

# SentryCore: A RISC-V Co-Processor System for Safe, Real-Time Control Applications

**Michael Rogenmoser<sup>1</sup>**      [michaero@iis.ee.ethz.ch](mailto:michaero@iis.ee.ethz.ch)

**Alessandro Ottaviano<sup>1</sup>, Thomas Benz<sup>1</sup>, Robert Balas<sup>1</sup>,  
Matteo Perotti<sup>1</sup>, Angelo Garofalo<sup>1,2</sup>, Luca Benini<sup>1,2</sup>**

<sup>1</sup> ETH Zurich, <sup>2</sup> University of Bologna

**PULP Platform**

Open Source Hardware, the way it should be!



@pulp\_platform 

[pulp-platform.org](http://pulp-platform.org) 

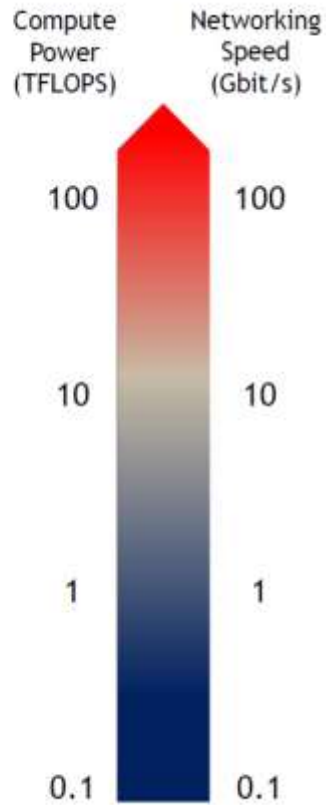
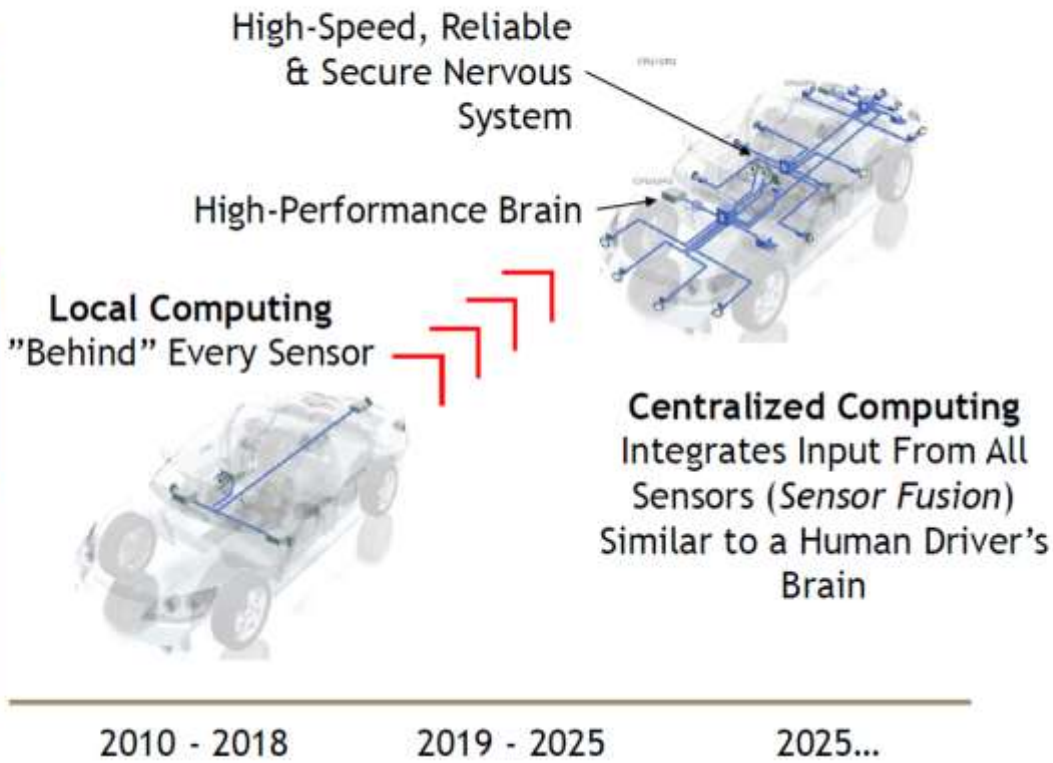
[youtube.com/pulp\\_platform](https://youtube.com/pulp_platform) 





# Advancing Automotive SoCs



[SCR'23]

## Path Towards Full Autonomy



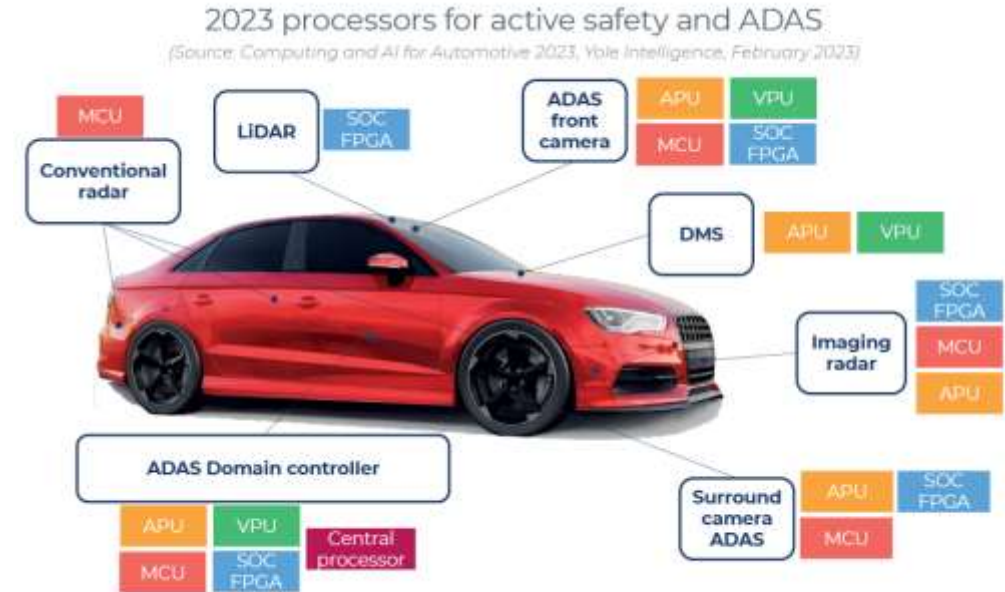
-  **Efficient**
-  **Secure**
-  **Real-time**
-  **Safe**



