

An Open-Source CVA6S+ based High-Performance, Cache-Coherent Cluster for 64b Automotive MPUs

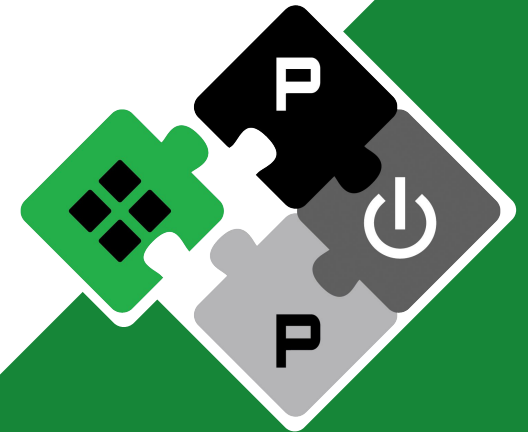
*Riccardo Tedeschi¹, Mehran Mohammadi¹, Aleksi Korsman²,
Filippo Grillotti³, Fabio De Ambroggi³, Elio Guidetti³,
César Fuguet⁴, Luca Benini^{1,5}, Davide Rossi^{1,6}*

University of Bologna, Italy¹
STMicroelectronics, Italy³
ETH Zürich, Switzerland⁵

Aalto University, Finland²
Univ. Grenoble Alpes, Inria, France⁴
Fondazione Chips-IT⁶

PULP Platform

Open Source Hardware, the way it should be!



@pulp_platform 

pulp-platform.org 

youtube.com/pulp_platform 

Introduction: Software Defined Vehicles



Next-gen Software-Defined Vehicles (SDV) are reshaping automotive electronics, moving complexity **from hardware to software** through **zonal controllers**.

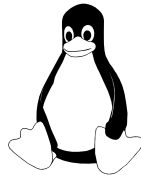
Introduction: Software Defined Vehicles



Next-gen Software-Defined Vehicles (SDV) are reshaping automotive electronics, moving complexity **from hardware to software** through **zonal controllers**.

Why?

- **Functional consolidation** reduces hardware and wiring costs.
- **Software and hardware** evolve on **independent life cycles**.
- **Rich OSes** fuel advanced tasks, from **infotainment** to **ADAS**.



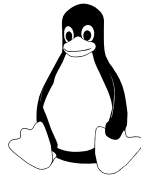
Introduction: Software Defined Vehicles



Next-gen Software-Defined Vehicles (SDV) are reshaping automotive electronics, moving complexity **from hardware to software** through **zonal controllers**.

Why?

- **Functional consolidation** reduces hardware and wiring costs.
- **Software and hardware** evolve on **independent life cycles**.
- **Rich OSes** fuel advanced tasks, from **infotainment** to **ADAS**.



How?

Fewer chips, more intelligence: from dozens of MCUs to a **few clustered high-IPC 64-bit MPUs**.



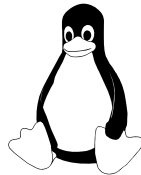
Introduction: Software Defined Vehicles



Next-gen Software-Defined Vehicles (SDV) are reshaping automotive electronics, moving complexity **from hardware to software** through **zonal controllers**.

Why?

- **Functional consolidation** reduces hardware and wiring costs.
- **Software and hardware** evolve on **independent life cycles**.
- **Rich OSes** fuel advanced tasks, from **infotainment** to **ADAS**.



How?

Fewer chips, more intelligence: from dozens of MCUs to a **few clustered high-IPC 64-bit MPUs**.



Good news:
Automotive IDMs are betting on RISC-V.

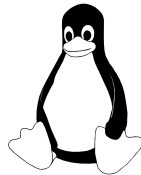
Introduction: Software Defined Vehicles



Next-gen Software-Defined Vehicles (SDV) are reshaping automotive electronics, moving complexity from hardware to software through **zonal controllers**.

Why?

- **Functional consolidation** reduces hardware and wiring costs.
- **Software and hardware** evolve on independent life cycles.
- **Rich OSes** fuel advanced tasks, from **infotainment** to **ADAS**.



How?

Fewer chips, more intelligence: from dozens of MCUs to a **few clustered high-IPC 64-bit MPUs**.



Good news:

Automotive IDMs are betting on RISC-V.

Bad news: few **open-source alternatives**



An Open-Source Cluster for automotive MPUs



Multi-core MPUs require **coherent memory** hierarchies to support **SMP** execution.

An Open-Source Cluster for automotive MPUs



Multi-core MPUs require **coherent memory** hierarchies to support **SMP execution**.

The **open-source landscape** today^{1,2,3}:

- **Custom HDLs**
- **Custom protocols**
- ↳ **Complex** third-party SoC **integration** and **certification**
- **Many-core platform** using **directory based coherence**
- ↳ **Excessive overhead** for embedded low core count clusters

¹J. Balkind et al. "OpenPiton: An Open Source Manycore Research Framework". In: SIGARCH Comput. Archit. News 44.2 (2016).

²P. Mantovani et al. "Agile SoC Development with Open ESP". In: Proceedings of the 39th International Conference on Computer-Aided Design. 2020.

³A. Amid et al. "Chipyard: Integrated design, simulation, and implementation framework for custom socs". In: IEEE Micro 40.4 (2020).

An Open-Source Cluster for automotive MPUs



Multi-core MPUs require **coherent memory** hierarchies to support **SMP** execution.

The **open-source landscape** today^{1,2,3}:

- **Custom HDLs**
- **Custom protocols**
- ↳ **Complex** third-party SoC **integration** and **certification**
- **Many-core platform** using **directory based coherence**
- ↳ **Excessive overhead** for embedded **low core count** clusters

We introduce the **Culsans cluster**:

- Synthesizable **SystemVerilog**
- Industry-standard **AMBA4 ACE**
- Based on industry-academia backed OpenHW's **CVA6** and **HPDCache**
- ↳ **Bring your own SoC**
- **Snoop-based MOESI** coherence
- ↳ **Optimal performance and cost** for **low core count (2-8)**

¹J. Balkind et al. "OpenPiton: An Open Source Manycore Research Framework". In: SIGARCH Comput. Archit. News 44.2 (2016).

²P. Mantovani et al. "Agile SoC Development with Open ESP". In: Proceedings of the 39th International Conference on Computer-Aided Design. 2020.

³A. Amid et al. "Chipyard: Integrated design, simulation, and implementation framework for custom socs". In: IEEE Micro 40.4 (2020).

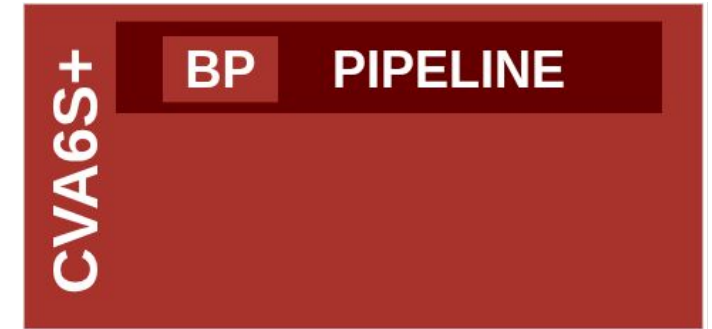
Contribution: the Culsans cluster



Three key components make up the **Culsans cluster**:

CVA6S+: RV64GC 6-stage dual-way superscalar core

↳ **Branch prediction (BP) exploration** to optimize area versus accuracy



Contribution: the Culsans cluster



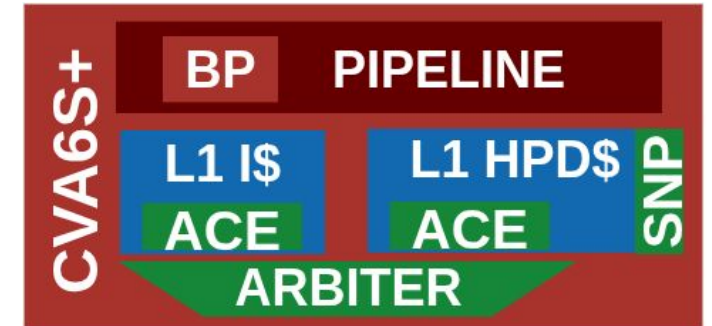
Three key components make up the **Culsans cluster**:

CVA6S+: RV64GC 6-stage dual-way superscalar core

↳ **Branch prediction (BP) exploration** to optimize area versus accuracy

HPDCache: pipelined, out-of-order, non blocking cache

↳ **Extended to support ACE** coherent transactions



Contribution: the Culsans cluster

Three key components make up the **Culsans cluster**:

