Date: 2020/9/22
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

A member asked about the status and availability of NMI spec. Krste replied that there were a few places needing updates, so it would be available in about 2 weeks.

#97 Proposed reformat of *cause CSRs

Confirmed that new behavior is NOT backwards-incompatible, but priv spec could be clarified.

In TG meeting, 9/22/2020 - adding these fields to mcause does not really affect backwards-compatibility, as the upper bits of the interrupt cause values were designated for platform use. The higher values of exception cause were also reserved. The text of the privileged ISA document should be clarified.

#99 New instruction proposal for “transient enable” of interrupts

To clarify:

This provides a single instruction "transient enable" of interrupts pending to allow them to be taken at this point in code, while before or after this instruction interrupts are disabled.

The instruction acts as a NOP, except that interrupts are temporarily globally enabled for duration of instruction.

The instruction does not change state, so if interrupts were enabled, they remain enabled afterwards.

EPC should point at the "transient enable" instruction, may get re-executed.
Value in xstatus.xpie would be that in xstatus.xie at time of interrupt.

Pros: this reduces code size and improves performance

Cons: implementation complexity

Does not directly relate to nxti but this is a space in the encoding that is otherwise of no use.

Might this be encoded in a different space, including WFI that temporarily re-enables interrupts?

#101 Allow xnxti to trigger on equal level as saved interrupt context

Discussed on whether reuse of same interrupt level already happens with mnxti - must keep behavior that levels control preemption in usual case.

#103 add Branch Bit Set Instruction to Fast Interrupt Repertoire

To clarify:

Proposal is to add a branch instruction that checks a bit whose position is encoded as an immediate in one of the branch operand.

Replacing "shift-left; branch-on-ltz/gtz" with branch of x[rs1][uimm5]

shift is compressible if destructive
andi, beqz/bnez, the beqz/bnez are compressible

Both of these forms need an available register whereas branch-on-immediate does not overwrite a register.

With B extension, rotate would avoid requiring a register, by rotating bits in place,. but would need three instructions for every bit test.