

Date: 2021/01/05

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>

Krste brought messages from TSC that we should finalize our spec this year in order to be included in the RVM22 profile. The features in the profile will be classified as: mandatory, optional, or unsupported (which means “don’t care” and does not intent to work with other features).

Existing basic interrupt controller (defined in the privileged spec) will be included in the RVM20 and becomes mandatory. So a member asked: when we include CLIC in RVM22, does it mean we must also include the mandatory feature in RVM20 (the basic interrupt controller) with CLIC? Or is a way to replace the mandatory feature in RVM20. This question needs further discussion and most likely will need the support from the Configuration Structure Task Group.

Issue discussed:

#108 usefulness of PUSHINT/POPINT

The TG discussed this during fast-int meeting. Some comments:

Code size is not a major concern with ISRs, but performance could be. Hardware stacking of CSRs might be a feature that is implemented without ISA support as part of the interrupt handling mechanism, and this would seem preferable to adding this CSR form of push/pop instruction.

While push/pop int instructions might allow more flexible handler design, most of the handler flexibility would be outside the save/restore mechanism in any case. It is not clear that the parameterization needed for stacking using the push/pop instruction would be any less than the equivalent parameterization in the hardware stacking mechanism.

General push/pop instructions could be useful in a handler, but would have to understand the impact on interrupt response time (if not pre-emptible) or interrupt throughput (if uses restart to handle preemption). A design that allowed resumable preemption through many interrupt levels could be difficult to support and require additional exposed state to be saved and restored.

Another proposal is to have a separate push/pop instruction only for interrupt CSRs, which would need less encoding space - the main advantage of this instruction would be to allow greater speed with a secondary advantage being fewer bytes of code to fetch in handler.

We think we can delay this discussion until other issues are resolved in the design.

#109 Add arch string for CLIC

Yes. As part of the arch profile work, we are breaking up the privileged architecture features into named extensions to allow the supported features to be listed clearly. The CLIC would fall into this category for RVM22, and should have appropriate named and versioned sub-extensions. The earlier versions had multiple options, so would want to not just list v0.9 for example, but have finer-grain distinctions on supported features.

#106 enhance support co-routines and thus allow level change already saved registers.

Since this is a relatively big proposal, some member suggested the author to give a presentation to describe the details.