Compliance Task Group Call – Agenda

Weds, Feb26 2019 8am Pacific →Standard← Time

See slides 9,10 for discussions and action items

Charter

The Compliance Task Group will

- Develop a framework for RISC-V tests, 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 method 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 – Stuart Hoad stuart.hoad@microchip.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)

Attendees

- Allen Baum (Esperanto)

Meeting Agenda

1. TestFormat Spec Poll results – passed, but with little participation
    1. 3 other pulls still outstanding
2. Progress on last meeting Action Items
3. Reviewing and Closing issues (slide 7), finding volunteers to close them
4. Coverage discussion (slides 12-13)
5. Looking towards the future
    - Getting more repository maintainers
    - Funding to get more tests/tools for tests, better coverage metrics
    - Transitioning to a standing committee – what is needed?
    - Research: using formal models to generate tests?

Action Items from last meeting

Allen: put test spec .pdf into compliance TG lists.riscv.org site **Done**

Allen: send poll for vote to accept test format spec v1.2.5 **Done**

Neel,Pawan - seen next slide

- document how to use riscof as separate pieces – available on branch
- investigate if possible to release as a docker image – others should do?
- look into a generator script instead of manual installation steps
  – available on branch
- remove hard-coded directory references from scripts
  – available on branch
- establish/document a std. directory structure for the riscof environment
  – needs more information

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	bug - needs fixing	reset code chg if mdeleg is RO?
#78	26-Jan-20	bobbl	RV_COMPLIANCE_HALT must contain SWSIG		change macros as described
#84	4-Feb-20	towoe	I-SW-01 corrupts .text region		change test
#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?
#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~~
#47	16-Feb-19	aprnath	Machine mode atomic extension tests?	not a bug?	should be closed

Discussion

1. Poll, passed with little participation
   Comment – new version of the test spec isn't crucial – getting more tests are.
2. Neel reviewed updated riscof framework (slide 7)
   1. improvements in setup and documentation
   2. New features: able to do custom test suite selection
3. Looked at other pull requests – all now merged
4. Looked at issue list
   1. Decided that many required test configuration, not possible in V.1, so will relabel them (and possibly close)
   2. Others require major test reorganization and re-writing, may need to wait until v.2
   3. Close #38 as not a bug

Decisions & Action Items

Decisions

Accept test spec v1.2.5

- new tests should match this format
- conflicts in changelog need fixing

Accept/merge pull requests 9, 83, 92 (done)

Clean up issues

Get serious about foundation support for test writing

Action Items

Anybody:Rv64i tests need to be re-written!

Allen: Need funding for tests!

Allen: contact Issue 84 submitter to reopen it

Allen: Re-order issue list into categories:

- not a bug, should be closed
- bug, unfixable in v.1, mark TBD?
- bug - should be fixed
- bug – has been fixed, should be closed

Allen: contact submitters of fixed issues to close

Next Meeting Agenda (in order of Priority)

- Review of issues (and getting owners to fix them)
- Next steps (and their ordering):
  - More tests?
    - Better coverage of existing tests vs. new tests (primarily priv level)
  - Coverage metrics?
  - Transitioning to v.2?
  - Funding?
  - 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**