



CORES TG – September 6 2022

Arjan Bink

Jérôme Quevremont

Davide Schiavone



OPENHW GROUP™
— PROVEN PROCESSOR IP —

© OpenHW Group

Agenda

- Core overview
- CV32E40X / CV32E40S status update
- CV32E20 status update
- CVA6 status update
- CV32E40Pv2 update
- Manifest file support

Core overview



Core	TRL	Privilege	ISA	Debug	Interrupts	Bus	Gate count	CoreMark / MHz	Target date
CV32E20	5	M, U (v1.11)	RV32I, RV32E, C, M	0.13.2	Basic	OBI	14K, 19K, -	-, 2.47, -	2023 Q1 (per August 2022)
CV32E40P	5	M (v1.11)	RV32I, C, F, M, Xpulp	0.13.2	Basic	OBI	-, -, -	-, 2.91, -	
CV32E40S	5	M, U (v1.12)	RV32I, RV32E, C, M, Xsecure, Zba, Zbb, Zbc, Zbkc, Zbs, Zca, Zcb, Zcmb, Zcmp, Zcmt, Zicsr, Zifencei, Zkt, Zmmul	1.00	Basic, CLIC	OBI	-, -, -	-, 2.91, 3.12	2023 Q1 (per July 2022)
CV32E40X	5	M (v1.12)	RV32I, RV32E, A, C, M, Xif, Zba, Zbb, Zbc, Zbkc, Zbs, Zca, Zcb, Zcmb, Zcmp, Zcmt, Zicntr, Zihpm, Zicsr, Zifencei, Zkt, Zmmul	1.00	Basic, CLIC	OBI	-, -, -	-, 2.91, 3.12	2023 Q1 (per July 2022)
CV32E41P	3	M (v1.11)	RV32I, C, F, M, Zca, Zcb, Zcmb, Zcmp, Zcmt, Zfinx	0.13.2	Basic	OBI	-, -, -	-, 2.91, -	
CVA5	3		RV32I, A, M				-, -, -	-, -, -	
CVA6	5	M, S, U (v1.10)		0.13.2	Basic		-, -, -	-, -, -	
September 6, 2022									





CV32E40X / CV32E40S status

Øystein Knauserud



CV32E40X

- ZC 0.70.5 design done
 - Including updates to RVFI
- Bugfixes and cleanups
- Updated user manual to 0.5.0
 - Added custom instruction – wfe
 - Added debug PC sampling interface
 - time_i inputs and time[h] CSRs

CV32E40S

- Weekly merges from CV32E40X
 - Zc 0.70.5 done, including Smstateen
- Bugfixes and cleanups
- Updated user manual to 0.5.0





CV32E20 Status

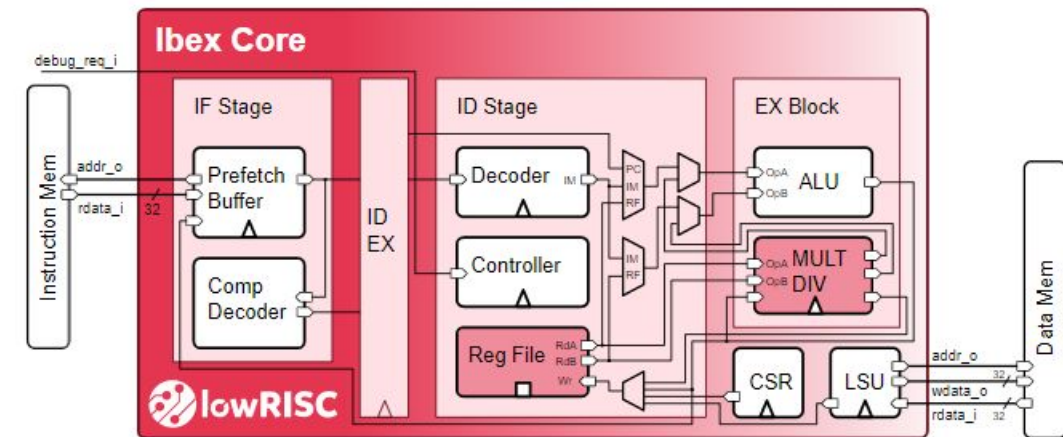
Joe Circello (NXP), Lee Hoff (Intrinsix),
Maarten Arts (NXP)