CVA6S+: RV64GC 6-stage dual-way superscalar core

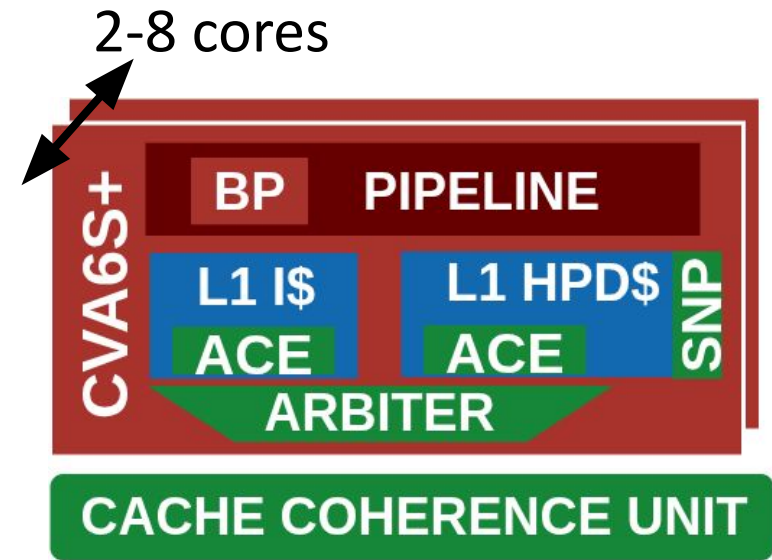
↳ **Branch prediction (BP) exploration** to optimize area versus accuracy

HPDCache: pipelined, out-of-order, non blocking cache

↳ **Extended to support ACE** coherent transactions

Cache Coherence Unit (CCU)

↳ **New open-source IP** enforcing snoop-based MOESI coherence over AMBA4 ACE



CVA6S+: branch predictor exploration



We consider **four predictors** to find the **best area-accuracy trade-off**:

CVA6S+: branch predictor exploration



We consider **four predictors** to find the **best area-accuracy trade-off**:

- **Bimodal:** scalar CVA6 default
simple 2-bit saturating counter per PC

↳ **Cheap** but **context-unaware**

CVA6S+: branch predictor exploration



We consider **four predictors** to find the **best area-accuracy trade-off**:

- **Bimodal:** scalar CVA6 default
simple 2-bit saturating counter per PC

↳ **Cheap** but **context-unaware**

- **Gshare:**
XORs PC with global branch history

↳ **Medium cost**, **captures correlations**

CVA6S+: branch predictor exploration



We consider **four predictors** to find the **best area-accuracy trade-off**:

- **Bimodal:** scalar CVA6 default
simple 2-bit saturating counter per PC

↳ **Cheap** but **context-unaware**

- **Private history:** CVA6S+ default
per-branch history table

↳ **Highly accurate** but **area-costly**

- **Gshare:**
XORs PC with global branch history

↳ **Medium cost**, **captures correlations**

CVA6S+: branch predictor exploration



We consider **four predictors** to find the **best area-accuracy trade-off**:

- **Bimodal:** scalar CVA6 default
simple 2-bit saturating counter per PC
↳ **Cheap** but **context-unaware**
- **Private history:** CVA6S+ default
per-branch history table
↳ **Highly accurate** but **area-costly**
- **Gshare:**
XORs PC with global branch history
↳ **Medium cost**, **captures correlations**
- **Tournament (gshare + bimodal):**
meta-predictor choosing the sub-predictor
↳ **Many configurations**,
balanced area-accuracy trade-off

CVA6S+: branch predictor exploration



We sweep across different {**predictor type, number of entries, history length**}.

CVA6S+: branch predictor exploration



We sweep across different {**predictor type, number of entries, history length**}.

- Benchmarks: **Embench-IoT**
- Technology: **GlobalFoundries 12nm FinFET**

CVA6S+: branch predictor exploration



We sweep across different {**predictor type, number of entries, history length**}.

- Benchmarks: **Embench-IoT**
- Technology: **GlobalFoundries 12nm FinFET**

- **bimodal - 128e**
- **private - 128e/3h**

References

Notation:
e = entries,
h = history bits

CVA6S+: branch predictor exploration



We sweep across different {predictor type, number of entries, history length}.

- Benchmarks: **Embench-IoT**

- **bimodal - 128e**
- **private - 128e/3h**

References

- Technology: **GlobalFoundries 12nm FinFET**

- **tournament - bimodal 128e + gshare 256e/6h**
- **tournament - bimodal 256e + gshare 256e/6h**

Identified via design space exploration

Notation:

e = entries,

h = history bits

CVA6S+: branch predictor exploration



We sweep across different {predictor type, number of entries, history length}.

- Benchmarks: **Embench-IoT**

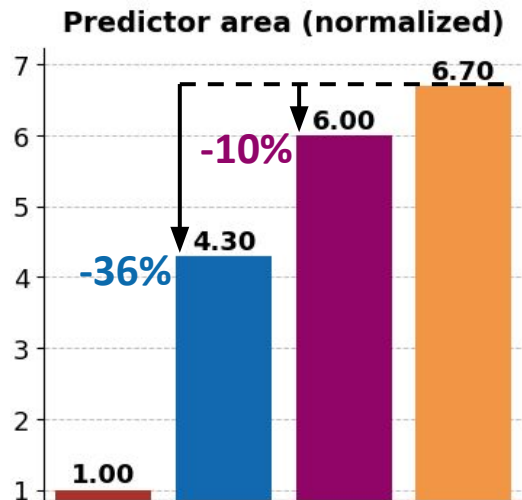
- **bimodal - 128e**
- **private - 128e/3h**

References

- Technology: **GlobalFoundries 12nm FinFET**

- **tournament - bimodal 128e + gshare 256e/6h**
- **tournament - bimodal 256e + gshare 256e/6h**

Identified via design space exploration



Notation:

e = entries,
h = history bits

Corner:

SS, 0.72V, -40°C

CVA6S+: branch predictor exploration



We sweep across different {predictor type, number of entries, history length}.

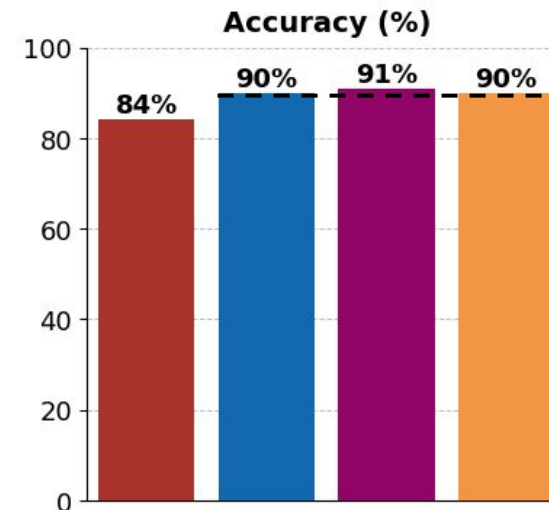
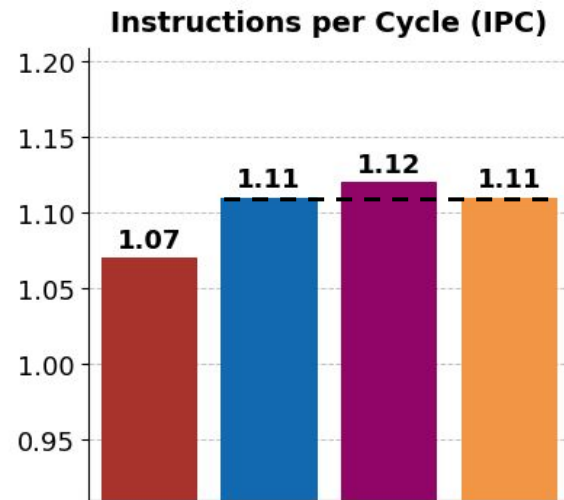
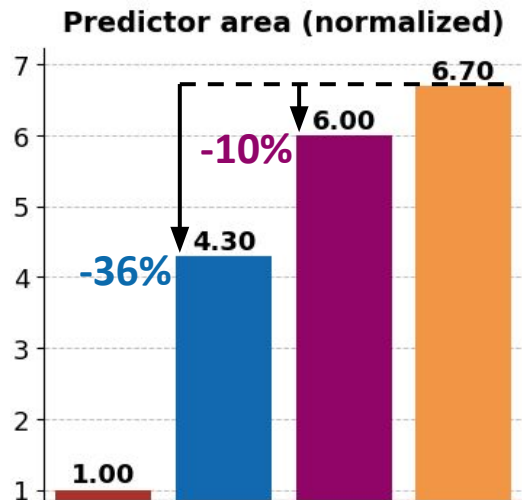
- Benchmarks: **Embench-IoT**
- Technology: **GlobalFoundries 12nm FinFET**

- **bimodal - 128e**
- **private - 128e/3h**

References

- **tournament - bimodal 128e + gshare 256e/6h**
- **tournament - bimodal 256e + gshare 256e/6h**

Identified via design space exploration



Notation:
e = entries,
h = history bits

Corner:
SS, 0.72V, -40°C

CVA6S+: branch predictor exploration



We sweep across different {predictor type, number of entries, history length}.

