ARCANE: Enabling High-Performance In-Cache Tensor Extensions in RISC-V

Flavia Guella, Vincenzo Petrolo, Michele Caon,
Guido Masera, Maurizio Martina

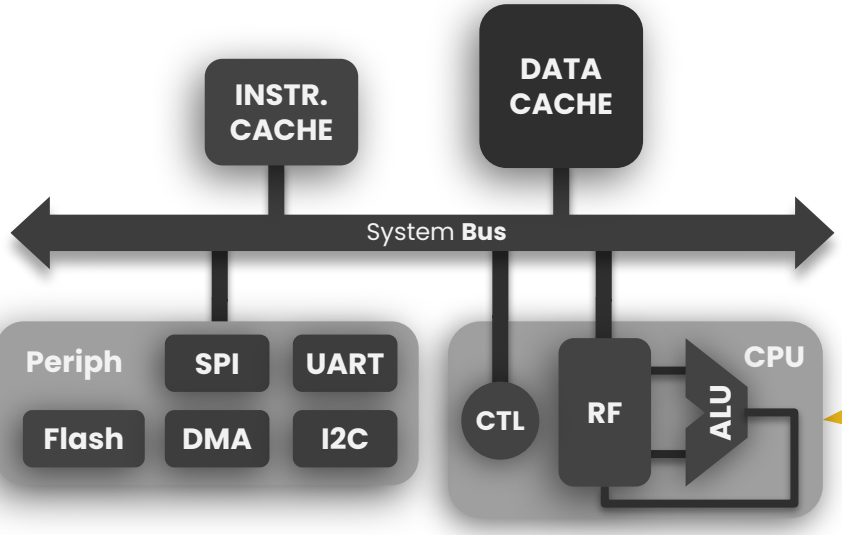
Politecnico di Torino

EPFL

RISC-V Summit Europe 2026,
Bologna, Italy

RISC-V®

Traditional SoC



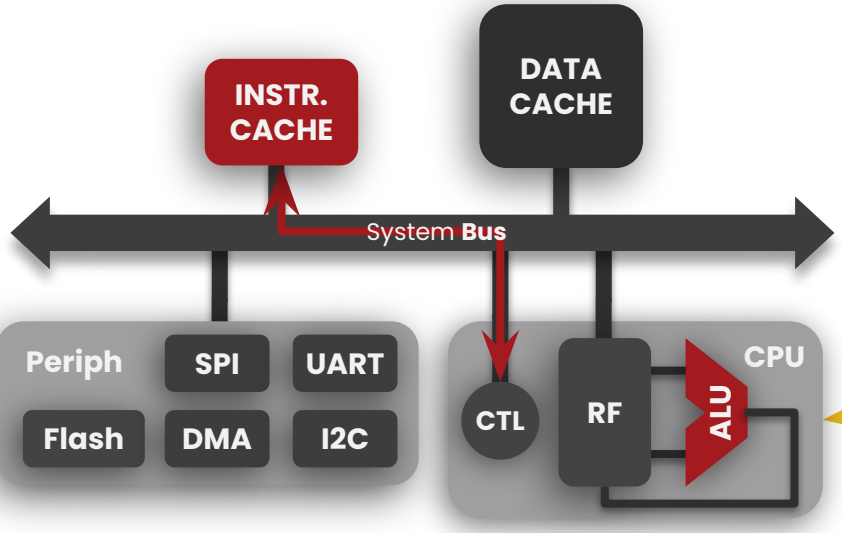
Data-driven tasks

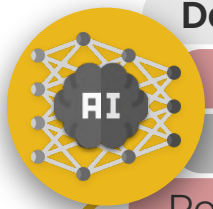
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

Traditional SoC





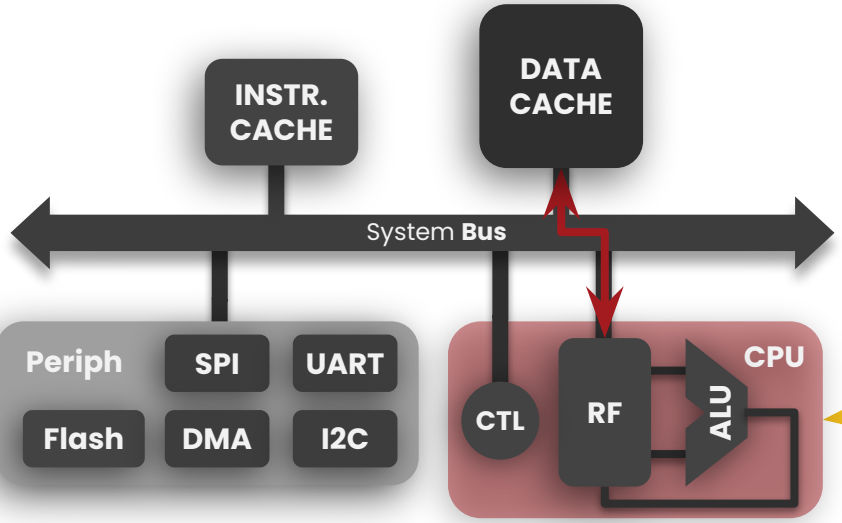
Data-driven tasks

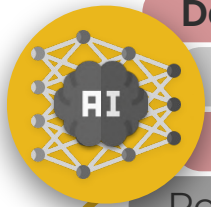
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

Traditional SoC





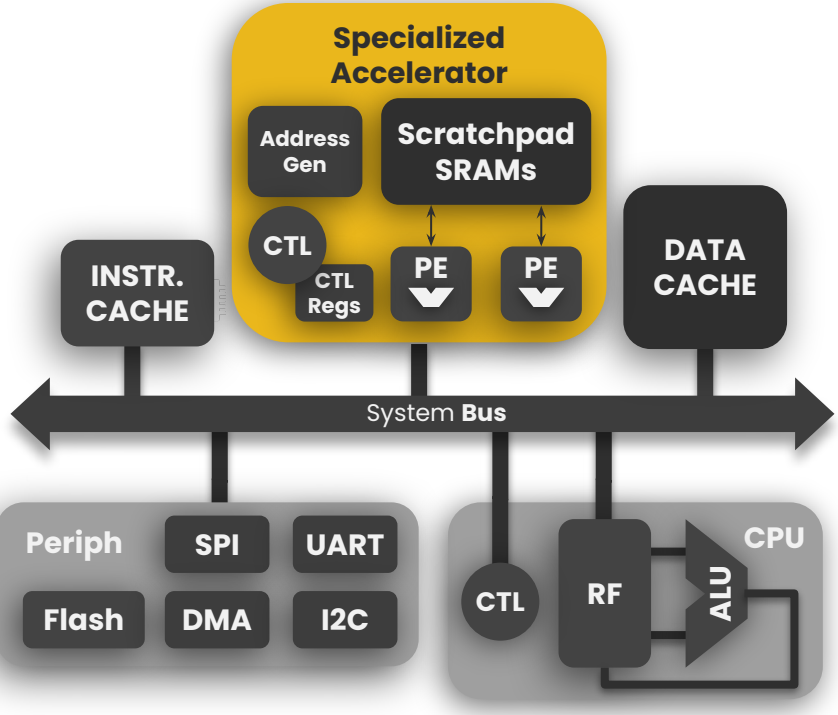
Data-driven tasks

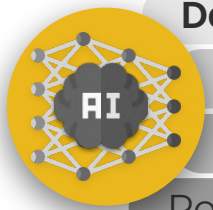
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

Heterogeneous Systems





Data-driven tasks

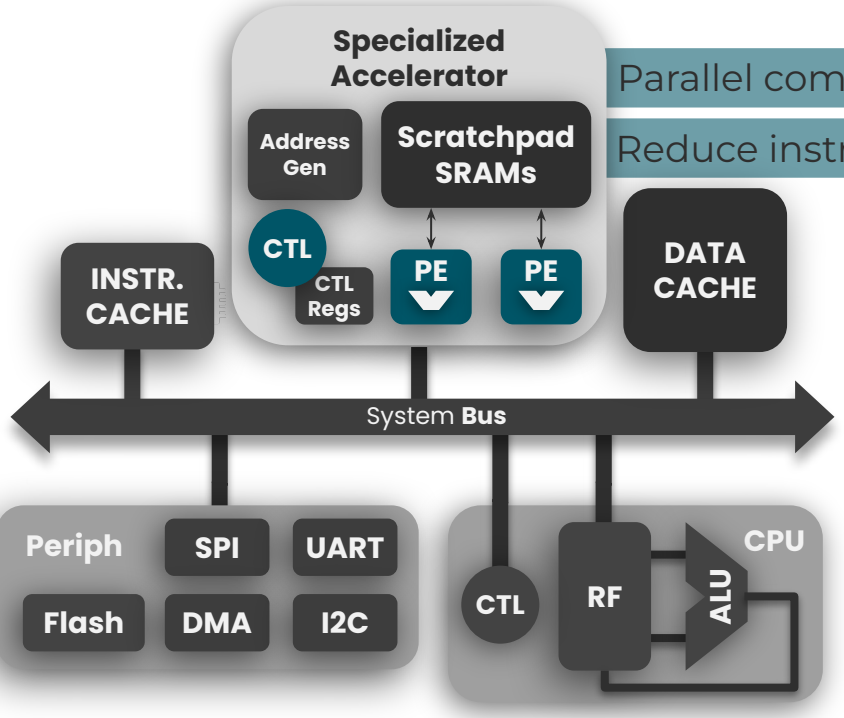
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