CV32E20 Project Details



- Project summary
 - Ultra-low-end core cloned from Ibex (fka ETH-Zurich ZeroRISCY)
 - Targeted at the low-end of RISC-V MCU roadmaps
 - Any “compute constrained socket”
 - Processor element in embedded SoC subsystems
- Planning for an initial two-phase development
 - Phase 1: RTL design of core starting from Ibex
 - RV32IMC and RV32EMC ISAs
 - OBI bus interfaces
 - CV32E40P-like interrupt interface
 - User and Machine privilege modes
 - Phase 2: Create core-complex aka “coreplex”
 - OBI to AHB-5 bridges (Intrinsix IP)
 - Interrupt controller design INTC (OpenTitan from PULP)
 - Debug design
- Project resourcing
 - Co-sponsored by NXP and Intrinsix
 - NXP contributes architecture, RTL design and verification resources
 - Intrinsix contributes bus gasket IP and verification resources

<https://github.com/lowRISC/ibex/blob/master/README.md>



Project Details Continued



- Core overview summary

Core	TRL	Privilege	ISA	Debug	Interrupts	Bus	Gate count	CoreMark / MHz	Target date
CV32E20	5	M, U (v1.11)	RV32I, RV32E, C, M	0.13.2	Basic	OBI	14K, 19K, -	-, 2,47, -	2023 Q1 (per August 2022)

- Optimized for minimum gate count and lowest power
- Future (CV32E20v2) design enhancements under consideration
 - Support for new compressed opcodes ("Zc"), possible inclusion of bit manipulation ISA
 - Exploration of 2-pin CJTAG debug interface
 - Exploration of ETH-Zurich's "Tiny FPU"
 - Low granularity Physical Memory Protection module (PMP)



