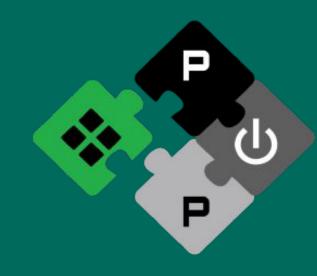


CVA6S+: A Superscalar RISC-V Core

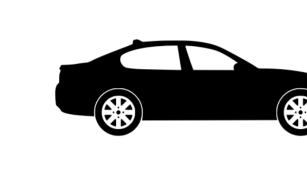
# with High-Throughput Memory Architecture



<u>Riccardo Tedeschi<sup>1</sup>, Gianmarco Ottavi<sup>1</sup>, Côme Allart<sup>3,6</sup>, Nils Wistoff<sup>2</sup>, Zexin Fu<sup>2</sup>, Filippo Grillotti<sup>5</sup>, Fabio De Ambroggi<sup>5</sup>, Elio Guidetti<sup>5</sup>,</u> Jean-Baptiste Rigaud<sup>3</sup>, Olivier Potin<sup>3</sup>, Jean Roch Coulon<sup>6</sup>, César Fuguet<sup>4</sup>, Luca Benini<sup>1,2</sup>, Davide Rossi<sup>1</sup>

ETHZURICH<sup>2</sup> MINES Institut Mines-Télécom<sup>3</sup> (Antica Antica Antic

• Growing demand for autonomy in critical applications: embedded automotive, industrial automation, and aerospace fields are driving the need for high-performance CPUs







**RISC-V Open-Source cores**: the ecosystem of

**<u>CVA6</u>** is a configurable 64/32 bit RISC-V core maintained by **OpenHW Group** with multiple industrial and academic partners

- $\rightarrow$  6 stages scalar pipeline: ×2 Fetch (IF), Decode (ID), Issue (IS), Execute (IE), and Writeback (WB)
- → In order dispatch, out of order completion, in order commit



