May 25, 2023 | RISC-V Control Transfer Records TG Meeting

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

Notes

- **Attendees**: BillM, Bruce, JohnS, RajneshK, RobertC, Snehasish, Ved, Beeman, KenD
- **Slides/video** here
- Opens
    - Snehasish: can we inval CTR RASEMU stack when we know CTR stack corruption will happen?
        - Don't have a user-mode way today to write CTR today
        - Don't know how many to pop. Better to pop all or none?
        - Would need to be fast, at least when CTR is not in use, otherwise SW won't use it
        - Ved: often unwinds aren't in apps, they're in libc or other shared libs
        - Snehasish: not going to be on a fast path when doing this unwind
            - Ved: Oracle DB cares about unwind for shadow-stacks, other architectures had to make them fast
        - Bruce: could have a compile switch
        - Robert: could set a flag somewhere, just to indicate that it happened
            - But could HW do it? Or SW?
            - Discussed this last week, can't write it to the last CTR entry bc will likely be popped
            - Could set a flag in some CSR, but how useful is that? By the time of the next sample, it's possible that the entire stack is valid, or none of it, or something in between
        - Longjmp is costly anyway, so adding instructions isn't so unreasonable
            - Popular in Go and C++ libraries
        - Including a user mode method gives users the option to do it
        - Shadow-stack has much higher overhead, CTR is a nice alternative
            - Though if SS is in use then profiling tools should use it instead of CTR
        - Beeman: perhaps CTR could lean on the sspinc instruction? Add it to the CTR extension, so both SS and CTR share a method for repairing the stack
            - SS's sspinc instruction is a "maybe op", almost a nop when SS disabled. Maybe could be active if SS or CTR are enabled.
            - Sspinc pops SS in shadow-stack mode, but in control-stack mode does something else
            - Robert and Snehasish prefer not to share sspinc, have a dedicated method
        - Let's discuss some options next week
    - S*csrind at the top of ARC's queue, hopefully was reviewed on Tuesday

- CTR freeze
    - Fixed bit positions vs last time
    - Priv mode makes this unnecessary for many usages. Only needed if LCOFI or breakpoint are handled in a mode that is being recorded.
    - Serves to ensure we don't overwrite records we want (from the workload) with records we don't (from the handler)
        - Kernel does not use vectored interrupts, common handler
    - Snehasish: when does freeze freeze?
        - At the time of the trap. RV doesn't have a precise sampling mechanism standardized, which might warrant freezing on some other (earlier) instruction. If/when such an extension is developed, we may need another freeze option (OFFRZ?)
    - Agreed that BPFRZ should be required, and LCOFIFRZ required if Sscofpmf is implemented
- **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