Heterogeneous Systems

Strengths



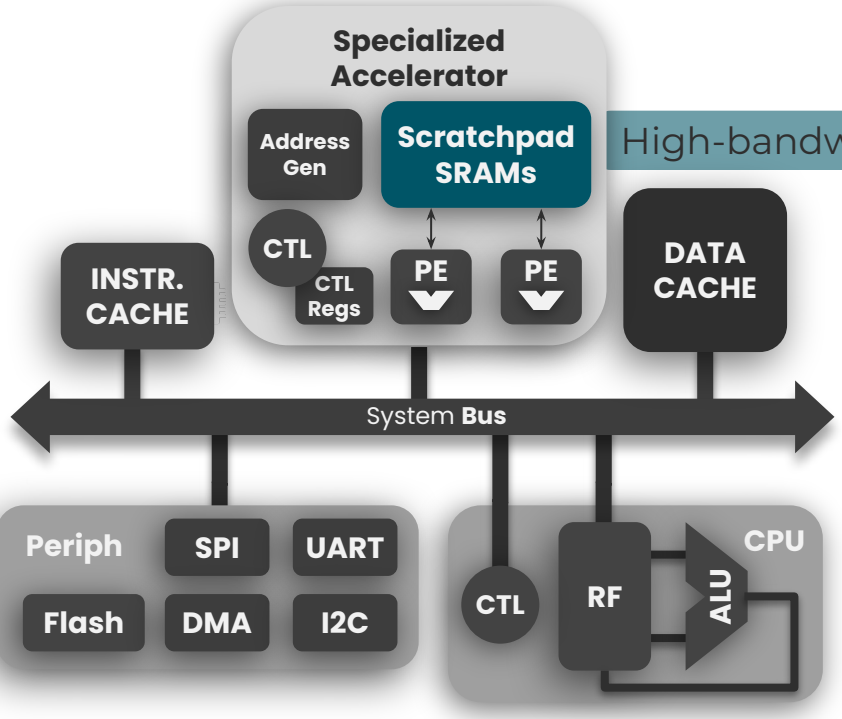
- Parallel computation
- Reduce instruction fetch cost

Data-driven tasks

- Massive **computation** requirements
- Memory **bandwidth** limits
- Poor energy **efficiency**

Heterogeneous Systems

Strengths



High-bandwidth scratchpad

Data-driven tasks

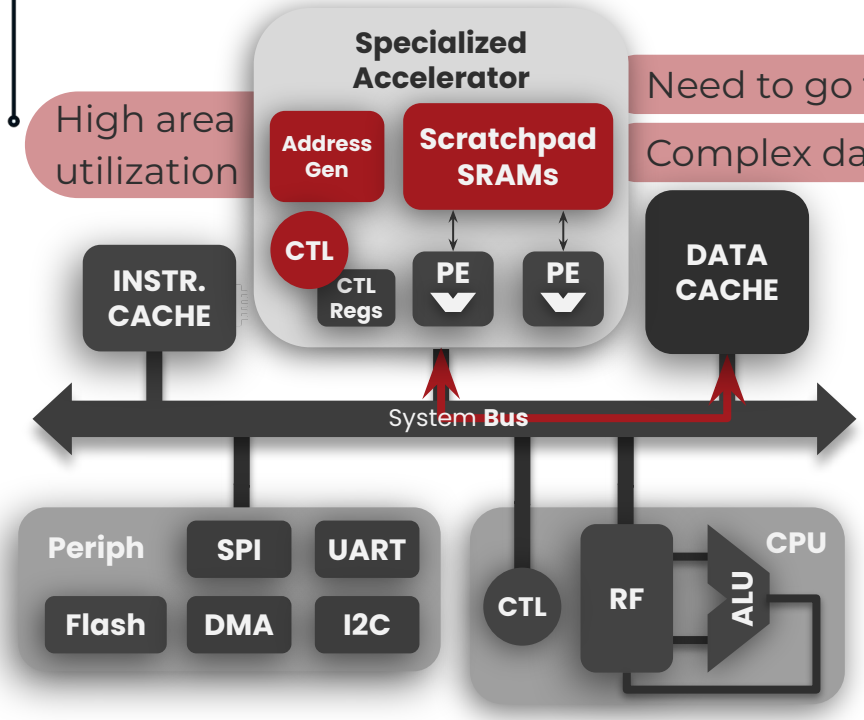
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

