Jul 6, 2023 | RISC-V Control Transfer Records TG Meeting

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

Notes

- **Attendees**: JohnS, RajneshK, RobertC, Snehasish, Beeman, Bruce, Greg
- **Slides/video** here
- Updates
    - New spec release, encompassing updates from last TG meeting
- Context filtering
    - Should CTR support something like Sdtrig and trace specs do? Only record based on match of xcontext CSR, or ASID/VMID?
    - Makes sense for debug features, but CTR analogs are already managed per context by in-target SW (Linux/perf)
        - Robert: but maybe RTOSs don't?
    - Beeman: could make it optional, but hesitant to add another CSR that has to be switched for a usage we're not sure will be used
    - Bruce: thinking of cases where CTR may be used as a "cheap trace"
    - Bruce: could have a single bit that, when set, uses context filtering from Sdext
        - Robert: could also have it mean Sdtrig trace actions (start/stop/notify) apply to CTR
    - Beeman: would lean towards this being custom, or in a debug CSR that SW doesn't have to manage
    - Robert to write up idea and share with the list
- Agreed on no user mode access for now
- CTR abbreviation
    - Got some feedback that CTR may be interpreted as short for counter, should we refer to it as CXR instead?
    - Rajnesh: in Linux, RISC-V PMU enabling uses ctr in variable names for counters
        - Robert: also gets used as short for control
    - Beeman: the extension name will actually be Smctr/Ssctr, CTR is just used for convenience
    - Agreed that there is some confusion, but not enough to warrant changing the abbreviation
- Variable depth
    - Added for two reasons
        - Could reduce context switch latency, requiring fewer register accesses
            - But LBRs only switched in call-stack mode, which is exactly when users need maximum depth (>=32)
        - Could allow hypervisor to "defeature" newer HW (with more CTR entries) to look like older HW (with less CTR entries), to support VM migration
            - If depth was fixed, migrating from 64e to 32e would just mean it always appears that CTRs aren't full. Should be okay.

      - Migrating from 32e to 64e could mean upper entries aren’t switched, and thus could leak info. But could provide SW guidance to always set CLR between save & restore.
  - Do we need variable depth? Or make it fixed?
    - Undesirable that DEPTH changes out from under VM, on migration
    - Addressing this means need variable depth, and HW that supports up to depth X must support all lesser depths
    - Snehasish: not sure this is a problem today
      - Beeman: would be good to know how KVM handles this. But possible that all modern Intel CPUs have same LBR depth of 32.
      - Snehasish: KVM and the VM don’t really handle LBRs, perf does
- Out of time, continue discussion on the email list

Action items

- □