Date: 2023/02/28

Task Group: Fast Interrupts

Chair: Dan Smathers

Co-Chair: Kevin Chen

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 (02/28/23)

Pull #297/ parameter discussion.

Meeting minutes

SAIL interrupt. mtimer interrupt sim works. code is in etrace github area. Used to generate trace messages.

Extension naming convention – z (unpriv) vs sm, ss, su (s for priv), next letter for mode. But legacy for sv. plan to write down at some time. naming chapter is out of date. Waiting for asciidoc conversion. want extensions to be additive. three stages of parsing extension string. 2nd does pass of implied (dependencies). 3rd does pass of incompatible (conflicts). order of extensions shouldn't matter.

Naming conventions – high level goal. for unpriv code, just want to know what is possible. want to support abstraction levels so just describe u-mode. profiles only look at s mode and lower. so supervisor and u-mode. name the features for what is visible in the mode. so don't want smclic. but what can user mode rely on. suppose have mcu. What can s see. Give names to those incr. Features. Even though switch is in m-mode, what can s-mode see. how can I find out what is there? you don't. You are told. the ability to turn it on is a different extension. z is all unpriv. dynamic discovery vs static discovery (told info, most common). can view extension names as compiler flags in a static model. toolchain folks are adament that only have access to csr names that compiler flag tells you you had. OS knows what values are legal and only writes those. need a name for things that are getting turned on and off. like a feature vector.

Issue #305/pull #306 – xideleg/xedeleg cleanup.

Issue #295 – should we leave as is or add extra gates. higher mode as threshold set. if take sw vectored, end up in different place than hw vectored. Logic depends on which mode you came from. workaround is keep intthresh 0 when leaving that mode. to keep hw simple in comparitor tree, don't look at inthresh when going to higher. but sw vectoring is filtered by the threshold.

like mprv. started clearing it when you left m-mode. a different option would be to clear thresh when leaving m-mode. would avoid the weird case. done on the mret/sret. use case of intthresh is to temp raise in critical sections. if in a lower mode, then not in a critical setion that needs it. Will copy similar text to priv spec for mstatus.mprv. (y!=M), set to 0.

Issue #290: will create a pull to create a trap.

Issue #304 – will split into 2 issues to separate cliccfg.nlbits question. will try to re-write how mem-mapped registers . platform can mandate to 4k boundaries if it is virtual memory but spec can leave it open. use same language as in m-mode. for cliccfg could put fields in different places in the word. put in one word with the byte for each priv mode. high byte is m mode, low byte is u-mode. byte 1 is s-mode.

Github updates since last minutes:

Specification updates since last minutes:

Open issue status:

Issues that can be closed? (Simpler **issues to discuss in red)**

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.

#305 – descriptions of xideleg and xedelg

Need more discussion:

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

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

#290 – additional hw vector table questions

#295 – xintthresh question

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

#304 – clarify changes to CLIC memory-mapped registers

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).