CORE-V Cores Roadmap

Davide Schiavone	**davide@openhwgroup.org**
Jérôme Quévremont	**jerome.quevremont@thalesgroup.com**
Arjan Bink	**arjan.bink@silabs.com**

OPENHW GROUP™
PROVEN PROCESSOR IP

CORE-V Family History

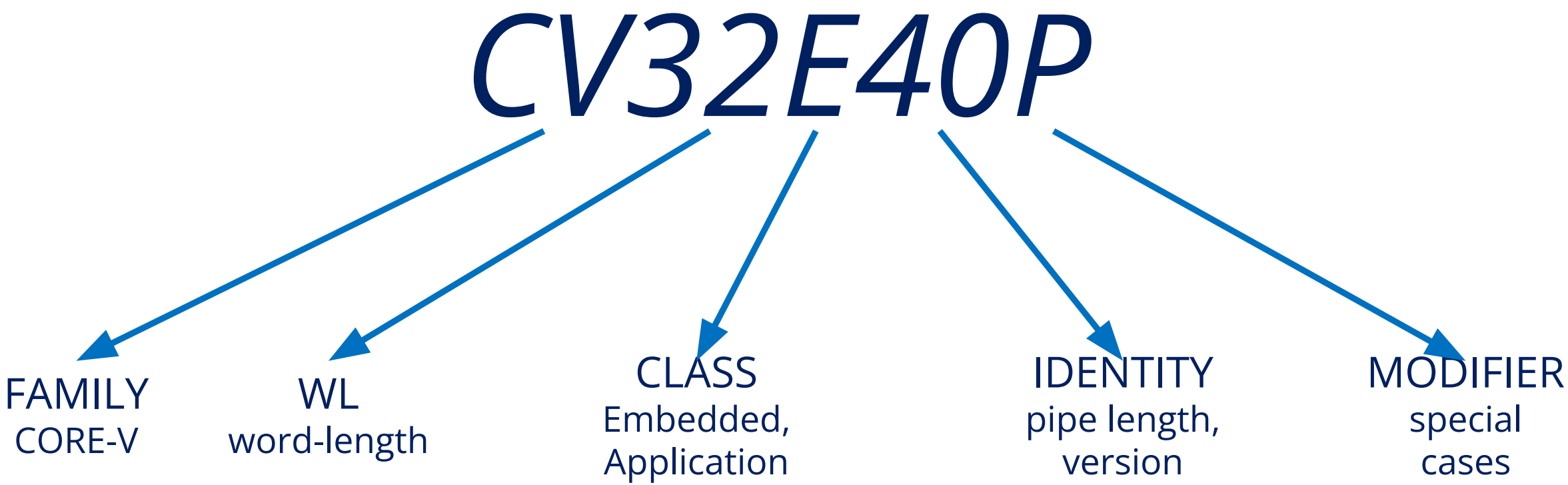
- Initial contribution of open-source RISC-V cores from ETH Zurich PULP Platform and the OpenHW Group is the official committer for these repositories
- OpenHW Cores Task Group has the mandate to develop feature and functionality roadmap for the CORE-V Family of open-source RISC-V processors
  - Chair: Arjan Bink, Silicon Laboratories
  - Vice-Chair: Jérôme Quévremont, Thales Research & Technology

Core	Bits/Stages	Description
CV32E40P (RI5CY)	32bit / 4-stage	A family of 4-stage cores that implement, RV32IMFCXpulp, optional 32-bit FPU, instruction set extensions for DSP operations including HW loops, SIMD extensions, bit manipulation and post-increment instructions.
CVA6 (Ariane)	32 & 64bit / 6-stage	A family of 6-stage, single issue, in-order CPU cores implementing RV64GC extensions with three privilege levels M, S, U to fully support a Unix-like (Linux, BSD, etc.) operating system. The cores have configurable size, separate TLBs, a hardware PTW and branch-prediction (branch target buffer, branch history table and a return address stack).

CORE-V P/Ns, Gates, TRLs

Reference Material

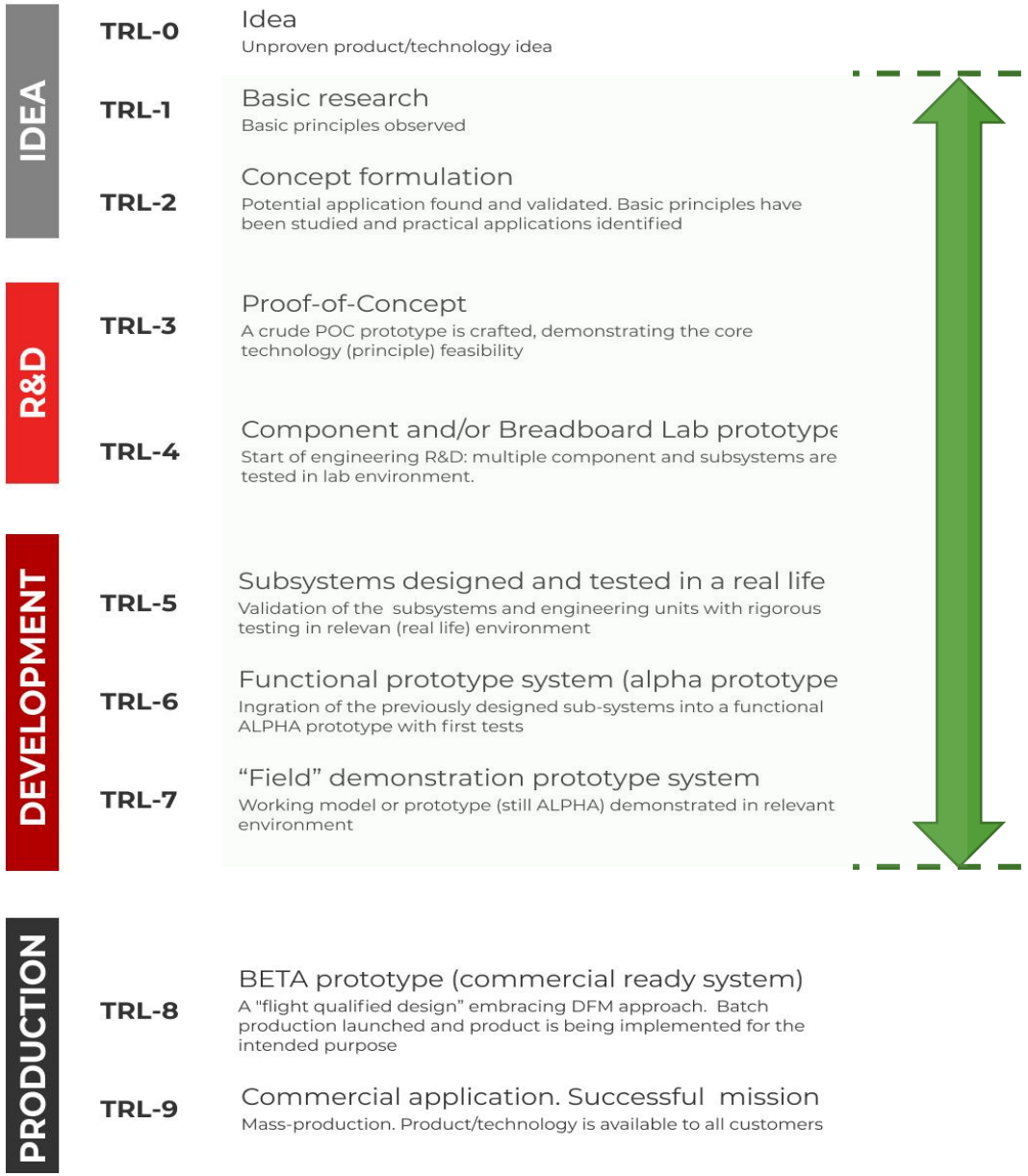
CORE-V Cores P/N Syntax



OpenHW Project Framework

Gate	Purpose	Criteria
PC Project Concept	Green-light of project concept by TWG	Proposed scope, initial view of the components and features, why do this project?
PL Project Launch	Full project launch approval by TWG	Outline of the requirements, features, components, project supporters, risks, high level schedule
PA Plan Approved	Communicate project plan to TWG, (allowing member participation and review	Project plan full checklist, project methodology. initial agile backlog, requirements specification
Project work		
PF Project Freeze	Completion of releasable project content	RTL Freeze checklist or other final checklist has been completed