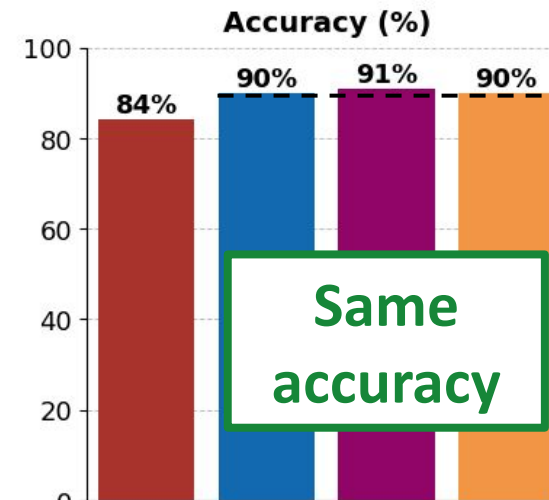
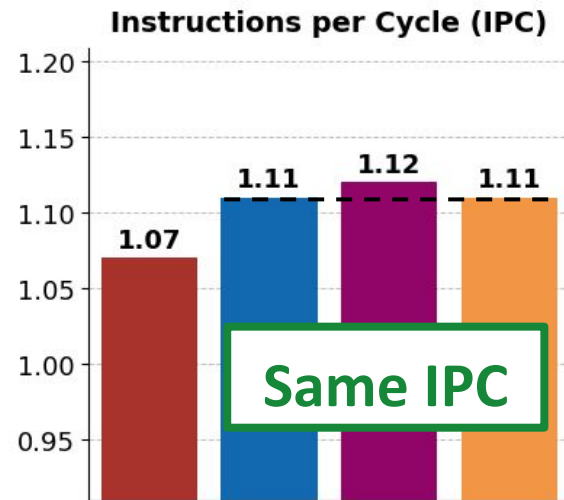
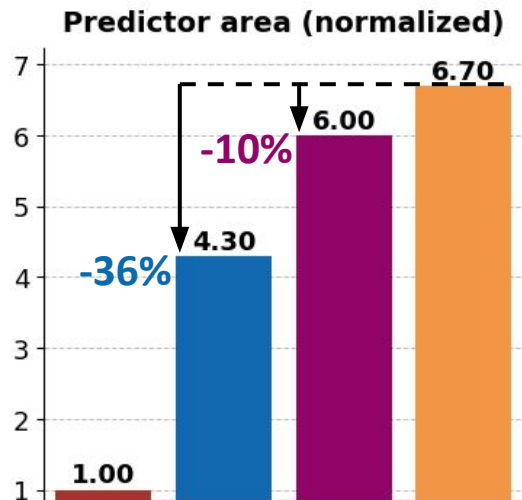
- Benchmarks: **Embench-IoT**
- Technology: **GlobalFoundries 12nm FinFET**

- **bimodal - 128e**
- **private - 128e/3h**

References

- **tournament - bimodal 128e + gshare 256e/6h**
- **tournament - bimodal 256e + gshare 256e/6h**

Identified via design space exploration



Notation:
e = entries,
h = history bits

Corner:
SS, 0.72V, -40°C

CVA6S+: branch predictor exploration



We sweep across different {predictor type, number of entries, history length}.

- Benchmarks: **Embench-IoT**
- Technology: **GlobalFoundries 12nm FinFET**

- **bimodal - 128e**

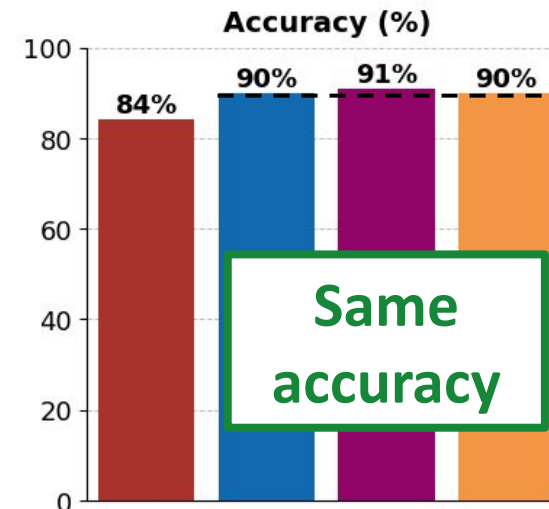
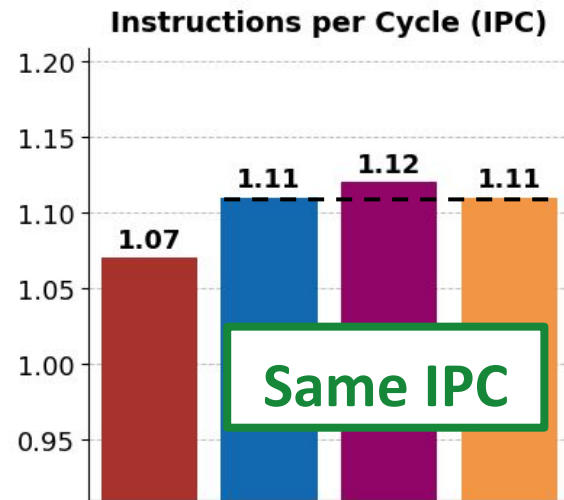
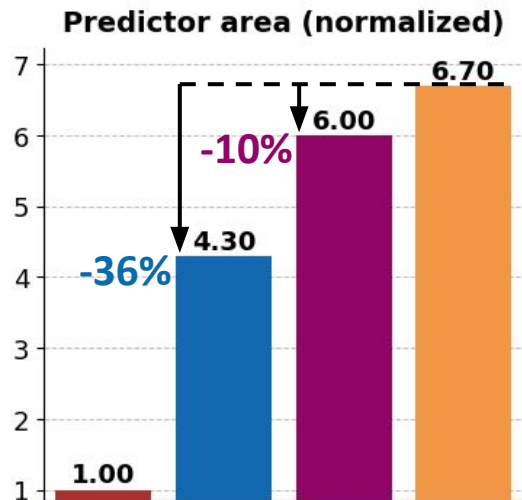
- **private - 128e/3h**

- **tournament - bimodal 128e + gshare 256e/6h**

- **tournament - bimodal 256e + gshare 256e/6h**



Tournament predictors: up to -36% area, iso-performance



Notation:

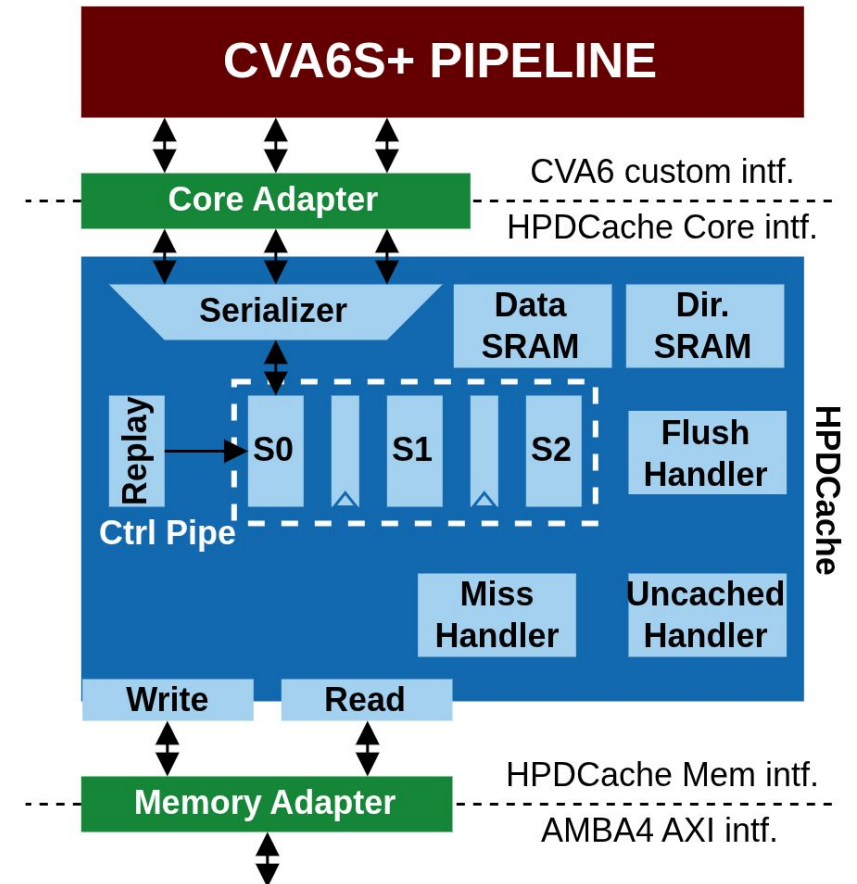
e = entries,
h = history bits

Corner:

SS, 0.72V, -40°C

HPDCCache: ACE extension

We extend OpenHW's HPDCCache to support ACE, covering both cache-side and interface changes:



HPDCache: ACE extension



We extend OpenHW's HPDCache to support ACE, covering both **cache-side and interface changes**:

