Reconfigurable Processor-Centric Accelerator for Safety-Critical Applications

CEI UPM
Centro de Electrónica Industrial

Luis Waucquez and Alfonso Rodríguez

luis.waucquez.jimenez@upm.es

RISC-V Summit Europe 2025

In the context of safety-critical systems in harsh environments, this work proposes a flexible RISC-V-based accelerator platform supporting Single, DCLS, TCLS, and staggered DCLS modes. A lightweight fault injection campaign demonstrates its fault tolerance and adaptability to diverse application requirements.

Accelerator Island Features

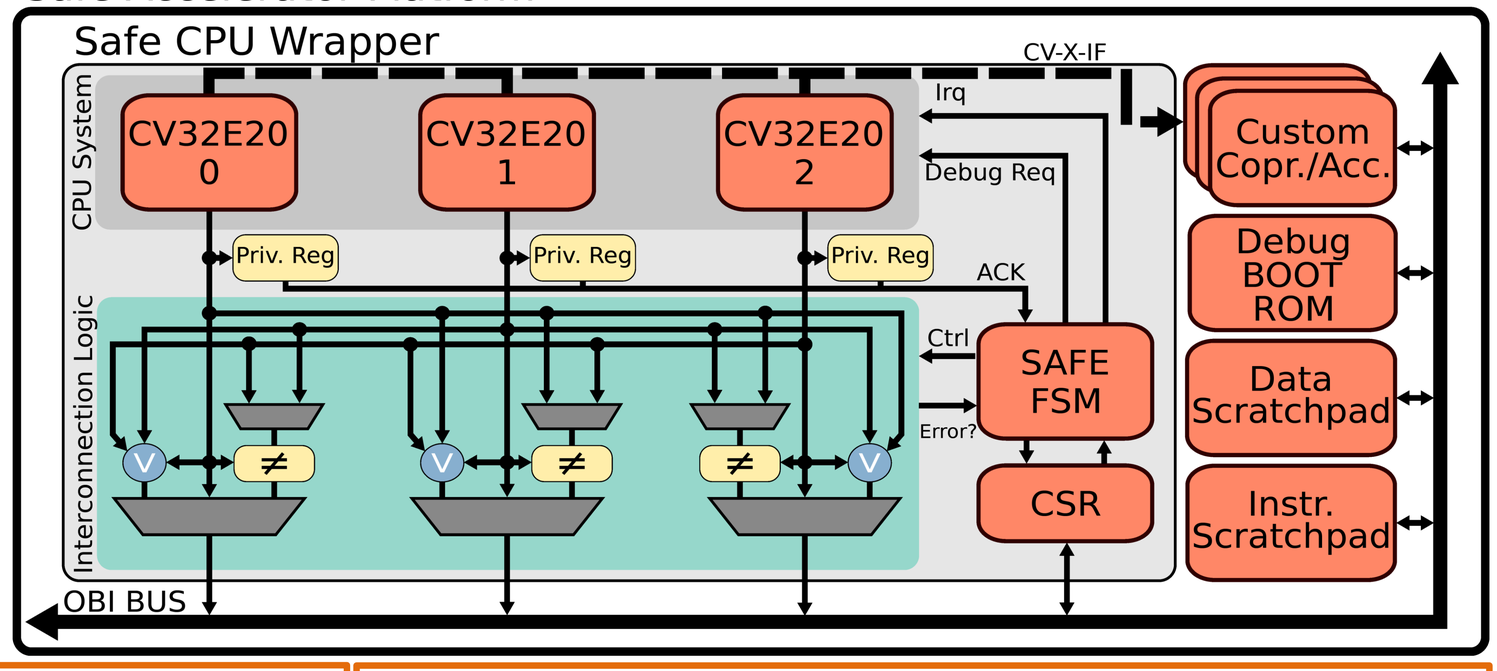
- **CPU System**: Multiple design options between **CV32E20**, **CV32E40P** and **CV32E40PX**.
- Separate instruction and data **scratchpad memories**.
- **System bus**: Based on the **Open Bus Interface** (OBI).
- Compatible with the **Core-V eXtension interface** (CV-X-IF) or **MMIO accelerators**.
- Fully **Open-Source** platform based on IPs collection from **X-HEEP**, **OPEN HW** and **PULP-PLATFORM** ecosystems.

RISC-V PULP OPENHW GROUP PROVEN PROCESSOR IP X-HEEP

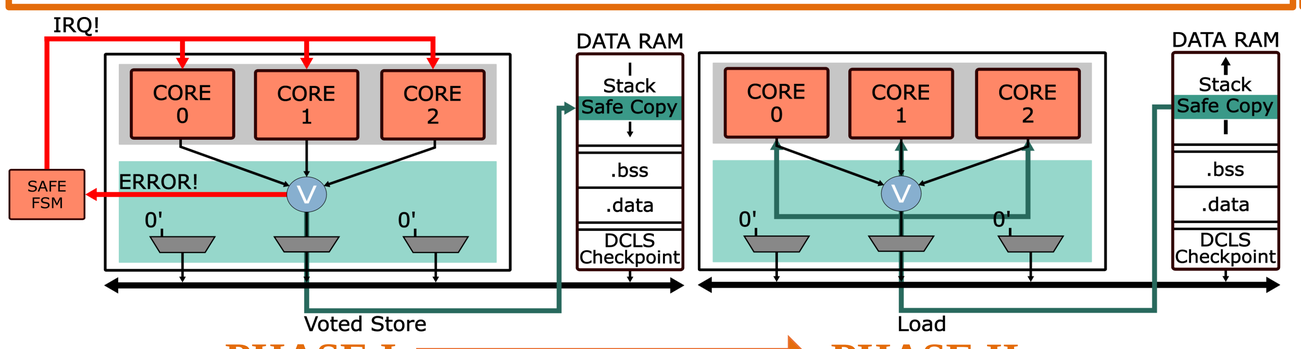
OPERATIONAL MODES

- **Single, TCLS, DCLS & DCLS with staggering**.
- **SW-Based** recovery implementation.