TRL Scale



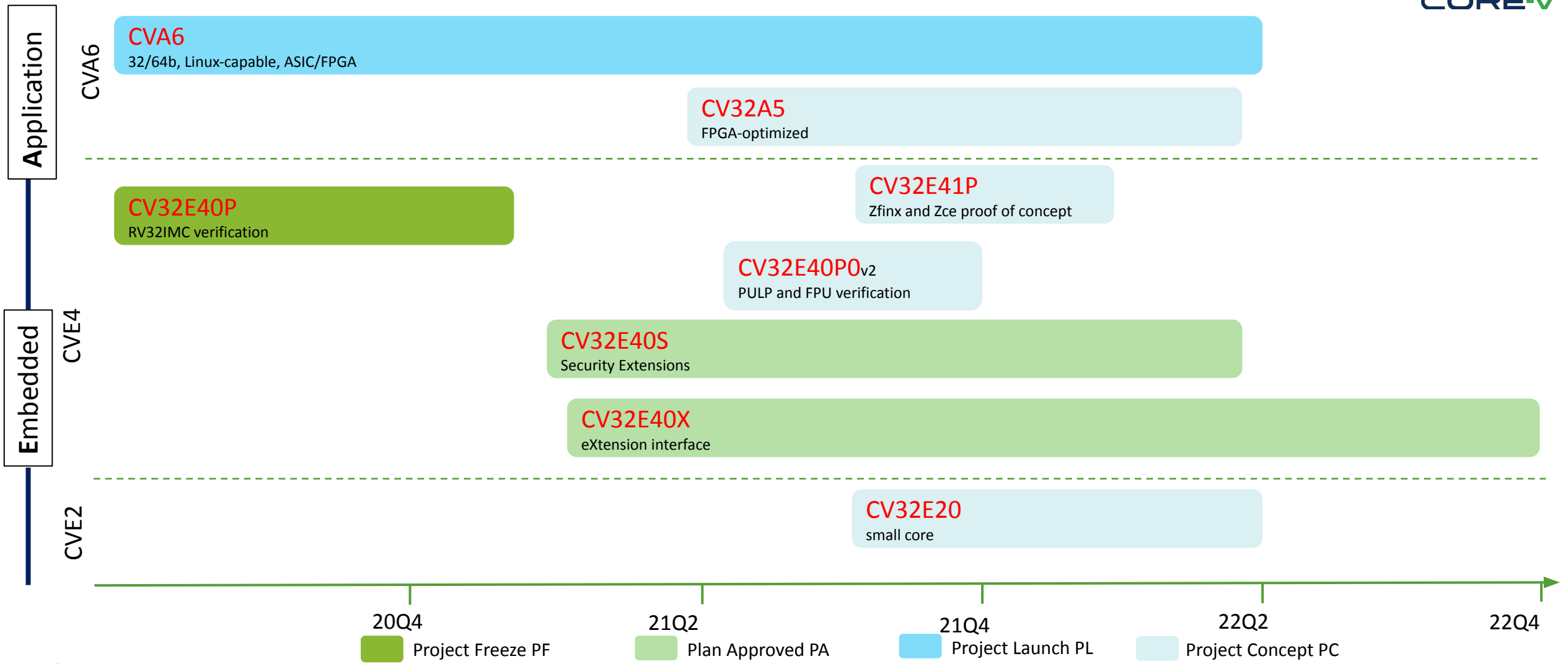
OpenHW Technology Outputs

OpenHW IP Adopters

TRL Levels as Utilized by OpenHW

TRL	OpenHW Utilization
1-Basic Principles Observed	•**OpenHW research projects** may target TRL-1 as project output, e.g. to develop novel approaches to core or accelerator architecture
2-Concept Formulation	•Core IP or accelerator development projects are typically initiated as TRL-2 concepts, identifying principles and applications of the IP •The **OpenHW Project Concept Gate** output includes a TRL-2 description of the Core IP
3- Proof of Concept	•Core IP or accelerator development projects will pass through TRL-3 **as the (RTL) design completes**. •Proof of concept is shown by core compilation and demonstration of basic operations (e.g. Linux booted, coremark results, hello-world)
4- Component Prototype	•Core IP or accelerator projects will pass through TRL-4 as they produce **preliminary PPA results** (via synthesis scripts for FPGA or ASIC) and/or **run preliminary application code**, such as an accelerator running machine learning code.
5- Subsystem Designed and Tested	•Core IP projects reach TRL-5 as they **complete full verification**. The OpenHW RTL Freeze checklist process verifies that the design is fully ready for industrial adoption.
6-Functional (Alpha) Prototype	•**OpenHW IP that is integrated into an MCU system** or other device reaches TRL-6 as prototype Silicon is **fabricated and demonstrated on a development board or other platform**.
7-Field Demonstration Prototype	•**OpenHW IP that is integrated into an MCU system** or other device reaches TRL-7 as prototype Silicon is fabricated, **deployed and demonstrated in the field**.

CORE-V™ Cores Roadmap



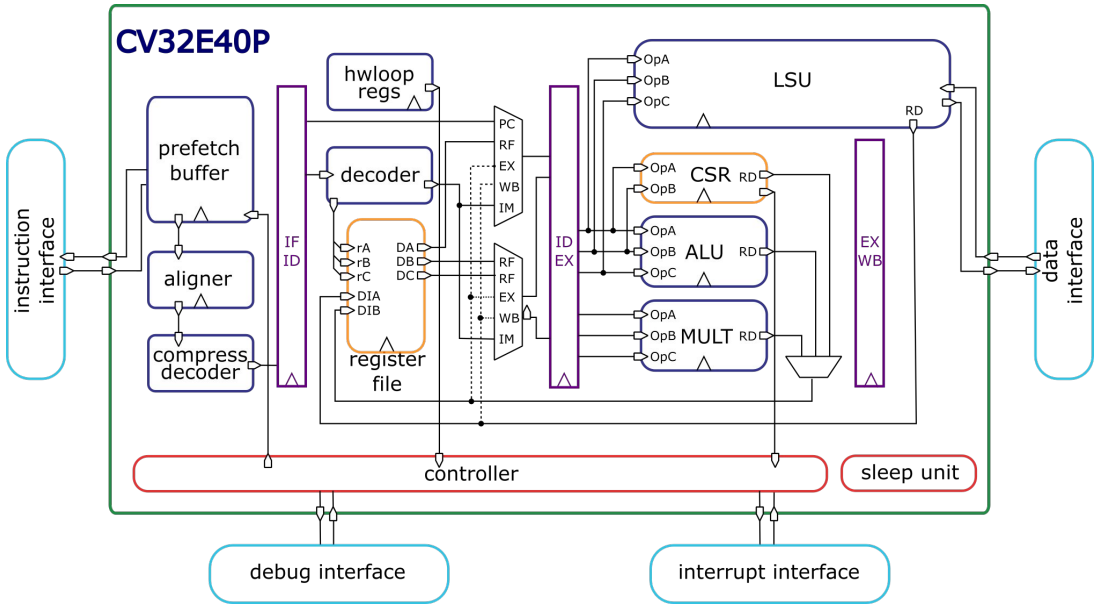
CORE-V Embedded-class Cores

CV32E40P – PF / TRL 5

- 4-stage, in-order, single-issue
- RV32IM[F]CZicount_Zicsr_Zifencei [PULP_XPULP][PULP_CLUSTER][PULP _ZFINX]
- M-mode, CLINT, OBI

