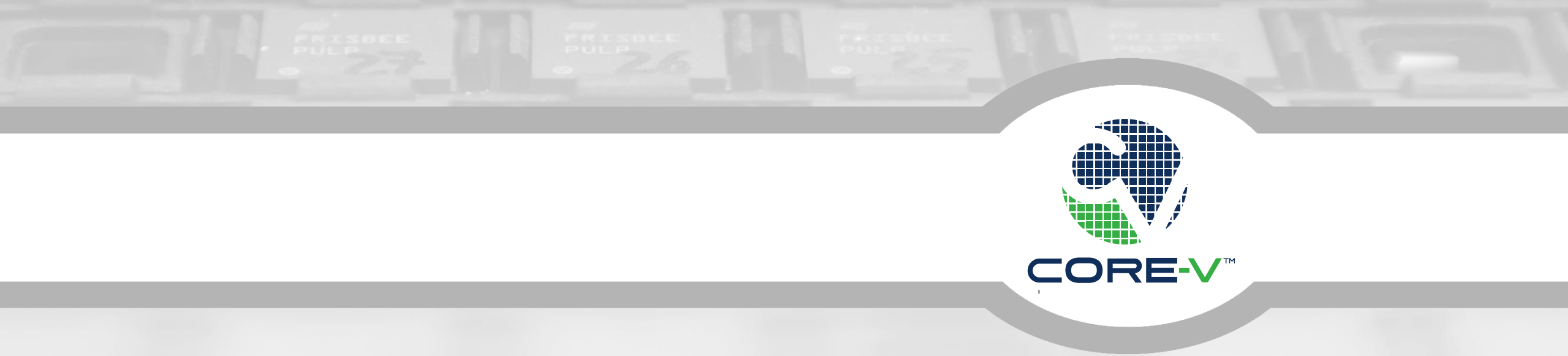


CORES TG – October 2 2020

Arjan Bink

Jérôme Quevremont

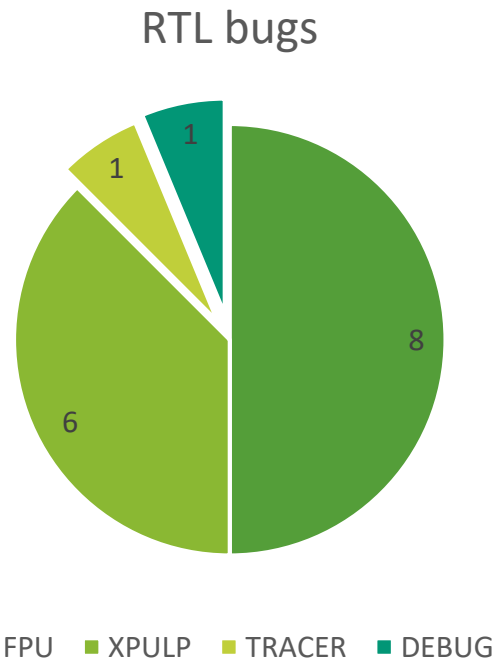
Davide Schiavone

Agenda

- CV32E40P status
- CV32E40P Roadmap
- CV32A6 Preliminary Project Launch (PPL)

CV32E40P issues

https://github.com/openhwgroup/cv32e40p/issues filter	#	Comment
is:issue is:open	48	Open
is:issue is:open -label:status:resolved	47	Non-closed
is:issue is:open -label:status:resolved label:component:rtl label:type:bug -label:waived:cv32e40p	16	RTL bugs



- 2 bugs related to RTL freeze target configuration
  - https://github.com/openhwgroup/cv32e40p/issues/529
  - https://github.com/openhwgroup/cv32e40p/issues/329
    - Likely a 'tracer' (non-RTL) issue; maybe already fixed; need help from issuer to get this closed
- RTL freeze for default params targeted this month

```
module cv32e40p_core import cv32e40p_apu_core_pkg::*;
#(
  parameter PULP_XPULP          =  0,
  parameter PULP_CLUSTER        =  0,
  parameter FPU                 =  0,
  parameter PULP_ZFINX          =  0,
  parameter NUM_MHPMCOUNTERS    =  1
)
```

CV32E40P RoadMap (from November on)

- SW TG proposed a Preliminary Project Proposal for supporting PULP extensions in the GNU Compiler toolchain
  - https://github.com/jeremybennett/core-v-docs/blob/jpb-gnu-tools-ppl/program/core-v-gnu-tools-ppl.md

We need to make Decisions →

CV32E40P RoadMap: Decision 1

- Change the encoding of CV32E40P of instructions that will conflict with RISC-V
  - See issue https://github.com/openhwgroup/cv32e40p/issues/452)
- If not done, then CV32E40P
  - won't have upstream, industry-standard GCC compiler support
  - will require the PULP compiler to be used for PULP instructionss

CV32E40P RoadMap: Decision 2

- PULP Extensions need to be verified
  - This decision depends on Decision 1, as verification needs to know what compiler and encoding we need to use
  - Reference model needed
    - Extending Imperas ISS?
    - Extending Spike?

CV32E40P RoadMap: Decision 3

- RVF Extensions need to be verified
  - The core implements the RVF extensions (optionally)
  - The HW needs to be verified
    - Depends on the external module, i.e., FPNEW from PULP Platform
      - Shall we fork it (should OpenHW own this)?
  - The Interface towards the FPU needs to be documented and/or re-designed
    - This work can be generalized for any external accelerator, thus used also by CV32E40. Or we can postpone it to CV32E40

CV32E40P RoadMap: Decision 4

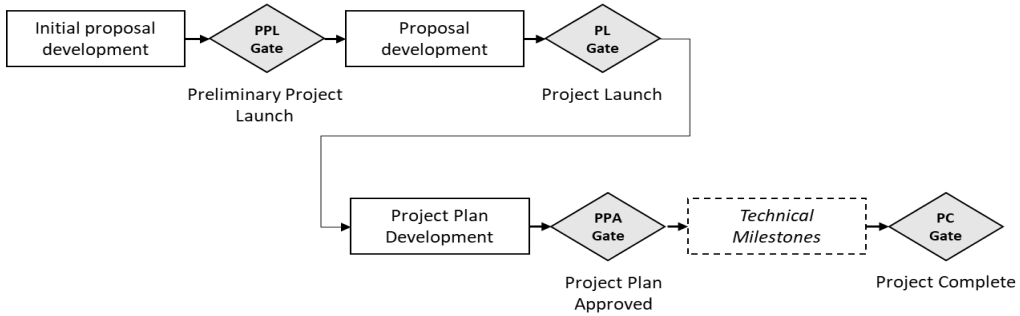
- Instruction and Data Bus-Error pins
  - Design this extra feature to handle bus errors

CV32E40 RoadMap: Extra Features

- Security
  - The PMP and user-mode will be added to the core
- CLIC
  - Change interrupt scheme to CLIC
- Accelerator Interface
  - Based on Decision 3
- RVB Extensions
  - When we are going to start with this project, we need to understand whether we want them or not (same as RVP, but less likely than RVB)

CVA6

- CV32A6 (32-bit version)
  - INVIA (Thales) contribution committed since 2020-09-11 (https://github.com/openhwgroup/cva6), no FPU MMU yet
  - LinkedIn communication a bit delayed, expected next week



- CVA6 PPL accepted by Technical WG 2020-09-28
  - Need to recruit project manager and verification team
  - Anticipate documentation and Vplan before next gates
  - Document here

Thank you!