CV32E20: Status



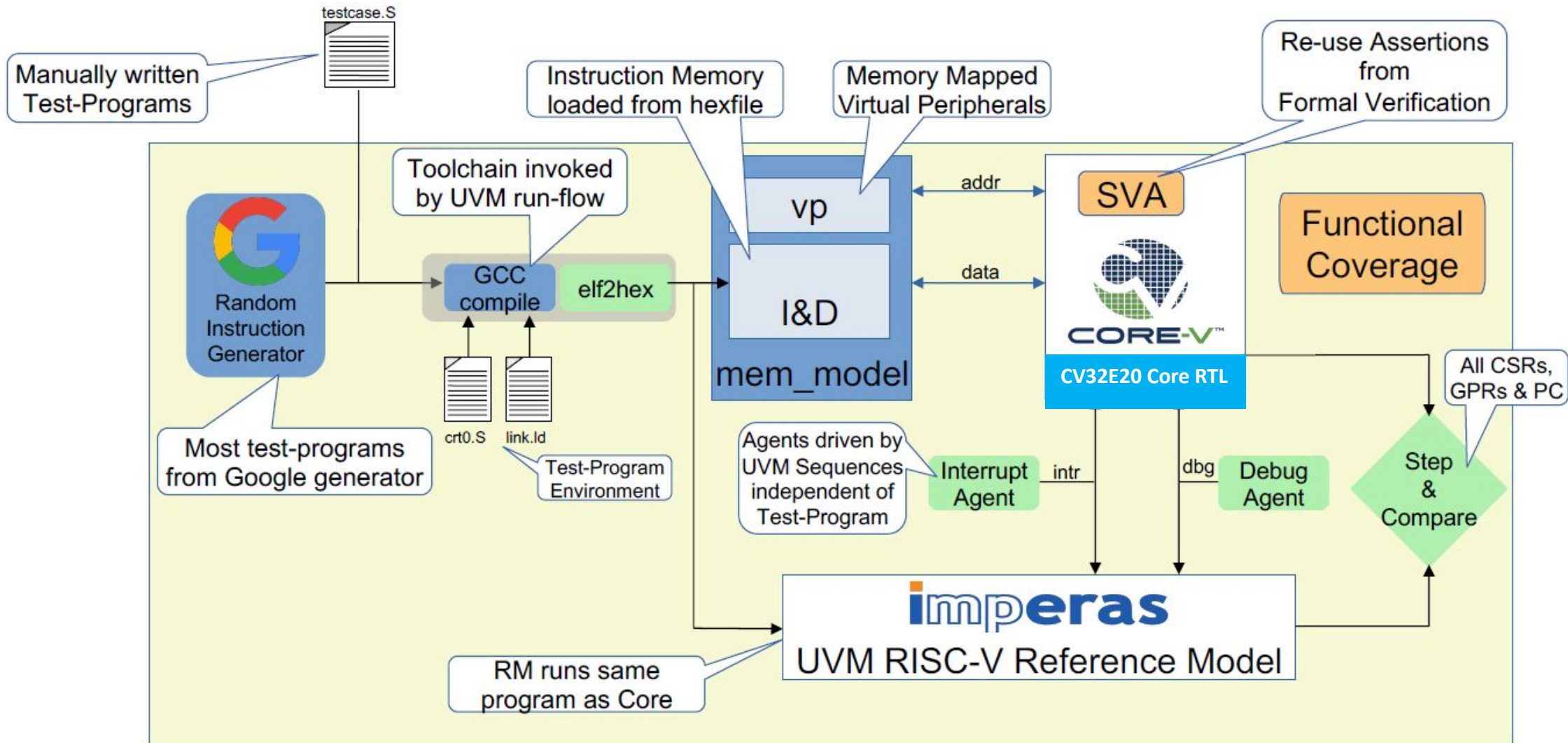
- Project gates
 - PC gate passed on 6/2021
 - PL gate passed on 2/2022
 - PA TBD
 - Intended technical details are fairly well defined
 - Development of initial high-level development timeline
 - 2 NXP resources (RTL design, Verification) have joined project
 - RTL design resource started at the beginning of August 2022
 - Project completion now estimated as end of 1Q2023
- Project meetings
 - Every Tuesday 9:00 EDT
- Mattermost channel - TWG : Cores : CVE20
- Github repository - github.com/openhwgroup/cve2



