Compliance Task Group Call – Minutes

Thu, Jan09 2019 8am Pacific →Standard← Time

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
  - Location is https://zoom.us/j/6213886723
- 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)

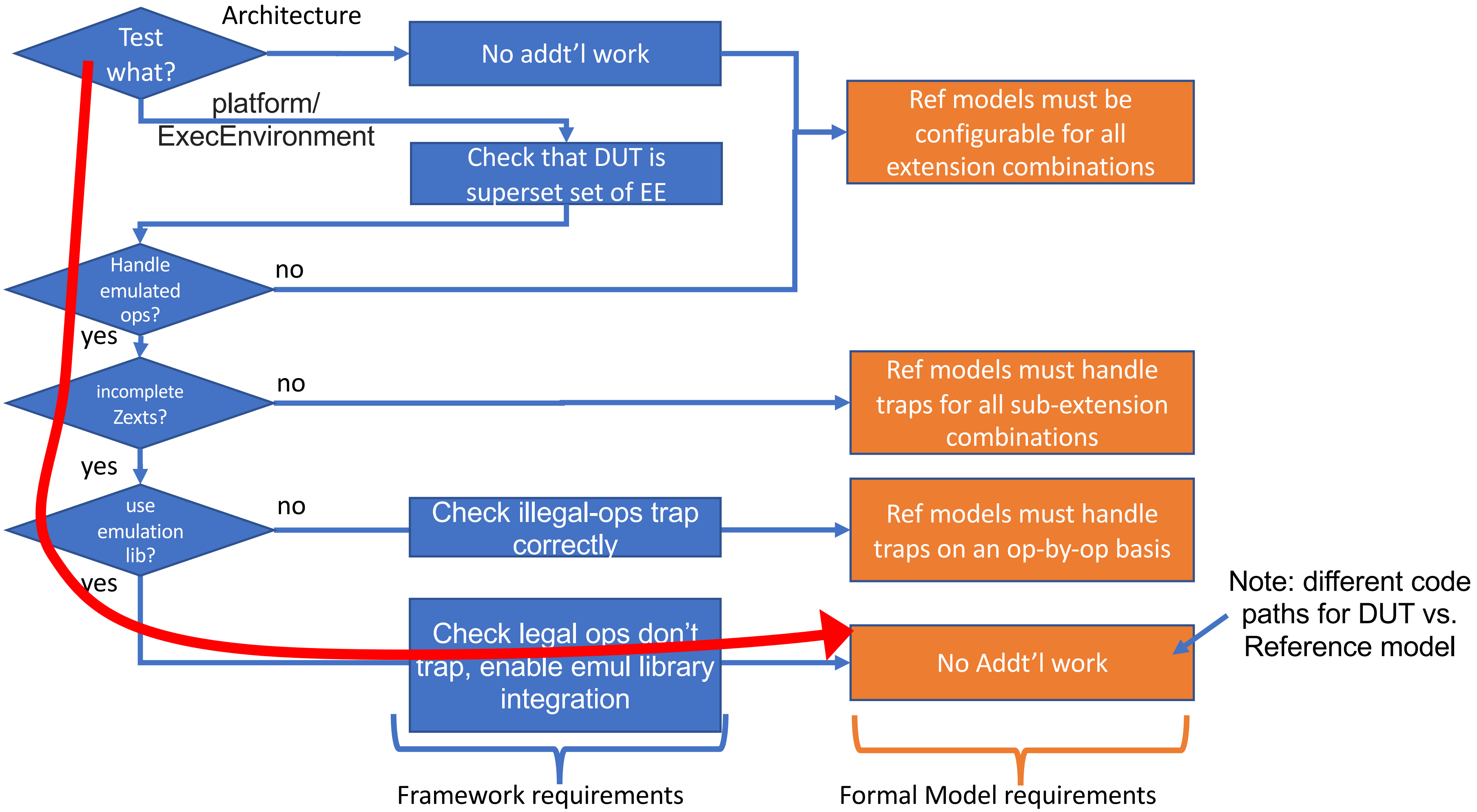
Attendees

Allen Baum	(Esperanto)
Premysl Vaclavik	(Codasip)
S Pawan Kumar	(IIT Madras)
Lee Moore	(Imperas)
Bill Mcspadden	(Seagate)
Ivan Alexandrov	(Syntacore)
Maxim Koksharov	(Syntacore)
Sergey Vasiliev	(Syntacore)
Mundkur	
Robin Norton-Wright	(Syzexion)

Meeting Agenda (in order of Priority)

1. Pull requests (if time permits)
   - 65: Test Format spec: first half reviewed
2. RISCOF review:
3. Reviewing and Closing issues (currently 23 open)
4. Emulated ops – the way forward
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?

Emulated Ops



Test what?
Architecture
No addt'l work
platform/ ExecEnvironment
Check that DUT is superset set of EE
Ref models must be configurable for all extension combinations
Handle emulated ops?
no
yes
incomplete Zexts?
no
Ref models must handle traps for all sub-extension combinations
yes
use emulation lib?
no
Check illegal-ops trap correctly
Ref models must handle traps on an op-by-op basis
yes
Check legal ops don't trap, enable emul library integration
No Addt'l work
Framework requirements
Formal Model requirements