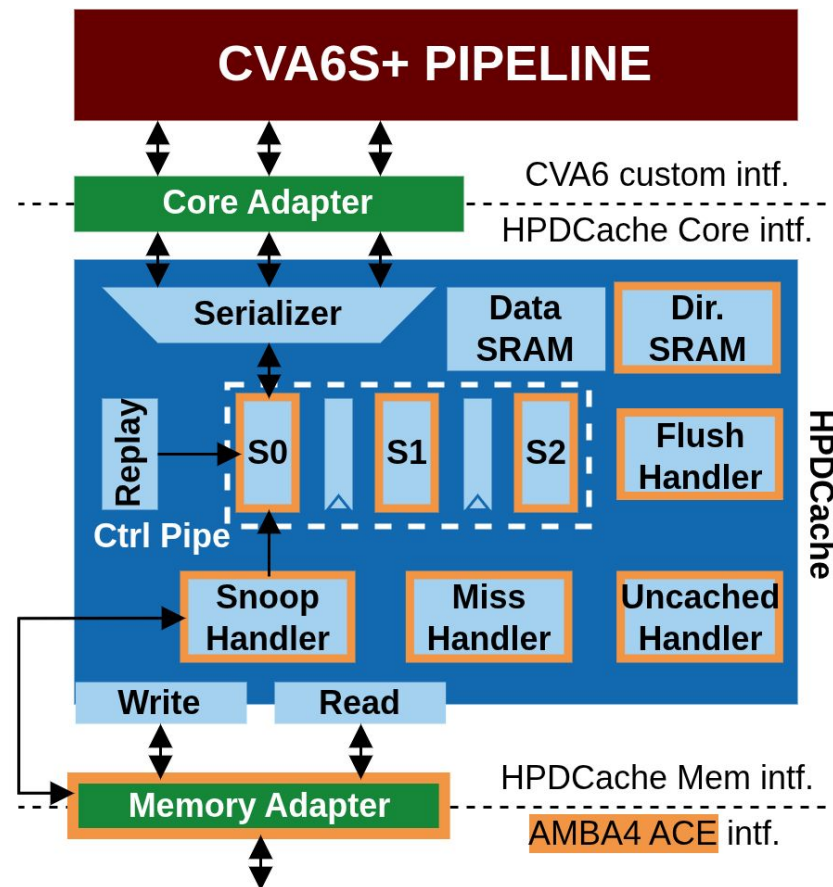
Cache side:

- New **snoop handler**
- **MOESI** cacheline state tracking
- Control pipeline **rework** for **snoop transactions**
- Coherent **ATOPs** and **LR/SC**

Interface:

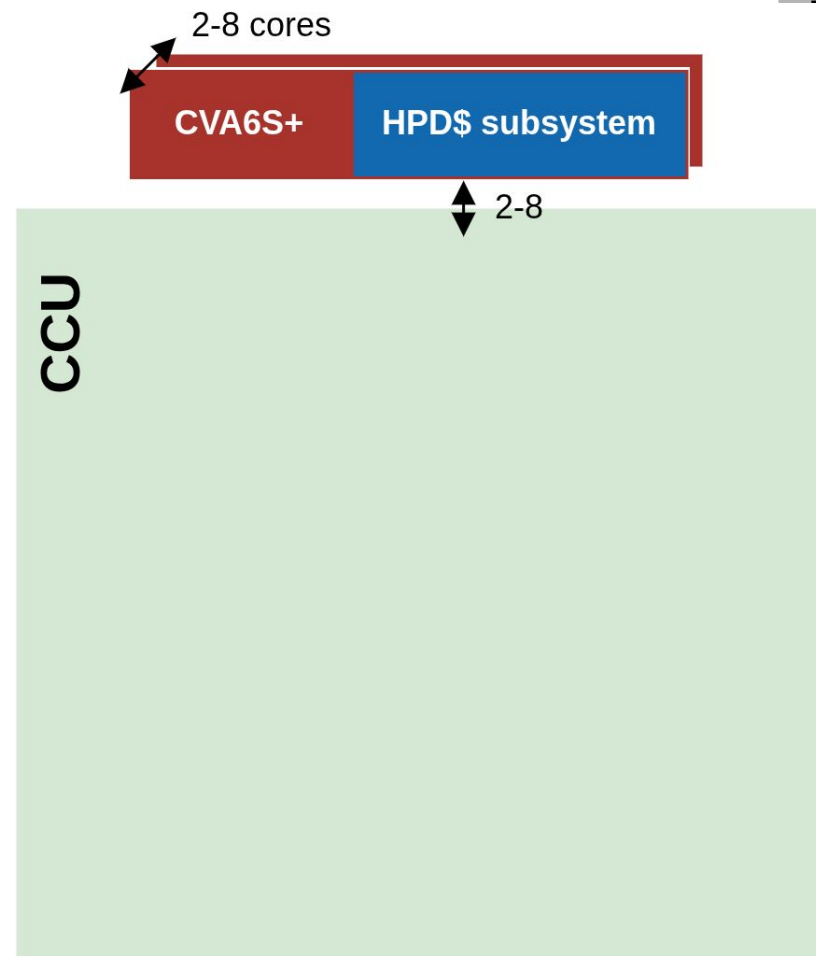
- **AXI to ACE transition** (Memory Adapter)

Enabling **direct cache-to-cache transfer of dirty data** across cores



CCU: architecture

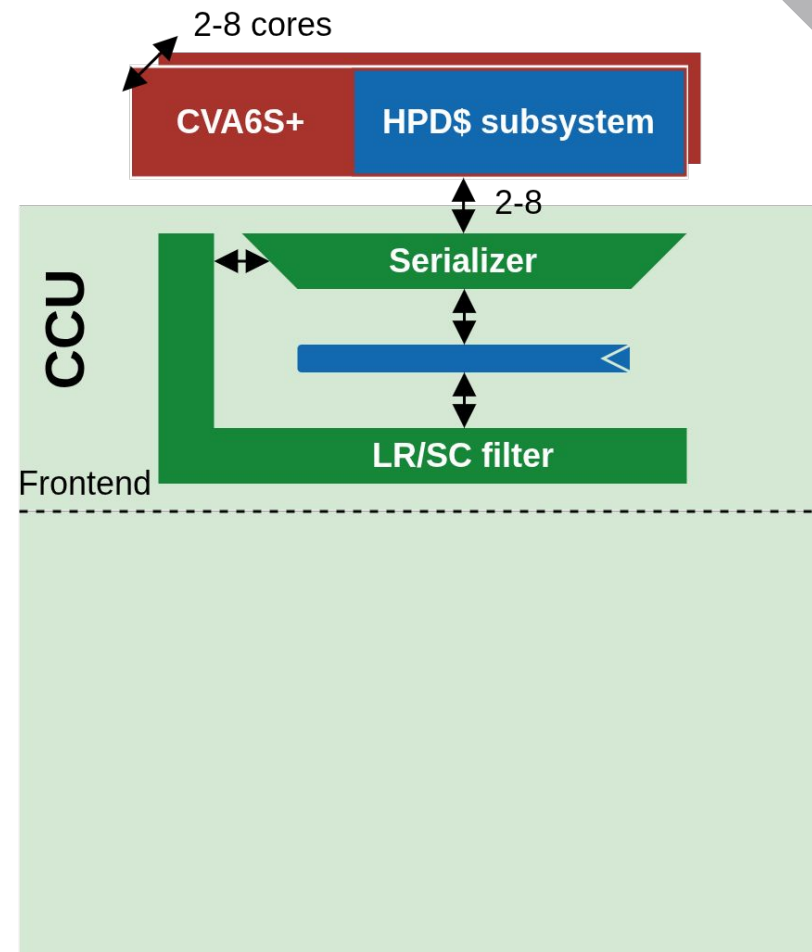
The **CCU** is a **new open-source IP** built around a **frontend/backend split** to implement a coherent interconnect:



CCU: architecture

The **CCU** is a **new open-source IP** built around a **frontend/backend split** to implement a coherent interconnect:

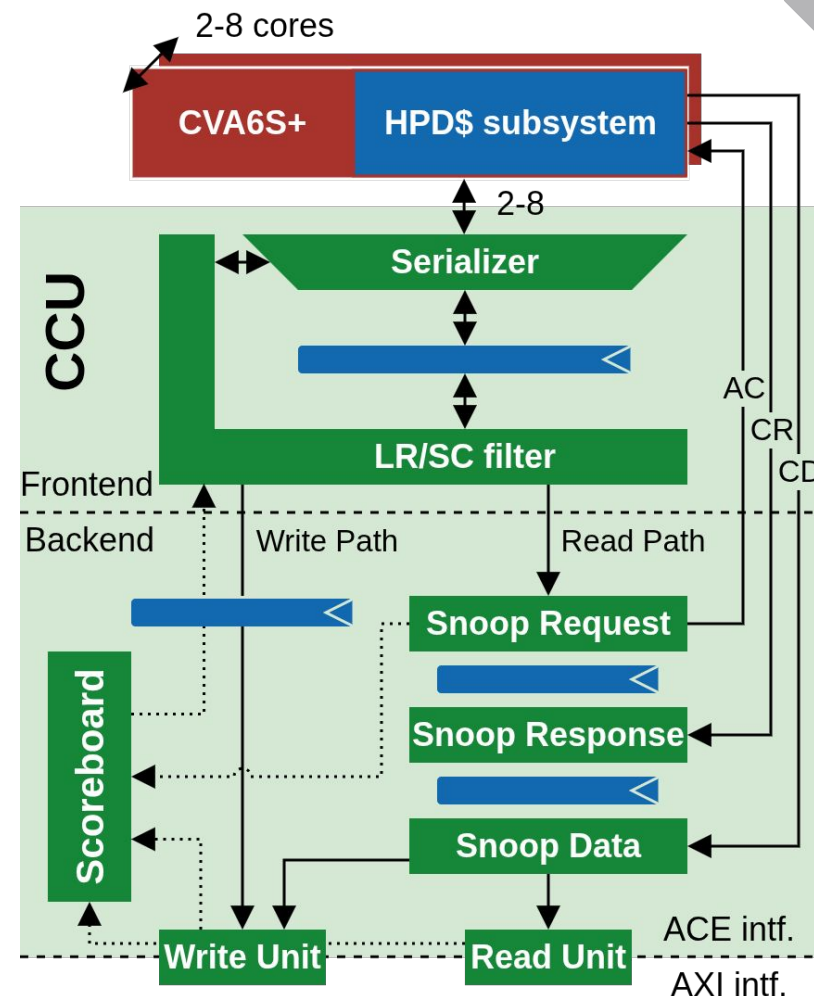
- **Core-side interface:** ACE
- **Frontend:** pipelined point of serialization for incoming transactions, LR/SC tracking