Heterogeneous Systems

Drawbacks



High area utilization

Need to go through system bus

Complex data management

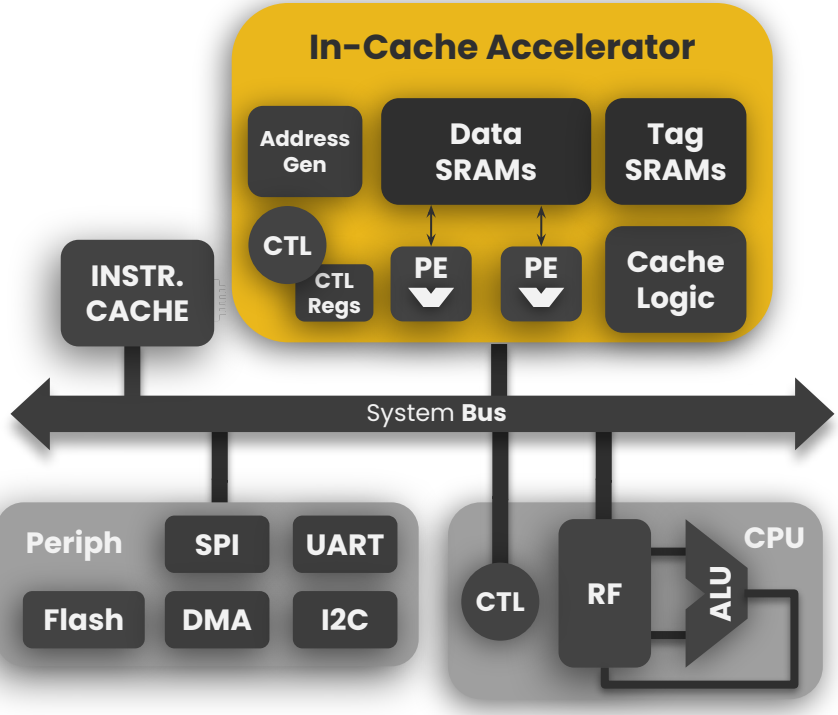
Data-driven tasks

Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

In-Cache Computing



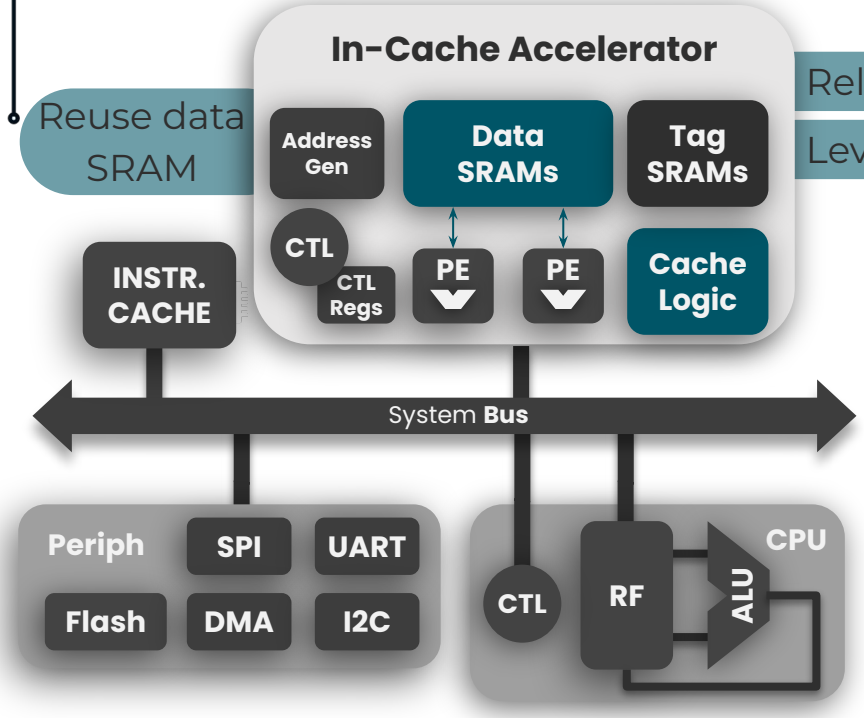
Data-driven tasks

Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

In-Cache Computing



Reuse data SRAM

Relieve system bus

Leverage cache logic

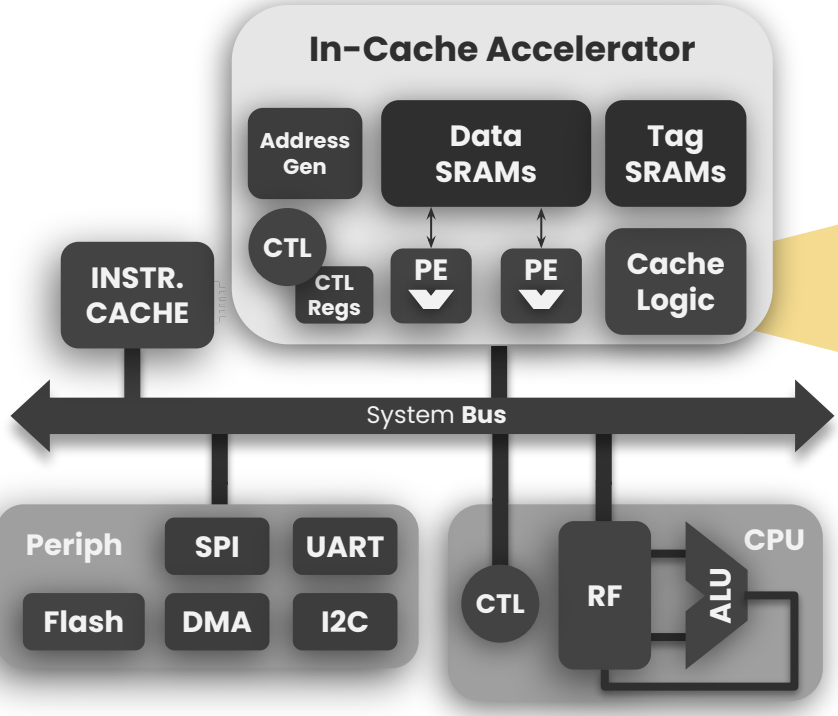
Data-driven tasks

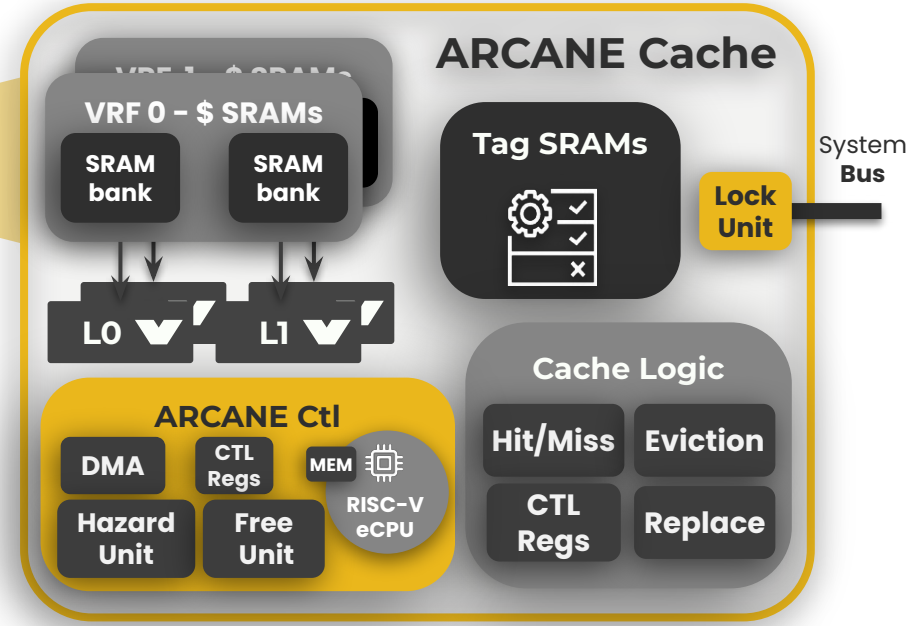
Massive **computation** requirements

Memory **bandwidth** limits

Poor energy **efficiency**

ARCANE: In-Cache Computing



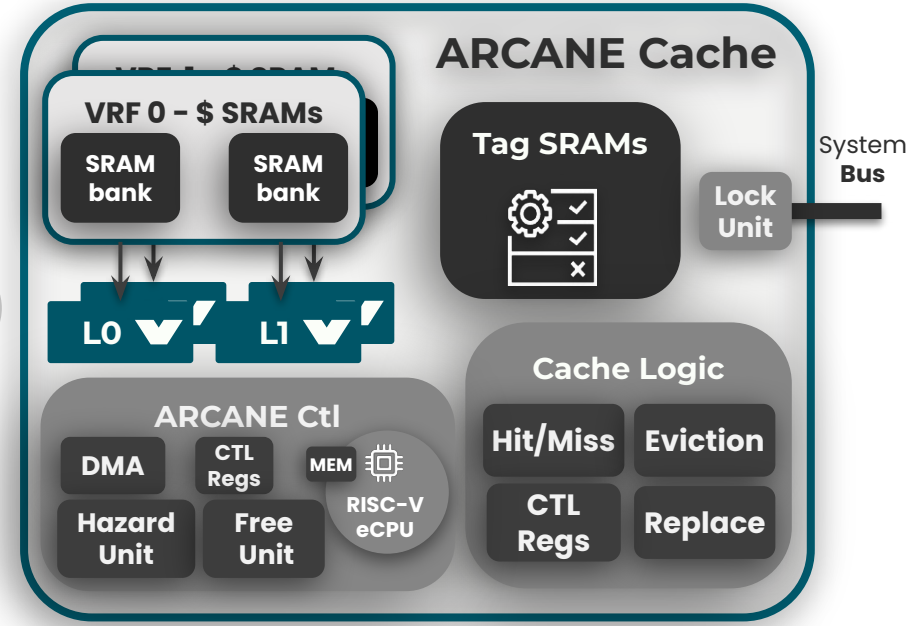


Computing Units

Multi-lane Vector Processing Units

