

## Aug 31, 2023 | 📅 RISC-V Control Transfer Records TG Meeting

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

### Notes

- **Attendees:** JohnS, Beeman, Atish, RobertC, BruceA, Snehasish
- No slides or video, just discussing latest spec [here](#)
- Snehasish: have just some cosmetic comments, will share in the coming days
- Bruce: Does WPRI make sense for the reserved for custom bits?
  - Beeman: I think so, that that SW that is not aware of custom extensions preserves them
  - Robert: would be better to split the field, so custom bits are separate
  - Would apply for other regs with bits rsvd for custom as well
  - Beeman to check with ARC about how to handle these bits
- Bruce: should be a control bit that enables trigger trace actions, optional
  - Bruce: goal is to enable trace on/off based on \*context match
  - Long discussion ensued
  - Would need at least another bit that indicates that last trigger action was “trace on”, so if we get context switched out and back we resume recording
  - Presume then that priv mode bits are all 0, so tracing is off by default? But then means trace on action enables tracing in all priv modes
  - Would have to configure some trigger to always match (e.g., mcontrol6 with execute=1 and entire address masked off), along with context filtering
    - Then could trigger “trace on” action on first instruction in the context (and every inst following, but no effect from that)
    - What would trigger “trace off” when leaving context?
  - Bruce & Robert to write up a full proposal
- Bruce: could have cycle count mode that switches between 16b binary cycle counter vs CCE+CCM
  - Benefit is precision at higher values
  - Existing spec allows for precision up to 8192 cycles, do we really need precision beyond that
    - Expect more value in differentiating between 80K and 1M, at reduced precision, than precision up to 64K but no way to know order of magnitude of latency beyond that
  - Adding another mode does add some complexity, HW needs to support & validate both, SW needs a branch to choose the right handler function, etc
  - Minimal implementation can implement simple  $\leq 13$ -bit binary counter (up to 8192)
  - Agreed that the benefit is small
- Robert: would be nice to have bit number(s) in the table with fields
  - Beeman: just following priv spec method
- Bruce: does setting CLR require writing 0 to each entry?

- Beeman: thinking HW will just have an internal valid bit per entry, such that reads return 0 when it is cleared. Then just clear those bits on CLR.
  - Beeman to add non-normative to describe
- Beeman to add non-normative text that \*iselect is logical and WRPTR is physical
- Robert: does WPRI require RMW? Or should those bits just be 0
  - Ran out of time, but this likely falls under the topic up top about WPRI. Beeman to ask.

#### Action items

