

Jun 29, 2023 | 📅 RISC-V Control Transfer Records TG Meeting

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

Notes

- **Attendees:** DavidW, RobertC, Snehasish, JohnS, Beeman, BruceA, GregF
- **Slides/video** [here](#)
- **Updates**
 - New [spec v0.1.1 released](#)
 - Reviewed updates to mctrstatus fields
 - Robert: suggest changing TOS to WRPTR, since not really a stack
 - Agreed
 - Bruce: does CLR require writing 0s to each entry?
 - Beeman: I imagine a uarch where there is a depth-width bit-vector, such that, for each entry, if the bit is 1 then reads return the array contents, but if the bit is 0 reads return 0. CLR then just 0s this bit-vector.
 - Discussion ensued on whether each sample needs to clear CTR, or have other indication of which entries are new vs the last sample.
 - Beeman argues that this isn't needed, that if some entries are shared between two samples that's fine because ultimately the tools just want to know the code path leading to the sample. Doesn't matter if it's similar to the last one.
 - Not sure if everyone was in agreement, can discuss via the email list
 - Performance event
 - Google user suggested that it may be nice to be able to precisely sample on a number of CTR entries
 - Not clear to the group how to use this
 - And RISC-V doesn't have a standardized precise sampling mechanism anyway
 - May include some non-normative text suggesting an event, but nothing stronger than that
- **Recording not-taken branches**
 - Could make it a non-default option. Mctrcontrol[36] aligns with that transfer type, but ROZ currently. Could make it an opt-in.
 - Not useful for PGO, but may be useful for collecting metadata on not-takens?
 - Snehasish: no concern as long as not on by default. autofdo/propeller convert a lot of takens to NT, so this would fill the CTR with useless information
 - Snehasish: could an implementation end up slowing down more to support this?
 - Agree that it would be very undesirable to slow down the default mode, by further restricting retirement to limit NT branches, in order to support this optional bit
 - Will add non-normative text about this

- And that there should be no slowdown when CTR is not enabled
- Recording transfer insts missed by the frontend
 - Lots of discussion trying to just explain the case, which feels like a good indication that this is too confusing to standardize 😊
 - Would not impact which transfer are recorded, would simply be a new ctrdata bit that indicates that this instruction incurred a uarch restart
 - Have custom bits that can be used for this, for implementations that believe this is interesting
- Recording priv mode
 - Greg: Couldn't expose V bit to a VM, would be a virtualization hole
 - Snehasish: PGO/FDO don't need this, used for one context at a time
 - Discussion ensued over value of adding more information for which we don't currently have a usage model
 - Beeman: the info is available but it's not free to add, uses 3 bits that we couldn't use for something else in the future. Also widens the storage array.
 - Snehasish: used ETM for autofdo, and the extra information triggered extra security review. So adding info is not free.
 - For debug, useful to know priv mode
 - Beeman: hesitant to add features for debug usage, given that trace already addresses that (better)
 - CTR could be used as a cheaper trace
- Out of time
 - As always, please email the list if you believe any of the topics we discussed above were not concluded to your satisfaction.

Action items

