NOTE: updated zoom info for next meeting and beyond.

Meeting link: https://zoom.us/j/95666422067

Passcode: 174425

Join link: https://zoom.us/j/95666422067?pwd=aHM1ZEJDcFJ3cGE0UWdTRmtaUXhsQT09

Date: 2023/08/15

Task Group: Fast Interrupts

Chair: Dan Smathers

Co-Chair: Jean-Baptiste Brelot

Number of Attendees: 5

Meetings Disclaimers Video :
https://drive.google.com/file/d/1y_XWJus8M5ZwSQ2cvEOzCjlOmsmXOnN4/view

Current issues on github: https://github.com/riscv/riscv-fast-interrupt/issues

Previous meeting minutes: https://github.com/riscv/riscv-fast-interrupt/tree/master/minutes

Fast Interrupt DoD (Definition of Done) Status:
https://wiki.riscv.org/display/HOME/Fast+Interrupts+TG

Next meeting agenda (08/29/23)

Issue #345/pull #346 – add Smstateen extension text about new ssclic CSRs?

Issue #339 – should mcause.mpil be updated on exceptions or only interrupts.

Issue #314 – shv==0 requirement for xnxti. more discussion needed. New CSR?

Issue #308/pull #325/pull #344– xnxti service loop – fix of pull #343 based on discussion.

Issue #307/pull #331 – Parameter cleanup (compare with pull #297?)

Issue #303 – How are CLIC-only CSRs and fields handled in CLINT mode

Meeting minutes:

Krste could not attend. Discussed ACT/SAIL/Spike.

ACT/riscof intended development flow.

Should smclint/ssclint just be added to priv directory instead of own directory? Is interrupt support required?

Current SAIL build is broken – something about isabelle – memory ordering properties. class of tools. take model and create assertions and see if certain properties hold true. Bill is not using latest SAIL model right now. why did compile error get past SAIL ci? isabelle is a completely separate tool.

When do a SAIL pull request, ci runs it on another machine.

Allen: When writing interrupt tests, look at env.h - prologue and epilogue macros that are instantiated. don't use fence. there is a fence macro. use rvmodel fencei macro. if don't define turns into nop or fence op.

If use the currently provided ACT interrupt handler, goes back to point of interruption. already saves signatures. epc tval, but not designed for nested interrupts. May not be sufficient for CLIC interrupt tests.

Trap signature area. regular area is sigupdate. trap handler is trap sig area. stores the vector and the interrupt mode it came from? so if vectored interrupts. for clic, modify arch-test.h to support more signatures. put in any special handling? ifdef in arch-test.h. if right value for clic, store these extra things.

In arch-test.h there is a special handler.

Macros in test_macros.h - goto m-mode macro, goto lower m-mode macro. handles weird details. if have virtual memory enabled and go to lower mode, returns to a physical address but want to return to virtual address that maps to physical address. arch_test.h includes test_macros.h

For interrupt coverage – when return from interrupt, is mprv automatically turned off. 2 ints – coverage. how to check interrupt nesting occurred with ACT coverage model? Need to make sure actually saw both interrupts and handled and what order.

Corrections by Allen Baum from notes above:

Some corrections:
It's not env.h, it's env/arch_test.h and env/test_macros.h
The trap handler cant handle horizontal nested interrupts, but it is possible to register a special handler that saves state, reenables interrupts,
then restores state (after waiting for an interrupt to happen maybe?) and returns - and that will enable horizontal nested interrupt testing.
Note that that registered routine just needs to get written once (as a macro with 3 versions, one for each interrupted mode),
so it can be reused in other tests. You can also use this to save any (currently unimplemented) extra state in the signature,
though it may be a little tricky because of needing to carefully handle the trap signature pointer in different modes.
(The pointer is a physical address, and needs to be offset (translated) into the VM mode of the handler.

OR, you can simply return from the interrupt and explicitly store the state into the regular signature region.

Github updates since last minutes:

Specification updates since last minutes:

Open issue status:

Issues that can be closed?

#307/pull #331 – NUM_INTERRUPT listed in two sections (text cleanup)

Need spec updates:

#158 – change CLICCTRLBITS to CLICMLPBITS. (resolve #226 first?)

#171 – CLICCFGLBITS parameter - related to #80, #158. (resolve #226 first)

Pull #286 – text update fix for #96/158/171/226/49 - allow multiple implementations for priv mode/level.

Need more discussion:

#102 - preemptible interrupt handler code (for section 7.2)

#226 – replace clicintattr.mode with xcliccfg.xlvl proposal (formerly #96)

#303 **-** How are CLIC-only CSRs and fields handled in CLINT mode?

#308/pull #325 – xnxti service loop behavior

#314 – shv==0 requirement for xnxti

#339 – should mcause.mpil be updated on exceptions or only interrupts?

Issues need to be worked:

#91 – DTS entry – have linux group review DTS example. If add CLINT compatibility option, just use same DTS entry as clint?

#107 - heritage of features. keep researching and adding references to bibliography.

#185 – SAIL model implementation of CLIC

#186 – CLIC architecture tests

#187 – QEMU CLIC implementation update

#221 – compressed clicintip/clicintie status registers (1-bit per interrupt) similar to AIA eip0-eip63? Add justification and create a pull with a proposal. Pull #270. add prior art reference.

#242 – SPIKE model implementation of CLIC

Issues punted for rev1, keep open for future enhancements:

#92/Pull #281 – hypervisor compatibility. still punted for rev1?

#99 – horizontal interrupt window. punted for rev1

#101 - xnxti to trigger on equal level. punted for rev1

#106 – allow level change. Punted for rev1

#108 – pushint/popint? defer to broader discussion on providing hardware stacking of interrupt contexts. Punted for rev1

#192 – allow mix of CLIC/CLINT at different priv modes punted for rev1

#248 – CLIC hypervisor mode (related to #92).

#**329 -** support for hw register save.