Ζ

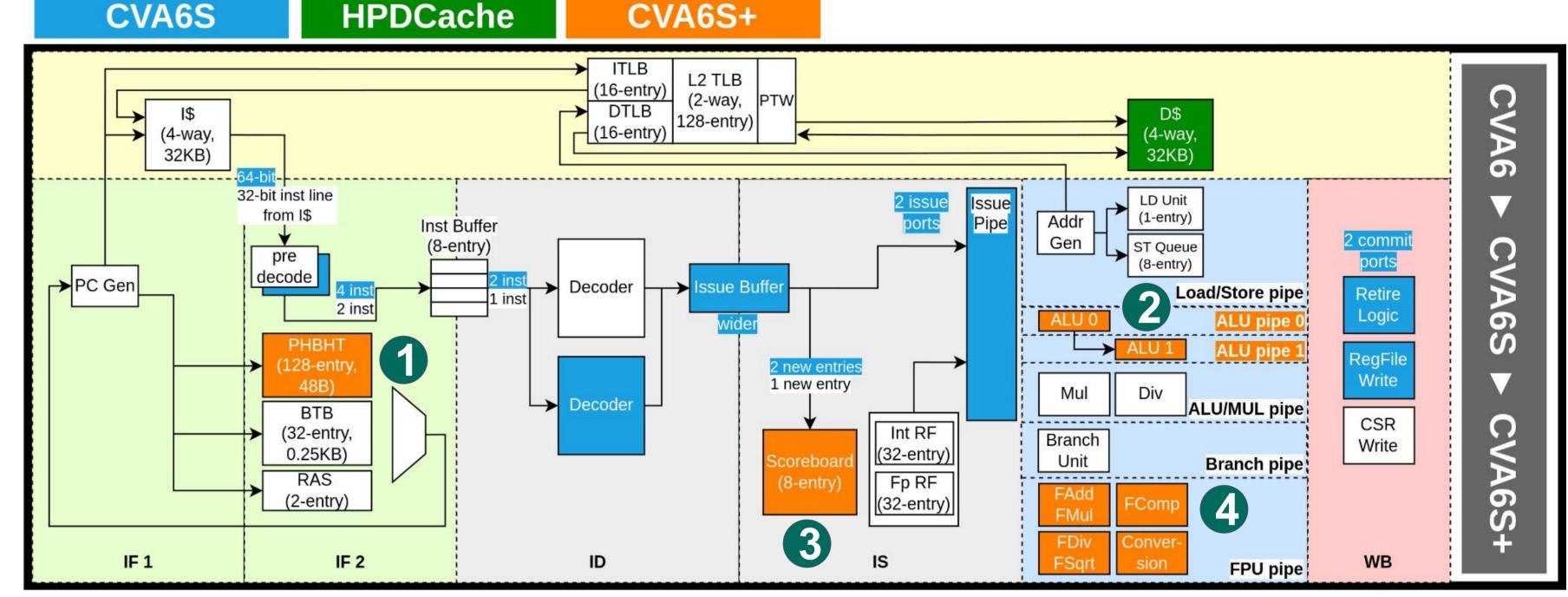
20

open-source RISC-V high-performance cores is growing (BlackParrot, BOOM, XiangShan, and Xuantie C910)

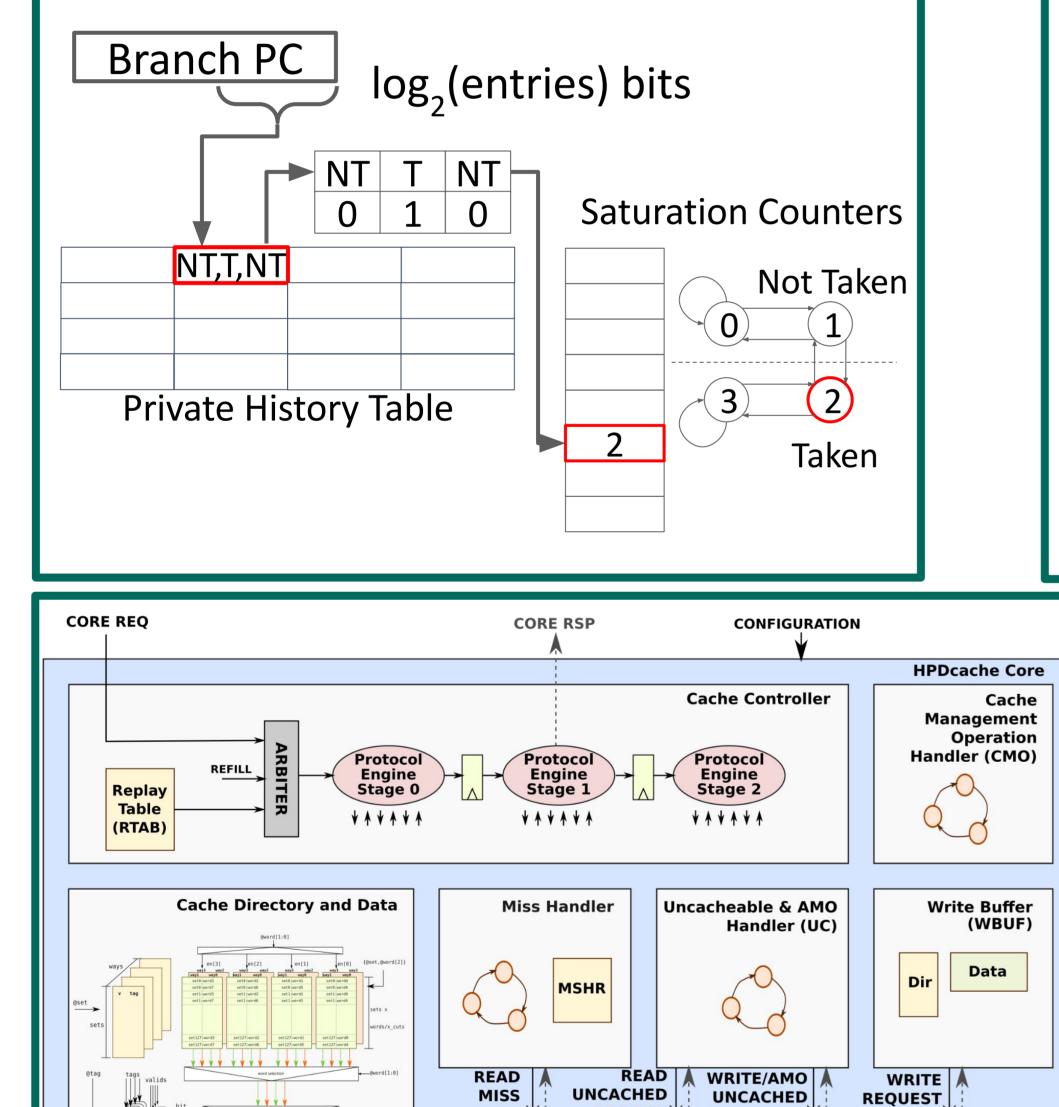
We present <u>CVA6S+</u>, extending CVA6S with: 1) Better branch prediction 3) Register renaming 2) ALU-ALU forwarding 4) FPU support