# Mixed Criticality Systems in automotive systems



- **Applications are becoming more demanding**
  - E.g., Machine Learning for ADAS
- **Federated approach not tenable**
  - Already quite stretched in current systems
  - ECUs, Entertainment systems, ADAS accelerators, ...
- **General trend toward full integration in single centralized SoC**
  - Allows for Domain Control
- **Safety and Real-time not negligible**



Efficient



Real-time



Secure



Safe



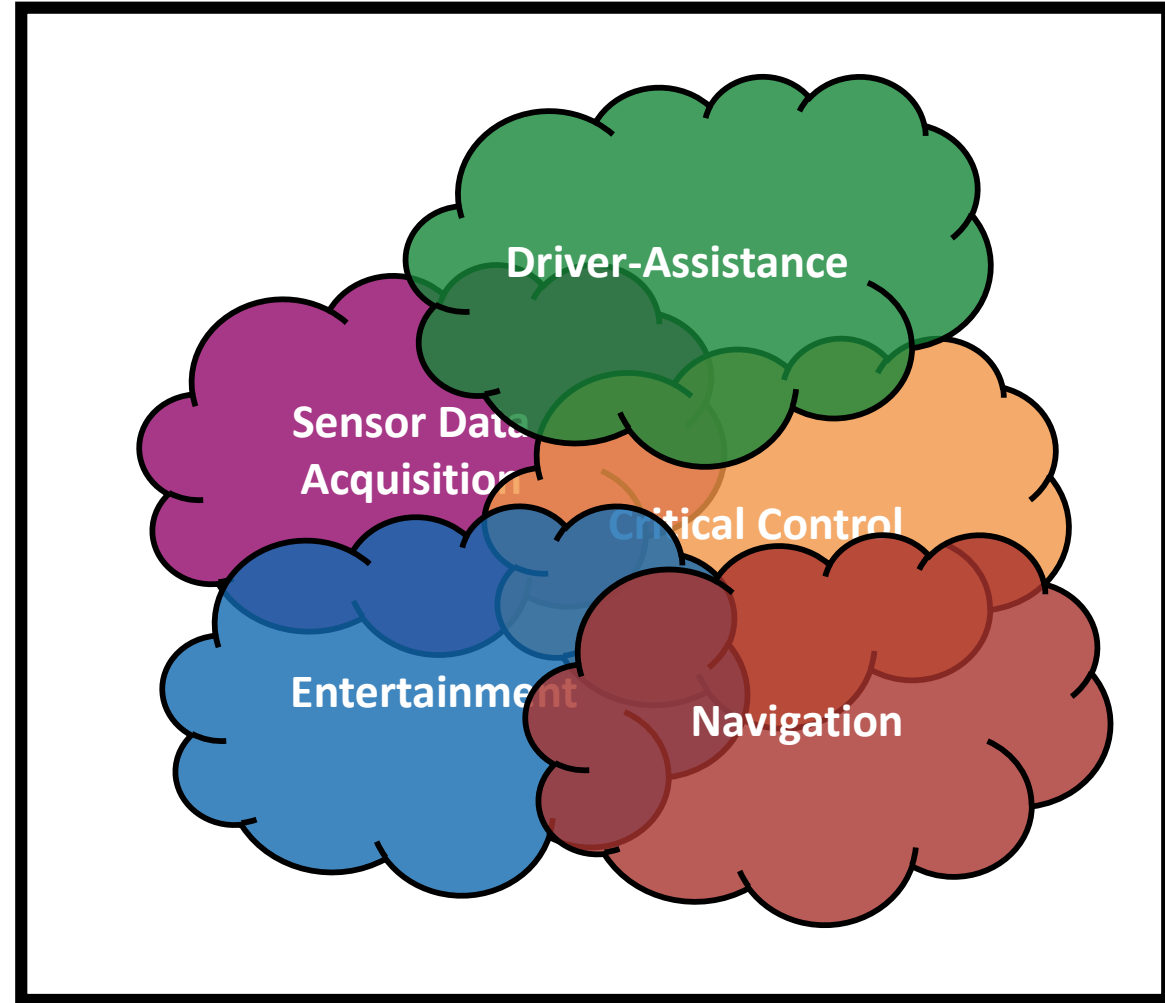
# Mixed-Criticality Systems



**Real-time**

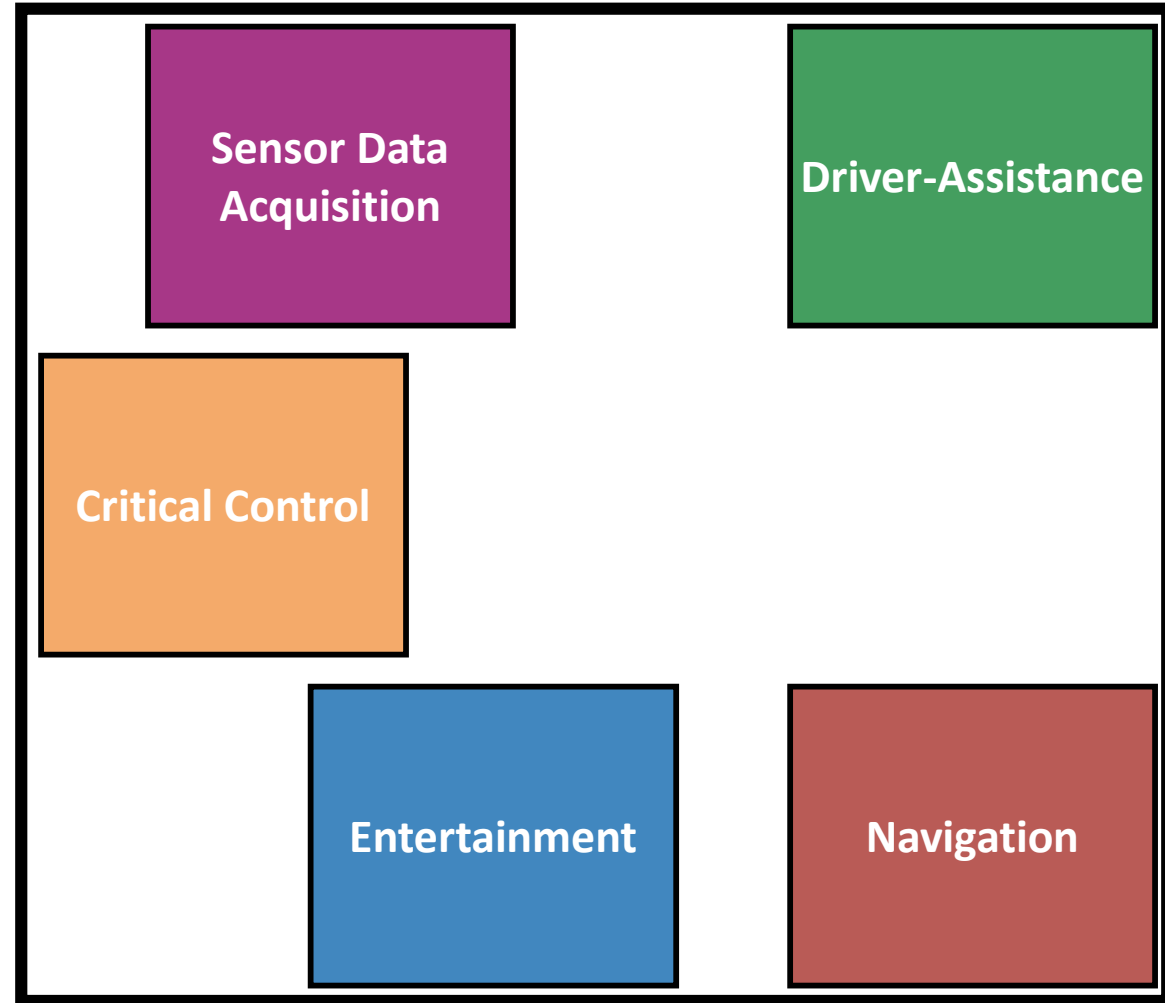


**Safe**



# Mixed-Criticality Systems

- **Isolation for no interference**
  - Isolate non-critical workloads
  - Isolate general-purpose OS
  - Ensure interference freedom for safety- and timing-critical tasks
