Compliance Task Group Call – Agenda

Thur, 27Aug2020 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 <u>and</u> 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 ???? 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/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/

Meeting Agenda

- Updates, Status, Progress
  - Policy/process updates (specifically Done policy)
    - https://drive.google.com/drive/folders/1EKGGxVN3s9wZkOcQLxwT_u_daB028wQx
    - https://docs.google.com/spreadsheets/d/1UL6F6ahNwFO69fecLJtnpZatx_PkS-Gdcvb8DdWhWUk/edit#gid=0
  - Adding Asia meeting times **– poll sent out**

- Discussion:
  - Compliance FAQ - any more comments?
    - write prose that defines what compliance means,
    - what things are we are going to do & what we won't do wrt SAIL
    - make a stab at what "passing" tests vs. "don't pass" tests means
    - delineate between priv & unpriv compliant, they are 2 separate things
  - 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
  - Coverage Spreadsheet
    - critique, review (See: Coverage Rules.xlsx in https://lists.riscv.org/g/tech-compliance/files/Review%20Documents)
  - 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 riscof" issues
  - 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 id time permits)

Discussion

Progress:

meeting calendars. How to subscribe and use new calendars docs here:
https://docs.google.com/document/d/18AtJDhPoyKWUpJddZDNDc4U6dg227hiSAhTGzPzxqUM/edit?usp=sharing
See Stephano's video on how to switch accounts and subscribe to calendars here:
https://youtu.be/zAFHXB5e4aU

Criteria for Adding Tests
- Chair: one criteria is size of code and data. Another is documentation, yet another is coverage. Current Pass/fail is stored signature match; in the future it will be a match of golden model results.
- Imp: Tests are typically 1k-1.5k max, but signatures could be 100s of MBs (vectors). The technique of hashing vectors and storing that as a signature has not been implemented yet.
- QC: what are the coverage goals? Specific instructions have specific functions. Marching ones, etc: do they make sense for all instructions? What do we want to do term term? Long term? We want to catch bugs/issues. Want to keep bad designs from being released.

Coverage Spreadsheet
- Chair: coverage spreadsheet should cover the interesting cases. There is a v2 of that spreadsheet in a separate tab here:
https://docs.google.com/spreadsheets/d/1POxWE2lQ3SXeY2Un8eNZteZZsdzLUnNNjXIANIUyFn0/edit#gid=0
  Please add missing cases , or restrict overly-broad cases on an op by op basis.
  Near term: corner cases are likely overspec'd, but some clarifications added to v2 (specifically immediate length and signed vs unsigned).
  Longer term: limit which of the interesting values are tested on an op by op basis
- QC: cases may not be amenable to a spreadsheet
- Imp: We took spreadsheet and then ran them against compliance testsuite. Coverage measure not good (Either existing tests are inadequate or blindly following spreadsheet creates impossible conditions that lower coverage metric)

- SH: Isn't the purpose to show the intent of the RISCV arch? Example signed v. unsigned multiply. We don't want to enter into verification space. Can't cover everything.
- QC : The goal is software interoperability. The better set of suites that are out there test for interesting boundary conditions. Not a bare minimum.
- Sail: question: suppose you took a branch coverage of the spec: what would you want beyond that? [branch coverage is which lines of code are executed in the formal model]
- Chair, Imp : we thought about that early on, but couldn't figure out how to filter based on which extensions - and features - are implemented. It's a good way to find coverage holes, though.
- Sail: SAIL model is structured into various feature/extension files. It could filter coverage based each file Can we define how we're going to measure coverage? Do we have to make changes to SAIL code? <<no changes needed for spreadsheet based coverage, but are for model code coverage>>
< walking through existing coverage spreadsheet >
- QC : is there a cover property file that is derived from spreadsheet?
- Imp : took spreadsheet and implemented in OVPSim. Can derive coverage. Go to github. Can see YAML coverage results.
 (github.com/riscv/riscv-compliance/blocb/master/riscv-test-suite/rv32i.extended.coverage.txt).
  Coverage from existing tests is poor.
- Chair : spreadsheet does not list result values, only src
- Sail : < walking through SAIL code coverage data here:
  https://alasdair.github.io/riscv_coverage.htmll
- <<discussion of how to make code coverage configurable by feature/extension

Decisions & Action Items

Decisions

Use JIRA for adding Action items
Jira#1: add Jira line to home screen

Outstanding Action Items- New

Ken take a stab at defining Risc-V compliance specifically, replacing "compliance if necessary.

<u>Chair</u>: send poll out for Asian friendly time, see if there is any response. Tentatively planning for meeting twice a month. <done, but only replies are from existing attendees>

<u>Chair</u>: Add correct link to updated coverage spreadsheet < done>

<u>Everybody</u>: comment, correct, and criticize coverage spreadsheet

Chair: add Jira issues for action items (#1:Stephano to add Jira link to home screen)

Old

<u>Imperas</u>: make pull request for updated assertion macro

<u>Imperas</u>: measure RV32I, vector max memory footprint, number of instructions executed

<u>Stuart</u>: write up coverage taxonomy

<u>Everybody</u>: 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

Ken take a stab at defining Risc-V compliance specifically, replacing "compliance if necessary.

Previous Action Items / Progress Update

- <u>ET</u> Base ISA coverage draft spec is uploaded - done – still needs more eyes to review

- <u>SH</u> will add file regarding coverage   - no progress….in progress

- <u>Imperas / Incore:</u> ensure headers, macros, dir structure match newest spec, assertions are not inline – waiting for assertion macro update, Imperas pull request

- <u>ET</u> to coordinate with Riscof to determine pipecleaning exercise - to be reviewed in TG

- <u>ET</u> to communicate with TSC about reorganization comments   - waiting TSC feedback

- <u>ET/SH</u> to talk w/ SAIL team about transitioning support – in/no 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 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 <X* seconds on simulator running on Intel Corei7 or equiv
- 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", "X" ← will vary by extension

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		
#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	Fixed in RISCOF	
#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	