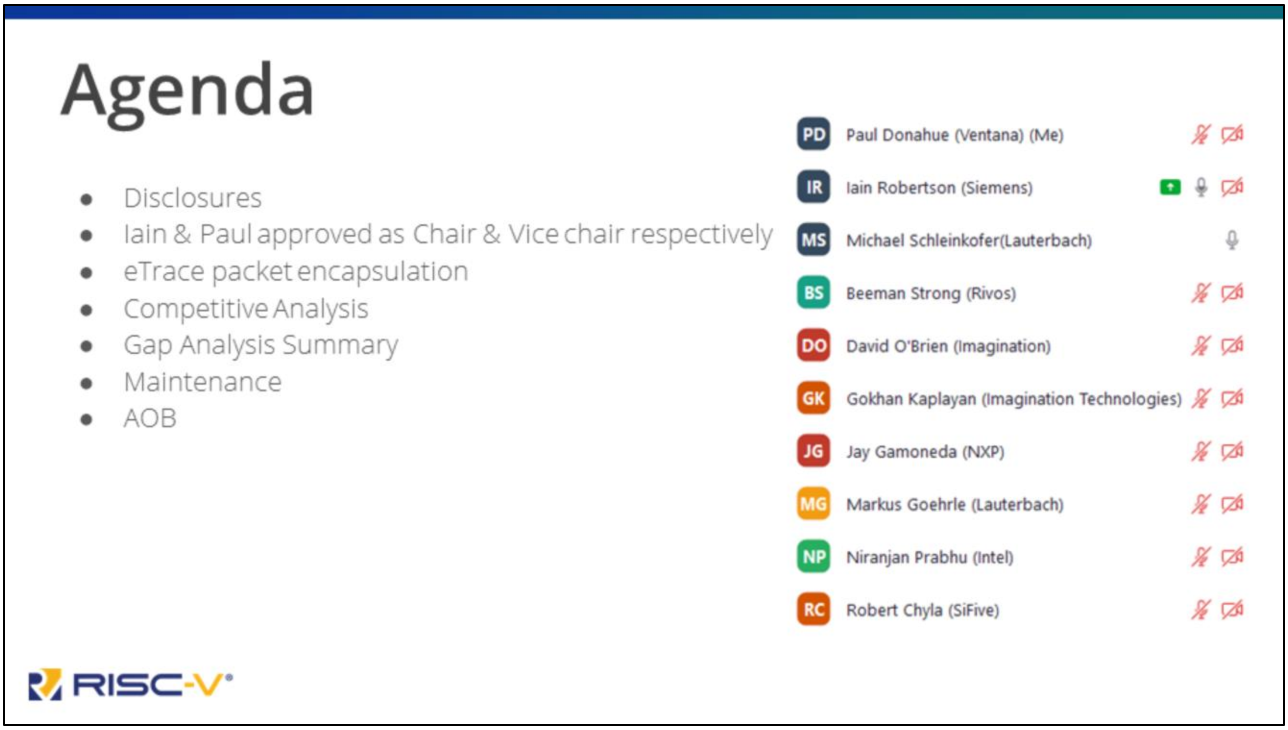
RISC-V®

DTPM SIG

04/12/2023

Meeting minutes are in the speaker notes for the relevant slide



Agenda

- Disclosures
- Iain & Paul approved as Chair & Vice chair respectively
- eTrace packet encapsulation
- Competitive Analysis
- Gap Analysis Summary
- Maintenance
- AOB

PD Paul Donahue (Ventana) (Me)

IR Iain Robertson (Siemens)

MS Michael Schleinkofer(Lauterbach)

BS Beeman Strong (Rivos)

DO David O'Brien (Imagination)

GK Gokhan Kaplayan (Imagination Technologies)

