Date: 2021/08/17

Task Group: Fast Interrupts

Chair: Dan Smathers

Co-Chair: Kevin Chen

Number of Attendees: ~7

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/TECH/Fast+Interrupts+TG

Next meeting agenda (08/31/21)

More open issue discussion?

Meeting minutes

Closed #162 with pull #163 – specify clicintie as bit 0.

#45 – when changing mtvec mode what happens to stvec and utvec. if both clic and clint are supported, then the low bits of stvec/utvec will be viewed as read-only in that mode.

If in clint mode and switch to clic mode and back, do aaa bits change?

If have both clic and clint, then both are restricted to 64 byte aligned. bit 1 of *tvec is controlled by bit 1 of mtvec (lower priv modes see it as read-only. bit 0 of lower-priv mode xtvec is unaffected by changes to bit 0 of mtvec. needs spec update.

#50 – add MPRV/SUM/MXR to m/scause? used in linux. linux doesn't care about turning off interrupts for 50 cycles. if not using linux, don't need to use these so don't need to save restore? if doing emulation, would probably want to turn off interrupts? for misaligned load/stores, emulating missing functionality. they can be saved and restored explicitly with a stack? may be required to turn off interrupts when you use them. in linux there are a lot of times that interrupts are turned off. maybe don't need to optimize for this case. especially in m-mode linux. MXR – make executable/readable for emulating instructions. maybe only for CLINT use-case? If it is really time sensitive, handler doesn't need to save/restore MXR/SUM and can ignore their setting completely. But MPRV may be more problematic. It is not desirable to lock out interrupts while MPRV is set so the handler needs a way to save/restore MPRV without accessing memory. Additional discussion via email after meeting.

Github updates since last minutes:

#50 – additional comments on proposal to add MPRV, SUM, MXR to m/scause

#162 – specify clicintie bit

#163 – pull request for #162

Specification updates since last minutes:

#163 - clarified clicintie bit position (added same wording as clicinip).

Open issue status:

Issues that can be closed?

#29 – Interrupt trigger in the trigger module – closed with pull #126?

Need spec updates:

#45 – all priv modes in clic or all in clint. Need spec update that only m-mode can select. Pull #145 tries to close. need to clarify stvec.mode, utvec.mode register behavior when mtvec.mode switches.

#48 – logic/state diagram for clicintip signals (related to #149?), add more clarification about hardware clearing of IP bits. only xnxti writes can clear IP bits.

#75 – move hw vectoring to separate section in spec - waiting until other spec updated before making this large text change.

#80 – version fields in clicinfo – remove clicinfo and make it a separate platform discovery mechanism? remove mclicbase too?

#86 – update xnxti pseudo-code and text

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

#148 – Elaborate address map possibilities. Platform should be doing the layout.

#154 – inhv clarifications

#158 – change CLICCTRLBITS to CLICMLPBITS.

#160 - can rewrite to allow for but not assume n-extension is available.

#162 – specify clicintie bit

Need more discussion:

#49 – reopened: prev spec had single control for multiple harts but current spec implies control per hart and implications.

#50 – xcause save privilege modification bits

#86 – auto clearing of edge interrupts

#96 – proposed reformat of cliccfg

#97 – proposed reformat of xcause CSRs

#98 – name of CLIC (related to #109?)

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

#31, #120 – WFI behavior – need to clarify text in pull #135. pull text has typos. Pull #144 tries to fix typos and clarify text. Discuss in TG. Tried to update pull text to clarify when interrupt is set to 0.

#155 – clarify inttrig details. Entire meeting discussed CLIC trigger details with Tim from debug group. CLIC, trigger module, and hart need to be tightly integrated since these features need to be in the core to get sync behavior. discussion on where the details are written down. E.g. CLIC spec describes set of interrupts selected for interface to debug, debug specifies what happens on that event (enter debug mode, start trace, stop trace). For now, keeping abstract action in CLIC, debug takes concrete action. try do discuss more over email reflector.

#158 – add option to reuse unused clicintattr.mode bits with clicintctrl to determine level value when 3 modes are supported but only 1 or 2 modes are selected by clicctrl.nmbits?

Issues related to work in other TGs:

#81 – programming clic in s and u modes – discussions in hypervisor group about a more general pattern of delegating configurable features

#91 – DTS entry – closed with pull #130? have linux group review DTS example.

#92 – hypervisor compatibility – goal is not to virtualize CLIC but to allow virtualization of device interrupts. Follow up with hypervisor group.

#109 – add arch string for CLIC – need appropriate named and versioned sub-extensions of clic.

#155 – clarify inttrig details. discuss with Tim from debug group. Invite Tim to fast-interrupt TG meeting to discuss.

Issues waiting on ratification (encoding/opcode consistency review needed)

#88 – CSR address mapping

Issues punted for rev1, keep open for future enhancements:

#82 – xcause register behavior with some modes in clic and some in clint.

#99 – horizontal interrupt window

#100 – reserve immediate bits

#101 - xnxti to trigger on equal level

#106 – allow level change

#108 – pushint/popint?