Aug 2, 2023 | RISC-V Control Transfer Records TG Meeting

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

Notes

- **Attendees**: Beeman, Snehasish, DavidW, JohnS, Rajnesh, VictorL, RobertC, BruceA
- **Slides** here (no video, sorry)
- vsctrcontrol
    - Last week said we'd make most bits pass-thru from mctrcontrol, now reconsidering
    - If we expect to support separate host+guest usage, it will be nice if the bits are not shared (so guest has separate config)
        - KVM adding support for this usage with LBR today, but requires switching LBR state on VM-exit/entry
        - Question is how much HW to add for this usage, which we don't expect to be the most common one
    - Rajnesh: Don't want the guest to be out of luck if host using CTR
    - Rajnesh: KVM only does save/restore of LBR on sched_out/sched_in of tasks, whether guest or host. Not on VM-exit/VM-entry
    - Leaning towards keeping vsctrcontrol as a separate config register, consistent with other vs* CSRs. But will include a comment and see what feedback we get.
- WRAP Behavior
    - Can't use WRAP bit to indicate number of valid entries for RASEMU mode, since WRPTR can underflow (when more returns than calls since started)
        - And believe we want that WRPTR underflow behavior for stack stitching
    - Could use it for default mode, but should be rare to not wrap between samples
        - Maybe with transfer type filtering, e.g. only recording exceptions?
    - So question is if WRAP is still useful
    - Also got feedback suggesting 2-bit WRAP scheme, which allows indication that we've not only overwritten some entries but also if we've overwritten all entries
        - Useful for stack stitching between b2b samples
        - Robert: also could be useful for non-RASEMU mode. Could stitch history if not full between samples, which could happen with sampling.
    - Robert: like the 2-bit scheme, wish we had it for trace
    - Beeman: let's sleep on this a bit
- Spec Comments
    - Should mireg[456] accesses cause exceptions or be ROZ?
        - Exceptions don't add much value since can't really emulate CTR entries
        - mireg[23] for entries beyond depth are ROZ, let's be consistent with those
    - Should we keep "External Trap" transfer type in ctrdata.TYPE
        - Using existing "Exception" vs "Interrupt" type would give more information
            - Can still tell external traps since target PC is 0

  - Let’s do that
- Discussed external trap possible dependency on EXCINH/INTRINH, and whether we need separate STE/HTE bits? Too deep, leave these as comments in the spec to be considered.
- Should we move V and MISP from ctrsource/ctrtarget to ctrdata?
  - Only needed if we might need support for 1B opcodes
    - Those would eat up too much opcode space to be practical
  - Keeeping them as-is allows software to skip reading ctrdata for many cases (that don’t depend on metadata)
    - Though can use PC=0 as equivalent of V
      - Not for RASEMU mode
  - Out of time, take this question up again next time

Action items

□