- ‘RTL Freeze’ achieved
  - RV32IMC extensions verified
  - Step&Compare with Imperas as reference model (100% coverage)
  - Interrupts and Debug

- Activities on RVFI interface
  - Facilitating sim-based step&compare verification FSM and formal verification



CV32E40P – preliminary PPA

- **CV32E40P**
  - COREMARK/MHz: ***2,91***

Technology	Frequency	Area	Comments
TSMC 16nm	909MHz	53,4kGE	No FPU

CORE-V™ MCU Tapeout 2nd half 2021

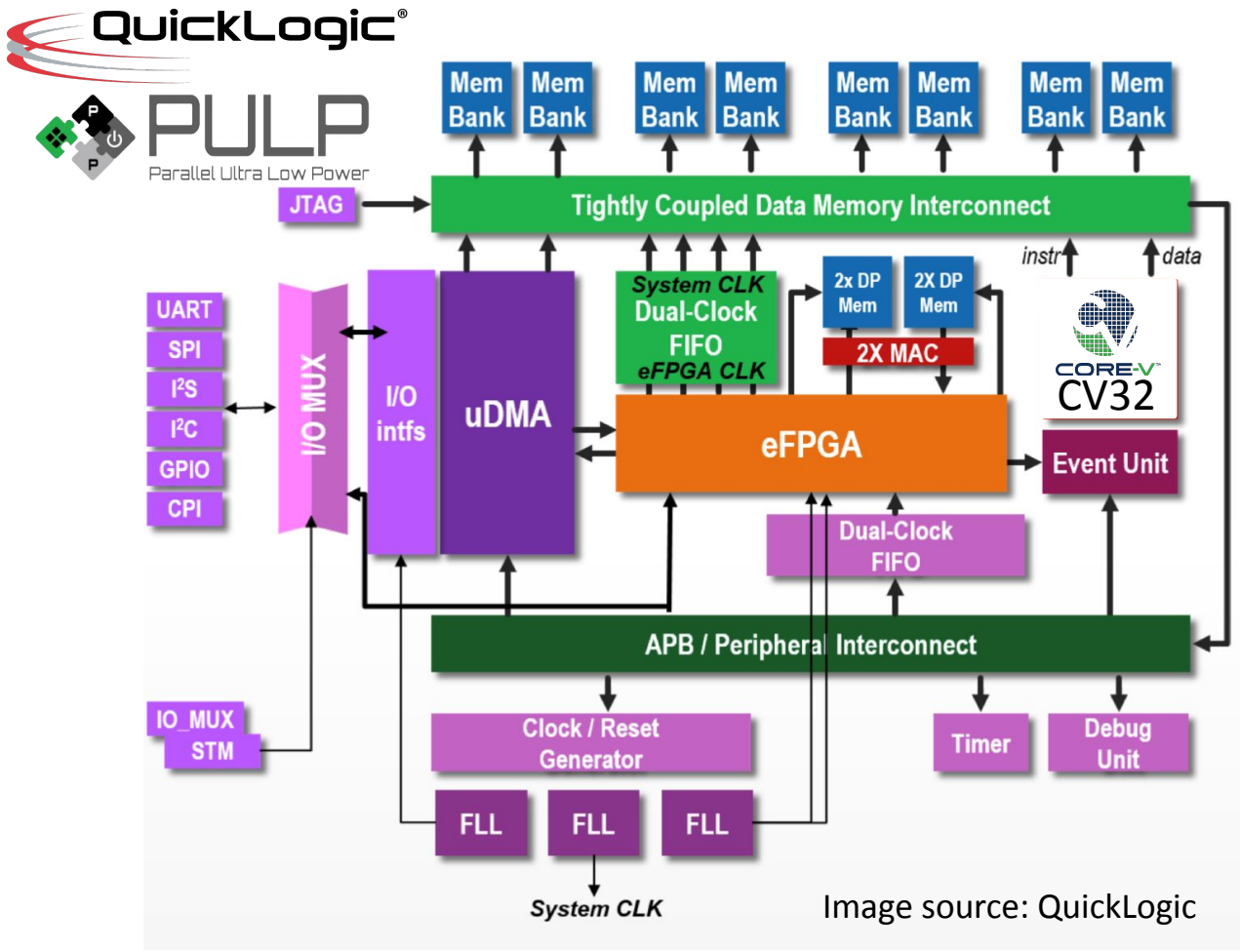
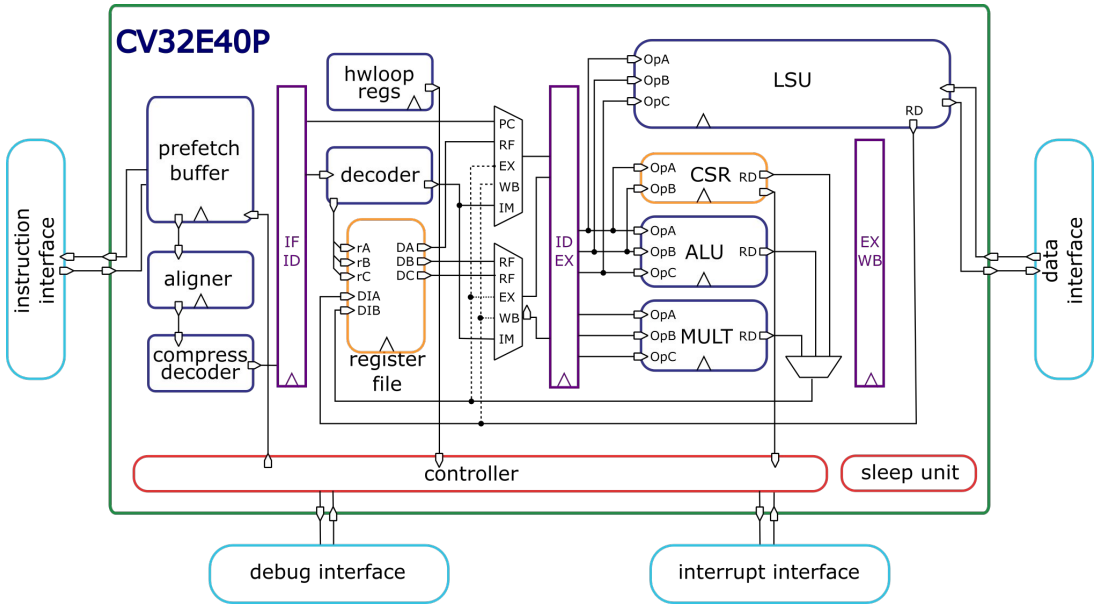


Image source: QuickLogic

- Real Time Operating System (e.g. FreeRTOS) capable ~600+MHz CV32E4 MCU
- Embedded FPGA fabric with hardware accelerators from QuickLogic
- Multiple low power peripheral interfaces (SPI, GPIO, I2C, HyperRAM, CAMIF, etc) for interfacing with sensors, displays, and connectivity modules
- Built in 22FDX with GF

CV32E40P:v2 – PL / TRL 3

- RV32PULP_XPULP extensions
  - Verification and Reference Model
  - Moving the existing instructions to the RISC-V custom space
  - SW support with upstream GCC and LLVM compiler
  - LEC to v1 when PULP_XPULP=0
- RV32F extensions
  - Verification
- CV-DBG
  - Moving, documenting, and verifying the external RISC-V Debug module
- Project goal: industrial grade (TRL 5)



CV32E40P:v2 – preliminary PPA

- **CV32E40P with PULP_XPULP**
  - COREMARK/MHz: ***3,19***

Technology	Frequency	Area	Comments
UMCL 65nm	560MHz	40,7kGE	No FPU – RI5CY based
Globalfoundies GF22FDX	938MHz*	N/A	No FPU – RI5CY based
TSMC 16nm	769MHz	84,5kGE	No FPU, 1 extra Perf. Counter
Xilinx Genesys 2	20MHz**	7kLUTs; 2,5kFFs; 7 DSP	No FPU
Zynq Ultrascale+	140MHz	8,4kLUTs; 2,7kFFs; 7 DSP	No FPU – RI5CY based

