Date: 2020/10/20
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

Issue discussed:

#105 Is mcause inhv writeable?

The inhv bits are only written by hardware during the table vector read operation. The inhv bits can be written by software, including when hardware vectoring is not in effect. The inhv bit has no effect except when returning from interrupt.

One use case for restart is when there is a bare-S mode running on top of a security M-mode monitor. The M-monitor can take M-mode interrupts in the middle of an S-mode interrupt sequence - the inhv bit allows the hardware to preempt the S-mode vector table read to take the M-mode interrupt. The M-mode interrupt handler does not have to be aware of the S-mode interrupt handler state and can simply mret as before, in which case the hardware will replay the table read.

An implementation could choose to make the interrupt/table-read sequence non-interruptible (at the cost of worst-case interrupt latency), so inhv is never reported. But some mechanism is still needed to indicate that the table read itself caused a trap, even if fatal. We use inhv to report this case, as the bit is used to support the other scenarios. Using a separate cause field for this case would prevent restart in other cases.

The cases when the table read exceptions are not fatal traps are when this code is running in a lower-privilege mode under some higher-privilege mode that takes action on the table read trap then resumes (.e.g, user-mode interrupts with S-mode paging, or S-mode guest OS on hypervisor, or U/S-mode code under M-mode that has access logging on turned on).

When the table read causes a horizontal trap, this usually signifies an error in the software configuration. These horizontal table read traps are not generally recoverable, so some separate debug mechanism will usually be required to determine the cause of the error. The table address will be available in mtval for debugger to inspect.

#104 Does a level==0 pending and enabled machine interrupt mask a non-zero pending and enabled user interrupt from occurring?

Yes, that's correct.

#89 Proposal: reorder lower 16 bits

We should update the spec to indicate these orderings are recommendations as part of a profile, not mandatory in all incarnations of the CLIC.