# Oct 17, 2024 | RV Performance Event Sampling TG

Attendees: Snehasish Kumar tech.meetings@riscv.org Beeman Strong

Notes

- **Attendees**: Beeman, Chun, Snehasish, Bruce, Atish
- **Slides/video** here (same slides as last week)
- Continue discussing Sspesa extension
- Recap Counter ID field, to uniquely identify for which counter overflow the sample PC was captured
    - >1 overflow for a single LCOFI very unlikely, and also not a known usage model to sample on multiple counters (generally other counters are just in counting mode)
- Should we snapshot context in addition to PC?
    - Scenarios where context changes in skid should be quite rare
    - Call-stack is more likely to be updated in skid, but modern implementations have limited skid enough to still make it reliable
- Illustrate how counter freeze could make it easy for an implementation to provide all that's needed for precise attribution
    - Simplifies HW needed if can freeze counter at the end of the cycle of overflow
- Why is precision within a cycle useful?
    - For INST.RET (in example), it probably doesn't matter too much
    - But if counting something problematic, like mispredicts or cache misses, want to be able to attribute the event to a specific inst
- Would we have to derive PC live, or offline?
    - Post-processing offline. Like trace, where decode (with disassembly) is done after the fact.
- So counters would freeze on interrupt or on overflow?
    - Propose here that they freeze on overflow at the end of cycle, so not precise (stop at 0). Both cheaper and more useful for this example.
        - So SW likely only needs to read lower 32b of counter
    - X86 has confusing FREEZE_ON_PMI, which actually freezes on overflow
        - This behavior broke Mozilla RR, since record & replay depends on getting all events counted, even during skid
    - Freeze on interrupt is preferred, for counters and CTR, if sample is being attributed to epc
        - Don't want to freeze until the inst to which sample will be attributed
    - Freeze on overflow is preferred with Sspesa, since attribution is to overflow PC
- Will discuss a more formal counter freeze proposal in more detail in later meetings
- CTR freeze changes
    - CTR freezes on interrupt (LCOFIFRZ)
    - With Sspesa, want to freeze on overflow, for reasons discussed

- Could change behavior of LCOFIFRZ when Sspesa is implemented, or add new mctrctl.OFFRZ bit
    - OFFRZ would be more explicit, avoids confusion that Intel has
    - But expect Sspesa will always be used when available, so why not just change LCOFIFRZ behavior when it's implemented?
        - Always post-process, minimal added work to get spc
    - Optionality sometimes adds confusion
    - Could have both bits defined but only implement one?
    - Or could rename LCOFIFRZ before ratification?  Or tweak text to allow for new behavior?
    - Beeman to draft something
- **Out of time**

Action items

- [ ]