- **Physically separate control processor**
  - Common solution for automotive SoCs



# Critical Control Requirements



Critical Control

- **Physical Task Isolation**

- Interference-freedom

- **Tight execution bounds**

- Low instruction fetch delay
- Low memory access delays
- Low interrupt delay
- Low context switching overheads



**Real-time**



**Safe**

- **Reliable Execution**

- Correct execution in the presence of faults



# Introducing SentryCore

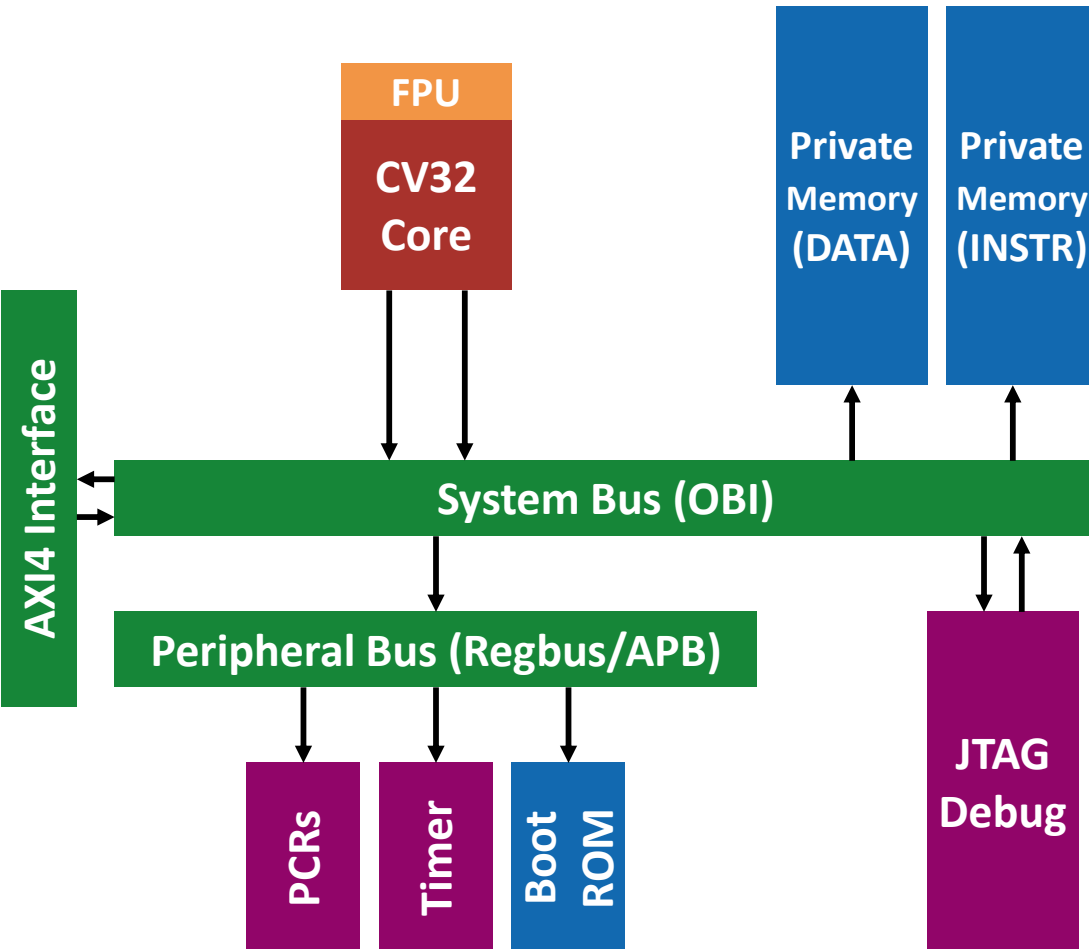


A standalone IP for reliable real-time compute

- **Physically separated mega-IP for self-contained processing**
- **Local instruction & data memory, CLIC support, and real-time extensions**
- **ECC-protected memory and Lockstep execution**