*freq in PULPissimo on SCMs and use of Forward Body-Bias **freq with FPU in PULPissimo, not max frequency

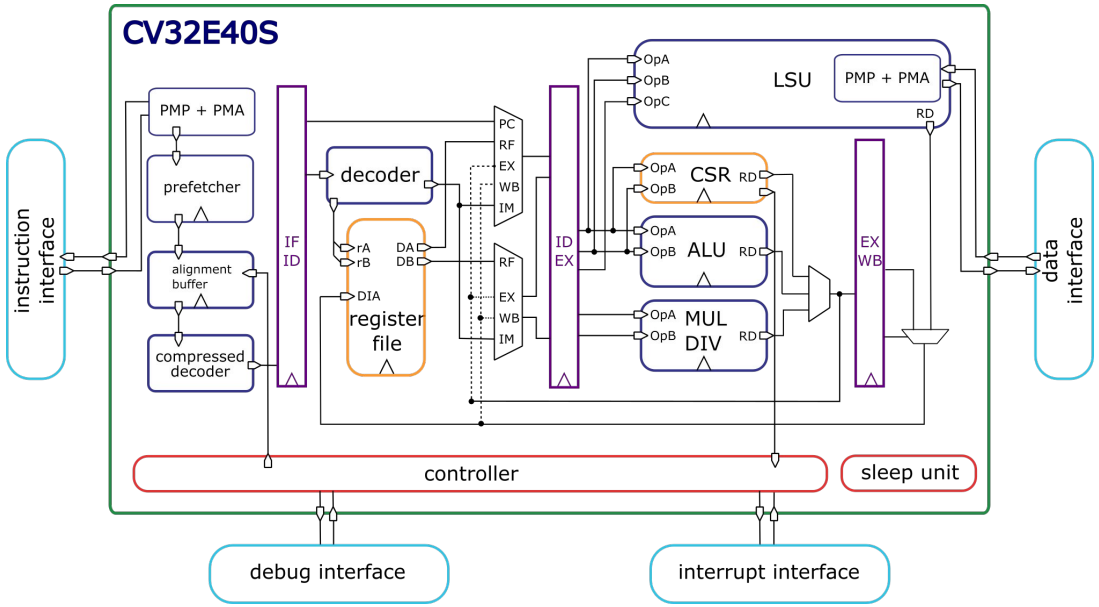
CV32E40S – PA / TRL 2

- 4-stage, in-order, single-issue
- RV32IMC[Xsecure]Zicsr_Zifencei[_Zce]
- M/U-mode, CLINT, OBI, ePMP, PMA, bus error

- Secure core
  - Reduction of side-channel attacks
  - _Zce (iff ratified)
  - PPA optimizations wrt CV32E40P

- Verification

- Project goal: industrial grade (TRL 5)



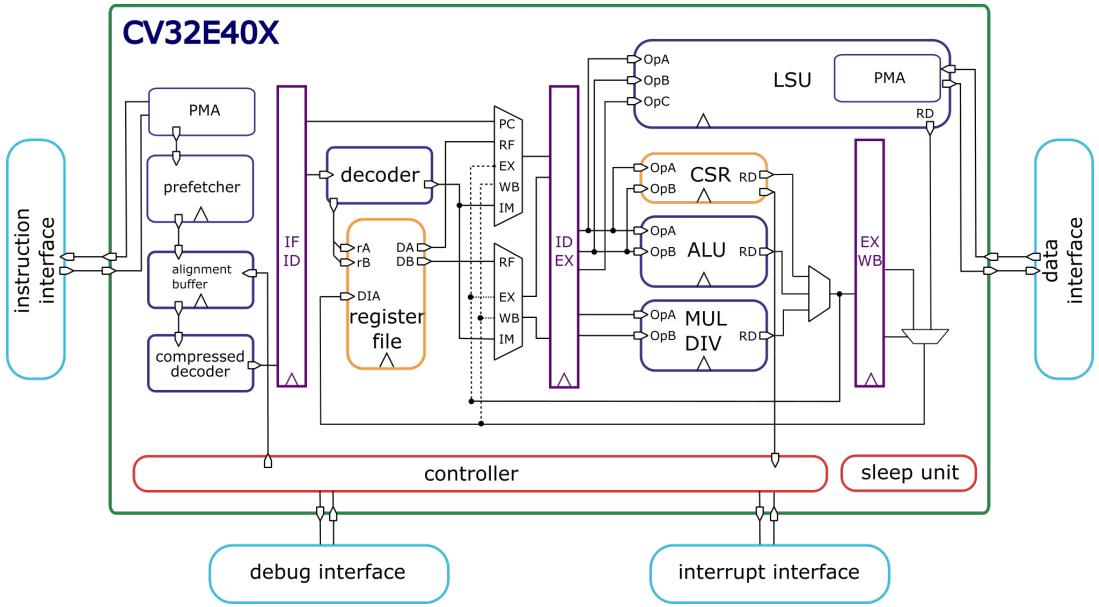
CV32E40X – PA / TRL 2

- 4-stage, in-order, single-issue
- RV32IMC[ABP]Zicount_Zicsr_Zifencei[_Zce][X]
- M-mode, CLINT, OBI, PMA, bus error

- Compute intensive core
  - CV-X-IF interface to offload custom extensions
  - _Zce, RV32B and RV32P (iff ratified)
  - PPA optimizations wrt CV32E40P

- Verification

- Project goal: industrial grade (TRL 5)



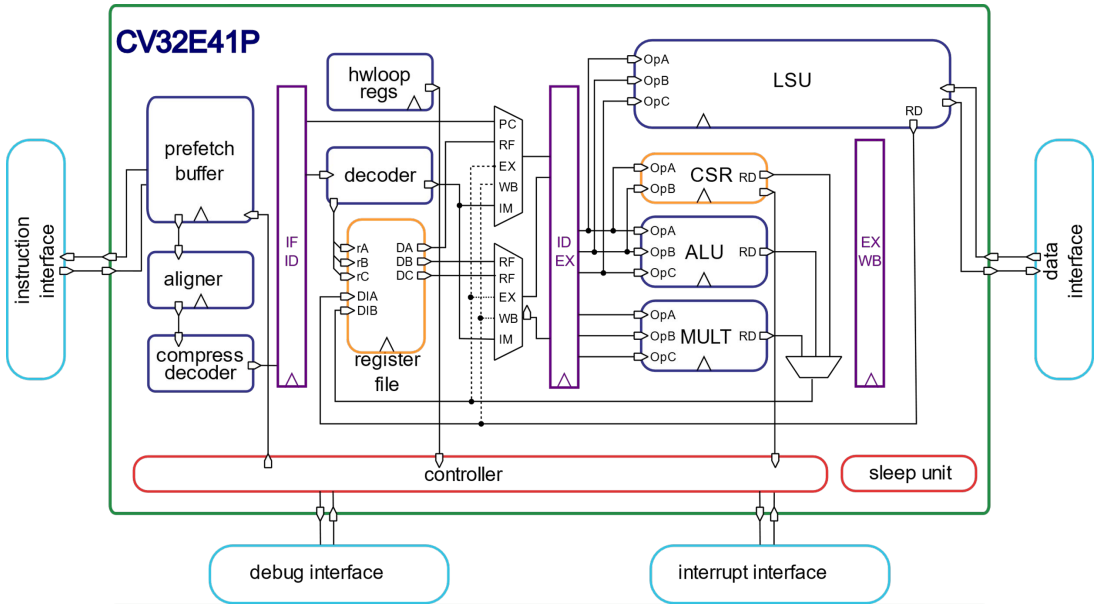
CV32E41P – PC / TRL 2

- 4-stage, in-order, single-issue
- RV32IMCZicount_Zicsr_Zifencei **[_Zce][{F, _Zfinx}]**[PULP_XPULP][PULP_CLUSTER]
- M-mode, CLINT, OBI