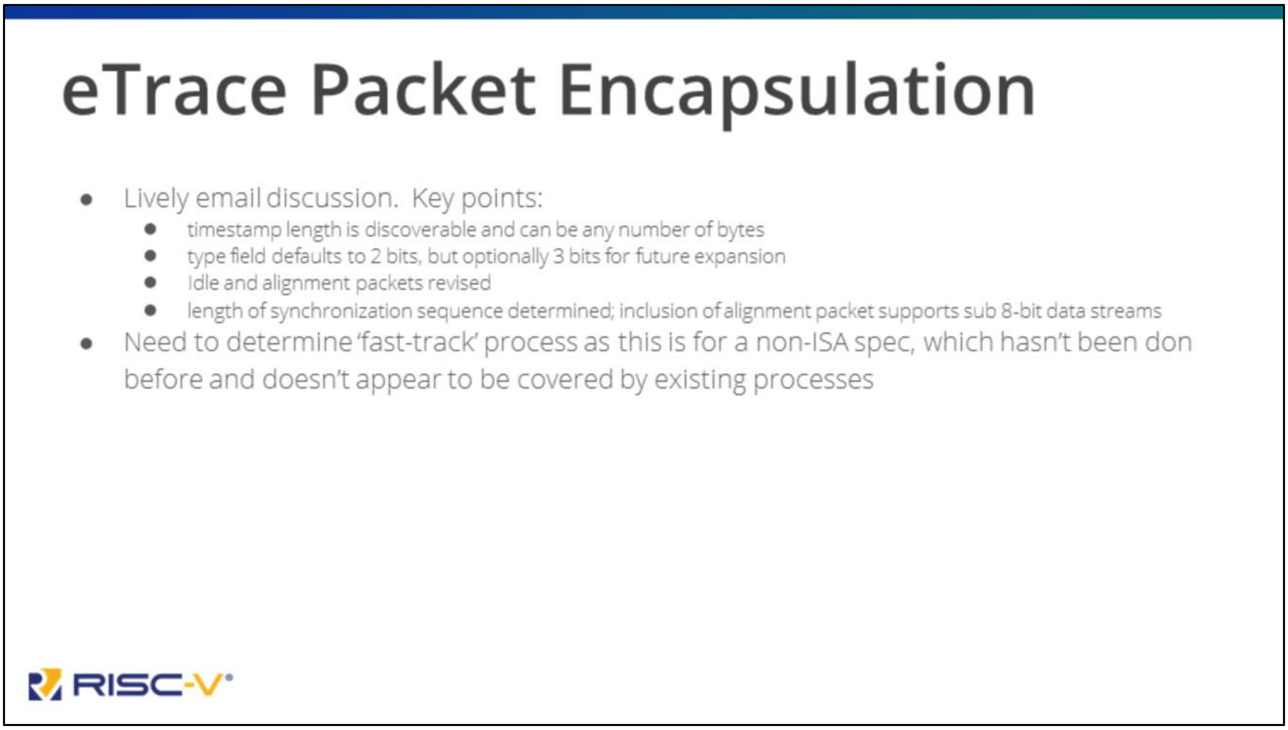
JG Jay Gamoneda (NXP)

MG Markus Goehrle (Lauterbach)

NP Niranjan Prabhu (Intel)

RC Robert Chyla (SiFive)