# Base Co-Processor



- **CV32E40P-based processor core**

- 32-bit RISC-V embedded processor
- RV32IMFC support + extensions

- **Tightly coupled Instruction & Data Memory**

- Avoids interference with other parts of the SoC

- **AXI4 Interconnect ports**

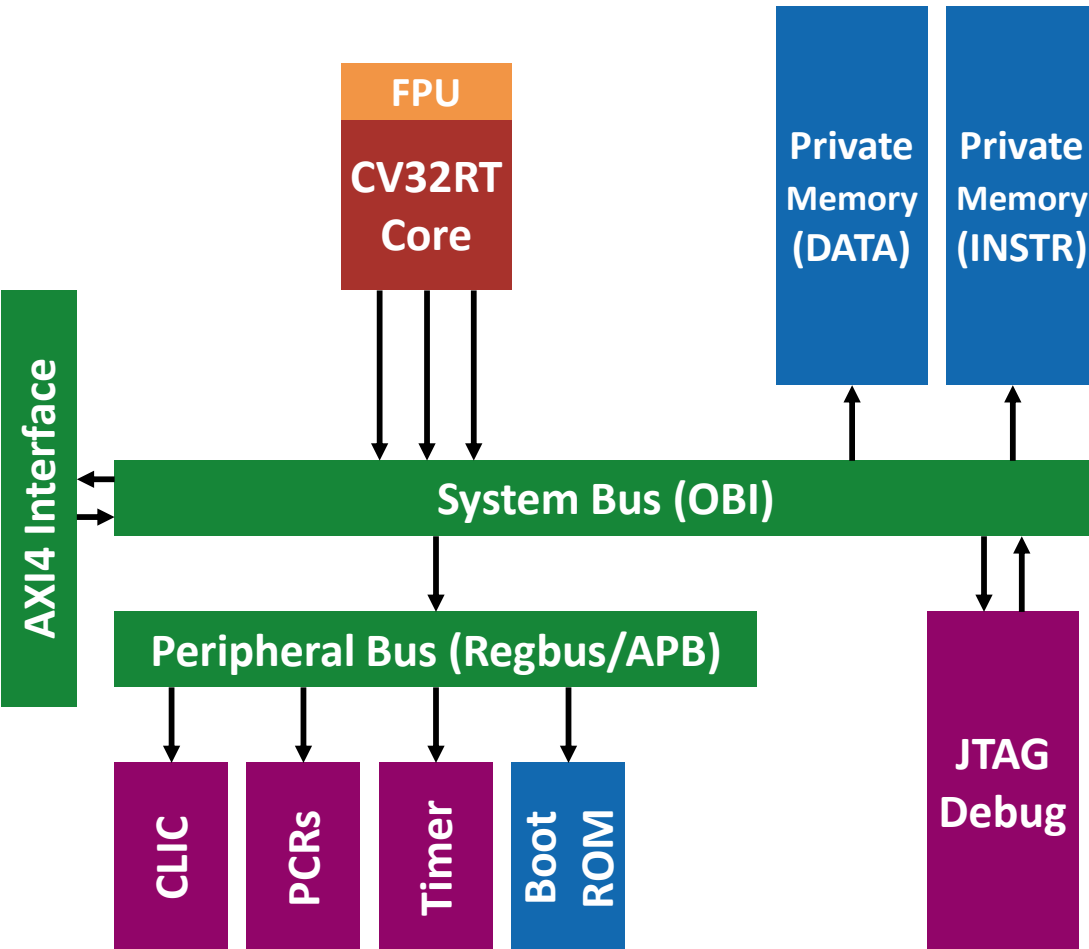
- Allows easy pluggability

- **Support Infrastructure**





# Real-time Capability



- **CV32RT [Balas et. al., 2023]**

- CV32E40P-based real-time core

- **Core-Local Interrupt Controller (CLIC)**

- Allows many fast interrupts
- Vectorized interrupt support
- Efficient configuration and extension

- **fastIRQ extension**

- Reduced interrupt latency in < 6 clock cycles
- Fast context switch in < 110 clock cycles