- Starting from CV32E40P fork

- Goals:
  - Proof of Concepts to demonstrate the PPA of Zce and Zfinx RISC-V draft ISA extensions

- Project goal: Proof of concept (TRL 3) *(next may be TRL 5)*

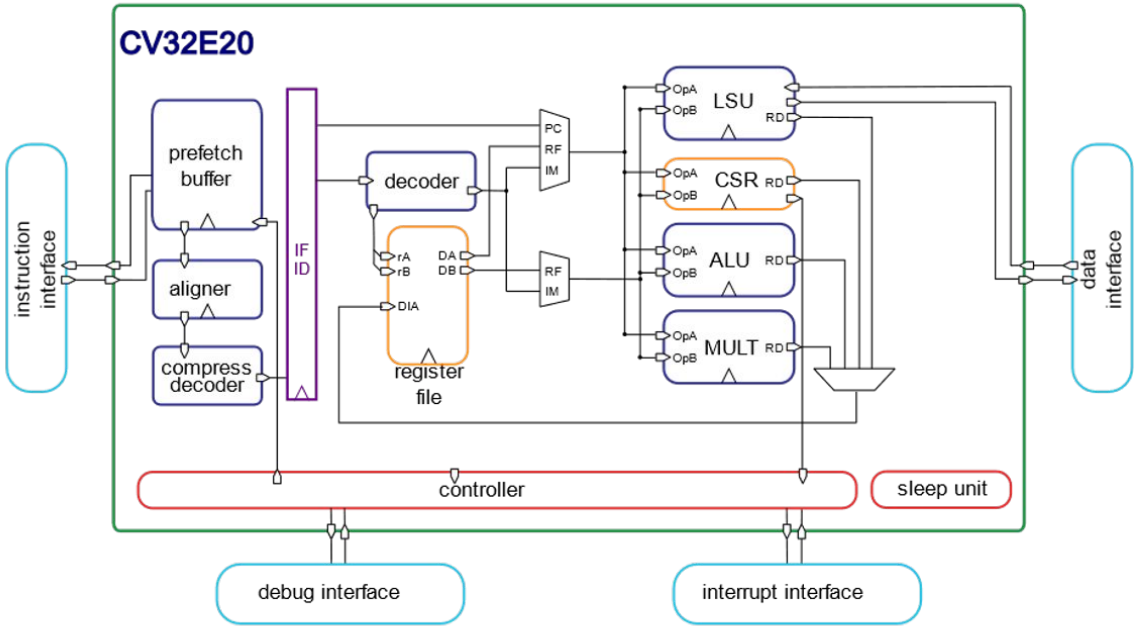


CV32E20 – PC / TRL 2

- 2-stage, in-order, single-issue
- RV32{I,E}[M]CZicount_Zicsr_Zifencei[_Zce]
- M-mode, CLINT, OBI

- Low area core
  - Optimized power and area for control-oriented applications
  - Starting point lowRISC Ibex (which started from ETH zero-riscy)
    - Clean-up parameters
    - Aligning IP interface with CV32E40* cores

- Project goal: industrial grade (TRL 5)



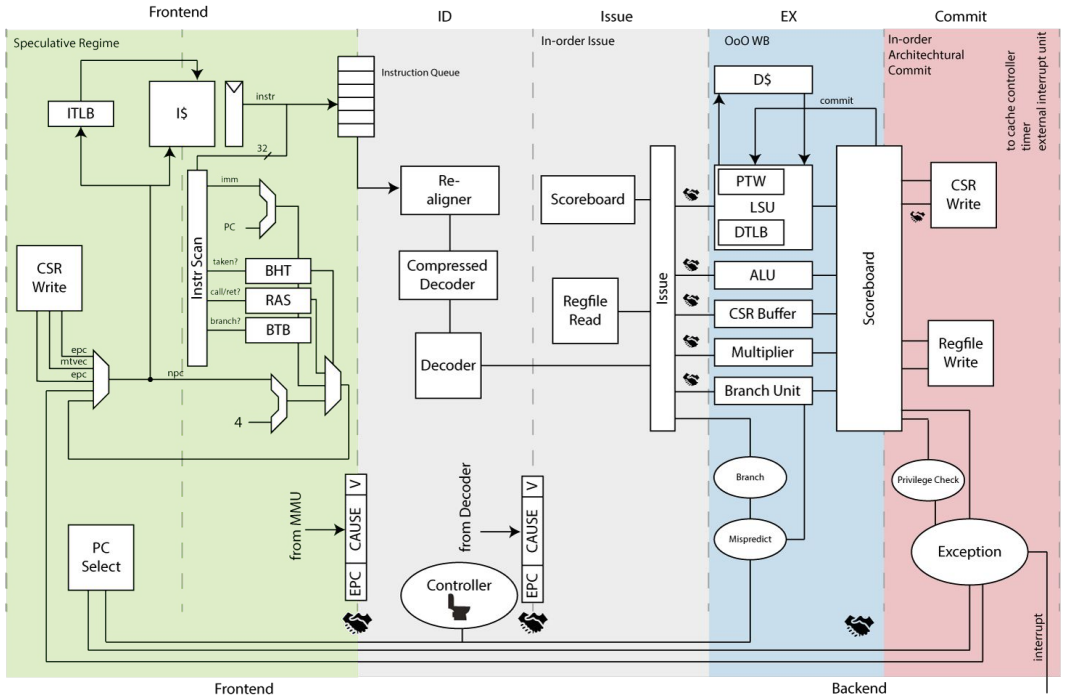
CORE-V Application-class Cores

CVA6

- 6-stage, in-order, single-issue
- RV{32|64}IMAC[FD]Zicsr
- M/S/U-mode, CLINT, AXI

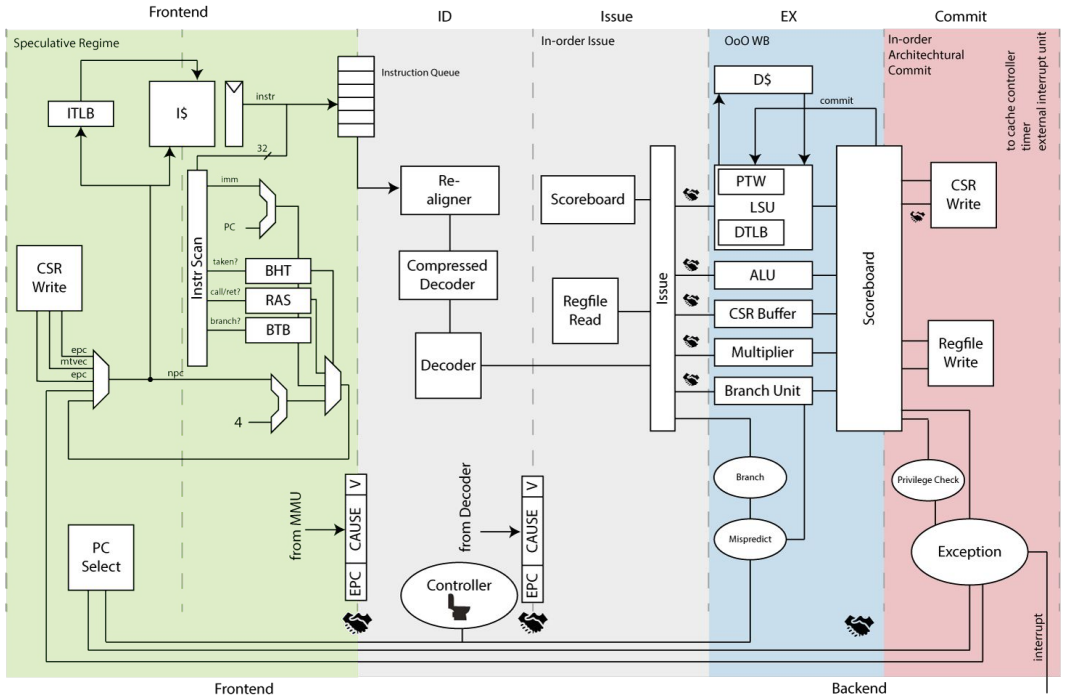
- Flexible application core
  - Linux-compatible thanks to MMU
  - 32 or 64 bit (CV32A6, CV64A6) from same RTL (64b from ETH, 32b from Thales)
  - L1 caches

