May 4, 2023 | RISC-V Control Transfer Records TG Meeting

Attendees: tech.meetings@riscv.org Beeman Strong Bruce Ableidinger

Notes

- **Attendees**: Beeman, Bruce, JohnS, Robert, Snehasish
- **Slides/video** here
- Latency discussion, continued
    - Agreed on 4 exponent bits and 12 mantissa bits in CC field
    - Propose allowing implementations to support 12..16 bits in CC field
        - Corresponds to physical counter widths of 12, 13, 15, 19, 27 bits
    - Will recommend 15 or 16 bit CC field (3-4 exponent bits) if supporting transfer type filtering, to allow for larger values for call/ret-only recording
        - Snehasish will get data soon on function timing
    - Agreed that SW can account for error when exp>1, better than requiring cost of HW rounding
        - Though the ideal value to add is not a whole number. E.g., for 3 bits cut off values can be 0..7, so ideal error correction value is 3.5
        - SW can choose how precisely it wants to account for error, spec should provide guidance
- RAS Emulation
    - Reviewed (again) proposal
        - Calls recorded normally
        - Returns “pop” youngest entry by clearing valid bit (ctrsource.V) and rotating stack
        - Changed since last time to not do anything special when stack is empty (still rotate the stack)
    - Could cycle counter sum the values, so that entry X has all cycles since call in entry X+1 (prior entry)?
        - Would require HW to add ctrdata.CC value to HW counter on RET that invalidates an entry, and RETs would not reset counter. Then each CALL entry would have total cycles since last entry on the stack.
        - Adds HW cost & complexity
        - Could say that if CCV=1 on CALL entries with RASEMU=1 means HW does this, else HW should clear CCV
    - Zcnt/Zcmp has additional insts defined as calls & returns, listed in N-trace spec. Also adds another branch.
    - Exceptions (where RETs are skipped) are a concern
        - Will result in some stale CALL entries left on the stack
        - Shadow stack has support for skipped RETs, find out how. Anything we could leverage?
        - Could inval entire stack on mismatching RET? Would inval stale entries, but also possibly some valid entries

        - Or set some bit in an entry that indicates that stack was corrupted?
        - Will consider options next week
    - Snehasish: Google doesn't use LBR call-stack mode, so no insight here. Google also doesn't use exceptions to skip RETs at all
- **Out of time**
    - Next week will consider options for exceptions, aim to close on RAS emulation mode

Action items

- ☑ ~~Beeman Strong~~ ~~Apr 27, 2023~~ ~~start thread about CC requirements~~
- ☑ ~~Beeman Strong~~ ~~Apr 27, 2023~~ ~~start thread about latency rounding~~