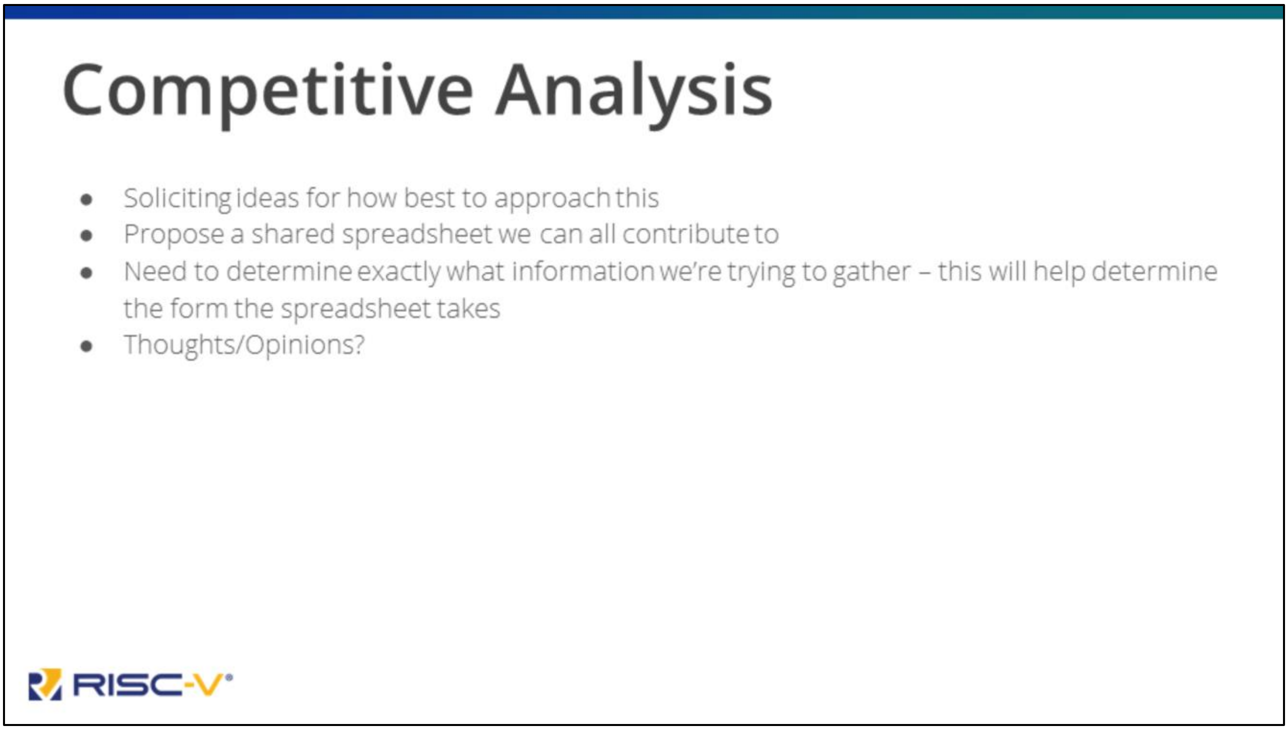
Attendees: see screenshot on the agenda slide



eTrace Packet Encapsulation

- Lively email discussion. Key points:
  - timestamp length is discoverable and can be any number of bytes
  - type field defaults to 2 bits, but optionally 3 bits for future expansion
  - Idle and alignment packets revised
  - length of synchronization sequence determined; inclusion of alignment packet supports sub 8-bit data streams
- Need to determine 'fast-track' process as this is for a non-ISA spec, which hasn't been don before and doesn't appear to be covered by existing processes

- There is support for integrating this as a chapter in the existing spec rather than having it as a separate document.
- Presentation of changes to the proposal since last meeting.
- Idle pattern: repeated 0x80 vs. repeated 0x00.
  - Michael discussed his original concerns
  - Robert: sequence of 0 makes more sense from the probe side. Also better for trace to memory.
  - Consensus: Original proposal of a sequence of 0 is preferred with a 0x80 at the end. It may result in an extra idle byte in practice. But the encoder will generally be able to change the type of the final byte rather than inserting an extra one, depending on the implementation.
  - We will stick with the original proposal of repeated 0x00 as the idle sequence.
- Iain will check with Jeff about the process for fast track