- Project goal: industrial grade (TRL 5)
  - Currently drafting specifications, entry point for next stages



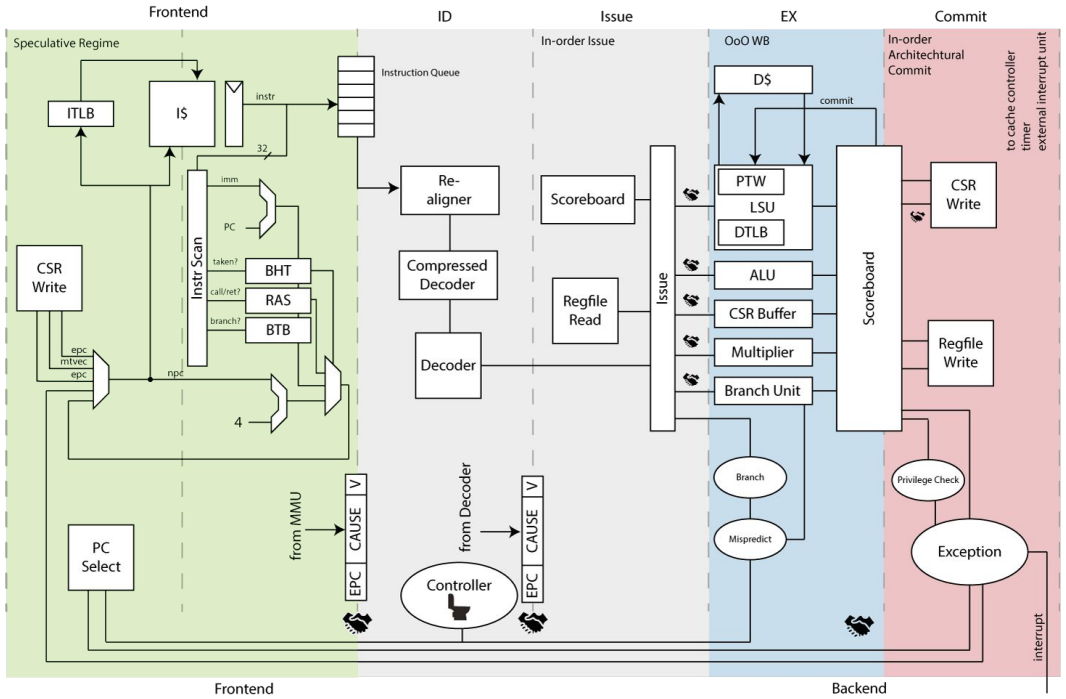
CV64A6 – PA / TRL 4

- Verification
  - RV64GC (RV64IMAFDC), debug, interrupts, privileges… to verify
  - sim-based Step&Compare, formal verification considered
  - RVFI interface
- New documentation
  - reStructured text-based as for the other cores
- FPGA optimizations
  - Target the Xilinx Genesys2, but not a Soft-Core!



CV32A6 – PA / TRL 3

- 32b version support for
  - Pipeline
  - MMU
  - Floating-point
  - Linux
- Verification
  - RV32IMAFC, debug, interrupts, privileges… to verify
  - sim-based Step&Compare, formal verification considered
  - RVFI interface
- PPA optimizations
  - ASIC
  - FPGA (vendor-independent soft-core)



CVA6 – preliminary PPA

- **CV64A6**
  - COREMARK/MHz: ***2,19***
  - DMIPS/MHz: ***1,53***

Technology	Frequency	Area	Comments
Globalfoundies GF22FDX	1,7GHz	210kGE	No FPU, No Caches
28nm	740MHz	211kGE	No FPU, No Caches
Zynq Ultrascale+	95MHz	39,7kLUTs 23,2kFFs 37 DSP 37 BRAM (cache)	

CVA6 – preliminary PPA

- **CV32A6**
  - COREMARK/MHz: ***2,25***
  - DMIPS/MHz: ***1,35***

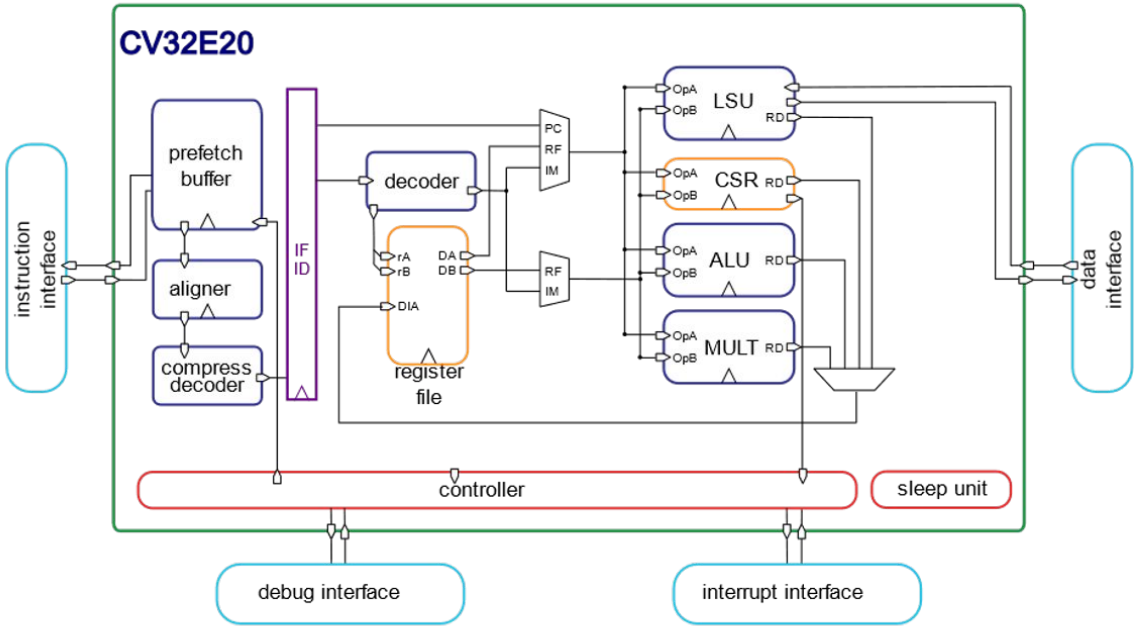
Technology	Frequency	Area	Comments
28nm	740MHz	87kGE	No FPU, No Caches, smaller CSR config., multicycle MUL, No MMU, No branch prediction, smaller scoreboard
Zynq Ultrascale+	120MHz	13,9kLUTs 9,3kFFs 4 DSP 32 BRAM (cache)	No FPU, No Caches, No MMU

CV32A5 – PC / TRL 3

- 5-stage, single-issue
- RV32I[M][A]
- out-of-order executions
- M, [S,U] Privileged support

- Soft-core
  - Optimized for FPGA
  - Starting from TAIGA from Simon Fraser University