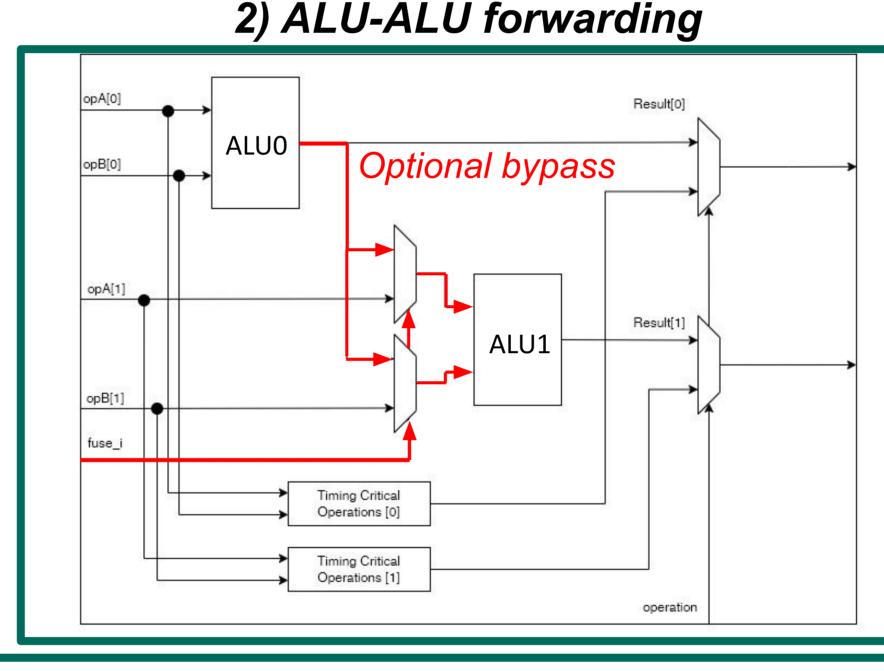
We integrate CVA6S+ with the the OpenHW Core-V High-Performance L1 Data Cache (<u>HPDCache</u>)

# **<u>CVA6S</u>** is the **superscalar dual-issue version** of CVA6 by Thales

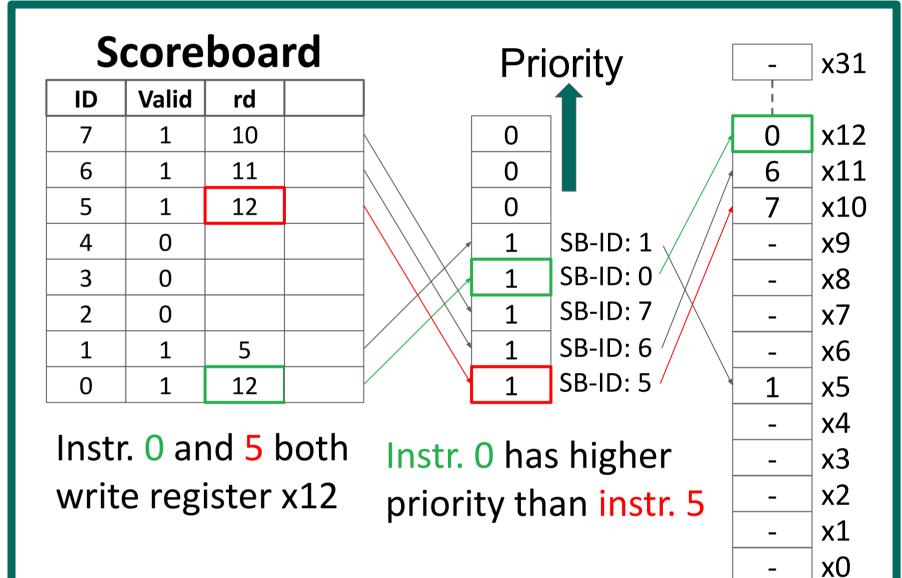


### 1) Private History Branch Predictor





# 3) Renaming scheme





BUTION

 $\rightarrow$  Pipelined µarchitecture, single-cycle read/write hit latency **HPDC** 

