Compliance Task Group Call – Minutes

Weds, Feb12 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)

(tentative) Meeting Agenda

1. Pull request (#65) TestFormat Spec
   - If no substantive comments, will approve pull , set up vote to ratify test spec 1.2.5
2. Pull request (#77) – fix for AUIPC test
3. Progress on last meeting Action Items
4. Reviewing and Closing issues (slide 7, currently 23 open)
5. Coverage discussion (slides 12-13)
6. 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 - update test format spec, removing TestPool Reference doc mentions, and move anything related to emulation to a “deferred” section - DONE

Simon – will re-review the test format spec (Allen will send out poll for approval if no further issues reported)

Simon – come up with test condition examples that are difficult to do manually (Vector spec?)

Neel,Pawan

- document how to use riscof as separate pieces
- investigate if possible to release a docker image
- look into a generator script
- remove hard-coded directory references from scripts, establish/document a standard directory structure for the riscof environment

Allen-document required changes for emulation to send to Neel, Pawan – DONE

Pull/Issue Status

Type	Issue#	date	submitter	title	status
Pull	#80	Feb 04, 2020	towoe	Report error in Ibex run target	bug - needs fixing
Pull	#79←	Feb 01, 2020	Xiretza	Ignore case when comparing signature outputs	bug - needs fixing
Pull	#83	Feb 07, 2020	pmundkur	Support F extension on RV32 sail-riscv-c.	new feature
Pull	#77<-85	Jan 20, 2020	bobbi	fix I-AUIPC-01 to write signature to correct	The fix for #77
Issue	#84<-85	Feb 08, 2020	byllgrim	I-SW-01 corrupts .text region	bug - needs fixing
Issue	#78	Jan 26, 2020	bobbl	RV_COMPLIANCE_HALT must contain SWSIG	bug - needs fixing
Issue	#76<--85	Jan 20, 2020	bobbl	I-AUIPC-01 test writes signature to wrong address	bug - needs fixing
Issue	#37	Jan 31, 2019	astimonov	rv32imc C.SWSP test5 writes a word outside the binary	bug - needs fixing
Issue	#30	Jan 21, 2019	RolfyYu	The golden results of rv32ui and rv64ui should be different	bug - needs fixing
Issue	#08	Oct 17, 2018	AnttiLukats	RV32I/I-IO.S bad file name	bug - needs fixing
Issue	#67	Sep 25, 2019	rongcuid	RV32I Immediate Operands error	fixed in commit cae8567?
Issue	#11	Oct 26, 2018	neelgala	illegal.S in rv32mi generates sup interrupt - may not be supported on all impls	fixed, close
Issue	#33	Jan 28, 2019	debs-sifive	rv32si/ma_fetch.S produces a different signature depending on RVC support	fixed?
Issue	#32	Jan 25, 2019	debs-sifive	breakpoint.s undesired behavior when trigger does not exist?	fixed?
Issue	#28	Dec 21, 2018	bluewww	I-SB-01 test war hazard (address register)	fixed?
Issue	#27	Dec 21, 2018	jlucnagel	Macros are checking side-effects	fixed?
Issue	#70	Oct 08,, 2019	towoe	Target error exit condition	not a bug?
Issue	#42	Feb 05, 2019	as-sc	Misaligned fetch bit must be excluded for RVC	not a bug?
Issue	#38	Jan 31, 2019	santhoshvlsi	I-LB-01 test - Load the data into X0 GPR register	not a bug?
Issue	#63	Aug 13, 2019	jeremybennett	Global linker script is not appropriate bug	TBD - fixed in riscof?
Issue	#47	Feb 16, 2019	aprnath	Machine mode atomic extension tests?	duplicate in v.2 for m/s/u
Issue	#72	Oct 26, 2019	vogelpi	Allow for non-word aligned `mtvec`	will be fixed in V.2
Issue	#45	Feb 12, 2019	debs-sifive	Reorganization of test suites for code maintainability	will be fixed in V.2
Issue	#40	Feb 04, 2019	debs-sifive	Usage of tohost/fromhost should be removed	will be fixed in V.2
Issue	#31	Jan 25, 2019	debs-sifive	I-MISALIGN_JMP-01.S outdated use of `mbadaddr` in trap handler?	will be fixed in V.2
Issue	#22	Nov 24, 2018	brouhaha	I-MISALIGN_LDST-01 assumes misaligned data access will trap	will be fixed in V.2
Issue	#16	Nov 07, 2018	neelgala	I-MISALIGN_JMP-01.S assumes compressed can be turned off	will be fixed in V.2
Issue	#09	Oct 22, 2018	neelgala	Setting SATP and PMP should be optional	will be fixed in V.2
Issue	#04	Jul 03, 2018	kasanovic	Section 2.3 Target Environment	will be fixed in V.2
Issue	#03	Jul 03, 2018	kasanovic	2.4 Processor configuration clarification	will be fixed in V.2

Discussion

1. Will we have only statically written tests or can we allow dynamical (run-time tool)generated tests?
   1. E.g. there are 90 configurations of vector implementations, and 8000 tests per configuration. Can we afford to **not** have dynamically generated tests?
   2. How does this work with the test filtering? Conceptually, it shouldn't be an issue, since the tests are generated according to the configuration, which is essentially the same as filtering based on a configuration
2. reject pull#80
   1. Needs a bit more feedback. The problem in general is that tests that fail to execute (e.g. build failure?) do not check results, therefore appear to be passing. I think we need to understand why standard tests were failing that way, fix it everywhere if there is a general problem, and fix the failure if it Ibex specific
3. Stephano (will be) in talks with YOCTO to see if they can support more test writing(/coverage metrics?)
4. The remaining controversial issue with the test spec is whether it is possible to easily manually generate test conditions
   1. Test conditions use YAMLconfig – for most cases, it is easy
   2. We may need syntax that has OR conditions, not just AND conditions
   3. There don't appear to be any issues with dynamically tests; the conditions used to generate them are effectively the filter conditions
5. Coverage: how do we design coverage rules? Where does compliance become DV? Still an issue
6. Formal checking vs instruction level test – how does this fit?
   1. Formal checking uses RTL, can't test SW implementations
   2. Doesn't fit with comparison of two models
   3. It would be great if we get to the point that we can test compliance formally, but unclear this will be uniformly possible on different implementations, so we still need to deal with the instruction level tests

Decisions & Action Items

Decisions

- No further changes requested for test format spec
- Pull #80 rejected until we understand the issue better

Action Items

- Allen: will put test spec .pdf into files section of compliance TG lists.riscv.org site
- Allen: will send poll out to vote on acceptance of the current spec v1.2.5

Next Meeting Agenda (in order of Priority)

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 *<u>category</u>*
- 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
<u>coverage & categories</u>

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