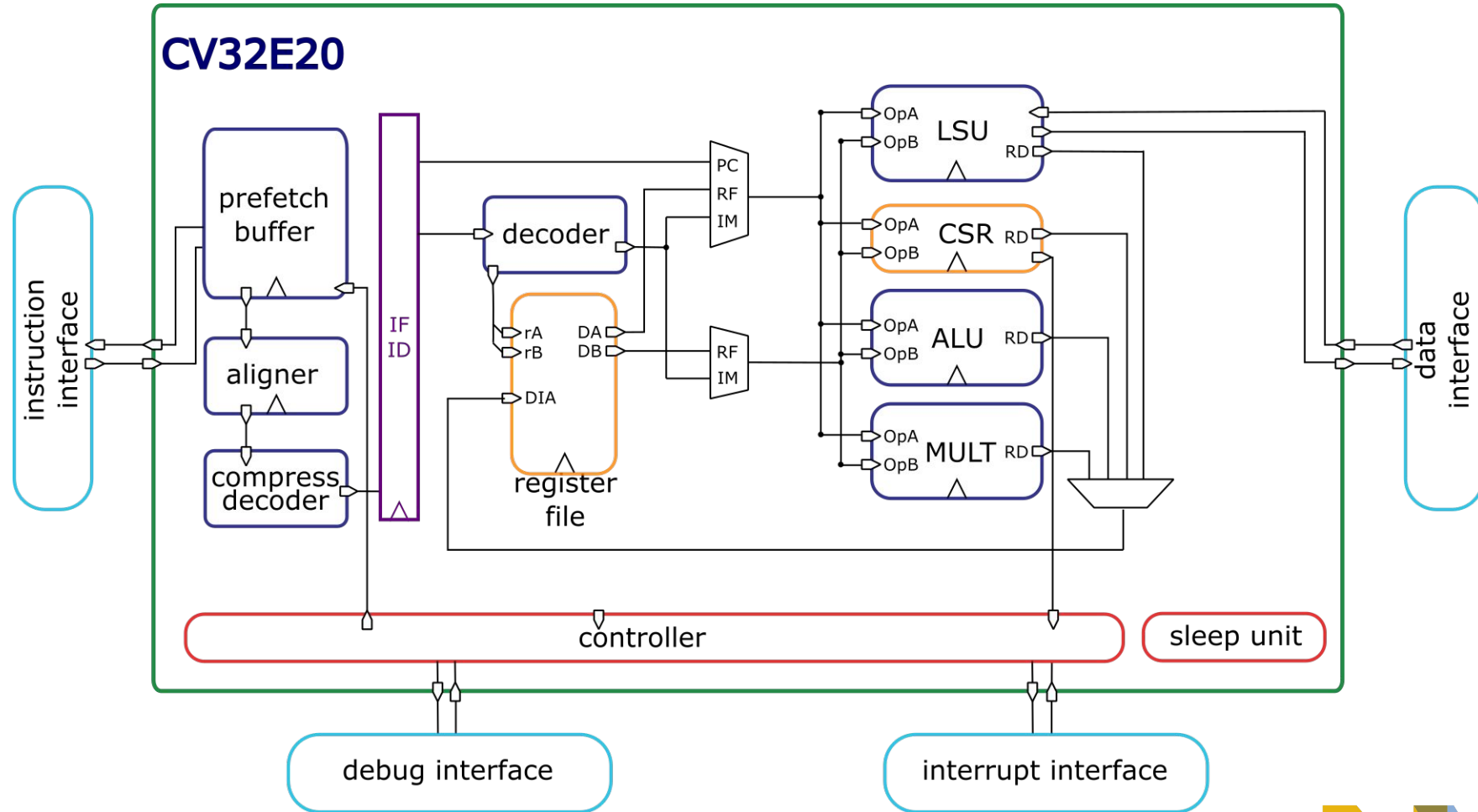
CV32E20: Current Activities

- Ongoing (weekly) discussions with multiple stakeholders
 - OpenHW Design (Davide Schiavone)
 - OpenHW Verification (Mike Thompson)
 - Imperas (Simon Davidmann)
 - Intrinsix + NXP CV32E20 teams
- Recent discussions focused on...
 - Core-V-Verif strategies
 - Intent is to fully leverage existing Core-V-Verif strategy and methodologies (see next page)
 - Imperas core configuration definition for reference model creation
- Creation of core specifications underway
- Initial design environment
 - “Pipe cleaner” activities at both NXP + Intrinsix for GitHub repositories

CORE-V Verification UVM Test Bench



CV32E20 Core Block Diagram





CVA6 status

Jérôme Quévremont

Cores TG meeting, 2022-09-06



OPENHW GROUP
— PROVEN PROCESSOR IP —

CVA6 highlights

- Project
 - Set up Kanban board to track progress of project tasks
- Core design:
 - Implementation of “generic” performance counters
 - FPGA optimization: -40% LUT -43% FF as of 2022-06-13
 - Investigating scratchpads
- SW:
 - Yocto support integrated
 - Updating FreeRTOS with latest kernel
- Verification:
 - Adding riscv-arch test suite
 - Set up dashboard for CVA6 regression based on core-v-verif
 - Integrated Linux boot in CI flow
 - CV-X-IF UVM agent PR'd, CV-X-IF verification on-going
 - Planning upcomping verification activities



CV32E40Pv2 status

Pascal Gouédo
Yoann Pruvost
Xavier Aubert



Project



- Project

- Design & Verification meeting
 - Wednesday 14:00 CEST every 2 weeks([lcal](#))
- Dedicated technical meetings when needed
- Reporting to Cores TG
- Next: Make a mid-term schedule

