Compliance Task Group Call – Minutes

Weds, 08 Apr 2020 8am Pacific →Daylight← Time

See slide 6 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 (M,A,F,D,Q,L,C,B,J,T,P,V,N)
  - 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://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
- Git repositories
  - https://github.com/riscv/riscv-compliance/
  - https://github.com/rsnikhil/Experimental_RISCV_Feature_Model
  - https://github.com/rsnikhil/Forvis_RISCV-ISA-Spec
  - https://gitlab.com/incoresemi/riscof (Shakti framework)

Meeting Agenda

- Review of issues (and getting owners to fix them)
  - Many issues are deferred to V.2
  - Issues only 4 that can't be dealt with more easily than v.2 versions
    #84 – fix is coming?
    # 4 – has been incorporated within RISCOF
    #27 – This has been fixed?
    #72 – Unclear this is can be easily fixed in a general manner until v.2
- New co-chair: Bill McSpadden, (Seagate)
- Progress:
  - Riscof: minor updates to riscv-config
    migrated riscof-plugins/riscof to use the latest version of riscv-config
    RVMODEL_BOOT decoupled from RVTEST_CODE_BEGIN
    RISCOF repository updated with TestFormat Spec
- Next steps (and their ordering):
  - More tests?
    - requires funding or volunteers
    - RFQ requires coverage metric
  - Transitioning to v.2?
- Other? – Covering for YAML-selected/configured tests

Discussion

1. Issue#72: How to handle MTVEC

- How to handle mtvec:
  Test suite for MTVEC must assume we can load trap handler;
  - where it gets put.
  - Need general solution, not just for mtvec
  - Use YAML file? Linker script that uses YAML generated macros
- Allen/Neel's plan is to have a standard trap handler. Tests swaps initial trap handler w/standard handler pointer, & restore on HALT
  - Misalign would then require no further changes

2. Issue#84: Sinple brute force fix is in place (pull #104)
   - But SB, SH need rewrite for coverage
   - Tests were auto-generated; say what needs fixing & they'll be fixed;
   - Get someone with SV knowledge to check coverage
   - SW test is a much better example; it was rewritten

3. issue#27 Contact submitter jlucnagel and see if can be closed

4. issue#4 can be closed now

5. Can we use one of the tests that use YAML parameters to pipeclean? Mtvec? Misalign?

6. Vector tests will use hash instead of individual values - the signatures are enormous
   This approach is compatible with both v.1 and v.2 framework

7. Coverage: Have we run coverage with new test versions?
   - How can we measure coverage for YAML selected/configured tests?
   - Need coverage for each test
     does new framework support this?
   - How to do this for different YAML? Use grift ?
   - No – use system Verilog, need a tool that refines its coverage points based on YAML (future topic)

NOTE: coverage applies to riscv test suites, and not to implementations - but test selection is model YAML specific.
- This is not used to measure the quality of an implementation, but to the riscv compliance test suite itself
- This means that we can measure coverage on any standard model (e.g. formal model or OVPSim), so doesn't need to be tied to anyone's implementation.

Could we crowdsource coverage models?
Have implementers look at coverage in google DV test suite (and their own) see if cover points are adequate, submit their own

Decisions & Action Items

Decisions

1. Start pipecleaning riscof flow with interesting tests: e.g.misalign
    1. This is related to issue #72
2. start looking into how we can crowdsource coverage modeL
3. --
4. Close Issues #4, #27, #84 (after verifying it works).
5. Add coverage to next meeting agenda

Action Items

1. Allen/Neel: start detailing steps needed to pipeclean misalign test, and any test that requires trap handling
2. Simon: get pointers to Google DV (?) cover points to start
3. nothing needed
4. Allen: will attempt to contact jlucnagel to close #27
5. Simon: will describe (& give a demonstration)? in 2 weeks (combine with #2?)

Action Items from last meeting

Anybody:Rv64i tests need to be re-written! – (deferred for funding)

Neel: eliminate or document OS dependencies (Debian vs. Ubuntu vs. Centos (vs. MacOS?) (ongoing)

Allen: contact submitters of fixed issues to close – (one bug remaining to be fixed, one likely is fixed, the remaining 8 will be deferred to v.2 – to hard to fix in v.1)

Allen: put coverage draft into https://lists.riscv.org/g/tech-compliance/files/Review Documents (done)

Riscof update details

RISCOF version 1.13.1 has just been released which includes the following changes/additions:

RISCOF: https://gitlab.com/incoresemi/riscof

RISCOF-PLUGINS: https://gitlab.com/incoresemi/riscof-plugins

Updates docs: https://riscof.readthedocs.io/en/latest/

- Docs have been significantly updated with more details and a better quickstart
- PYTHONPATH no longer needs to be set for the plugins. All the information is picked up from the config.ini itself.
- RISCOF now checks that both signatures are not empty before comparing them
- RISCOF has now shifted to use all the new tests (with better coverage) from github
- RISCOF now allows people to run custom suites using the --suite. This mean for developers who want to write new tests or for those who do not wish to run the entire suite but only a sub-set.
- The plugins now have a makeutility available which can be used to quickly generate makefile. The spike_parallel and riscvOVPsim plugins have been updated to use this utility. The user can also now choose what make command to use (e.g. pmake, make -j, etc). Default is make which should work on all distros
- Print the help command when no argument are specified
- Print version on execution
- The --setup command now generates all collaterals (including a python plugin template) required to execute RISCOF
- All the hard-coded paths in the spike_simple plugin have been removed and should now work as is on any system with changes only happening in config.ini