CCU: architecture

The **CCU** is a **new open-source IP** built around a **frontend/backend split** to implement a coherent interconnect:

- **Core-side interface:** ACE
- **Frontend:** pipelined point of serialization for incoming transactions, LR/SC tracking
- **Backend:** pipelined snoop channel handling (AC/CR/CD) and writebacks
- **Memory-side interface:** AXI

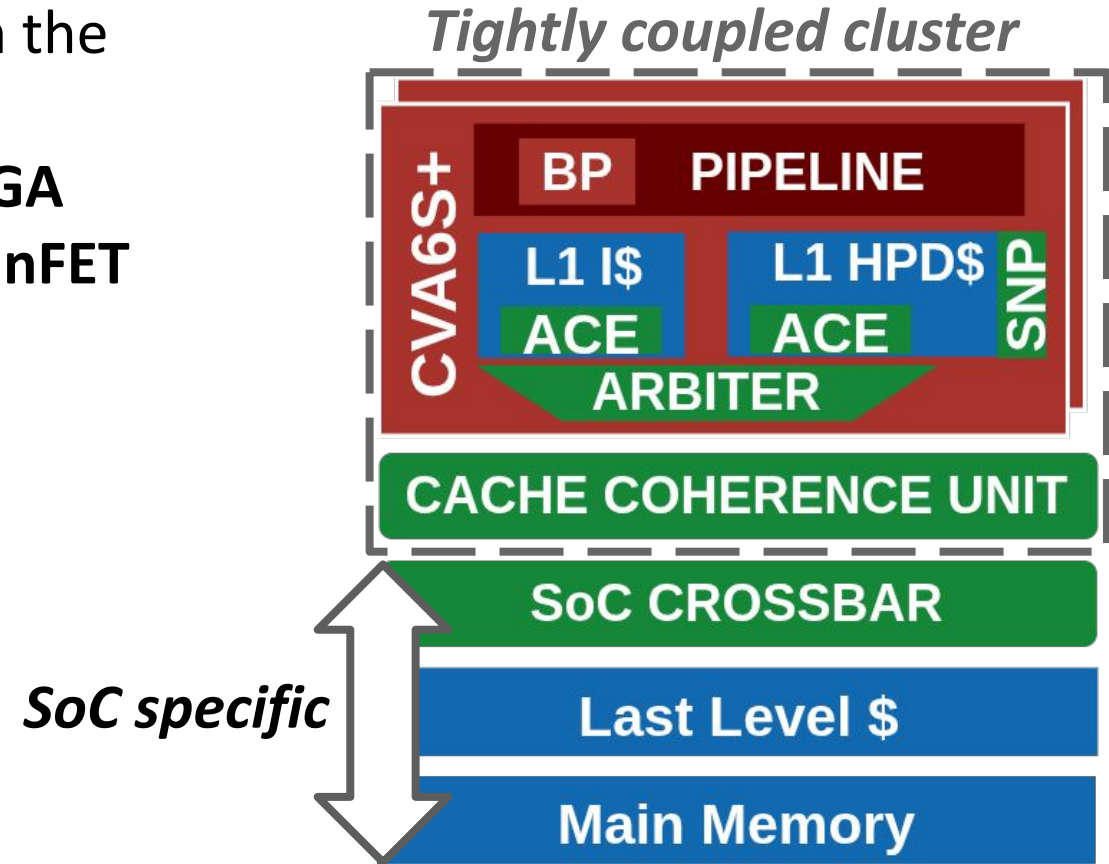


Culsans: experimental setup



The **dual-core coherent cluster** is integrated in the **Linux-capable Cheshire SoC**

- Functional validation on **Xilinx VCU128 FPGA**
- ASIC synthesis in **GlobalFoundries 12nm FinFET**



Culsans: experimental setup



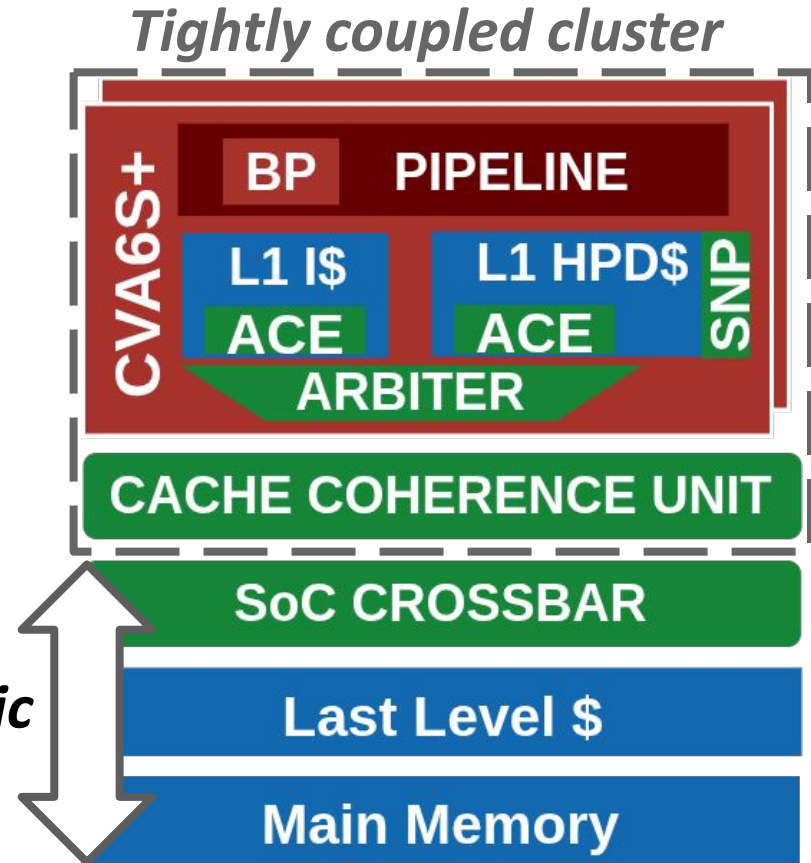
The **dual-core coherent cluster** is integrated in the **Linux-capable Cheshire SoC**

- Functional validation on **Xilinx VCU128 FPGA**
- ASIC synthesis in **GlobalFoundries 12nm FinFET**

Cheshire's **software stack** is extended to support:

- **SMP Linux**: full symmetric multiprocessing
- **RTEMS**: real-time POSIX compliant OS for mixed-criticality workloads

SoC specific



Culsans: experimental setup



The **dual-core coherent cluster** is integrated in the **Linux-capable Cheshire SoC**

- Functional validation on **Xilinx VCU128 FPGA**
- ASIC synthesis in **GlobalFoundries 12nm FinFET**

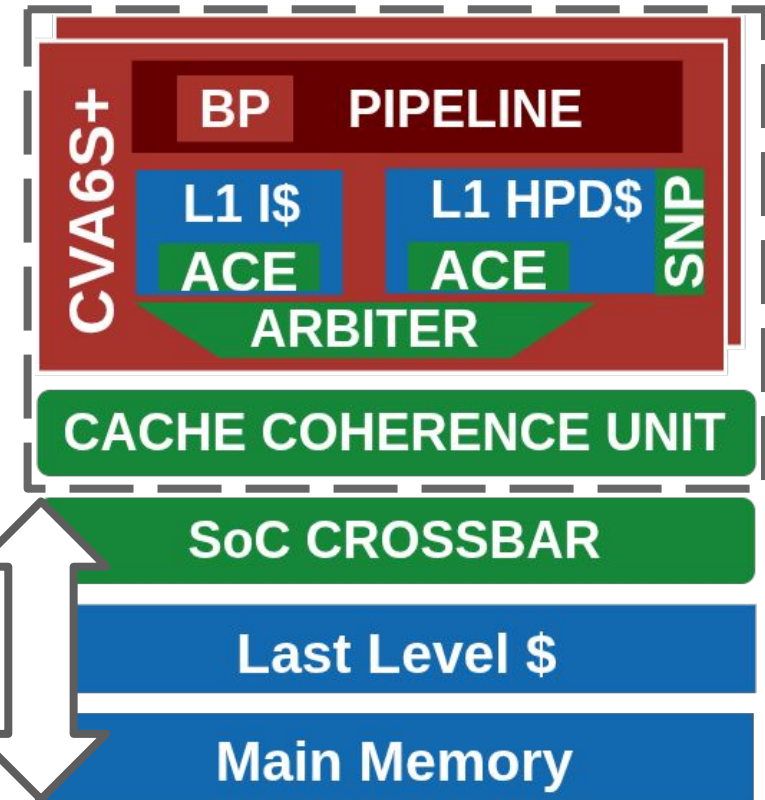
Cheshire's **software stack** is extended to support:

- **SMP Linux**: full symmetric multiprocessing
- **RTEMS**: real-time POSIX compliant OS for mixed-criticality workloads