- $\rightarrow$  Out-of-Order Execution & **Non-Blocking Pipeline**
- → Supports both WB and WT policies on a cache line-level granularity
- The **scoreboard** is a **circular**  $\rightarrow$ buffer
- → RAW hazards need to know the **newest instruction** to correctly **forward data**

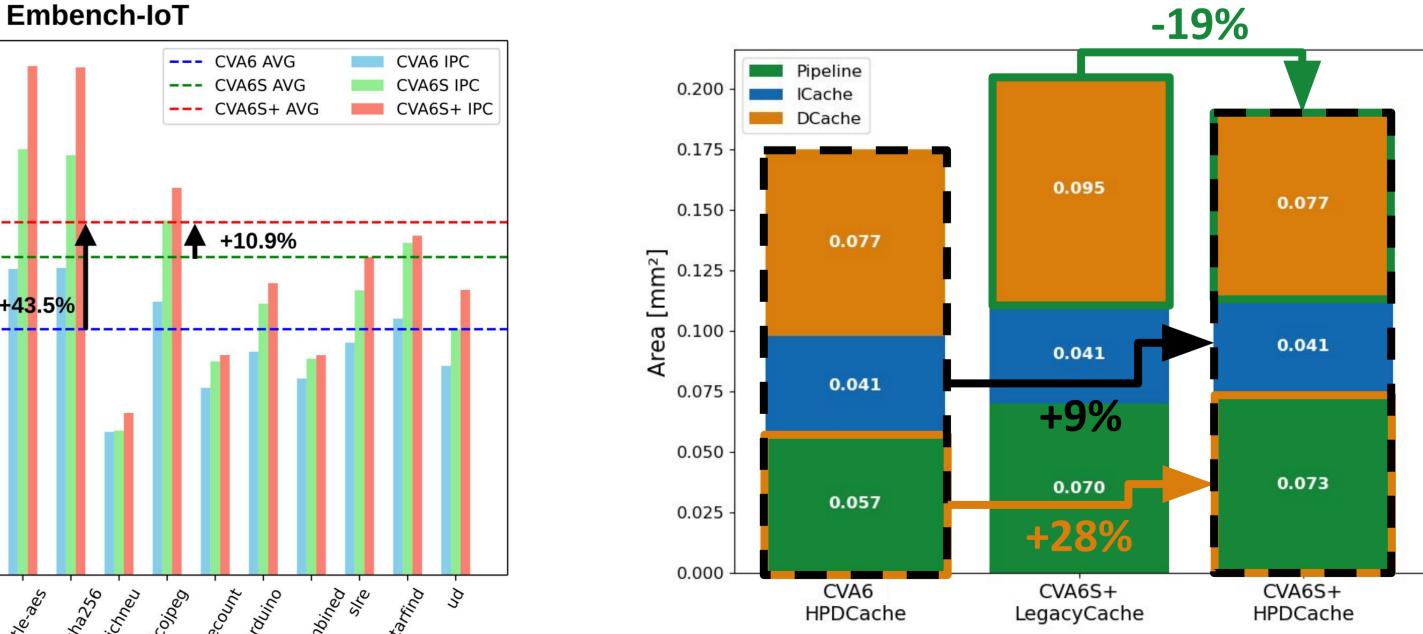


ac

he

overview

Rotate the entries based on the **commit pointer** and forward data accordingly



#### RaiderSTREAM

MEMORY

READ REQ/RSP

**REQ/RSP** 

Arbiter

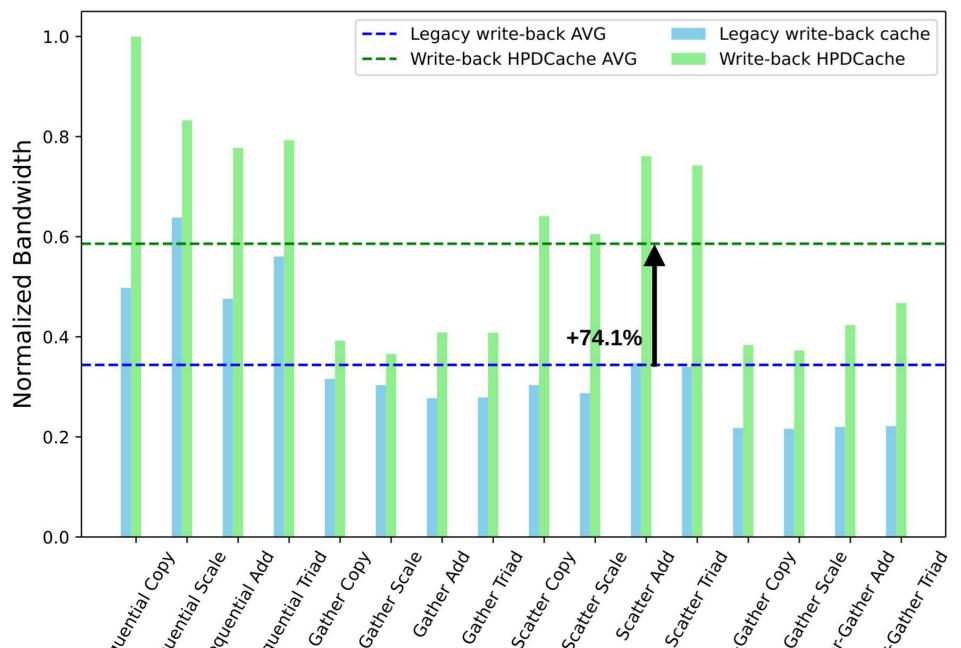
REQUEST