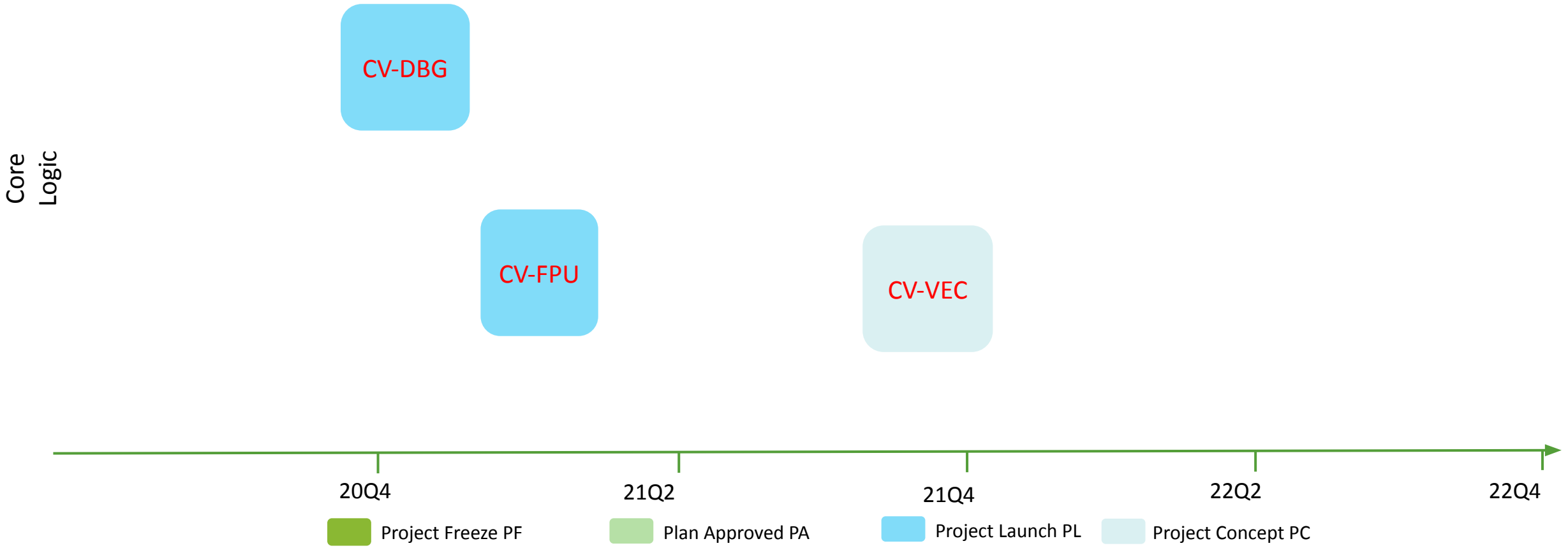
- Project goal: research (TRL 4)



CORE-V Core Logic Blocks

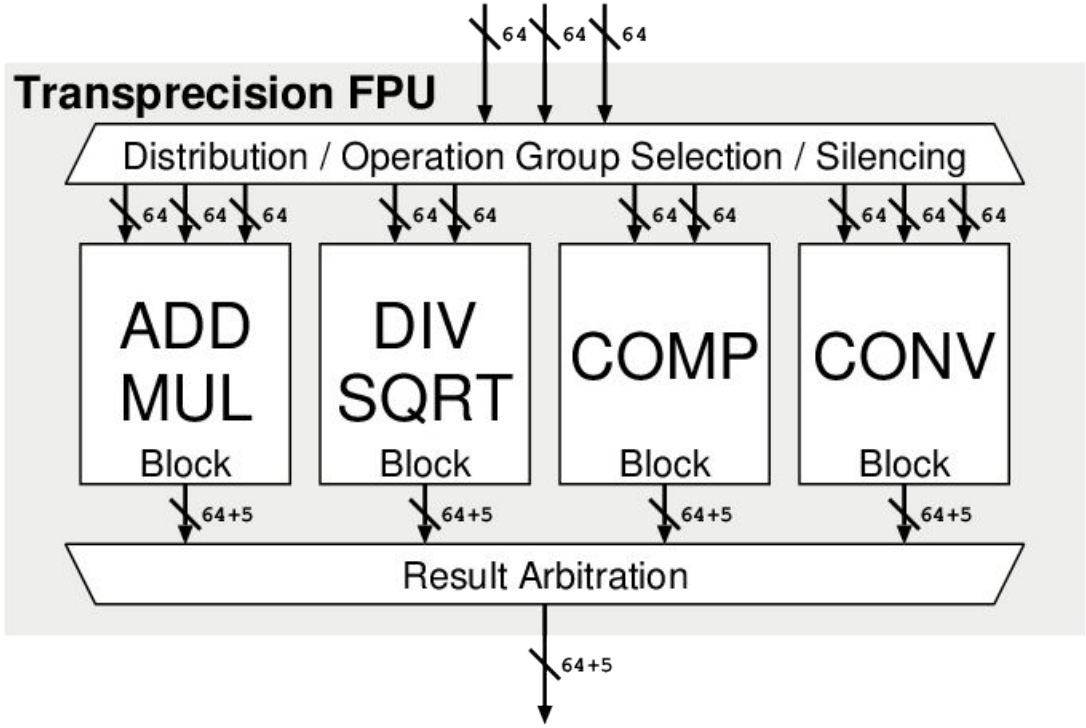
IP surrounding the CORE-V COREs

CORE-V Core Logic Blocks



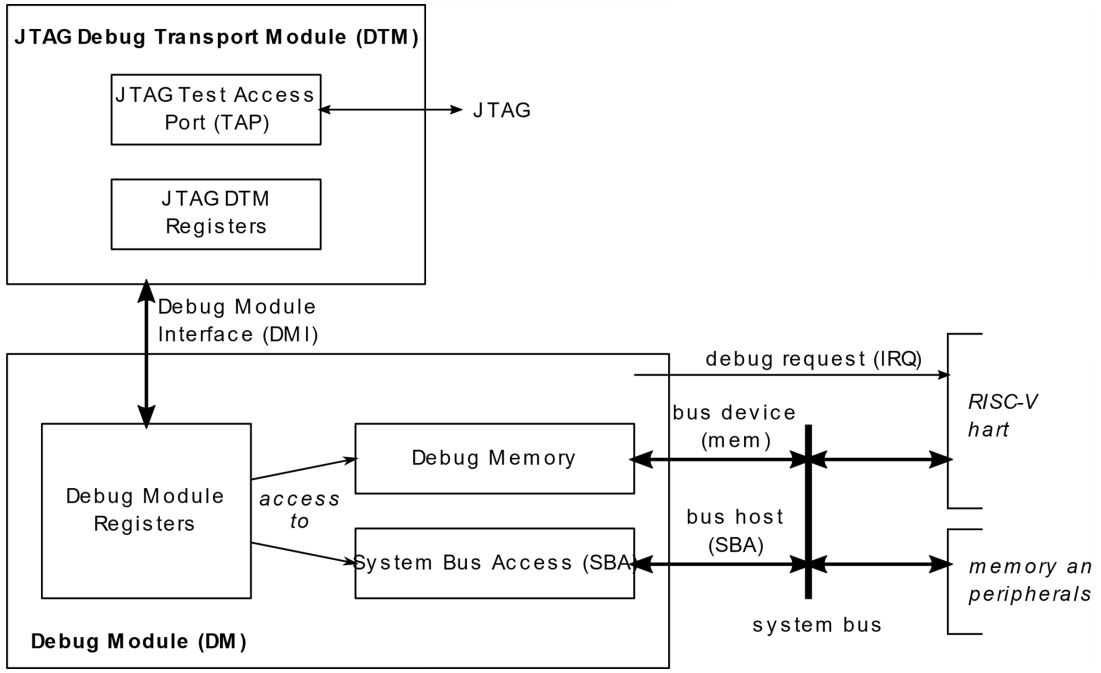
CV-FPU – PL / TRL 3

- Standalone co-processor that computes floating-point RISC-V RVF and RVD extensions
  - Parts of other projects as
    - CV32E40Pv2, CV32A6, CV64A6
- Possible starting point ETH *fpnew:*
  - Documentation
    - resTructured text documentation
  - Verification stand-alone or as part of the cores
- Project goal: industrial grade (TRL 5)