Sliced Vector Register File reuses data-cache SRAMs

Tunable VRF size, #lanes, #VPUs

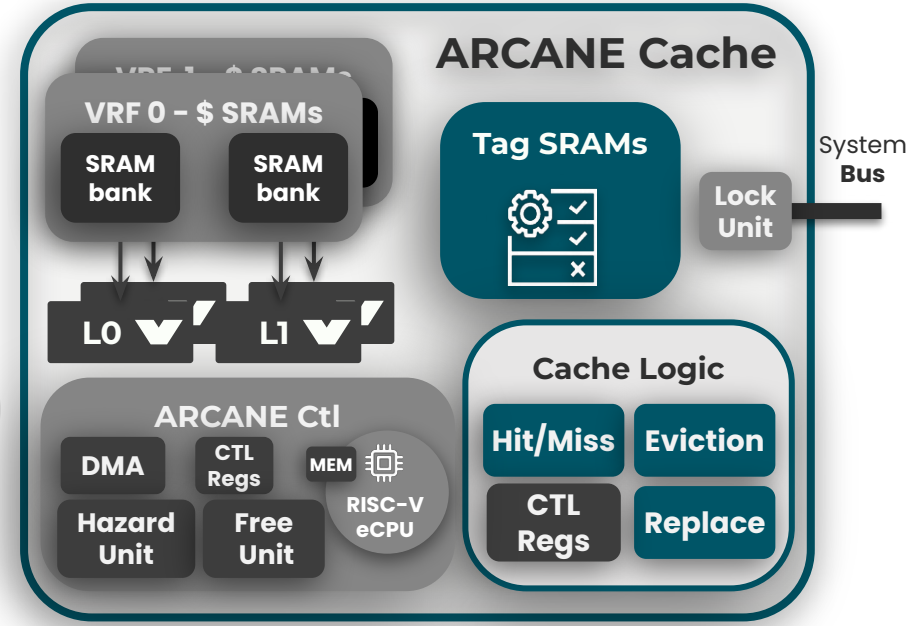


Cache Logic

Additional Status Flag

Concurrent computing and cache mode

Fine-grained selection of usable cache lines and lines busy computing

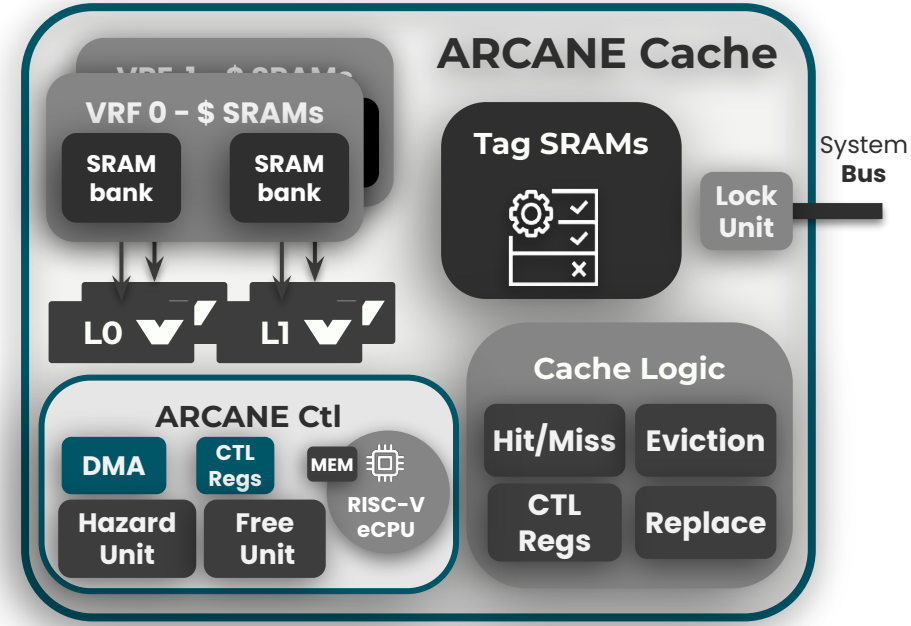


Data Management Logic

Data Management Logic

Reorganizes data according to VRF layout for desired kernel

Uses cache logic to check for hit on requested data



Data Management and Cache

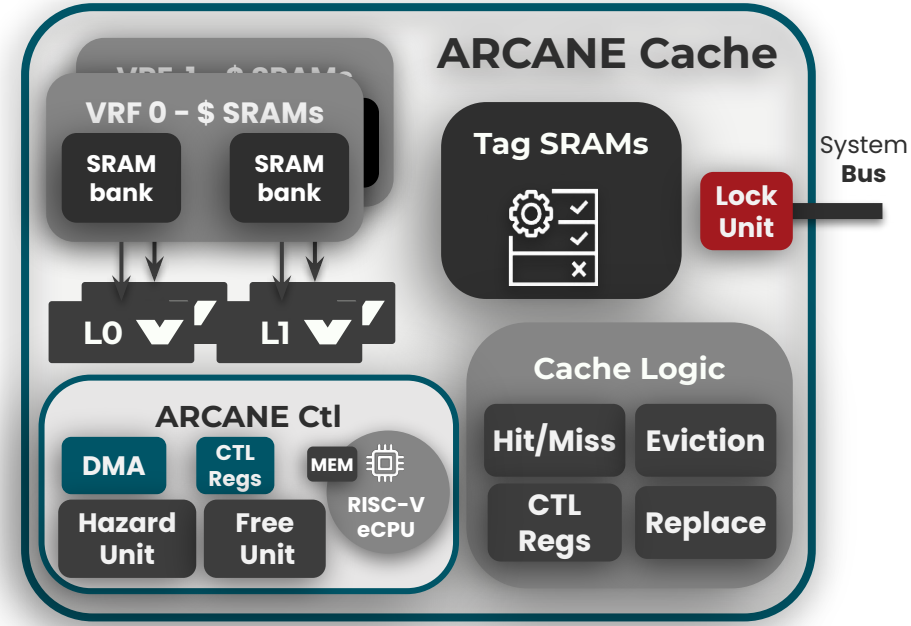
Data Management Logic

Reorganizes data according to VRF layout for desired kernel

Uses cache logic to check for hit on requested data

STOP

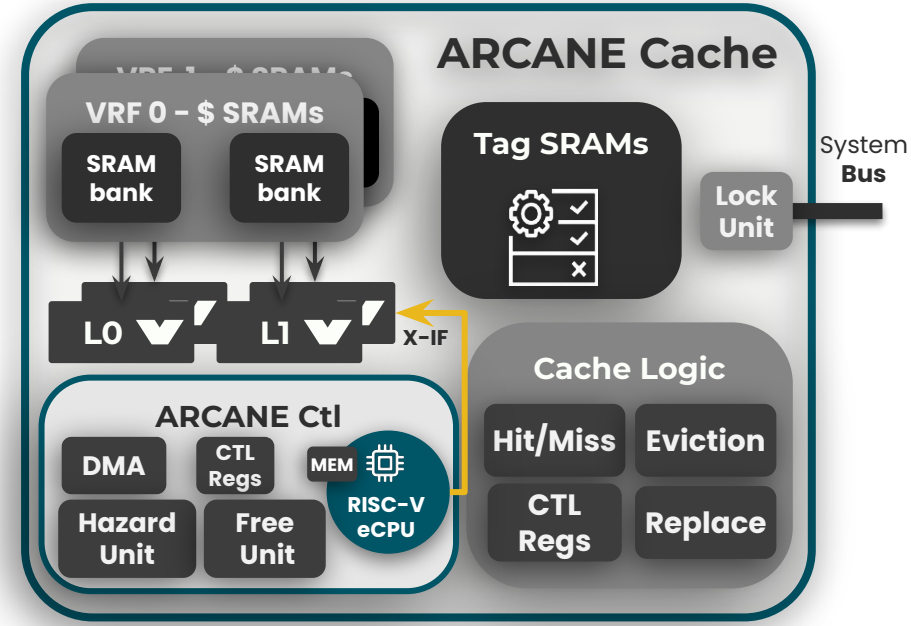
During data movement cache cannot answer system requests



