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: 2023/12/05

Task Group: Fast Interrupts

Chair: Dan Smathers

Co-Chair: Jean-Baptiste Brelot

Number of Attendees: 8

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 (12/19/23)

Issue #349/pull #364 – discuss moving CLIC memory mapped registers to indirect CSRs

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.

Meeting minutes:

New meeting time poll: Frequency and duration. for krste all times are bad. let others drive it. this cadence seems about right.

Issue #349 – using indirect CSRs for clicintctrl/attr/ip/ie. Hesitate to implement both indirect and CSRs and memory mapped. choose one or the other.

AIA – one register to implement. try to be compatible with AIA? in terms of msi – keep that consistent. restrict memory map for software interrupts. just because have msis, don’t open up the whole memory map to everything.

Don’t' have to have indirect csrs. understanding indirect csr as base. memory map just opened up enough to do software interrupts.

How should we pack the bits? if ditch memory map, just pack it

With csrs, can put 4 bits in each register, can touch any for free. Because of immediate. can put all 32 bits. if setup a mask, can set/clear each one.

In terms of packing, think about software use cases. When set miselect – index. Ok to have several different scales. ip/ie as bit packed registers. because bit-packed can use same scale.

The fields that need more, can have different range and different shifting amount. mireg, mireg1. those that need different number of bits. bitpacked ones one range. byte packed in a different range. With a different offset. indexing by miselect. should be binary scaling.

Miselect and then field in csr. which mireg are you using. awkward to do that from software. diff miregs are fields in the structure. so keep ie in mireg, ip in mireg2. two arrarys. could overlay ranges but don't do that. if don't do memory mapped, don't need bytes.

If do RV32/64, do upper and lower. up to extension to decide. make them always 32 bits? penalty if want to turn off 64 interrupts in a shot, due 2. usualy split because have address related value.

For issue #314:

Look up previous artwork for skipping save restore. single bit in csr to say skip by 16 bits.

Can it be derived from context from bits we are already changing? e.g. if interrupt level is non-zero. maybe that works? global mode setting to operate that way or not. so don't need to do anything in the code. if interrupt level is 0, need to save/restore. if interrupt is non-zero, skip save/restore.

But every handler needs. so have save/restore code static. if interrupts enabled and not in a handler, jump to save code, nexti – software handler jump in that middle loop.

Would open avenues for lower interrupt latency with hwvectored interrupts to avoid save/restores.

Discussing issue – read that would increment the select csr so do read/read/read. Because reads can't do increments. read causing side effects.

Development branch for ACT tests to be created. not there yet. should happen soon. read_csr in coverage.

Github updates since last minutes:

Specification updates since last minutes:

Open issue status:

Issues that can be closed?

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.

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

Need more discussion:

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

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

#308/pull #325 – xnxti service loop behavior

#314 – shv==0 requirement for xnxti

#349 – discuss moving CLIC memory mapped registers to indirect CSRs

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

#360 - Requested Tentative stateen bit (refer to greg email on how to interact with AR).

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.

#348 – discuss location of inhv for future if enabling CLIC per mode. xcause still ok?