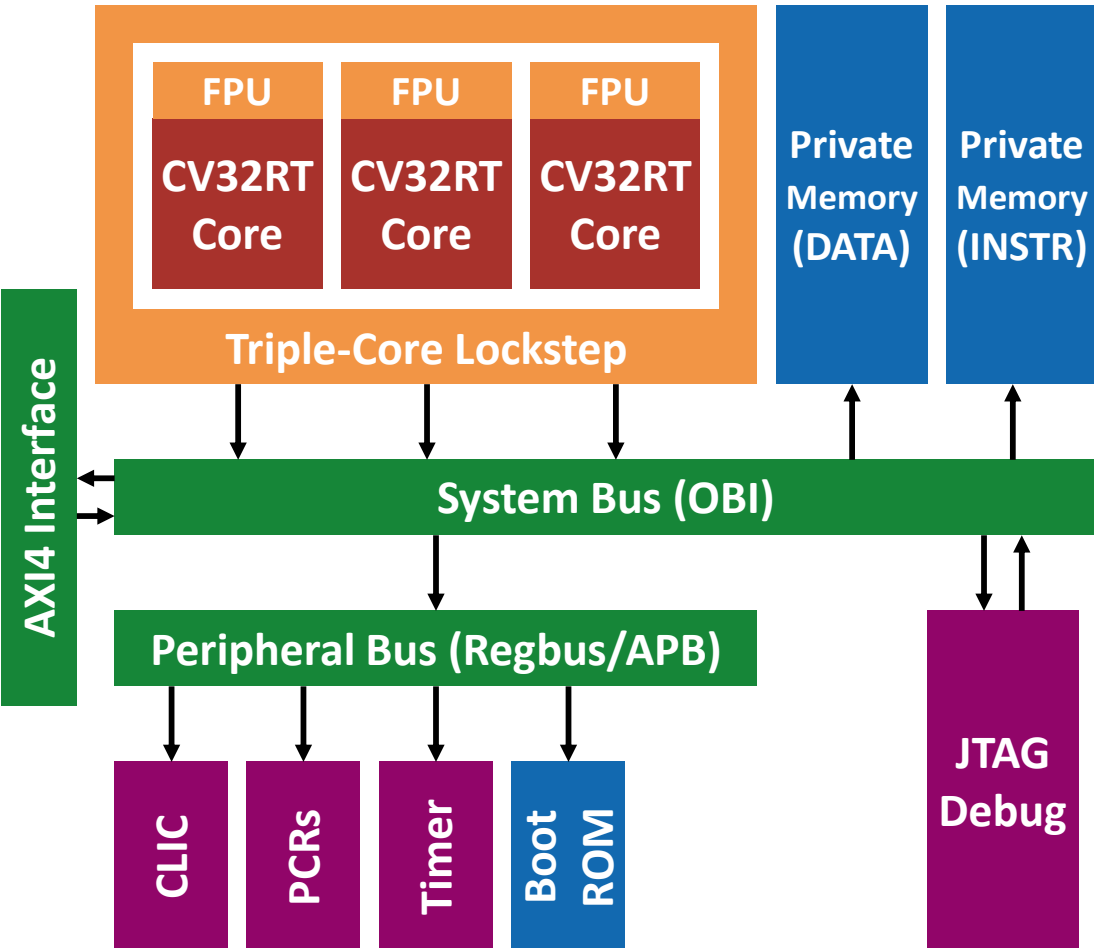


# Built-in architectural Reliability

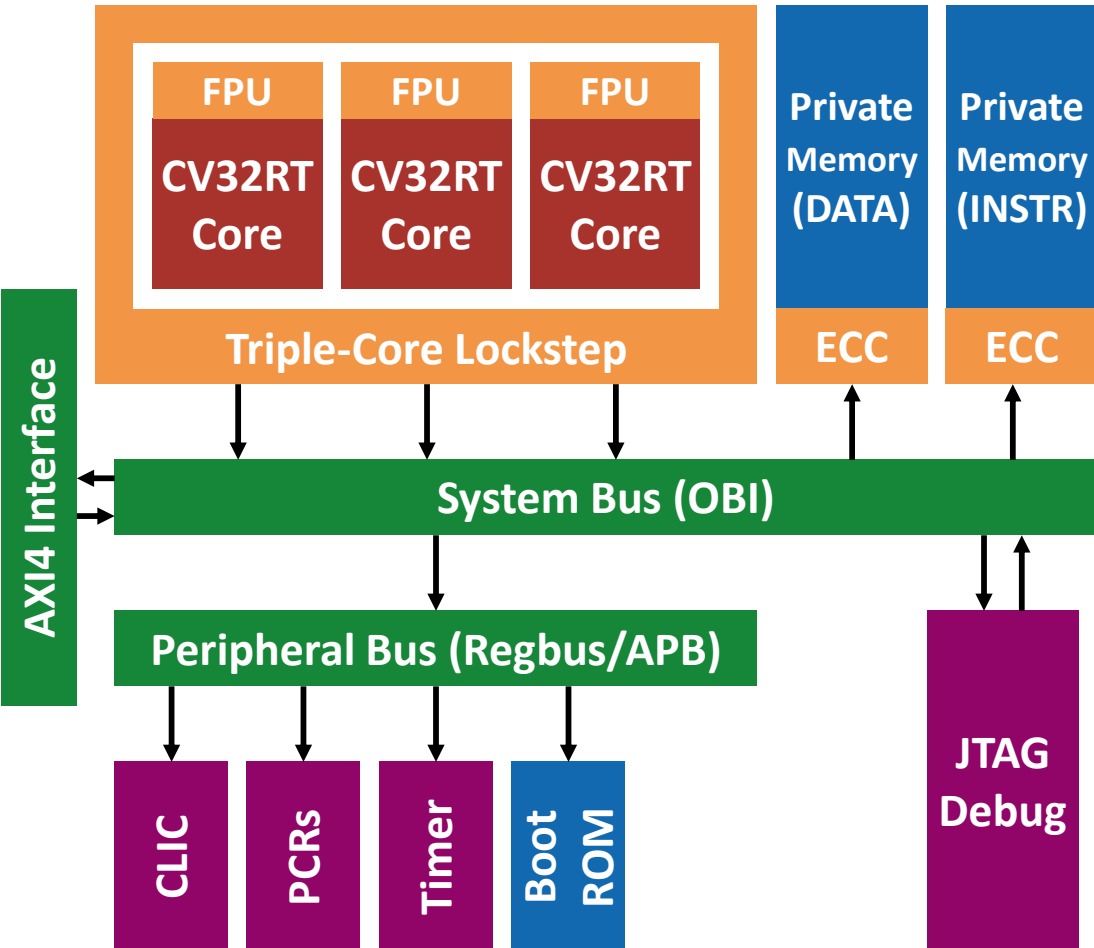


## • Triple-Core Lockstep

- Identical inputs/instruction execution
- Majority Voters on all outputs
- Dedicated re-synchronization to correct errors in < 600 clock cycles



# Built-in architectural Reliability



- **Triple-Core Lockstep**

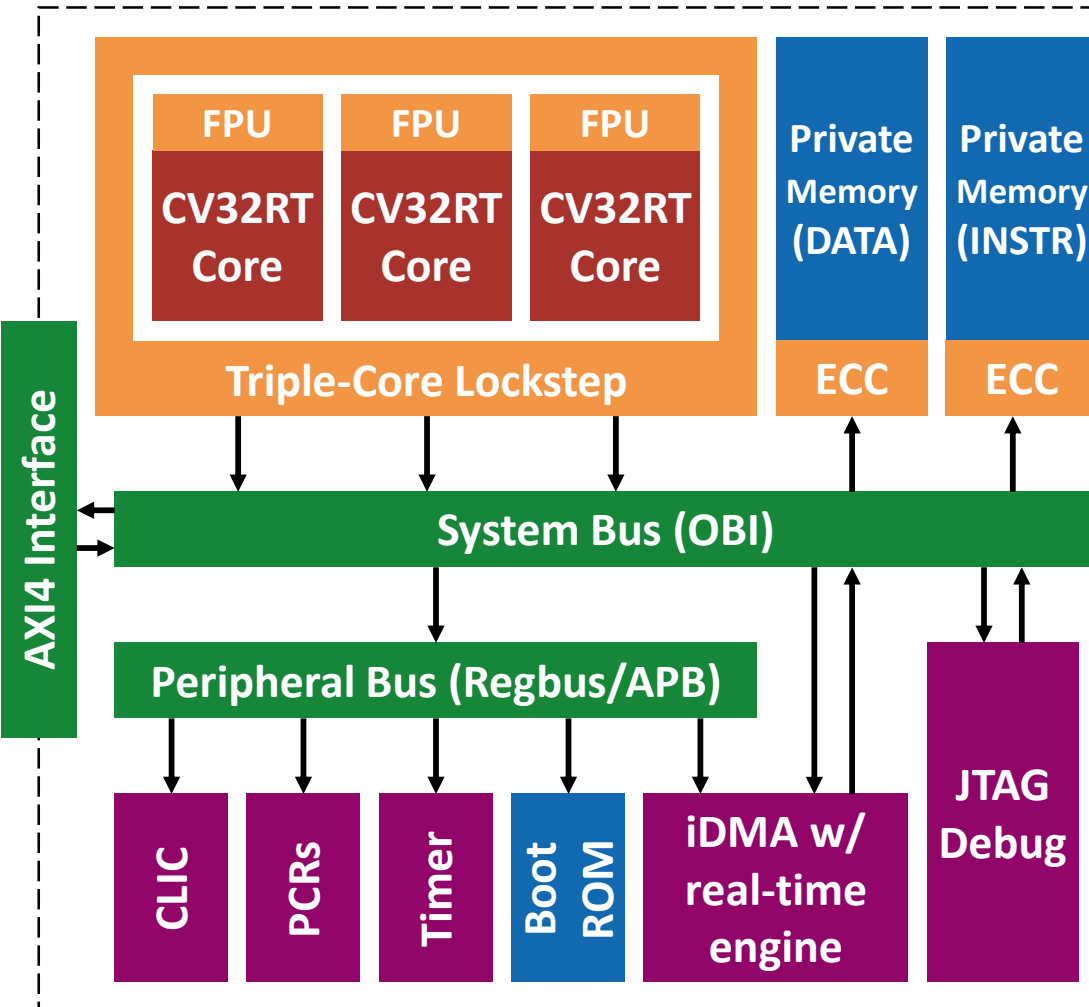
- Identical inputs/instruction execution
- Majority Voters on all outputs
- Dedicated re-synchronization to correct errors in < 600 clock cycles