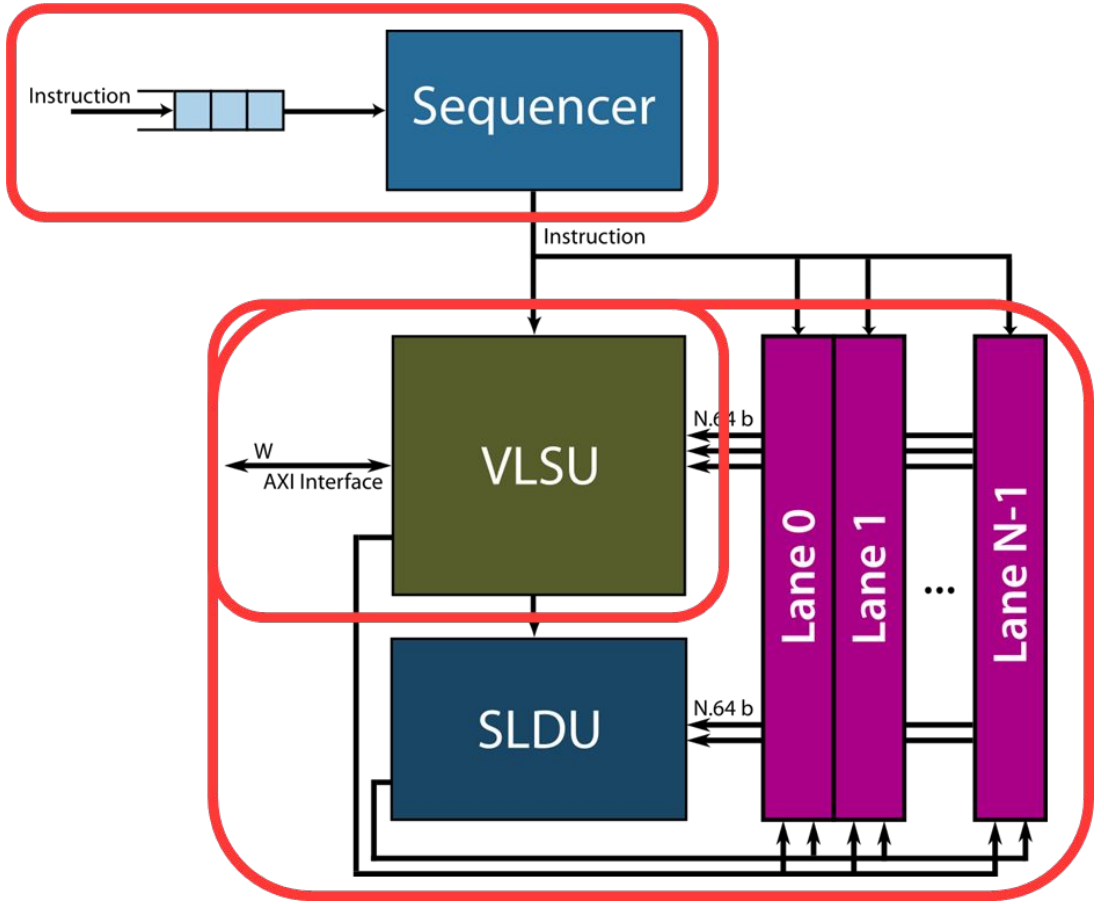
CV-DBG – PL / TRL 3

- Standalone IP that implements RISC-V Debug Module
  - Debug Transport Module, and ROM for Execution based debug
  - Parts of other projects as
    - CV32E40Pv2, CV32A6, CV64A6
- Possible starting point ETH *riscv-dbg:*
  - Documentation
    - resTructured text documentation
  - Verification stand-alone or as part of the cores
- Project goal: industrial grade (TRL 5)



CV-VEC – PC / TRL 3

- Standalone co-processor that computes vectorial instructions based on RISC-V RVV extensions
- Starting point ETH ara
  - RVB vector extensions and mixed-precision
  - SW support for ML applications
  - Implementation in Globalfoundries 22FDX
  - Compatible with CV64A6
- Part of OpenHW Accelerate research program with Mitacs, CMC Microsystems, Polytechnique Montréal & ETH Zurich TRL 4

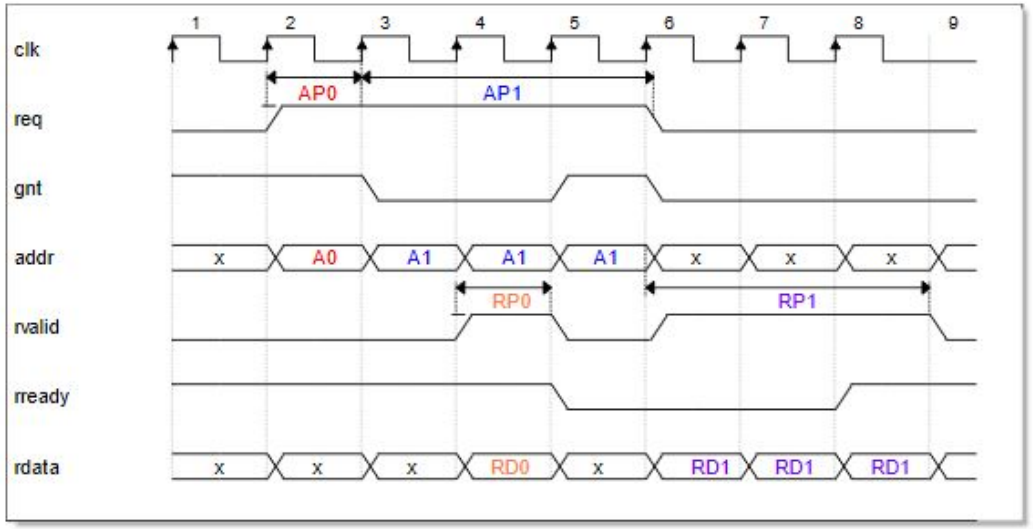


CORE-V SPECs

Specifications for RISC-V COREs

Open Bus Interface (OBI) Spec

- Memory Bus spec for RISC-V cores
  - Handshake based on ARM AMBA AXI
  - Coupled Read-Write channels
  - Easily translatable to ARM AMBA protocols
    - AXI, AHB
  - Based on: *https://raw.githubusercontent.com/openhwgroup/core-v-docs/master/cores/cv32e40p/OBI-v1.0.pdf*



CV-X-IF Spec

- Bus spec to allow RISC-V cores to offload RV extensions
  - CV32E40X will leverage this interface the most
  - CV32E40P and CVA6 will evaluate it
  - Based on: *https://github.com/openhwgroup/core-v-xif*
- Verification
  - Formal verification needed to test the spec
- Allows the cores to be agnostic about the custom extension definition

Mapping RISC-V Profiles

CORE-V	RISC-V Profile	Comment
CV64A6	RVA20	
CV32A6	RVM20_32	RISC-V does not have a 32b application profile
CV32A5	RVM20_32	
CV32E40P CV32E40Pv2	RVM20_32	
CV32E40X CV32E40S	RVM20_32	TBC: Zce support to qualify for RVM22_32*
CV32E20	RVM20_32	
CV32E41P	RVM20_32	Zce will likely qualify CV32E41P for RVM_22_32*

*RVM22_32 under definition

- OpenHW Group is a not-for-profit, global organization driven by its members and individual contributors where HW and SW designers collaborate in the development of open-source cores, related IP, tools and SW such as the CORE-V Family of open-source RISC-V cores

- OpenHW Group & CORE-V Family of open-source RISC-V cores for use in high volume production SoCs
  - Visit the OpenHW CORE-V Cores GitHub repository and contribute
  - Learn more at OpenHW TV

- Follow us on Twitter @openhwgroup & LinkedIn OpenHW Group