Multithreaded benchmarks:

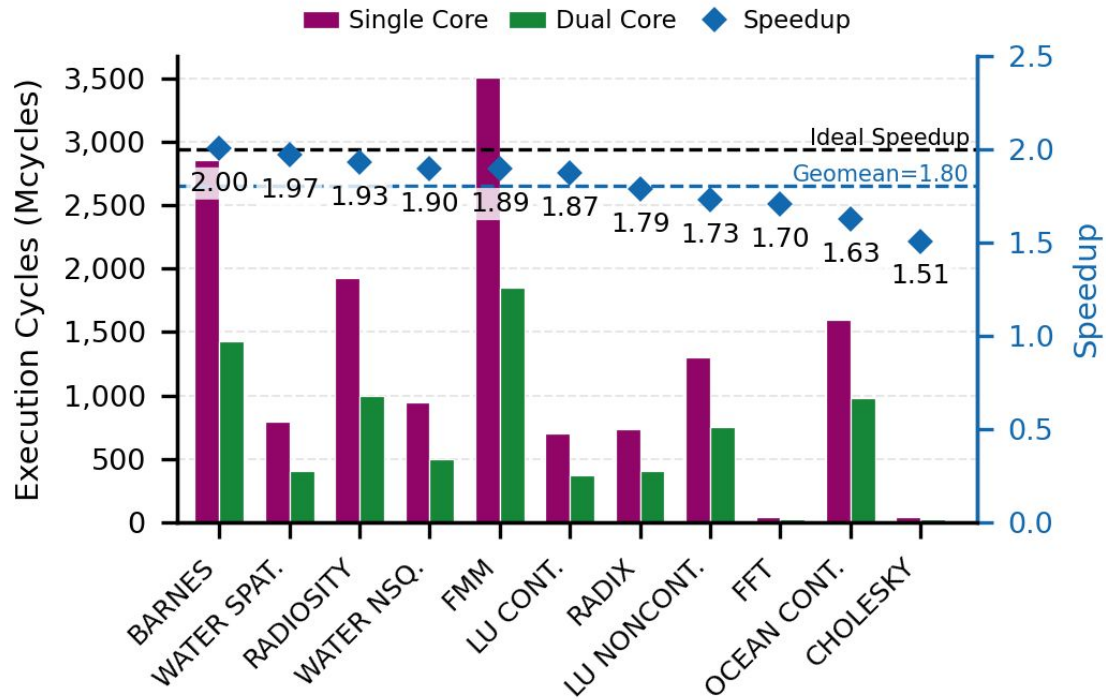
- **Splash-3**: suite of **11 benchmarks**, parallelized via **pthread**

Tightly coupled cluster



SoC specific

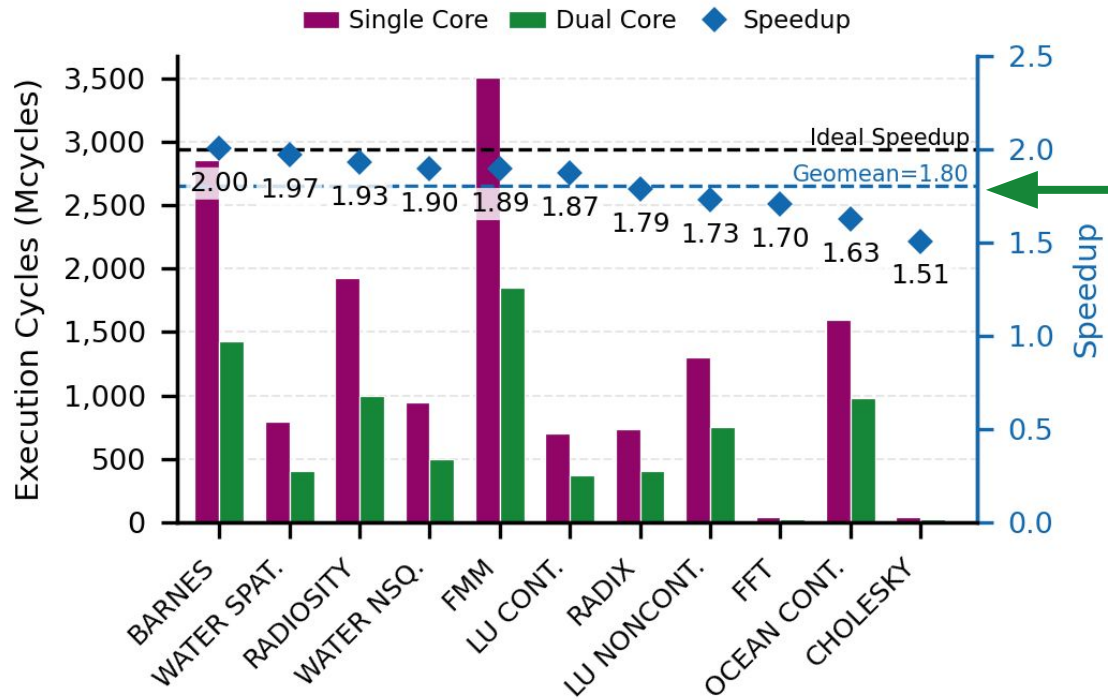
Culsans: performance evaluation



We evaluate the **speedup** of the **dual-core cluster** over a **single core baseline**.

- Benchmarks: **Splash3 on SMP Linux**

Culsans: performance evaluation

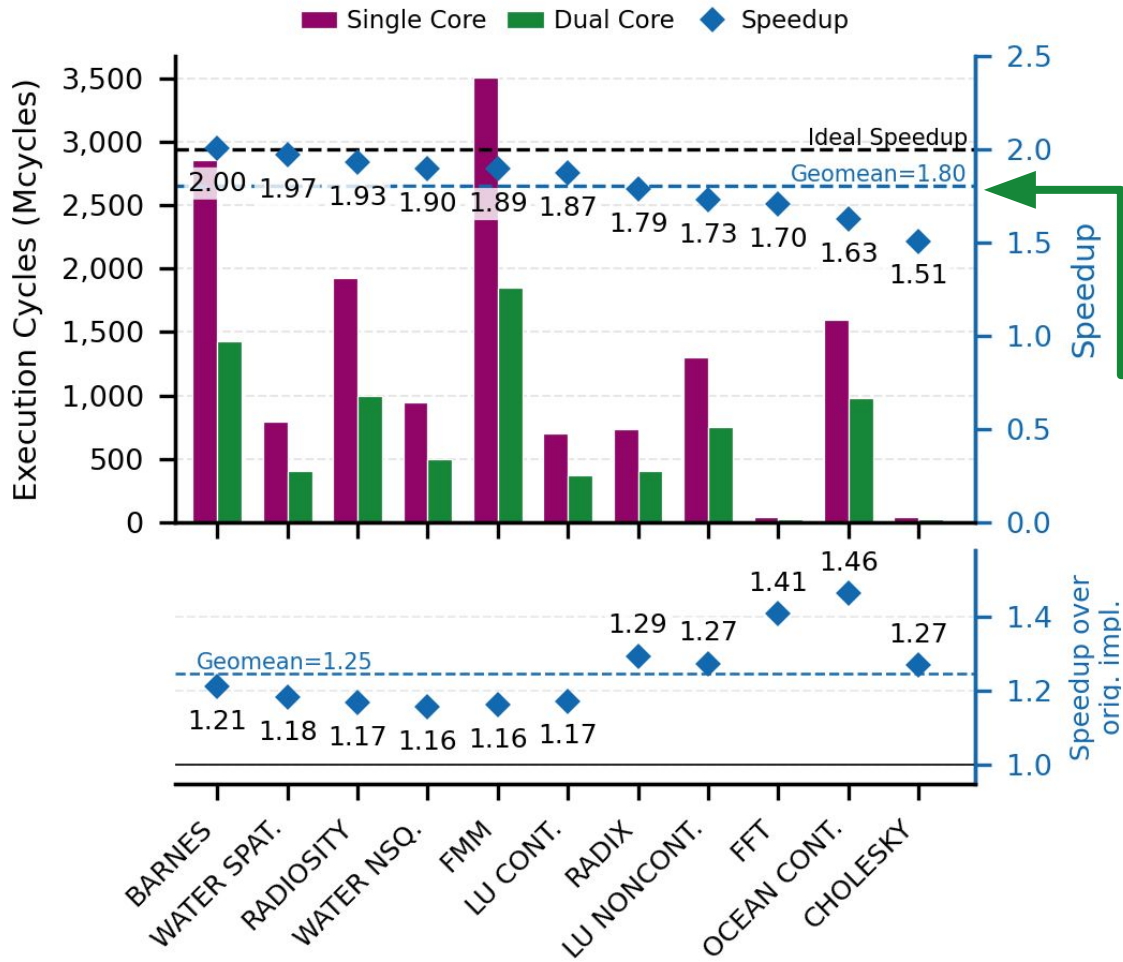


We evaluate the **speedup** of the **dual-core cluster** over a **single core baseline**.

- Benchmarks: **Splash3 on SMP Linux**

1.80x geomean speedup over single core

Culsans: performance evaluation



We evaluate the **speedup** of the **dual-core cluster** over a **single core baseline**.