- **ECC-protected Memory**

- 39-32 Hsiao Code for single error correction, double error detection (SECDED)
- Integrated read-modify-write for sub-word writes
- Scrubber for each bank for continuous correction



# Sensor Data Acquisition



- **Independent DMA unit**

- Transfer of data in and out of the mega-IP
- Connection to the surrounding AXI system

- **Sensor Data Collection**

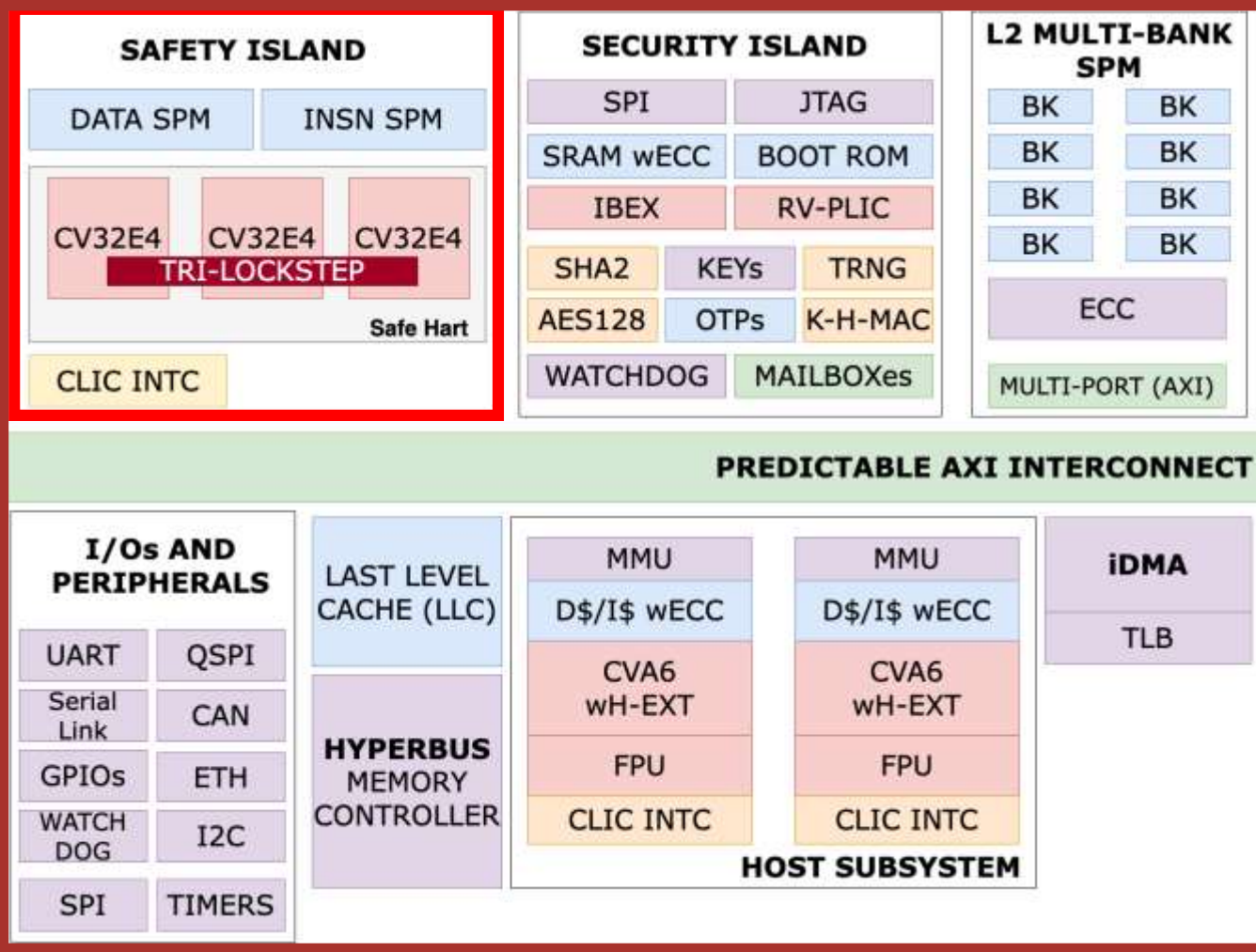
- Independent unit for data gathering
- Periodic trigger for collection



