Date: 2023/04/25

Task Group: Fast Interrupts

Chair: Dan Smathers

Co-Chair: Jean-Baptiste Brelot

Number of Attendees: 6

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: (new location)
https://wiki.riscv.org/display/HOME/Fast+Interrupts+TG

Next meeting agenda (05/09/23)

Discuss Issue #329 – support for hw register save

Issue #322/pull 330 – clean up xcliccfg text

Issue #321/pull 327 – clean up setting xintthresh to min (vs. 0) text

Issue #320/pull 328 – RNMI CLIC interaction details

Issue #317/pull 326 – add text to code examples about non-x registers

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

Issue #308/pull #325 – xnxti service loop – more discussion needed.

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

Meeting minutes:

Krste and Jean-Baptiste are not able to attend today. Discussing ISA status checklist items and path to freezing spec.

Major items on Freeze Checklist:

- **RISC-V SAIL**

Saravana introduction. will be working on interrupts and SAIL. From networking background. Coming up to speed on SAIL model. Went through cookbook model. Starting with PLIC. PLIC is non-ISA. interrupt registers PLIC is almost completely MMIO addresses. wires from PLIC into interrupt registers. That I/f is sort of architectural. hook up pseudo event generator to wiggle the wires. if hook up PLIC have to connect to PLIC.

From a hart perspective there are a lot of interrupt wires coming in. how do they get there. incoming write addr turns into wiggling a wire.

Event generator relative to retired count of a hart. Assert the wire 5 instruction retires from now. As independent from implementation as possible. instruction retires are relatively deterministic.

SAIL status – 1 more pull request for vector. do have backlog. 7 or 8 prs that are pending. some PRs may affect the same file. instruction extension usually just a separate file. CLIC will add CSRs and is more complicated (touch at least 3 files). RNMI. PRs priority things that are ratified.

- **RISC-V Tests**

If there is a PR for ACT should have been tested and compared to Spike. riscv-non-isa/riscv-arch-test (github.com)

Ovpsim may be a good alternative to spike with initial development while spike is being updated. have to do in environment that generates interrupts.

Regression testing – interaction with CLINT.

Would CLIC get its own folder under riscv-arch-test/riscv-test-suite/rv32i?

In one place but say execute if RV32I or RV64I or don't run test.

Test plan – need to write something that stimulates clic sources, turns into wires.

ACT – read/understood/implemented the spec. not DV coverage (lots of combinations). but do want some cross-coverage.

- **RISC-V Tests Input**

riscv-software-src/riscv-config: RISC-V Configuration Validator (github.com)

Schema_isa.yaml - gives allowed configuration on risc-v and allowed values. WARL fields give allowed ranges. 10k lines in file.

Examples/rivc32i_isa.yaml is compared against schema_isa.yaml to see if it is allowed.

When add new CSRs, add PR to add to schema_isa.yaml and then will be checked against example implementation.

Schemas/schema_platforml.yaml - memory mapped registers (like clicintctl, etc.)

Run python scripts that look at your implementation and see if it is valid. e.g., if have d then have f- extenstion, etc. checks if WARL field is valid.

Write anything, read legal. mapping from what is illegal to what is legal. mapping is arbitrary. so prefer (easiest) if implementers implement when write something illegal, don't write. that works easiest for describing in this file.

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)

#317/pull #326 – clarification of trampoline examples

#320/pull #328 – clarify CLIC with RNMI

#321/pull #327 – clearing xintthresh when fewer than 8 thres bits are implemented.

#322/pull #330 – xcliccfg field text cleaning. text needs to be cleaned up due to newly added fields.

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

#329 – support for HW register save?

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 (probably waiver)

#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 required if sail is waived?

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).