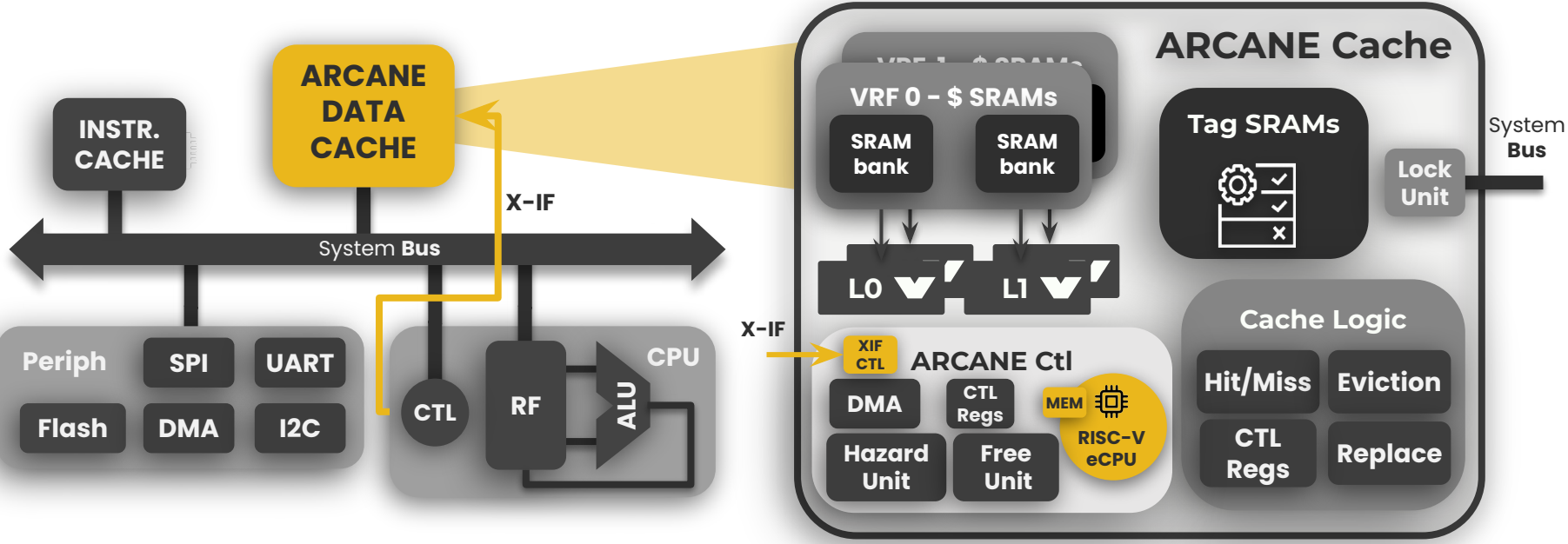
Computing and Control

RISC-V Embedded CPU

- Custom subset of Zve32x (C-V-XIF)
- Orchestrates other modules



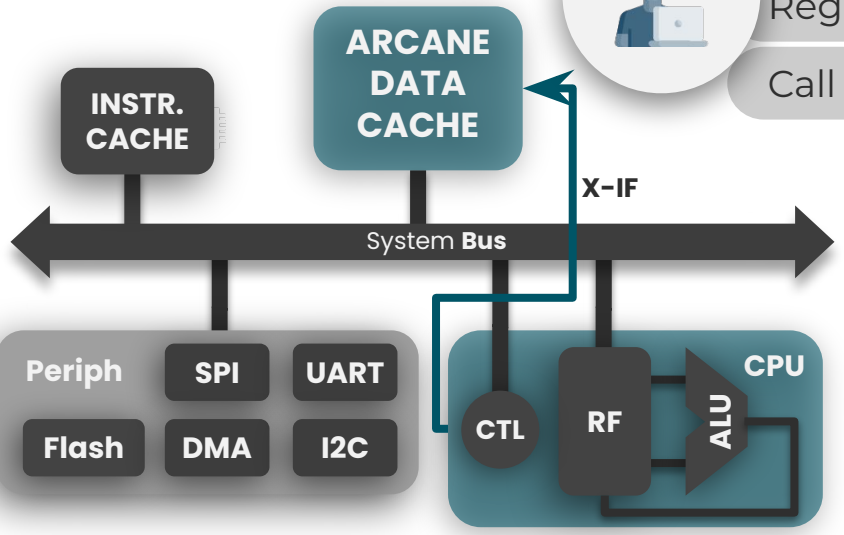
ARCANE: In-Cache Computing



Programming Model

Tensor-based ISA extension

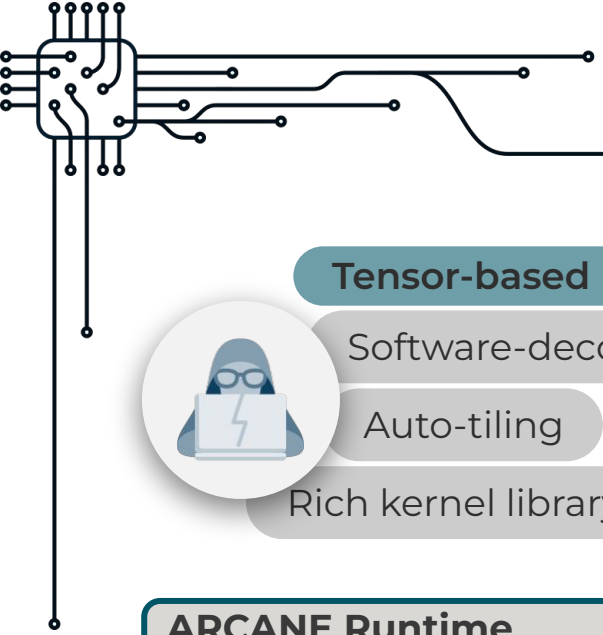
- Software-defined ISA
- Register 4-dimensional tensors
- Call a registered kernel



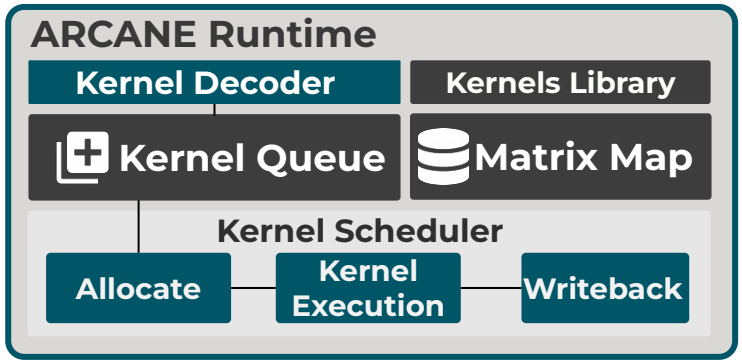
main.c

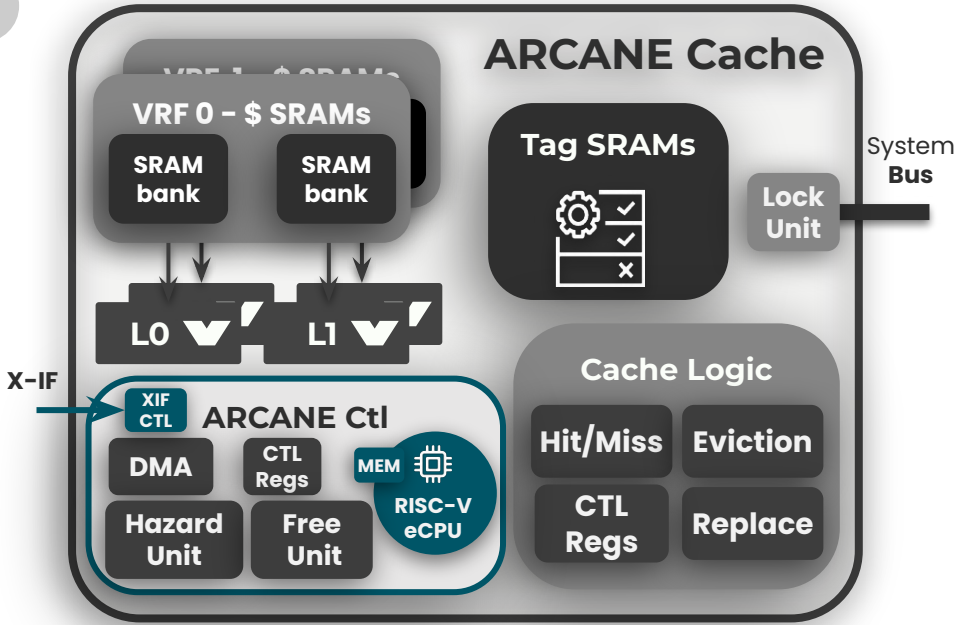
```c
int main(void) {
int A[rowsA][colsA] = {...};
int F[rowsF][colsF] = {...};
int R[rowsR][colsR];

// Reservation
_xmr_w(m0, A, rowsA, colsA);
_xmr_w(m1, F, rowsF, colsF);
_xmr_w(m2, R, rowsR, colsR);

// Matrix Kernel
_xmk0(m2, m0, m1);
...

// Access to results
print(R[0][0]);
...
```

Programming Model



- **Tensor-based ISA extension**
- Software-decoded instructions
- Auto-tiling
- Rich kernel library