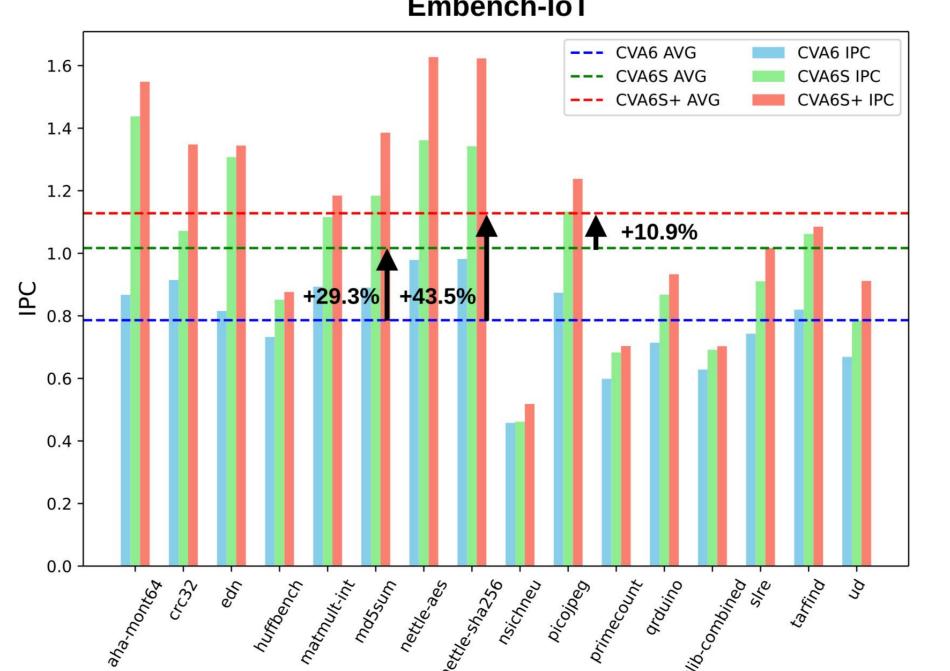
Arbiter

MEMORY /

WRITE REQ/RSP

REQ/RSP





# RESULTS

# **HPDCache vs Legacy cache subsystem**

- → Same pipeline (CVA6S+)
- → RaiderSTREAM: working set 2× cache size
- → +74.1% bandwidth with HPDCache
- -> Cache area is reduced by 19% due to

## better SRAM organization

# CVA6S+ vs CVA6/CVA6S

- → Same cache subsystem (HPDCache)
- → Embench-IoT: working set fully cached
- → +43.5% IPC vs CVA6 / +10.9% IPC vs CVA6S
- → Pipeline area: +28% / Total area: +9%
- → Max. Frequency: 1090 MHz (-0.5% vs CVA6)

