Date: 2021/02/16
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:

#111 Strange behaviour for faults accessing trap vector table

There are two separate issues here - packing of other bits into cause is already recorded as #97, so please continue that discussion on that issue.

Any horizontal trap in an interrupt handler is non-resumable until exception state has been saved. Using either mtval or xepc does not change this (#87).

The reason to support resumable faults on vector tables is in case a more privileged level has virtualized/emulated the RTOS+application using the CLIC, and could have paged out the vector table (or just wants to track accesses to it).

Krste believes the original motivation to use xepc instead of mtval was that the original CLIC proposal used instruction fetch instead of data fetch to access the vector table, and effectively switched to a new ISA for vector table fetches which only had a single instruction: "jump to address encoded in instruction". This trick avoided need to route memory load data to PC, which is a path not otherwise present in processors, and also allowed restart to be handled as an instruction fetch restart. Using xepc instead of mtval avoided the need for new hardware path from mtval to pc.

It would be more consistent to use mtval to hold the faulting table address, as this would match the existing privilege architecture spec.

#31 Does Interrupt Level Affect WFI?

Krste clarified that, if a higher privilege interrupt occurs, it will not resume WFI but start servicing this new high privilege interrupt instead.

Also, a member had a nice proposal: use “0” for threshold to replace “-1” if we want the “no filtering mode,” which allows any interrupts to wake up WFI. This is cheaper (no cost at all) and also very intuitive since this usage is in agreement with the original definition.