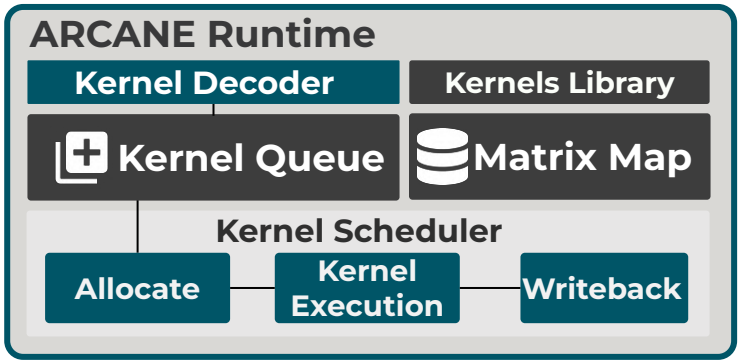


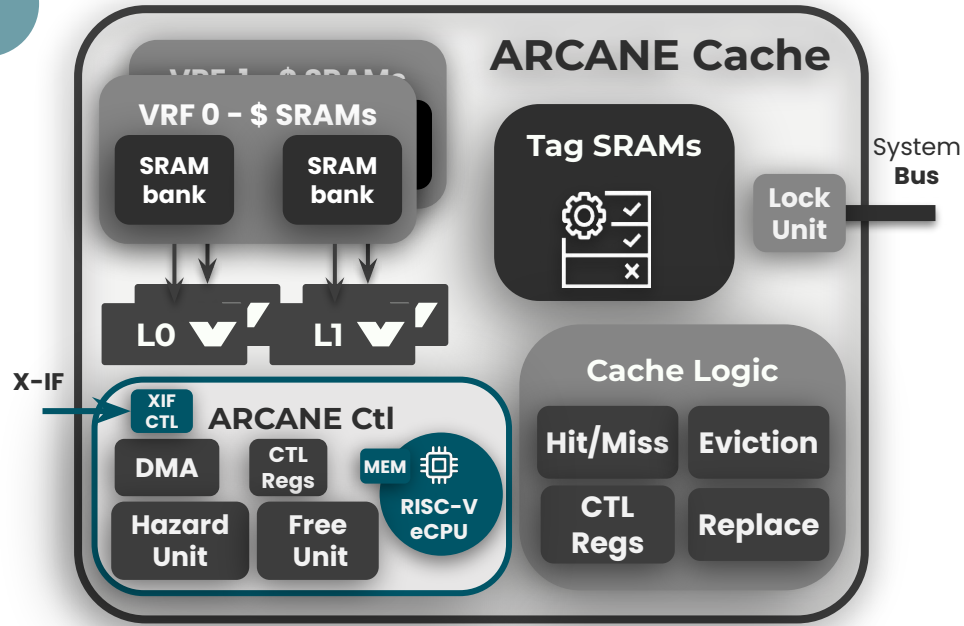


Programming Model

Programmable

Data-management is now **transparent** to the host CPU





Programming Model

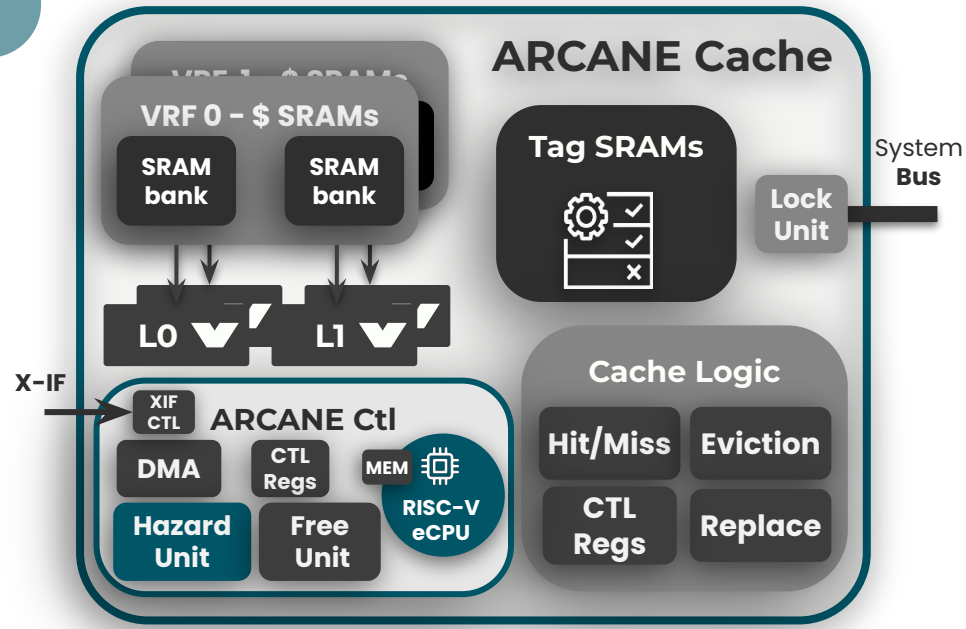
Programmable

Data-management is now **transparent** to the host CPU

No explicit synchronization

main.c

```
int main(void) {
int A[rowsA][colsA] = {...};
int F[rowsF][colsF] = {...};
int R[rowsR][colsR];

// Reservation
_xmr_w(m0, A, rowsA, colsA);
_xmr_w(m1, F, rowsF, colsF);
_xmr_w(m2, R, rowsR, colsR);

// Matrix Kernel
_xmk0(m2, m0, m1);
...

// Access to results
print(R[0][0]);
...
```



Programming Model

Programmable

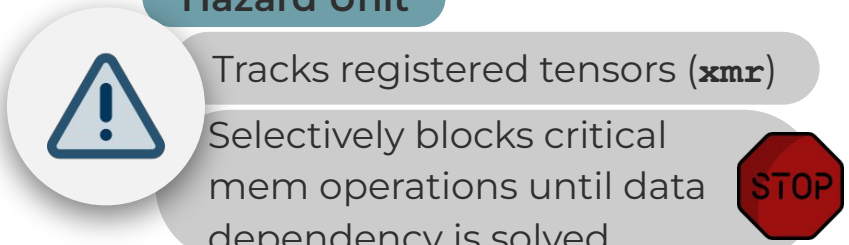
Data-management is now **transparent** to the host CPU

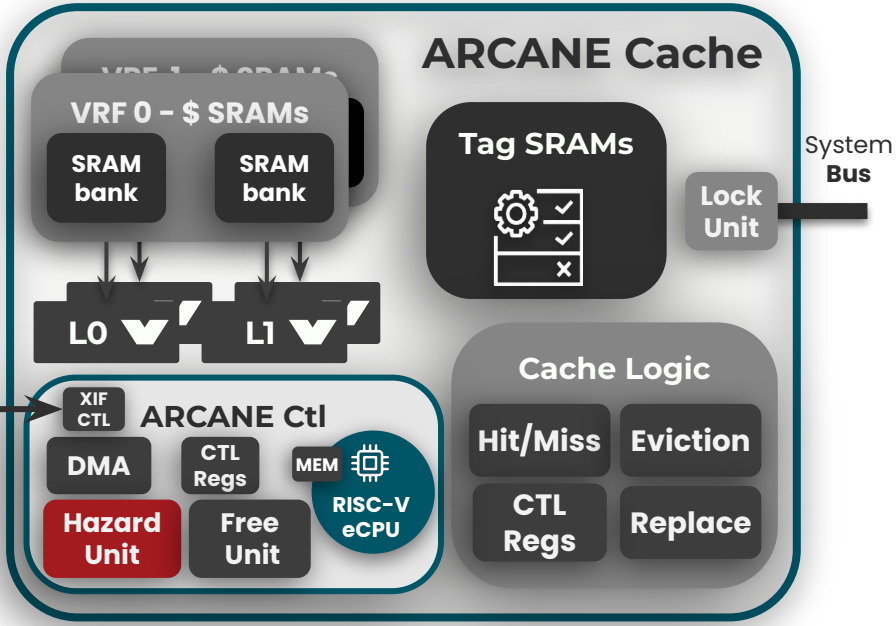
No explicit synchronization

Hazard Unit

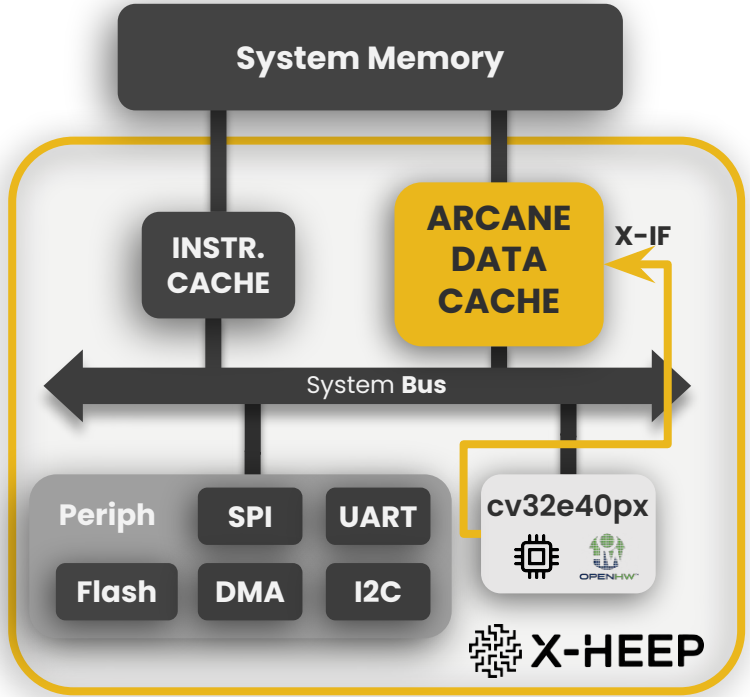
Tracks registered tensors (`xmr`)

