Date: 2021/03/02
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/issues
Previous meeting minutes:
https://github.com/riscv/riscv-fast-interrupt/tree/master/minutes

Issues discussed:

#31 Does Interrupt Level Affect WFI?

Briefly discussed the subtle behavior change of using “0” to replace “-1” for “no filtering mode,” which allows any interrupt to wake up WFI. For the cost of 1 mode bit, it may not justify creating new behavior difference which might confuse users. Nevertheless, after further discussion, we decided to bring this new WFI mode option to a higher level as described below.

This appears to be a more general issue than only for the CLIC. Other interrupt controllers, including the original CLINT scheme, might want to have a way to change WFI behavior to ignore all lower-privilege interrupts when executed in a higher-privilege mode. For example, we could add another mode bit in a status CSR to modify WFI behavior in this way for all local interrupt controllers.

If we address this in the CLIC spec, we might find that later we'll have to define the interaction of the CLIC setting with the other more-universal setting. Suggest we raise this as a privilege group issue, and possibly worth fast tracking. OTOH this might be difficult to get agreement across the different local interrupt controller designs (CLINT, CLIC, AIA).

The WFI-mode bit when clear would preserve existing behavior.
The WFI-mode bit when set could have the form of 1) always ignore lower-mode interrupts, or 2) respect xie bits of lower-mode interrupts, so higher-mode can modify yie bits of modes y<x to get desired behavior.

Action is to raise this issue on priv committee.

After the meeting, Krste further proposed to create a new instruction for the new WFI mode instead of creating a mode bit in CSR. More details can be found in the discussion of this github issue.

#120 Incorrect spec of behavior on WFI (Section 6.3)

Based on previous discussion of behavior for WFI (#31), the description in the current spec is no longer valid:

A higher-mode threshold setting can cause lower-mode interrupts that are enabled and above the lower-mode threshold to not wake up a WFI in the higher-mode. When a currently pending higher-mode interrupt is below the threshold set in the higher mode, the higher-mode interrupt will effectively mask the lower-mode interrupt in the interrupt priority reduction tree, but will not cause WFI to exit in the higher-mode due to the higher-mode threshold setting. Current text does not capture this behavior correctly.

When the higher-mode threshold is set to zero, it cannot mask any pending and enabled lower-mode interrupts.