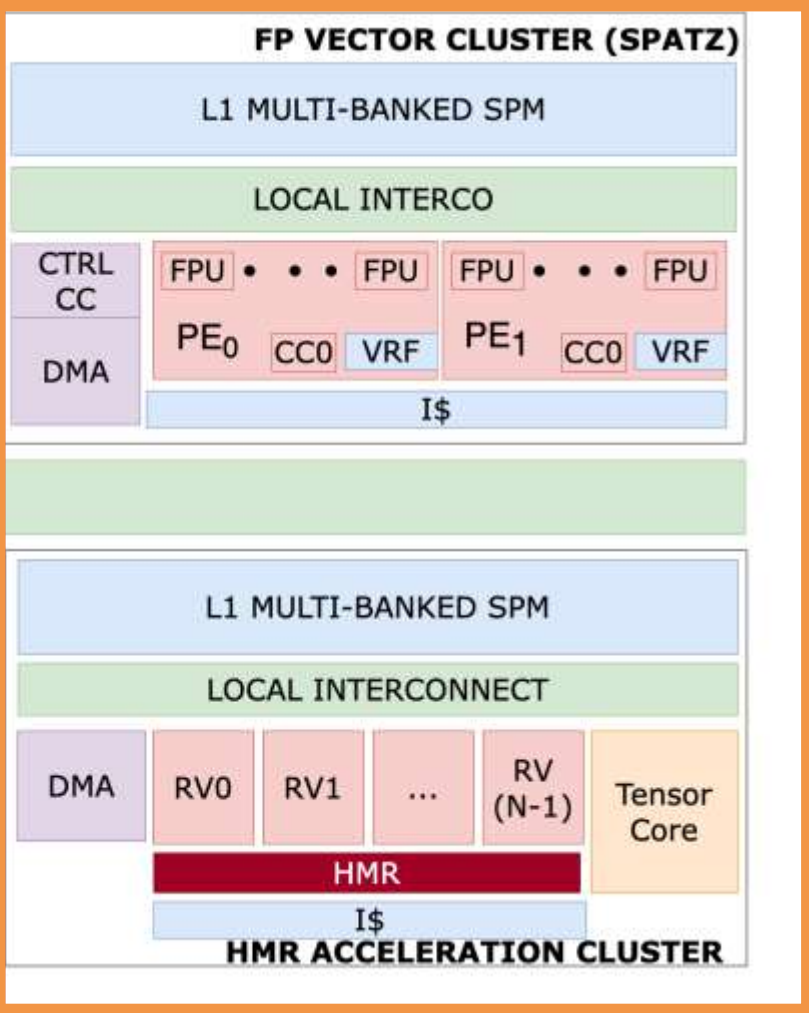
# Integrating into Carfield



## Main Computing and I/O System

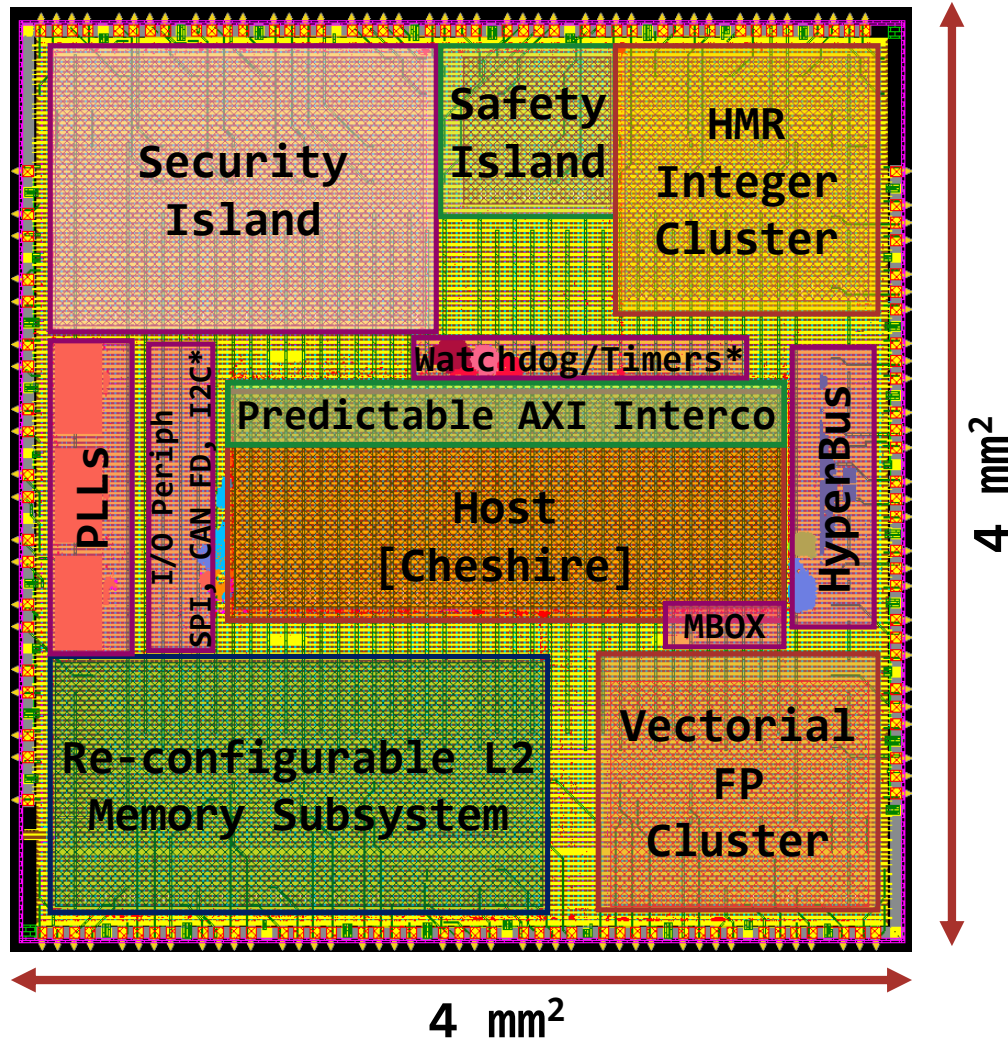


## Accelerators Domain





# Carfield SoC Flooplan – Taped out 11/2023



- **Host [Cheshire]**
  - Dual-Core 64-bit RISC-V processor; **2.45 mm<sup>2</sup>**; 600 MHz;
- **Security Island**
  - Low-power secure monitor; **1.94 mm<sup>2</sup>**; 100 MHz;
- **Safety Island**
  - **0.42 mm<sup>2</sup>**; 500 MHz
- **Re-configurable L2 Memory Subsystem**
  - 1MB; **2.33 mm<sup>2</sup>**; 500 MHz
- **HMR Integer Cluster**
  - **1.17 mm<sup>2</sup>**; 500 MHz;
- **Vectorial FP Cluster**
  - **1.14 mm<sup>2</sup>**; 600 MHz;
- **Hyperbus**
  - 2 PHY, 2 Chips; 200 MHz; Max BW **400 MB/s**