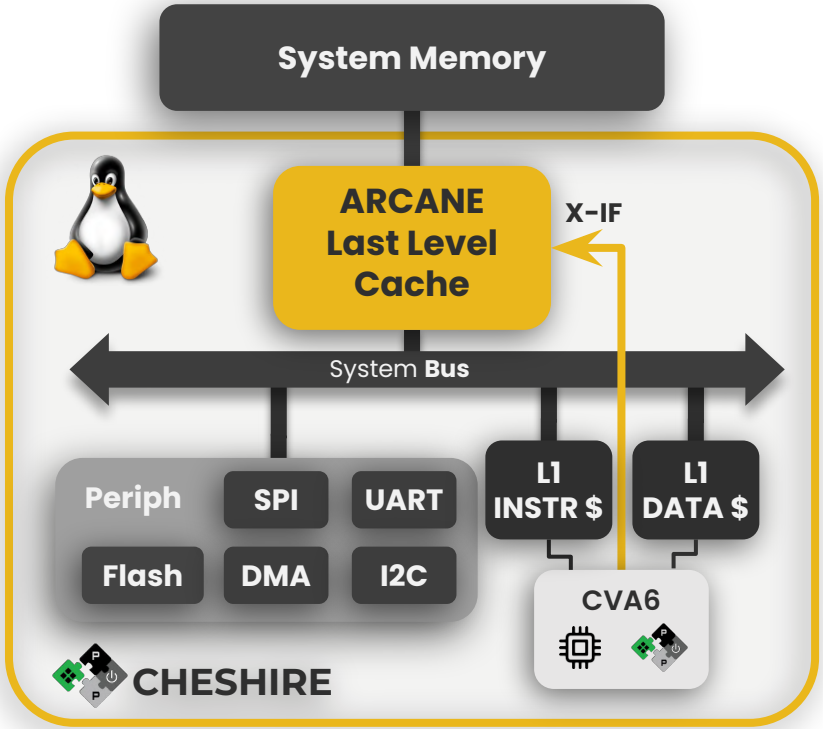
Selectively blocks critical mem operations until data dependency is solved





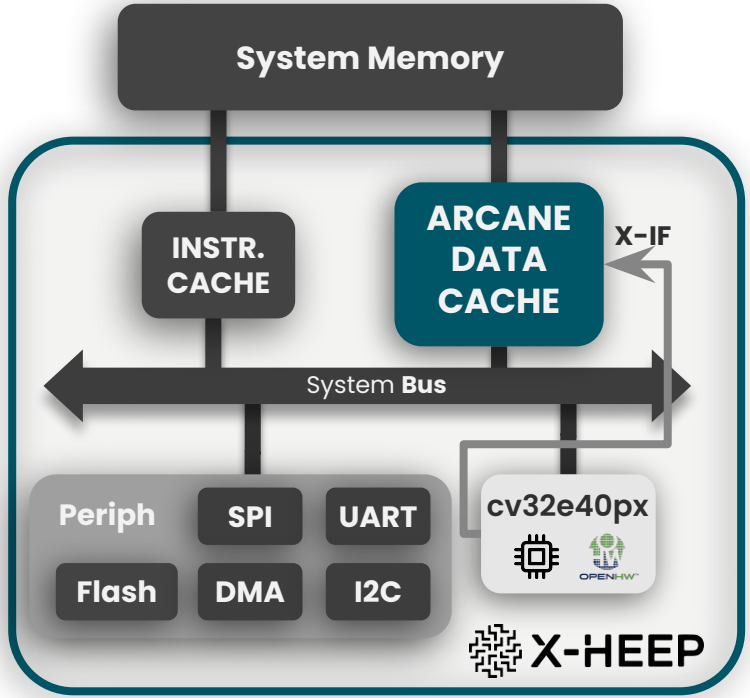
System Integration

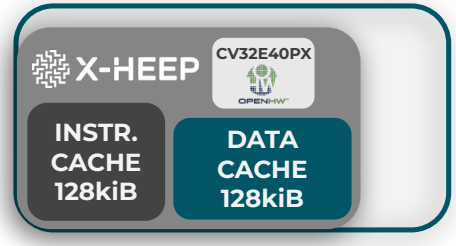


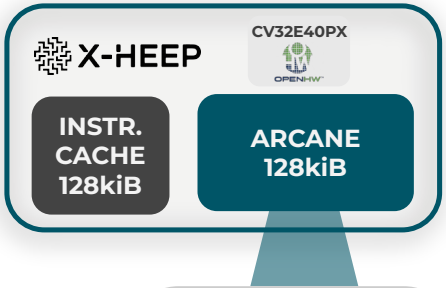


Ottaviano *et al.*, "Cheshire: A Lightweight, Linux-Capable RISC-V Host Platform for Domain-Specific Accelerator Plug-In"
Machetti *et al.*, "X-HEEP: An Open-Source, Configurable and Extendible RISC-V Platform for TinyAI Applications"

