Dec 8, 2022 | RISC-V Control Transfer History TG Meeting

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

Notes

- **Attendees**: StasN, Bruce, Beeman, RobertC, AtishP, AllenB, Greg, Deepak, JohnS
- **Slides/video** here
- Charter review
  - Reviewed current charter on github, as well as edits suggested by Snehasish
  - Snehasish: leave the details out of the charter
  - Bruce: no requirement that HW sample transfer history
    - Beeman: that sentence is referring to SW collecting transfer history with samples
    - Robert: HW doesn't sample PC
  - Allen: best to keep charter general, talk about what you want to do not what to do
  - Atish: may want to remove prototype statement, normally req'd but sometimes there are waivers
    - Robert: and may use spike
  - Greg: okay to remove from charter, but one of first tasks should be to make a plan, for prototype, SAIL, etc
  - StasN: what about perf for call-stack reconstruction?
    - Beeman: wasn't sure we wanted to make that a requirement
    - Robert: could include Linux perf, but don't need to say call-stack
  - StasN: interested in filtering by transfer type too, agree it shouldn't be in charter but let's remove priv mode as well
  - Robert: call-depth is not circular
  - Settled on this charter revision, please respond if you have feedback or concerns:

    Profile guided optimizations depend on accurate branch profiles to guide compilation and effect performance improvement. Low overhead hardware control transfer recording has been used by tools such as AutoFDO to generate aggregate branch profiles without the drawbacks of instrumentation based profiling. Recent research on post link optimizers have also leveraged control transfer history to greatly improve optimization results. Finally, collection of control transfer history with each event sampled by a profiling tool, such as Linux perf, or a debugger, results in richer sample data.

    The Control Transfer History Task Group shall develop a RISC-V Control Transfer Records (CTR) extension that enables recording of recent non-speculative transfer history to a small, per-hart on-chip ring buffer. This mechanism will be controlled by privileged software, or a debugger, and will include support for *filtering. For each transfer recorded, the source and target address will be recorded, as well as optional metadata.*

- Competitive Landscape
    - Reviewing details of Intel LBRs, AMD BRS, ARM BRBE, Power9 BHRB, and SiFive PP, see table in slides
    - Extensions are mostly remarkably similar to each other, highlighted diffs in bold
    - Snehasish: user mode reads allows getting an understanding of context, for context-specific optimizations
        - Stack unwind is slower, 50-80ns
        - Whisper paper (MICRO22) is emulating this, by inserting insts to track context, but adds ~10% overhead
    - Snehasish: 32 entries is limiting for call-stack mode, have to fall back to walking call-stack
        - Bruce: recursive code can be a problem
    - Bruce: PP cycle timer to freeze has some randomization, can auto-reload
        - Can push PP to trace, so no need software interaction
- **Finished slide 11, will continue next time**
    - Including initial discussion of requirements vs optional/custom extensions

Action items

- [ ]