent (WARL fields), or when the signature store will be executed more than once, or when a signature store is at a branch convergence point

- tests need to run in Sail with assertions enabled to be accepted

Outstanding Action Items- New

[Chair]: start discussion on the use of LI, LA pseudo-insts in tests

[Chair]: start discussion on assertion generation in tests

[everybody]: comment on base ISA cover points:

https://github.com/incoresemi/riscv-compliance/tree/dev/coverage

Old

Imperas: make pull request for updated assertion macro

Stuart: 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

Previous Action Items / Progress Update

- 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
- Chair coordinate with Riscof to determine pipecleaning exercise - to be reviewed in TG
- Chair to communicate with TSC about reorganization comments - waiting TSC feedback
- Configuration Structure TG vs. Riscv-Config: - discussions underway – see https://github.com/riscv/configuration-structure/ and new profile group

Note: initials are company abbreviations

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 – first cut

Tests must:

- conform to current standard of test spec (macros, labels)
- run in framework
- run in SAIL with assertions enabled and not fail any tests
- generate a valid signature using SAIL (that can be saved and compared with another 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*
- improve coverage
- use only standard instructions ←are LA ,LI allowed?
- 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", "X" ← will vary by extension, base ISA expected to be <8K. Tests of JAL max*

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

Pull/Issue Status

Issue#	Date	submitter	title	status	comments
#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	
#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 has been written	Close when test is merged
#125	15-jul-20	ShashankVM	Request to stop hosting closed source code on riscv repo	under discussion	
pull#128	29-jul-20	nmeum	grift: update for new directory structure		Who can review this?
pull#129	31-jul-20	nmeum	sail-riscv-ocaml: Disable RVC extension on all devices not using it		Who can review this?
#132	15-aug-20	davidmlw	Why not just use mepc for mret?	answered	Should be resolved
#135	04-sep-20	MikeOpenHWGroup	Request for a Tag on this Repo	assigned	

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		
CSC-1	20/Aug/20	Ken Dockser	Come up with names for the tests suites that we are creating		1st 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