Synthesis Results





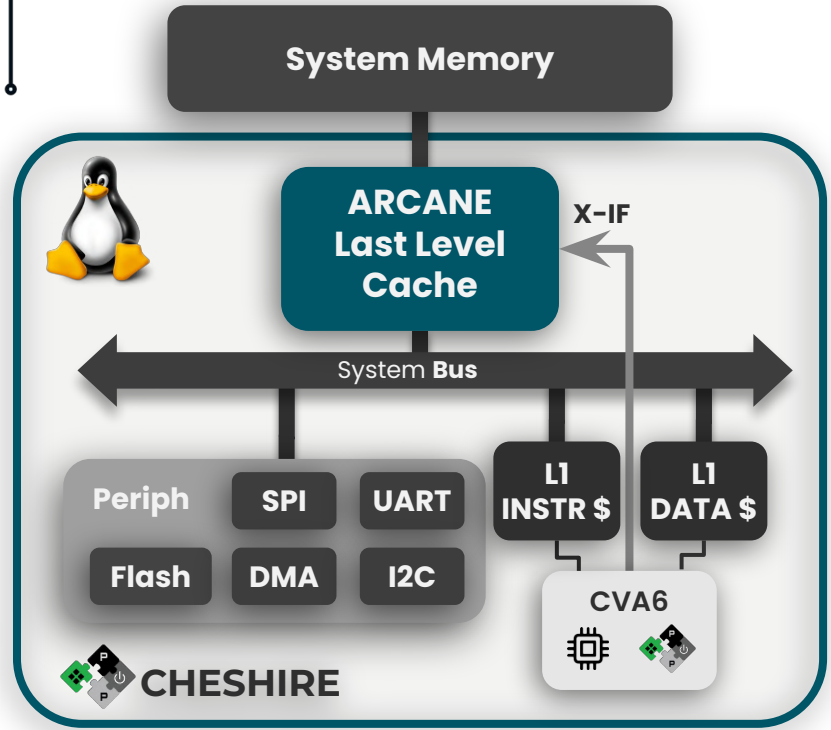


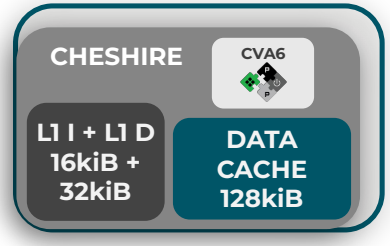
1.4x X-HEEP area @65nm

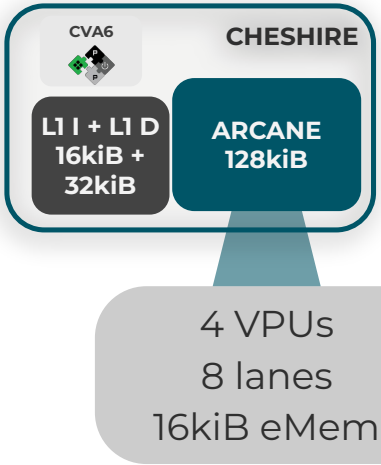
4 VPUs
8 lanes
16kiB eMem

6x speed up on CV32E40P

Implementation Results



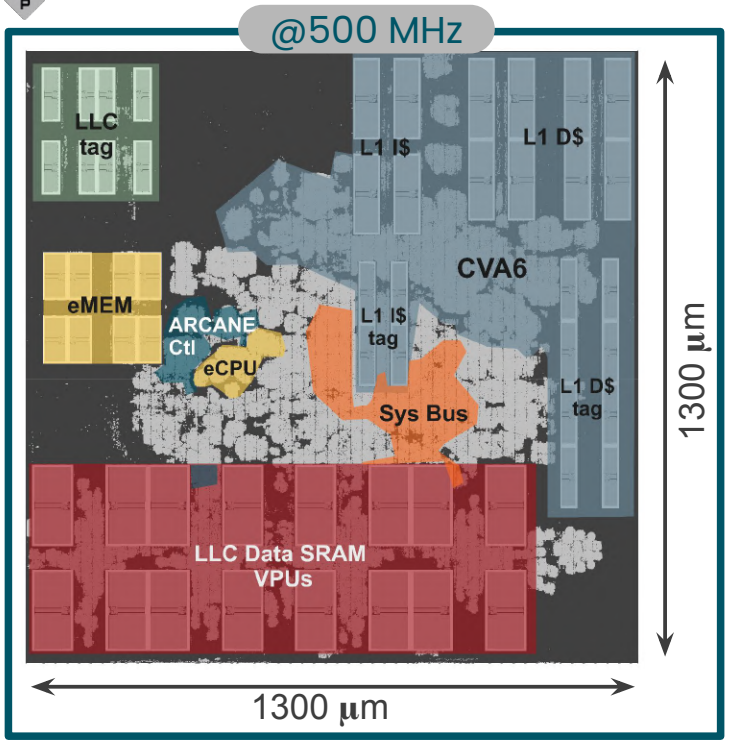


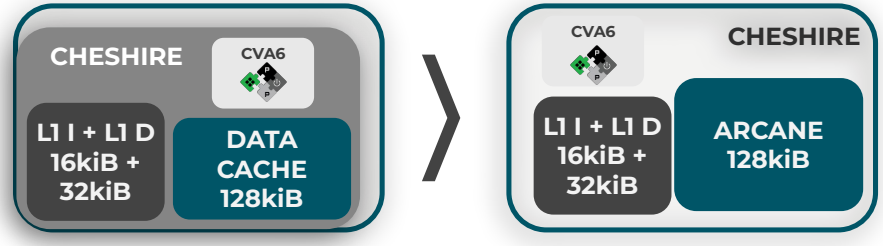


1.2x Cheshire area
@22nm

Implementation Results

CHESHIRE





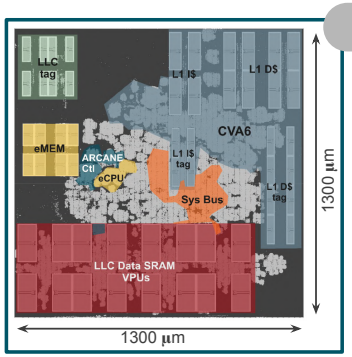
1.4x LLC area

46% due to the VPU logic

36% due to eCPU and eMEM,
18% to Cache Ctl

Benchmarking

CHESHIRE



@500 MHz

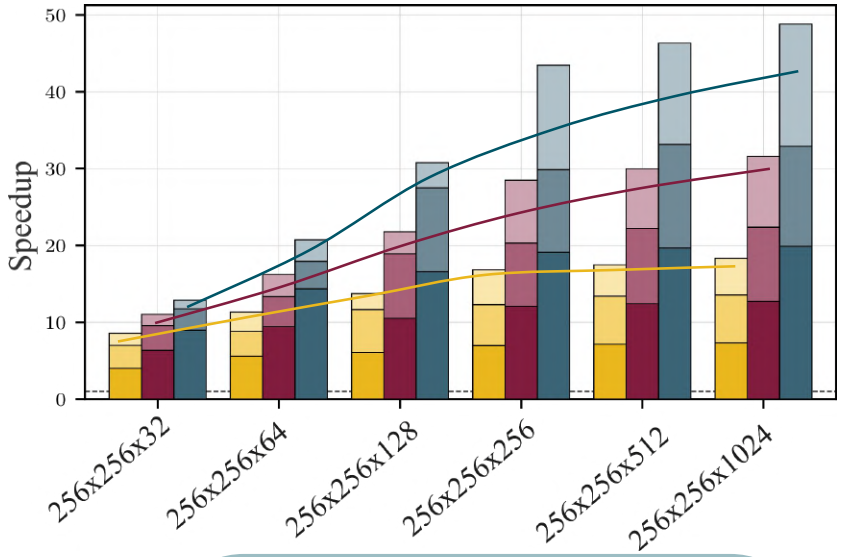
int8 int16 int32

ARCANE, 2 lanes

ARCANE, 4 lanes

ARCANE, 8 lanes

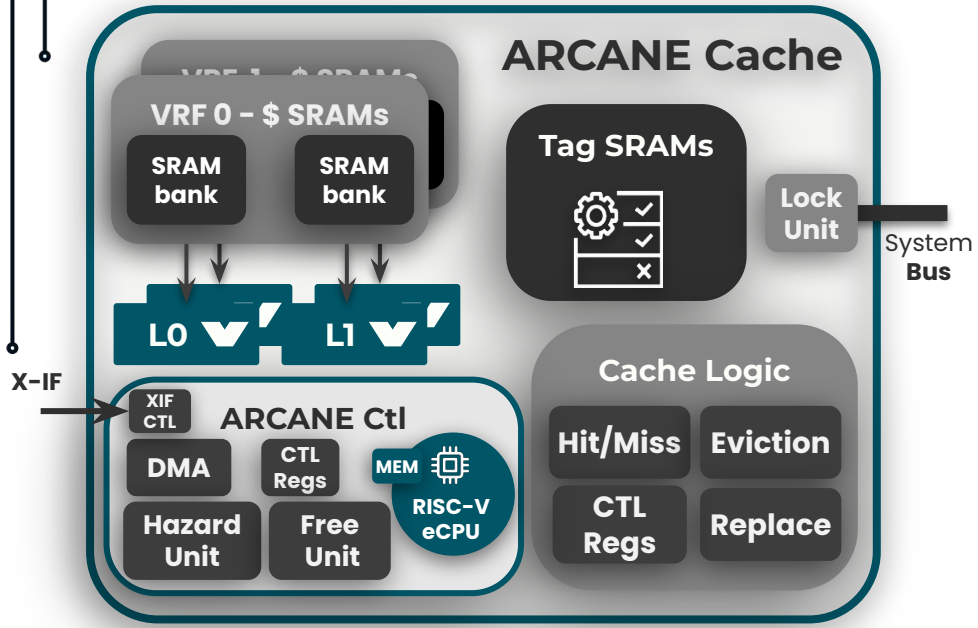
MatMul



10x area efficient
14x energy efficient

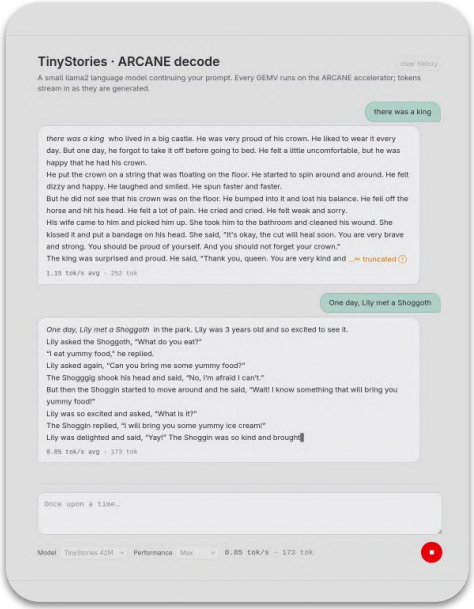
Up to 50x speed up on CVA6

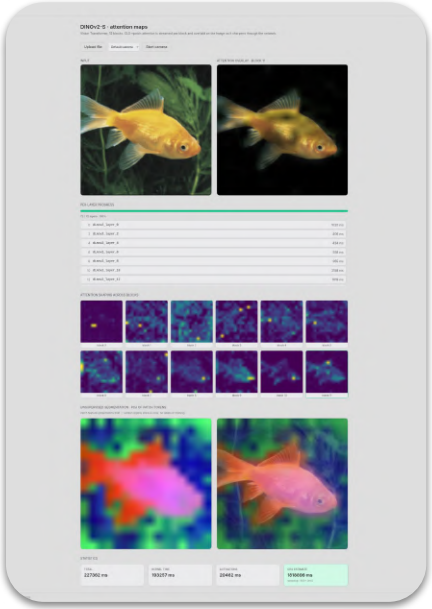
Conclusions



- **Drop-in replacement** of a cache
- **Flexibility** and Programmability
- **Scalable** - from embedded to application-level systems

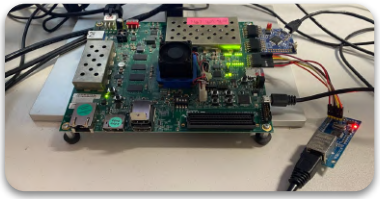
Demo



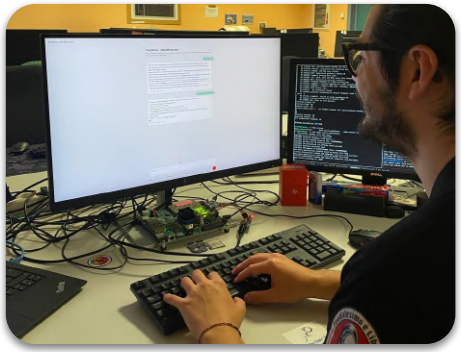


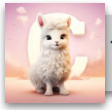
The hardware

Cheshire+ARCANE @ 80 MHz



Check out our DEMOs!





TinyStories on llama2.c

Meta DINOV2 ViT

Thank You

Contacts:
flavia.guella@polito.it
vincenzo.petrolo@polito.it

Politecnico di Torino

EPFL

RISC-V Summit Europe 2026,
Bologna, Italy

RISC-V®