Note: different code paths for DUT vs. Reference model

Issue Status

Issue#	date	submitter	title	status
#37	Jan 31	astimonov	rv32imc C.SWSP test5 writes a word outside the binary	bug - needs fixing
#30	Jan 21	RolfyYu	The golden results of rv32ui and rv64ui should be differ	bug - needs fixing
#08	Oct 17, 2018	AnttiLukats	RV32I/I-IO.S bad file name	bug - needs fixing
#67	Sep 25	rongcuid	RV32I Immediate Operands error	fixed in commit cae8567?
#11	Oct 26, 2018	neelgala	illegal.S in rv32mi generates supervisor interrupt - might not be supported on all implementations	fixed, close
#32	Jan 25	debs-sifive	breakpoint.s undesired behavior if trigger doesn't exist?	fixed?
#33	Jan 28	debs-sifive	rv32si/ma_fetch.S produces a different sig if RVC supported	fixed?
#27	Dec 21, 2018	jlucnagel	Macros are checking side-effects	fixed?
#28	Dec 21, 2018	bluewww	I-SB-01 test war hazard (address register)	fixed?
#38	Jan 31	santhoshvlsi	I-LB-01 test - Load the data into X0 GPR register	not a bug?
#42	Feb 05	as-sc	Misaligned fetch bit must be excluded for RVC	not a bug?
#70	Oct 08	towoe	Target error exit condition	not a bug?
#63	Aug 13	jeremybennett	Global linker script is not appropriate bug	TBD - fixed in riscof?
#47	Feb 16	aprnath	Machine mode atomic extension tests?	will be fixed in v.2, should currently work
#03	Jul 03, 2018	kasanovic	2.4 Processor configuration clarification	will be fixed in V.2
#04	Jul 03, 2018	kasanovic	Section 2.3 Target Environment	will be fixed in V.2
#09	Oct 22, 2018	neelgala	Setting SATP and PMP should be optional	will be fixed in V.2
#16	Nov 07, 2018	neelgala	I-MISALIGN_JMP-01.S assumes C-ext can be turned off	will be fixed in V.2
#22	Nov 24, 2018	brouhaha	I-MISALIGN_LDST-01 assumes misaligned data access traps	will be fixed in V.2
#31	Jan 25	debs-sifive	I-MISALIGN_JMP-01.S outdated `mbadaddr use ` in handler?	will be fixed in V.2
#40	Feb 04	debs-sifive	Usage of tohost/fromhost should be removed	will be fixed in V.2
#45	Feb 12	debs-sifive	Reorganization of test suites for code maintainability	will be fixed in V.2
#72	Oct 25	vogelpi	Allow for non-word aligned `mtvec`	will be fixed in V.2

Discussion

1. Trying riscof: issues primarily w/running in VM (I had similar issues)
    1. Pmake not available w/ CentOS, needs fixing
    2. VMs need to allocate sufficient mem (>2GB for spike)
    3. Docs need to be more idiot-proof (dir structure would be a small help)
2. 4.3 Assertion value in UPD must be optional
3. 4.3: require test_case param1 to match #ifdef param?
4. 4.3.2.2 requires #ifdef/endif pairs, but 4.3.5 prohibits them: change wording so that only ifdefs are those surrounding test cases
5. 4.4 Q:Test pool ref doc is it necessary? Do any fields require manual updates? Could be regenerated for each run (must be forchanges since last run, or to integrate custom tests?) I generated by riscv-config, then format needs to be further documented.

5. 4.4.1 If it remains, the example description must be updated to include all the fields of each entry, where the content comes from (e.g. test_case macro, repository hash) – Q: anything needs to be manually updated?
6. appendixA.b – fix macro references as in 4.3.4 Are there any other includes required? [no, all are optional]
7. Appendix A.d.a fix testcase a32 → a3
8. Emulation libraries: if we allow them, then we need github hash as part of the compliance report. Basic question: compliance of EE or just arch? The decision tree cost is onetime for loading library vs continued cost of maintaining formal model

Decisions & Action Items

Decisions

- RVTEST_BASEUPD macro will be added
- Email discussions will be started regarding whether the test pool reference doc is necessary (or necessary to be exposed)
- Email discussions will be restarted for emulated op handling

Action Items - Framework

- Allen will continue discussion of emulated ops handling:
  - - should the compliance framework check for platform compliance or just architectural compliance?
  - - if platform compliance, should it handle emulated ops w/emulation or just that it traps properly
  - If emulation: do we need to add vendor emulation libra
- Pawan will
  - enable riscof to be run with either make or Pmake (& document how to do each)
  - Add RVTEST_BASEUPD

Action Items - TestFormat spec

Allen will fix typos in TestSpec

- Allen will start email regarding necessity of test pool reference doc in the spec
- Once this is resolved, a vote will be held to ratify the spec

Next Meeting Agenda (in order of Priority)

1. Pull request (#65)
   - Review comments on emulated op handling (slide 6)
   - Review comments on need for test pool reference doc
   - If no substantive comments, will set up vote to ratify test spec 1.2.4
2. RISCOF reviews:
3. Reviewing and Closing issues (currently 23 open)
4. 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?

Backup from previous discussions

State of Compliance

- The de-facto compliance framework is the existing GIT repository
- There have been significant updates to this lately
  - a separate config repository that enables precise description of implemented options, including more precise WARL definition
  - more model support
  - more coverage support
- This will be formalized as a v.1 compliance framework
- V.2 is under review
  - updated test spec format, designed to be easier to write compliance tests (under review)
  - updated framework (riscof) (under review)
    - implements the updated test spec format,
    - specifically enables dynamic signatures, instead of static or self checking
  - formalizes WARL description
    - Limits illegal→legal mappings, making formal models and tests easier to write

Future Discussions

1. Coverage metric (slide 12)
2. WARL definition – (slides 13-16)
   https://riscv-config.readthedocs.io/en/latest/yaml-specs.html#warl-field-restriction-proposal

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

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 [0x4000001:0x3FFFFFFF] -> 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 [0x4000001:0x3FFFFFFF] -> unchanged
```