- Mattermost channels

- TWG : Cores : CV32E4*P
- TWG : Verification

- Resources

- Pascal Gouédo
 - Specification, Design, Verification
- Yoann Pruvost
 - Design, Verification (core-v-verif & Formal)
- Xavier Aubert
 - Verification leader
- 1 Sub-contractor
 - Formal Verification (with OneSpin support)

- OpenHW staff

- Mike
 - Verification technical support
- Davide
 - Architecture & Design technical support



Documentation

- Specification
 - Full review on-going
 - Put final re-encoding table in PR #452
 - Update [instruction set extensions.rst](#) with last re-encoding table
- Next
 - Clarify rendered documents sources & versioning (latest?)
 - Add a Revision History section
 - Create an issue for missing MSTATUS.FS & MSTATUS.SD (when Zfinx = 0)
 - Move from PULP_ZFINX to RISC-V Zfinx
 - Hardware loops

Design



- Core + CV-FPU wrapper is now the new CV32E40P top level (APU interface not visible anymore)
- Re-encoded PULP instruction available for internal Formal verification task
 - Post-increment and Register Indexed LD/ST
 - General ALU/Bit Manipulation/MAC
 - 124/220 SIMD instructions
 - Next
 - Push internal work to cv32e40p repo
 - Remaining SIMD instructions once re-encoded table update finalized
 - Hardware Loops once specification finalized, reviewed and approved
- Define Strategy about unsupported features for RTL cleanup
 - PULP_SECURE, PMP, A_EXTENSION, privlv, uhartid, ...



Verification



- E40P core-v-verif environment
 - Now supports PULP/CLUSTER/FPU/ZFINX through new config files
 - Switched to RVFI support (based on E40X test-bench implementation)
 - COREV-DV tests generator
 - Started to add PULP instructions
- Next
 - PULP re-encoded instruction support
 - Add ISS options to enable XPULP v1 or v2
 - Add Floating Point compliance tests in core-v-verif
- Longer term
 - Switch E40Pv2 test-bench to RVVI/Imperas-DV framework once fully functional on E40X



Formal Verification

- RV32IMC
 - 1 rule violation
 - RV32F (all but DIV/SQRT results)
 - 16 rules violations
 - F instructions decoding
 - F instructions result impact on pipeline
 - FFLAGS not correctly set
 - Missing MSTATUS.FS and SD (when Zfinx = 0)
 - FMUL.S wrong result
- CV-FPU pb
- Core Pb?
- PULP instruction
 - Post-increment and Register Indexed LD/ST : Hold bounded
 - General ALU/Bit Manipulation/MAC
 - 8 rules violations to analyze
- Re-encoding strategy suspected

Tools



- SW toolchain
 - Embecosm business contract finalized
 - Toolchain development for re-encoded PULP instruction will start on October 1st, 2022
 - Deliveries:
 - November 1st, 2022
 - Establish baseline RV32IMFCZciscr functionality, test and benchmarking environment
 - Implement the PULP 8- and 16-bit SIMD extensions
 - December 1st, 2022
 - Implement the PULP bit manipulation extensions
 - Implement the PULP 8- and 16-bit SIMD ext
 - January 1st, 2023
 - Implement Zfinx support
 - Benchmarking
 - Later on
 - Hardware Loop support
 - ...





Manifest File

Davide Schiavone



Today

- **CVA6:**
 - FLIST, FUSESOC, SRC_FILE, BENDER
- **CV32E40P:**
 - FLIST, SRC_FILE, BENDER
- **CV32E40X/S:**
 - FLIST
- **CVE2:**
 - FUSESOC, SRC_FILE
- **CVA5:**
 - no file (inside scripts)

What to do?

- **flist:**
 - easily integrate in existing core-v-verif
 - no tool dependency
 - no scripts generated
- **src_files, bender:**
 - solve packages and generates scripts for compilation, synthesis (**legacy PULP**)
- **fusesoc:**
 - solve packages and generates scripts for compilation, synthesis (**from Olof Kindgren**)

Thank you!