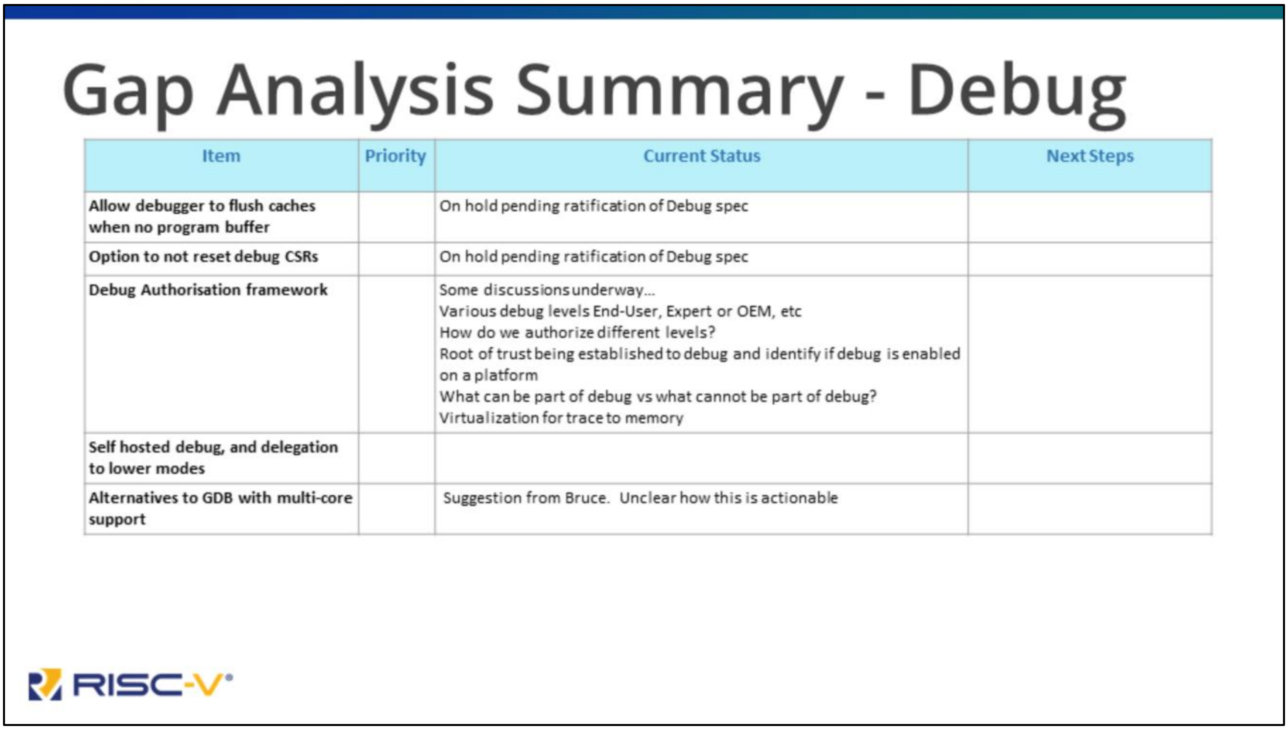


Competitive Analysis

- Soliciting ideas for how best to approach this
- Propose a shared spreadsheet we can all contribute to
- Need to determine exactly what information we're trying to gather – this will help determine the form the spreadsheet takes
- Thoughts/Opinions?

- Niranjan: Has a link (to a document that's not a spreadsheet) that he can share to get started. What is the end goal?
- Robert: Should give users an overview of what they would be gaining or losing by migrating to RISC-V. It's about all trace technologies, not just PC trace.
- Iain: Good idea. Also can be used to determine strengths and weaknesses to prioritize areas to recommend plugging gaps.
- Beeman also has a starting point.
- Niranjan and Beeman will share their documents and then we can massage it into a spreadsheet. Iain will ask Jeff about where we can put things on Google Drive (update – see link below).
  - Niranjan: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/secure-coding/intel-debug-technology.html
  - Beeman: Competitive analysis for Perf Analysis SIG: https://docs.google.com/spreadsheets/d/1NtWr70UC9GfW3kOP9tGJYA34ecHxySXW_EQ3RsfMR18/edit?usp=share_link

- The DTPM SIG's workgroup folder on the RVI Google Drive is at for risc-v members/Workgroups/Debug Trace Performance Monitoring: https://drive.google.com/drive/folders/1DBShFvr4Zkitl8QbyRjolL0GS7QI7LYG?usp=share_link



