Jun 13, 2024 | Performance Event Sampling TG

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

Notes

- **Attendees**: Beeman, Snehasish, Bruce, Dmitriy, RobertC, Atish, Chun
- **Slides** here (forgot video :()
- Review Charter
    - Should it include the extension names?
        - AI Beeman to check, prefer to wait if possible
    - Typo: supoprt, will fix
    - Ready to send to HC
- Requirements for precise event sampling extension
    - Typo: sample PC for events that don't support precise attribution should be older than inst, not younger
    - Multi-instruction retire may require reporting something more than simply a PC value
        - Since implementations may not have the PC for every inst at retire
        - If one inst in retire group gets a cache miss, likely to retire first in group, so should always be the first PC in the cycle
            - But some implementations cluster insts up front, then make the whole cluster wait until all insts have completed. So could be an inst in the middle of the cluster.
        - Trace ingress has retire blocks, which get PC of the first inst, number of insts, and number of half words
            - Could use some similar information here, but want to be able to uniquely identify any inst
    - So would handling concurrent overflows mean need 29 sample PCs? 29 CSRs?
        - Yes, though likely implement only one per counter implemented
        - Could implement fewer sample PCs than counters? Gets complicated then
        - Probably prefer to keep this simple and just have 1 sample PC, at the expense of some corner cases. But we'll discuss.
    - What if I want to capture loads where access takes >100 cycles?
        - You want the inst sampling extension
- Requirements for inst sampling
    - Here could collect memory latency, and use filtering to only collect those that are >100 cycles
    - DLA (data linear address) in PEBS is very useful
    - Could we filter samples based on an addr? Technically possible but not sure how useful it would be
    - Want to do what SW wants, but craft the ISA to ensure it's implementation-friendly

- Could be useful to have a buffering mechanism, rather than needing an interrupt every time
  - Didn't add that yet because generally want to collect call-stack. PEBS most often used with PMI on every sample. Buffer could be a follow-on.
  - With PEBS load latency, can be useful to buffer samples together so can get a higher sample rate
    - PEBS is much less intrusive than a PMI interrupt, probably an order of magnitude. Much of that is due to PMI interrupt handler overhead, esp if using NMI. Is RISC-V as bad?
    - SSE is faster than delivering interrupt to S-mode, bypasses the triage code. But SSE still has an extra mode change, so 10s-100s of cycles?
  - **Out of time**, continue this discussion

Action items

- [ ]