Dec 5, 2024 | RV Performance Events TG

Attendees: Beeman Strong, Dmitriy Ryabtsev, tech.meetings@riscv.org

Notes

- **Attendees**: Beeman, Hasan, Chun, Bruce, Daniel, Snehasish
- **Video** here (no slides)
- Reviewing new changes to PR #18, largely the addition of INST (instruction events)
    - Includes instruction events from prediction_*, rvv_*, and cache_* json files
    - Reviewed control transfer type breakdowns
    - Reviewed memory inst data source breakdowns
        - What constitutes remote?
            - Added non-normative guidance in spec, left up to implementation but one option is same/different NUMA node
            - Should be based on latency
                - Doesn't aim to capture every latency difference, just the "big" ones
            - Implies vendors will publish arch details, may not want to
                - The flexibility in definition should allow vendors to create whatever appearance they want
            - Should we included shared cache indication?
                - Since these events are intended to imply latency, I don't think shared is relevant
            - The flexibility here will make it hard to compare implementations
            - Are more options possible?
                - Yes
            - This will require some thought
    - Do sub-events sum to parent?
        - In some cases yes. Need to clarify which in the spec.
    - PDF of spec here
- Events to add?
    - Snehasish suggested CTR_INSERTS, to count insts/operations that update CTR, regardless of CTR configuration
        - Would be dependent on implementation of Smctr/Ssctr. Already have some events dependent on extension implementation
    - Trace can be configured to stall retirement, to avoid buffer overflow. Could count trace stall cycles.
        - Good idea
    - Cycles without retirement could be useful to identify instructions that delay retirement
        - Sampling on cycles will hit these, but will also hit insts that are used most frequently

        - Doesn’t provide any causality, just identified insts that may be good targets for optimization
        - Could also count cycles where only 1, 2, etc insts retire, for histograms
            - Gets rather microarchitectural, and hard if retire width is large
            - Probably more useful for architects than developers
            - Probably better accomplished with an event threshold
    - SiFive also counts watchpoints (inst/data addr match), to determine how many times an inst at a given PC retires
    - ALU/etc utilization is useful
        - Very microarchitectural, not sure we want to try to standardize such events
        - But probably wise to give guidance that such custom events are useful for core bound breakdown

Action items

- [ ] rdb197@gmail.com - Aug 8, 2024 - Include Bfloat16
- [ ] rdb197@gmail.com - Aug 8, 2024 - Check on single/double vs 32/64 terminology for FP events
- [x] ~~rdb197@gmail.com - Aug 8, 2024 - Consider adding events for VSETVL, div/sqrt, etc~~
    - Added in PR #18
- [ ] Beeman Strong - May 23, 2024 - check on idea to count remote HITMs locally
- [ ] Beeman Strong - Dec 5, 2024 - Clarify when sub-events sum to parent