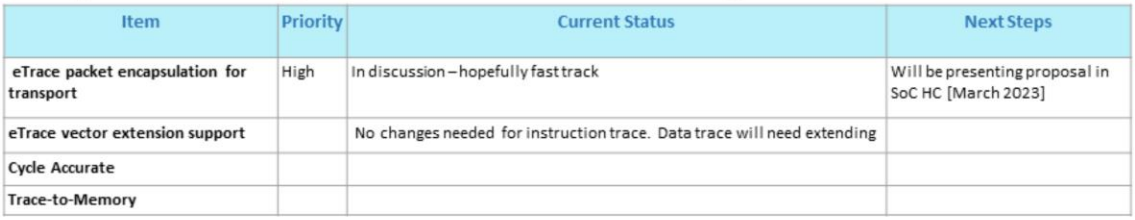
Gap Analysis Summary - Debug

Item	Priority	Current Status	Next Steps
Allow debugger to flush caches when no program buffer		On hold pending ratification of Debug spec	
Option to not reset debug CSRs		On hold pending ratification of Debug spec	
Debug Authorisation framework		Some discussions underway… Various debug levels End-User, Expert or OEM, etc How do we authorize different levels? Root of trust being established to debug and identify if debug is enabled on a platform What can be part of debug vs what cannot be part of debug? Virtualization for trace to memory	
Self hosted debug, and delegation to lower modes			
Alternatives to GDB with multi-core support		Suggestion from Bruce. Unclear how this is actionable	

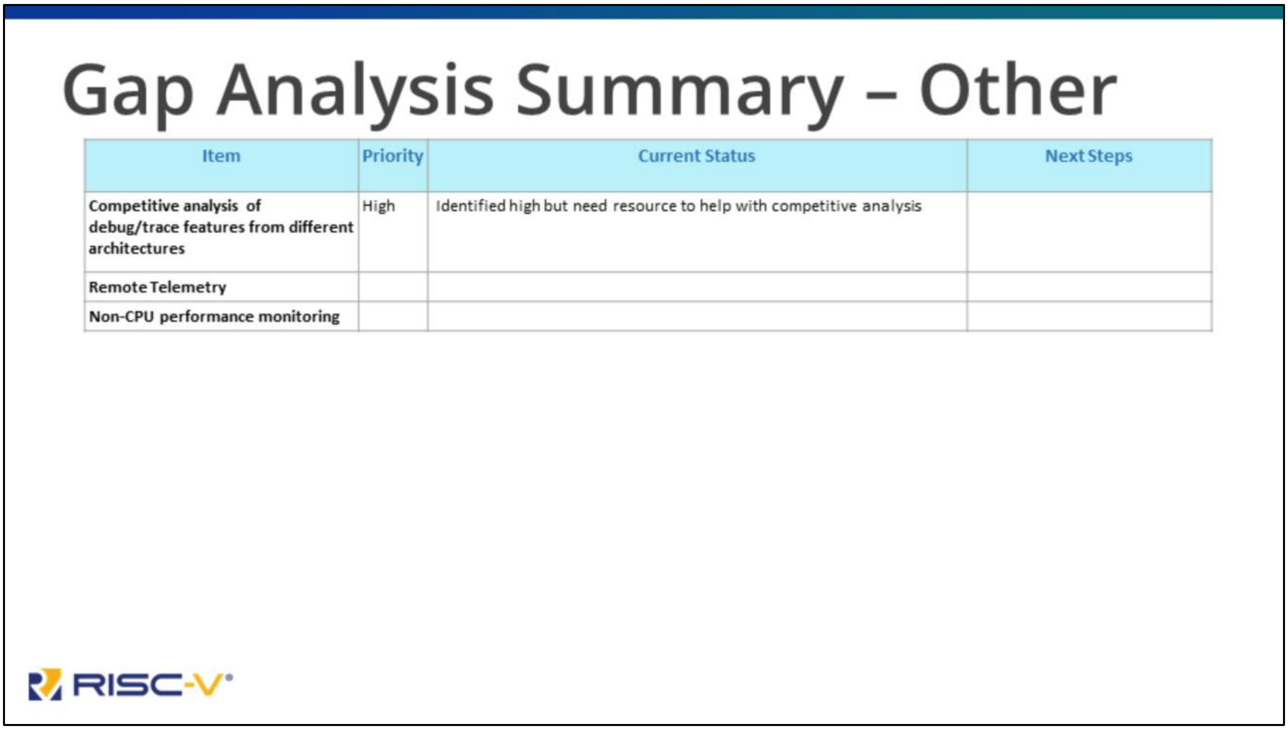
RISC-V®

- Debug spec gap analysis:
  - Gokhan: What's the timeline for having the debug spec contemplate how to flush the instruction cache from an external debugger when the program buffer is not implemented?
  - Paul: There is a github issue on the debug spec. It was deferred until the next revision of the spec (after 1.0 which will effectively freeze on Friday). Other implementations have used custom features for this.