Pull/Issue Status

Issue#	date	submitter	title	status	
#04	3-Jul-18	kasanovic	Section 2.3 Target Environment	??	will be fixed in V.2
#30	21-Jan-19	RolfyYu	The golden results of rv32ui and rv64ui should be different	bug	most rv64 tests need rewriting
#08	17-Oct-18	AnttiLukats	RV32I/I-IO.S bad file name		change file name…
#37	31-Jan-19	astimonov	rv32imc C.SWSP test5 writes a word outside the binary		will be fixed in V.2
#40	4-Feb-19	debs-sifive	Usage of tohost/fromhost should be removed		change makefile.include? Fixed in V.2?
#42	5-Feb-19	as-sc	Misaligned fetch bit must be excluded for RVC		reset code chg if mdeleg is RO?
#78	26-Jan-20	bobbl	RV_COMPLIANCE_HALT must contain SWSIG	bug - needs fixing	change macros as described
#84	4-Feb-20	towoe	I-SW-01 corrupts .text region		change test
#95	03-Mar-20	dsw	doc directory does not build		?? title & document history mismatch
#96	05-Mar-20	dsw	how to run all RV64 tests, e.g./rv32ui/rv64ui, through spike ?		Documentation fixup?
#90	11-Feb-20	towoe	Report target execution error		will be fixed in V.2
#03	3-Jul-18	kasanovic	2.4 Processor configuration clarification		
~~#09~~	~~22-Oct-18~~	~~neelgala~~	~~Setting SATP and PMP should be optional (closed)~~		
#16	7-Nov-18	neelgala	I-MISALIGN_JMP-01.S assumes compressed can be turned off		
#22	24-Nov-18	brouhaha	I-MISALIGN_LDST-01 assumes misaligned data access will trap	bug - not fixable in v.1	will be fixed in V.2
#31	25-Jan-19	debs-sifive	I-MISALIGN_JMP-01.S outdated `mbadaddr` in trap handler?		
#63	13-Aug-19	jeremybennett	Global linker script is not appropriate bug		
#72	26-Oct-19	vogelpi	Allow for non-word aligned `mtvec`		
~~#11~~	~~26-Oct-18~~	~~neelgala~~	~~illegal.S in rv32mi : S-mode int not always supported (closed)~~		will be fixed in V.2
#27	21-Dec-18	jlucnagel	Macros are checking side-effects		
#28	21-Dec-18	bluewww	I-SB-01 test war hazard (address register)	fixed?	Close?
#32	25-Jan-19	debs-sifive	breakpoint.s undesired behavior when trigger does not exist?		
#33	28-Jan-19	debs-sifive	rv32si/ma_fetch.S has a diff. sign depending on RVC support		
#67	25-Sep-19	rongcuid	RV32I Immediate Operands error		commit cae8567?
#97	05-Mar-20	dsw	rv64i/Makefrag, variable rv64i_sc_tests contains SRAIW twice	low priority bug	
#45	12-Feb-19	debs-sifive	Reorganization of test suites for code maintainability	low priority bug	will be fixed in V.2
~~#38~~	~~31-Jan-19~~	~~santhoshvlsi~~	~~I-LB-01 test - Load the data into X0 GPR register (closed~~	~~not a bug?~~	~~closed~~

Next Meeting Agenda (in order of Priority)

- Next steps
  - More tests?
    - Better coverage of existing tests vs. new tests (primarily priv level)
  - Coverage metrics?
  - Transitioning to v.2?
  - Funding?
  - Get more repo maintainers
  - Other?

Backup from previous discussions

Draft Test Coverage Proposal (unpriv)

Classes of things we want to test for

- Decode
  - Immediate – test all bits in either polarity will affect output
  - Register specifiers – test that changing any bit will affect output, ensure all regs are tested
  - Variations – test values of opcodes suffixes that have any string after a "." in its opcode
- Register combinations
  - Destructive (dest = either src) and non-destructive
  - Non-updating (i.e., targeting X0), or non-supplying (X0 as an input)
  - All registers (or immediate bit) should be used per instruction *category*
- Special and exception cases
  - Explicitly defined (e.g. shifts>=XLEN & RD=X0
  - Implicitly defined – corner cases
    - Maximal and minimal inputs, or creating maximal outputs
    - Inputs that special case outputs (mostly FP cases, also. shiftamt>=XLEN)
    - Outputs crossing value boundary (e.g. address cross word/page/superpage/VA boundary, FP crossing exponent boundary)

proposed coverage & categories

Arith[I],	*W1/0,crys*
Logical[I],	*W1/0*
Shift[I],	*W1/0/msk,+*
Auipc,Lui,	
Ld,St,	*W1/0, bndXing*
Br,	*W1/0, bndXing*
Jmp ,	*W1/0, bndXing*
Ebreak/ Ecall	

W1/0= walking 1/0

BndXing=: boundary crossing

This works for 32i base ops – what do we need to add for priv modes? Mem model? Sequential Dependencies? Other extensions?