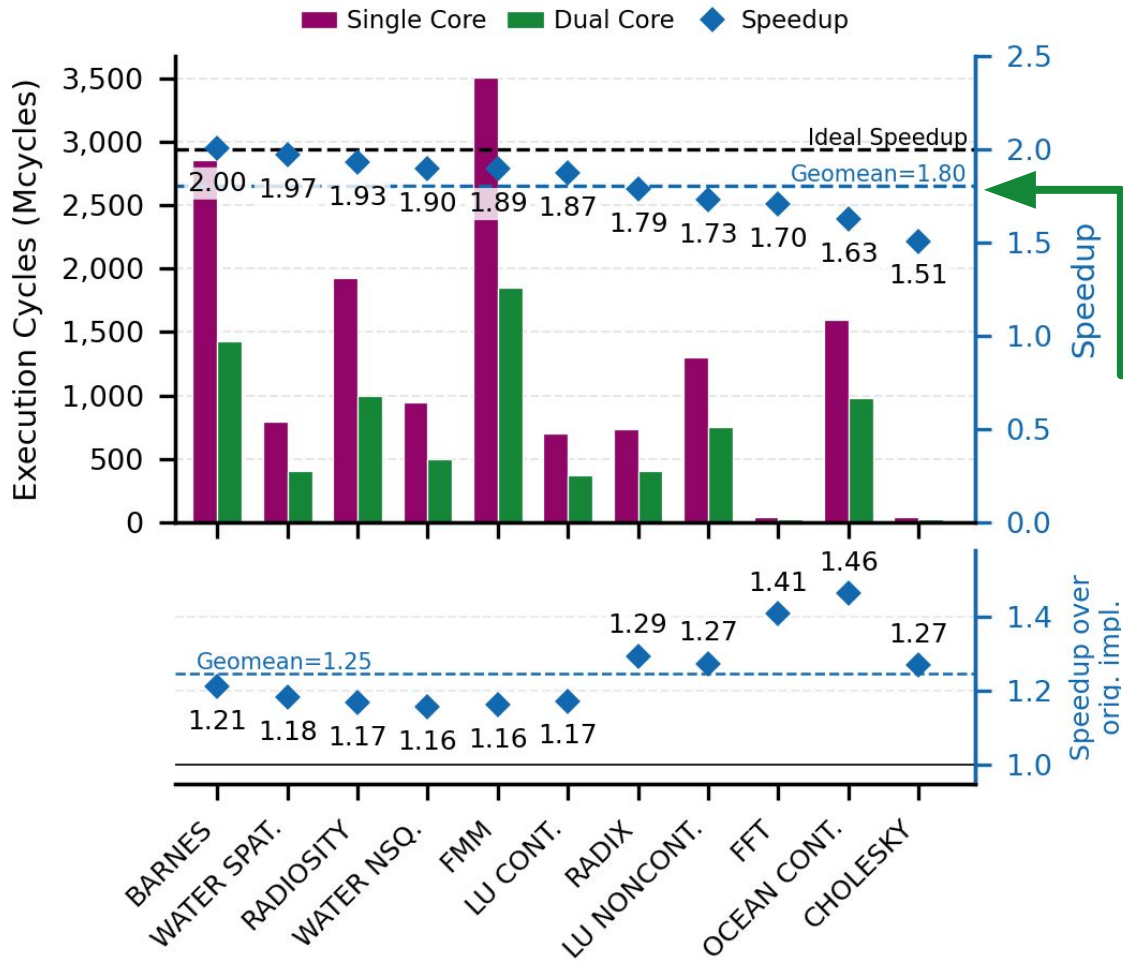
- Benchmarks: **Splash3 on SMP Linux**

1.80× geomean speedup over single core

We evaluate the **speedup** of the **dual-core cluster** using **{CVA6S+, HPDCache}** versus **{CVA6, legacy WB cache}**¹:

¹WIP paper: Tedeschi, R., et al. "Culsans: An Efficient Snoop-based Coherency Unit for the CVA6 Open Source RISC-V application processor," in *WiPIEC Journal - Works in Progress in Embedded Computing Journal*, vol. 10, no. 2, pp. 4, 2024.

Culsans: performance evaluation



We evaluate the **speedup** of the **dual-core cluster** over a **single core baseline**.

- Benchmarks: **Splash3 on SMP Linux**

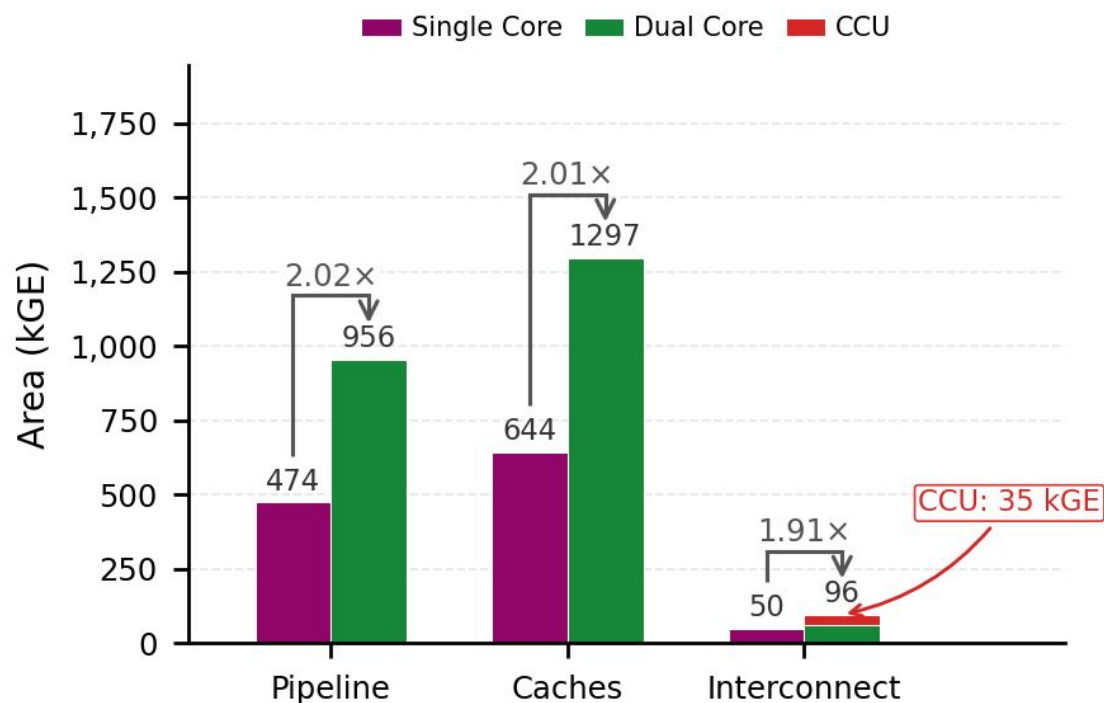
1.80× geomean speedup over single core

We evaluate the **speedup** of the **dual-core cluster** using **{CVA6S+, HPDCache}** versus **{CVA6, legacy WB cache}**¹:

1.25× speedup with CVA6S+ and HPD\$

¹WIP paper: Tedeschi, R., et al. "Culsans: An Efficient Snoop-based Coherency Unit for the CVA6 Open Source RISC-V application processor," in *WIPIEC Journal - Works in Progress in Embedded Computing Journal*, vol. 10, no. 2, pp. 4, 2024.

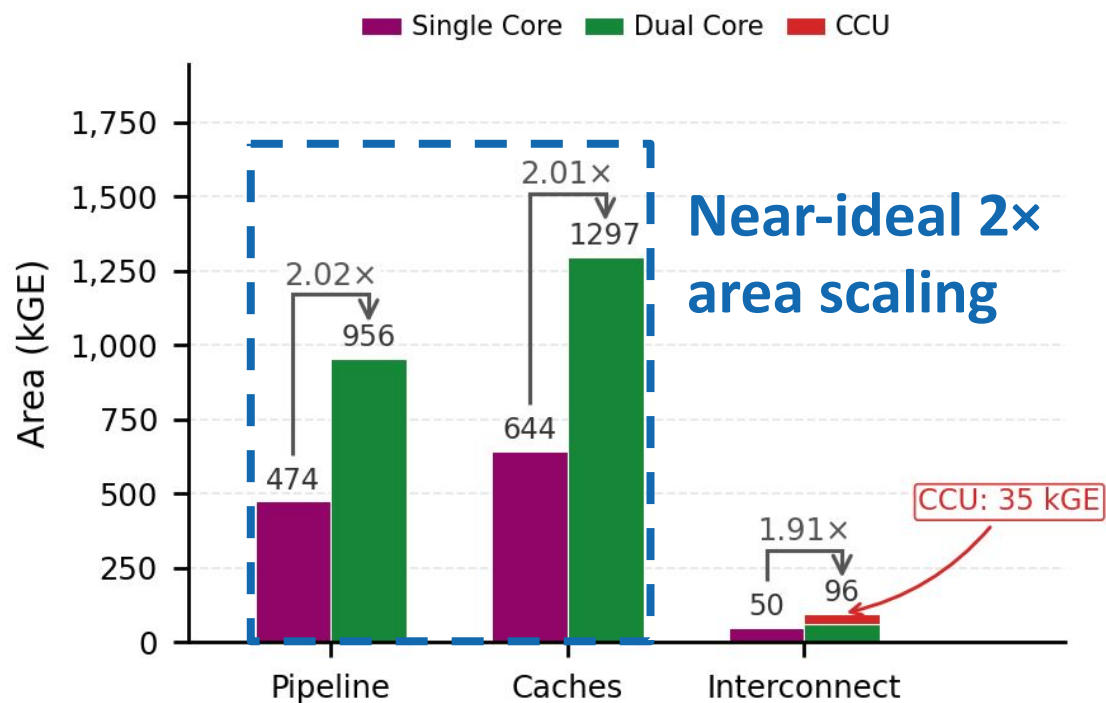
Culsans: area estimation



We evaluate the area of the **dual-core cluster** over a **single core baseline**.