Gap Analysis Summary - Trace

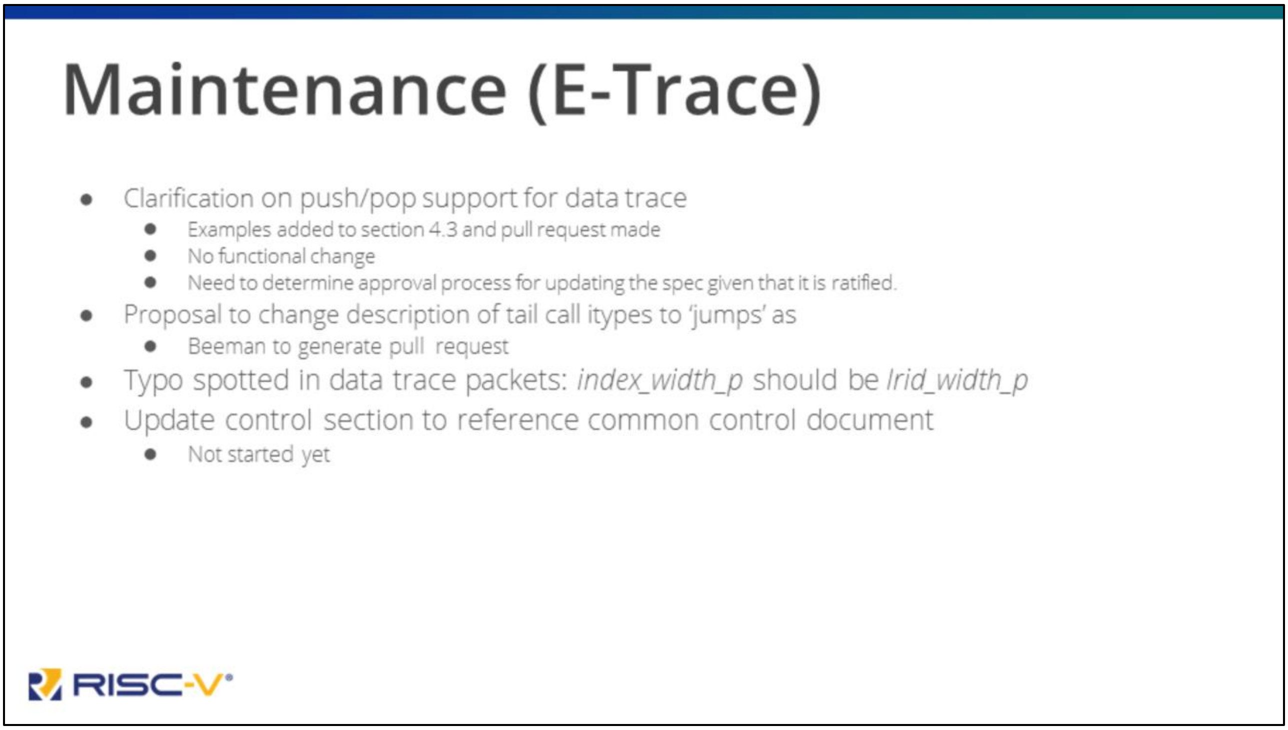


Item	Priority	Current Status	Next Steps
eTrace packet encapsulation for transport	High	In discussion – hopefully fast track	Will be presenting proposal in SoC HC [March 2023]
eTrace vector extension support		No changes needed for instruction trace. Data trace will need extending	
Cycle Accurate			
Trace-to-Memory			