Need a review of existing (non-RISC-V) compliance specs

Draft Test Coverage Proposal (more, incl priv)

- Forwarding: result of one op can be used as the source of the very next instruction
  - Need at least a case within and between instruction classes
- Changing non-reg state used by an op, immediately followed by op that uses it, e.g. :
  - changing the rounding mode for an FP op
  - writing into the instruction stream, followed by a fencei affecting the next ifetch
  - changing a page table entry or PMP entry, or SATP affecting the next access
  - changing xEPC or xSTATUS followed by xRET
  - changing MISA followed by any op enabled or disabled by it
  - changing xTVEC, xDELEG, xIE followed by a trap
  - write once behavior (PMP-lock)
- Ops that change non-reg status, immediately followed by op that tests it, e.g.:
  - FP status after an FP op
  - xSTATUS.FS,XS fields after FP, Vector or other coprocessor op
  - xCAUSE, xEPC, xTVAL, xPP after an interrupt or exception

RISCV-CONFIG

- Examples & definitions
  - https://github.com/riscv/riscv-config/tree/master/examples
  - https://github.com/riscv/riscv-config/tree/master/riscv_config/schemas
  - https://github.com/riscv/riscv-compliance/tree/master/riscv-ovpsim/config-yaml/examples
- Validator
  - https://github.com/riscv/riscv-config/blob/master/riscv_config/checker.py
- Example integration of converter (OVPsim)
  - https://github.com/riscv/riscv-compliance/tree/master/riscv-ovpsim/config-yaml
- WARL, YAML
  - https://riscv-config.readthedocs.io/en/latest/

RISCV-CONFIG WARL Syntax

WARL: {optional items in curly braces}

- dependency_fields: [list] – use this when legal/illegal values depend on other fields (in list)
- legal: [<warl-string>{,<warl-string>*}]
- wr_illegal: [<warl-string>{,<warl-string>*}] -> update_mode

where <warl-string> is either ”&” separated list of rangehi:rangelo lists

{*[dependency_value] ->*} field-name1[bit#hi:bit#lo] in [legal-range-list]

*{ & field-name2[bit#hi:bit#lo] in [legal-range] }**

or ”&” separated list of bitmasks

{*[dependency_value] ->*} field-name1[bit#hi:bit#lo] bitmask [mask, fixval]

{ & field-name2[bit#hi:bit#lo] bitmask [mask, fixval] }*

(can’t mix ranges and bitmasks)

RISCV-CONFIG WARL Example1

```
#         When base of mtvec depends on the mode field.
WARL:
  dependency_fields: [mtvec::mode]
  legal:
    - "[0] -> base[29:0] in [0x20000000, 0x20004000]”                  # can take only 2 fixed values when mode==0.
    - "[1] -> base[29:6] in [0x00000:0xF00000] & base[5:0] in [0x00]" # 256 byte aligned when mode==1
  wr_illegal:
    - "[0] -> unchanged”
    - "[1] wr_val in [0x2000000:0x4000000] -> 0x2000000”           # predefined value if write value is in this range
    - "[1] wr_val in [0x4000001:0x3FFFFFF] -> unchanged”           # predefined value  if write value is this range

#         When base of mtvec depends on the mode field. Using bitmask instead of range
WARL:
  dependency_fields: [mtvec::mode]
  legal:
    - "[0] -> base[29:0] in          [0x20000000, 0x20004000]"     # can take only 2 fixed values when mode==0.
    - "[1] -> base[29:0] bitmask [0x3FFFFFC0,  0x00000000]”        # 256 byte aligned when mode==1
  wr_illegal:
    - "[0] -> unchanged”                                           # no illegal for bitmask defined legal strings.

”
```

RISCV-CONFIG WARL Example2

```
# no dependencies. Mode field of mtvec can take only 2 legal values using range-descriptor
WARL:
  dependency_fields:
  legal:
    - "mode[1:0] in [0x0:0x1]                          # Range of 0 to 1 (inclusive)"
  wr_illegal:                                          # default to 0 if not a legal value
    - "0x00"

# no dependencies. using single-value-descriptors
WARL:
  dependency_fields:
  legal:
    - "mode[1:0] in [0x0,0x1]                          # also Range of 0 to 1 (inclusive)"
  wr_illegal:
    - "0x00”
```

- "[1] wr_val in [0x2000000:0x4000000] -> 0x2000000 & wr_val in [0x4000001:0x3FFFFFF] -> **unchanged**