Modules marked with (\*) are not in scale



# Implementation in Intel16

- **SentryCore Configuration**

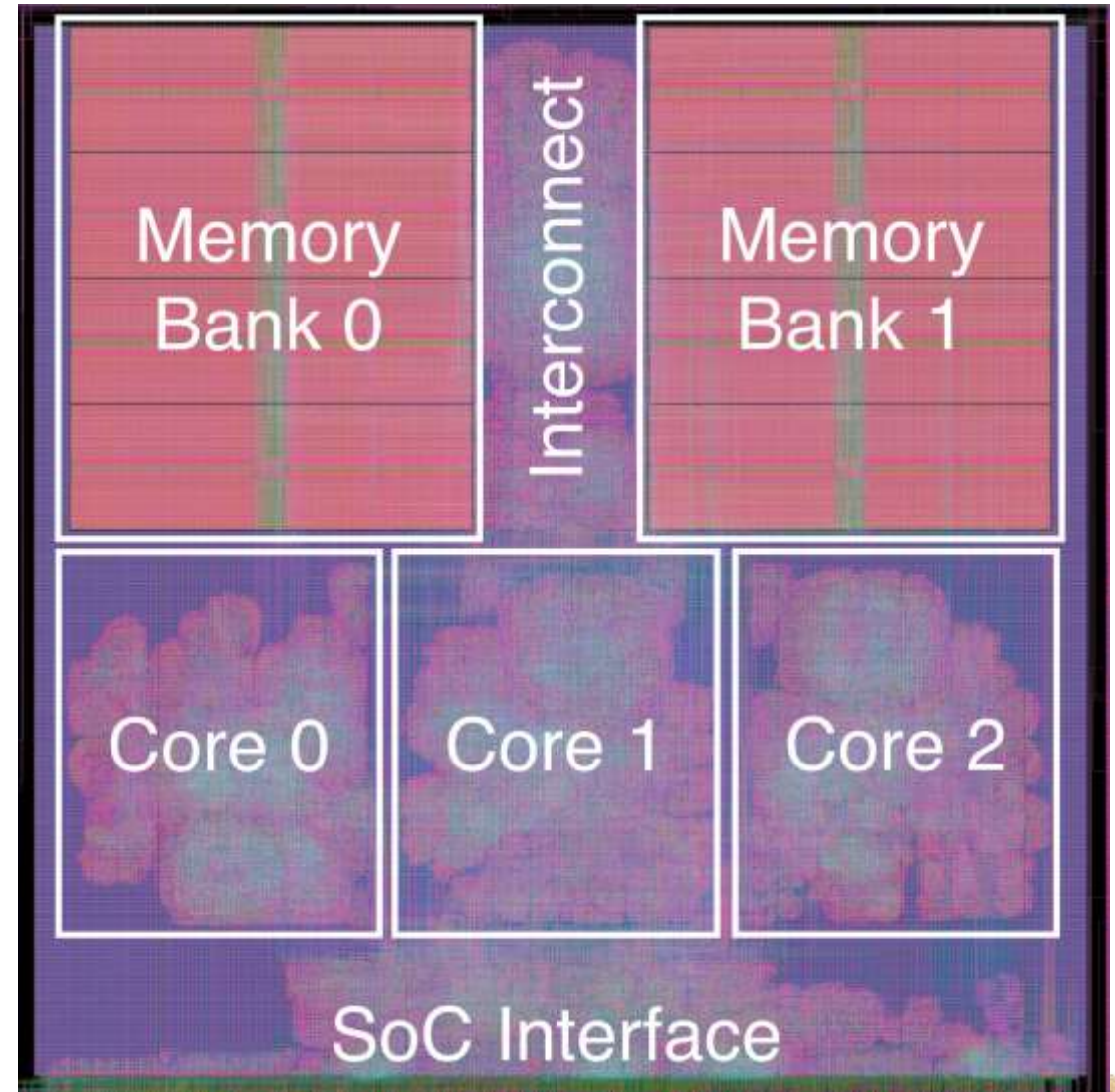
- CV32RT, FPU (32bit), CLIC
- TCLS, ECC Memory
- 64 bit AXI interface, no DMA
- Total 128 KiB ECC-protected Memory

- **Physically separated TCLS Cores**

- 20  $\mu\text{m}$  margins
- Avoids multi-bit error from single particle

- **Implementation Results**

- Clock Frequency: **500 MHz**
- Area: **0.42 mm<sup>2</sup>**
- Power (preliminary): **50-70 mW**



# Software Support



- **Simple bare-metal runtime supported with multiple boot modes**
  - JTAG usable for debuggability
  - Preloaded mode for system-integrated booting
- **Dedicated support for multiple real-time operating systems (OSs)**



**RTIC**  
Real-Time Interrupt-  
driven Concurrency

- **Most popular real-time OS**
- **ASIL-D certified**
- **Designed in Rust**
  - Memory-safe implementation





# Conclusion



- **SentryCore: Safe & real-time 32-bit RISC-V mega-IP**
- **Self-contained IP for physically partitioned mixed-criticality systems**
- **Real-time capable with local memory, CLIC, and custom extensions**
  - Low interrupt (<6 cycles) and context switch (<110 cycles) latencies
  - Dedicated DMA for data collection
- **Reliable execution with Triple-Core Lockstep and ECC-protected Memory**

[github.com/pulp-platform/safety\\_island](https://github.com/pulp-platform/safety_island)



# Future Work



- **Expansion of the SentryCore IP**
  - Options for alternate cores - planned Q3 2024
- **Full SentryCore reliability**
  - Interconnect - planned Q4 2024
  - Control Registers, CLIC - planned Q2 2025
  - DMA, Timers - planned Q2 2025
- **Support from the RISC-V community!**
  - Testing of the SentryCore IP - GitHub issues!
  - Extension of the SentryCore IP - GitHub pull requests!
  - Safety certification

[github.com/pulp-platform/safety\\_island](https://github.com/pulp-platform/safety_island)





**Thank You!**