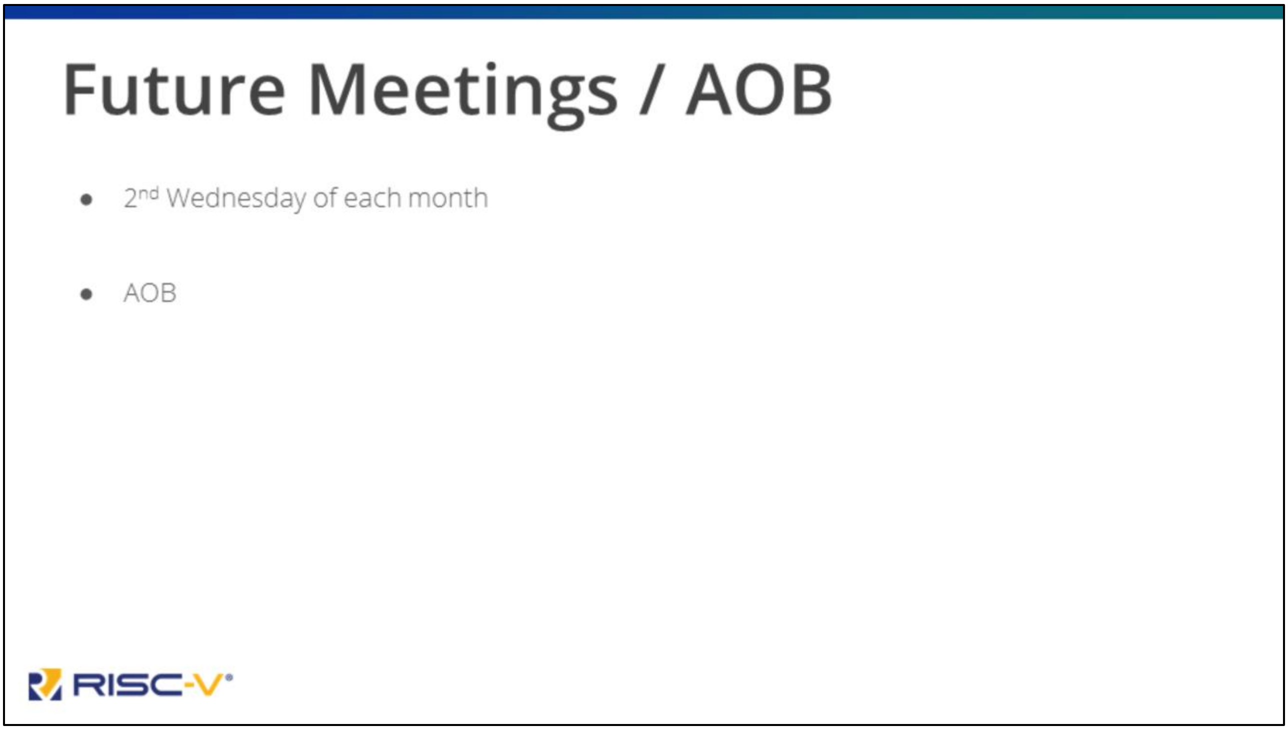
Gap Analysis Summary – Other

Item	Priority	Current Status	Next Steps
Competitive analysis of debug/trace features from different architectures	High	Identified high but need resource to help with competitive analysis	
Remote Telemetry			
Non-CPU performance monitoring			



Maintenance (E-Trace)

- Clarification on push/pop support for data trace
  - Examples added to section 4.3 and pull request made
  - No functional change
  - Need to determine approval process for updating the spec given that it is ratified.
- Proposal to change description of tail call itypes to 'jumps' as
  - Beeman to generate pull request
- Typo spotted in data trace packets: *index_width_p* should be *lrid_width_p*
- Update control section to reference common control document
  - Not started yet



Future Meetings / AOB

- 2nd Wednesday of each month
- AOB

- Timestamps:
  - Michael: Control of timestamps within encapsulation. Do these timestamps replace the timestamp in format 3 packets?
  - Iain: During the original E-Trace work, someone asked for the timestamp to be included. It's optional. So if you use timestamp in the encapsulation then you probably don't want to include it in format 3 packets. The encoder can then control which packets have timestamps rather than only allowing them in format 3 packets.
  - Some packets always come close together in time so you don't need time on each of them.
  - Beeman was envisioning that format 3 could include full timestamp and encapsulation has deltas.
  - Timestamps are "a Pandora's box with a can of worms inside."

Thank You

RISC-V®