- Topographical Synthesis
- Tech: GF 12 nm FinFET
- Corner: SS, 0.72V, -40°C

Culsans: area estimation



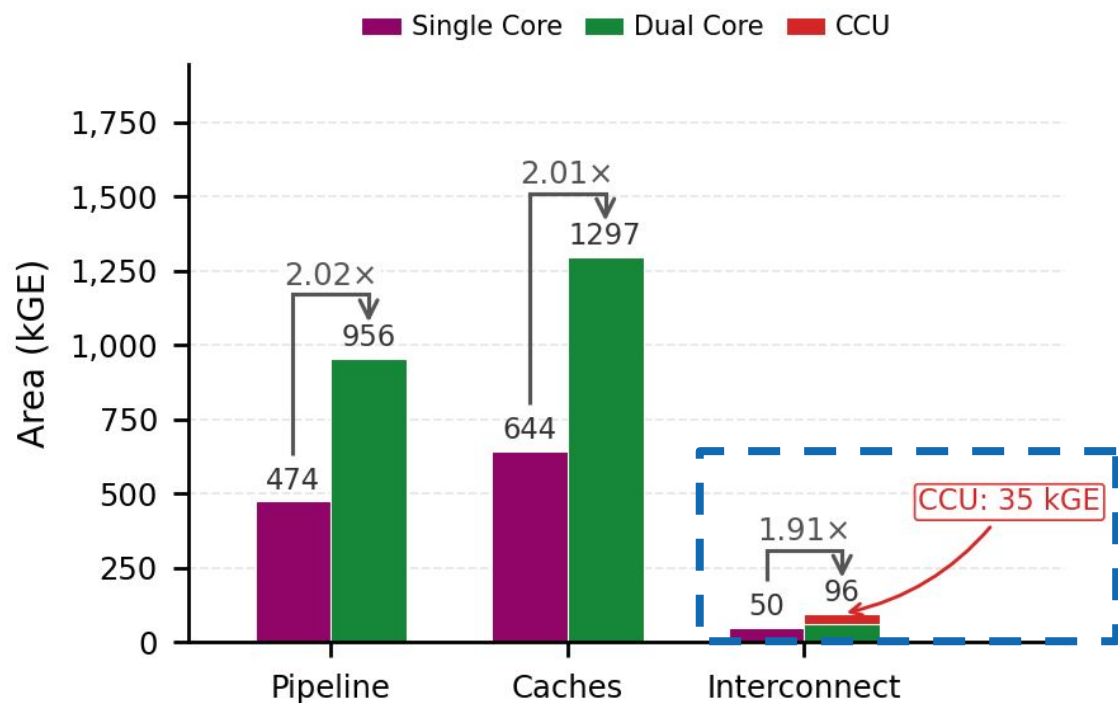
Near-ideal 2x area scaling

We evaluate the area of the **dual-core cluster** over a **single core baseline**.

- **Pipeline:** <2% per-core **area overhead**
- **Caches:** **minimal cost** for **ACE support**

- **Topographical Synthesis**
- **Tech:** GF 12 nm FinFET
- **Corner:** SS, 0.72V, -40°C

Culsans: area estimation

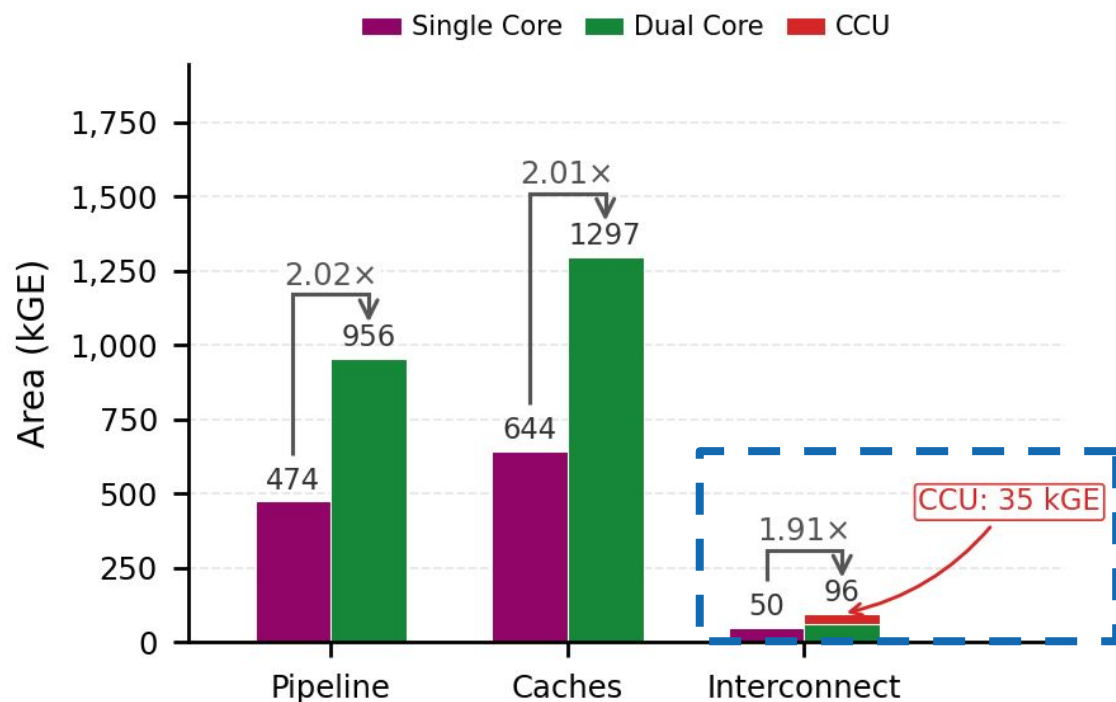


We evaluate the area of the **dual-core cluster** over a **single core baseline**.

- **Pipeline:** <2% per-core **area overhead**
- **Caches:** **minimal cost** for **ACE support**
- **Interconnect:** **35 kGE CCU** overhead

- **Topographical Synthesis**
- **Tech:** GF 12 nm FinFET
- **Corner:** SS, 0.72V, -40°C

Culsans: area estimation



We evaluate the area of the **dual-core cluster** over a **single core baseline**.

- **Pipeline: <2%** per-core **area overhead**
- **Caches: minimal cost** for **ACE support**
- **Interconnect: 35 kGE CCU** overhead
- **No frequency degradation** (850 MHz)

- **Topographical Synthesis**
- **Tech: GF 12 nm FinFET**
- **Corner: SS, 0.72V, -40°C**

Conclusion



We introduce **Culsans**, an open-source cache-coherent cluster for automotive 64b MPUs, built on OpenHW's **CVA6S+** and **HPDCache**.

Conclusion



We introduce **Culsans**, an open-source cache-coherent cluster for automotive 64b MPUs, built on OpenHW's **CVA6S+** and **HPDCache**.

- **CVA6S+ branch predictor optimization:** 10-36% predictor area reduction with no accuracy loss
- **HPDCache extended** to support AMBA4 ACE coherent transactions
- **New open-source CCU** enforcing snoop-based MOESI coherence over AMBA4 ACE

Conclusion



We introduce **Culsans**, an open-source cache-coherent cluster for automotive 64b MPUs, built on OpenHW's **CVA6S+** and **HPDCache**.

- **CVA6S+ branch predictor optimization:** 10-36% predictor area reduction with no accuracy loss
- **HPDCache extended** to support AMBA4 ACE coherent transactions
- **New open-source CCU** enforcing snoop-based MOESI coherence over AMBA4 ACE
- The cluster achieves **1.80× geomean speedup** on **Splash-3** vs. single core
- **<2% per-core area overhead**, with the CCU requiring only **35 kGE**

Conclusion



We introduce **Culsans**, an open-source cache-coherent cluster for automotive 64b MPUs, built on OpenHW's **CVA6S+** and **HPDCache**.

- **CVA6S+ branch predictor optimization**: 10-36% predictor area reduction with no accuracy loss
- **HPDCache extended** to support AMBA4 ACE coherent transactions
- **New open-source CCU** enforcing snoop-based MOESI coherence over AMBA4 ACE
- The cluster achieves **1.80× geomean speedup** on **Splash-3** vs. single core
- **<2% per-core area overhead**, with the CCU requiring only **35 kGE**

Future work: scalability to 4-8 cores, power characterization, functional safety

Thank you! Questions?



Riccardo Tedeschi
riccardo.tedeschi6@unibo.it



DEI – Università di Bologna
Viale del Risorgimento 2
Bologna, Italy

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

@pulp_platform 

pulp-platform.org 

youtube.com/pulp_platform 



youtube.com/pulp_platform