Date: 2020/5/26
Task Group: Fast Interrupts
Chair: Krste Asanovic
Co-Chair: Kevin Chen
Number of Attendees: ~10
Current issues on github:
https://github.com/riscv/riscv-fast-interrupt/blob/master/clic.adoc

Issues discussed:

#27 Pending bit position and behavior of SW setting pending bits

Pending bit is located in bit 0 (LSB).
Writing pending bits with SW will definitely set/clear edge-triggered interrupts. In contrast, for level-triggered interrupts, SW writing to pending bits is ignored completely.

As a side note, in the course of our group discussion, some members expressed that setting pending bits to activate level-triggered interrupts is a useful feature for testing. Another suggestion was to define it as "undefined" to give each company the freedom to decide whether they need this testing feature. However, we finally concluded that writing to pending bits for level-triggered interrupts will be ignored for the following reasons:

1. This "sticky-bit" behavior is against the traditional concept of level-triggering, so it can create confusion for users.
2. To have this sticky property, an extra latch is needed so the HW cost will increase.
3. SW interrupt handler will need an extra step to manually clear this bit in addition to the mandatory clearance of the external level-device. This extra step may be easily forgotten and also add more instructions and latency.
4. Defining the writing behavior entirely as "undefined" can create nightmare for compatibility and compliance tests.
5. Lastly, users can easily add a simple external testing module if they really need this testing feature.

#30 mcause format for exceptions under CLIC mode

For exceptions, in CLIC mode, mcause has "CLIC format." In other modes, it has the original format.

#29 Specification of debug triggers for CLIC interrupt

After discussing with the chair of the Debug Task Group, he said CLIC should support the existing standard triggering behavior but it is fine for CLIC to create additional triggering conditions.

However, later in our group meeting, we decided that existing interrupt triggering conditions should fit our needs in the current stage, so we will not create additional triggering conditions such as “upon pending” or “pending and enabled.”

Therefore, the triggering condition according to the current Debug Spec is: “the trigger only fires if the hart takes a trap because of the interrupt.“

Also, when the trigger fires, “the requested action is taken just before the first instruction of the trap handler is executed.“

#31 Does Interrupt Level Affect WFI?

The global interrupt enables in xstatus.xie determine whether a pending-and-enabled interrupt causes a trap (jump to trap handler). Any pending-and-enabled interrupt must cause WFI to be exited (whether or not xstatus.xie is set).

Individual interrupt has to be enabled by individual enable bits to be taken. If pending interrupt is in higher-privilege mode, then higher-privileged xie bit is ignored (always enabled). WFI does not have to retire for an otherwise enabled and pending interrupt for current privilege mode with interrupt level below the CLIC threshold for the current privilege mode. For interrupts pending-and-enabled for a higher privilege than current mode, the higher-privilege's CLIC threshold is ignored and treated as if set to lowest value for purpose of determining whether to take the interrupt, i.e., trap will be taken into higher-privilege mode, even if higher-privilege global interrupt enable is 0.

If global interrupt bit for current privilege in mstatus is only thing preventing trap from being taken, then WFI must wake up.

Threshold only applies to current privilege mode.

Define pending-and-globally-enabled.