Jun 1, 2023 | RISC-V Control Transfer Records TG Meeting

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

Notes

- **Attendees**: Beeman, JohnS, RajneshK, Snehasish, RobertC, Bruce, Ved
- **Slides** here (no video, sorry!)
- Opens
    - Almost have an initial spec ready, just navigating the adoc build process. Hope to have something for review this week
    - Snehasish: what's the process after the spec is ready?
        - AR review (ratification plan estimates 6 weeks), then freeze, 30-day public review, the various votes, including BoD, before ratification. Est ratification in November.
- Introduction
    - Rajnesh: Rivos, working on PoC of CTR
- Resumed discussing CTR stack corruption and potential solutions
    - Stack unwind could be repaired, but stack switching probably just wants CTR stack cleared
    - Stack switch is a problem when done in user mode. Perf will switch CTR state when RASEMU=1, for kernel context switch.
    - Indication of corruption could be a bit in some CSR, or could mark all subsequent calls as good somehow
    - Want to keep complexity of RASEMU limited, it's not our primary usage. So any solution should be simple.
        - LBR doesn't have any solution for corruption
    - Snehasish: believe call-stack mode usage may suffer due to these corruption issues, so solving them may increase adoption
    - SW clearing could involve a new user mode CSR that just exposes the CLR bit
    - HW clear on return mispredict implementation likely would require arch tolerance of some false positives
    - Bruce: Recursive code can quickly blow away the stack history
    - Snehasish: Datacenter call stacks (C++) typically >32 anyway, so almost always have partial history
    - Next week will review a proposal for both SW and HW stack clearing, next level of details
- Out of time

Action items

- [ ] May 11, 2023 - Beeman Strong - check with Atish if expect perf to always get LCOFIs in M-mode, to work around interrupt masking