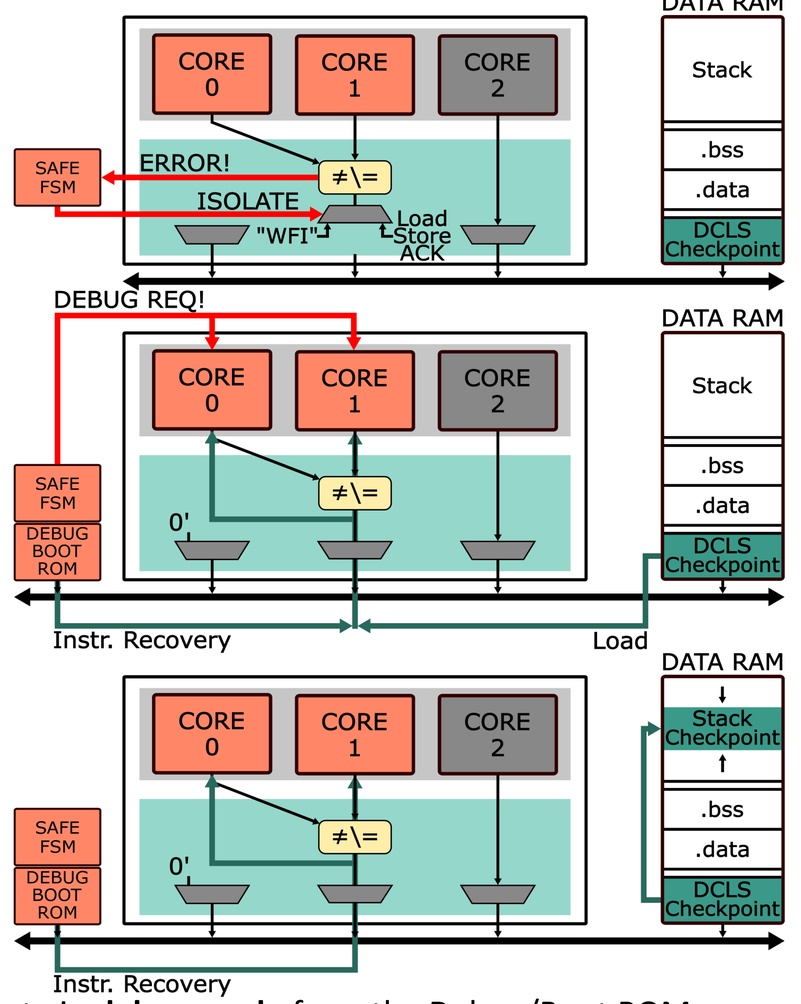
Safe Accelerator Platform



SW Recovery TCLS



SW Recovery DCLS



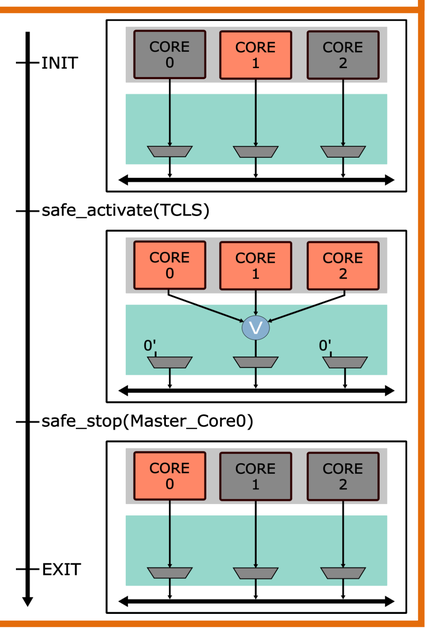
PHASE I → PHASE II

- **SW Recovery ISR** : Core context store (voted) & load operation.
- **Core Context**: Registers File & Control Status Registers.
- A single error can be masked to prevent the recovery operation until another error.
- **No need to restore stack context** in TCLS configuration.

- **SW Recovery** : Execute in **debug mode** from the Debug/Boot ROM.
- **Core Context**: Registers File & Control Status Registers from checkpoint.
- **Stack Context**: Restore from the checkpoint.
- **WFI injection** and **Load/Store ACK** emulation for **cores disconnection**.
- Execution interrupted when error is detected, recovery procedure is executed.

Safety SW Template & Flow Execution

```c
#include "Safety.h"

int main(){

   /* Non-critical tasks here... */

   //Enter Safe mode (TCLS,DCLS,LOCKSTEP)
   safe_activate(TCLS);

   /* Critical tasks here... */

   //Exit Safe mode (MASTER_CORE0/1/2)
   safe_stop(MASTER_CORE0);

   /* Non-critical tasks here... */

   return 0;
}
```



Experimental Results

Modes	Entry [CYCLES]	Exit [CYCLES]	Recovery [CYCLES]	Store-Context [CYCLES]	Overhead Base	Overhead Bit-flip	Overhead CMF
SINGLE	-	-	-	-	0%	-	-
TCLS	301	54	180	-	0%	0.75%	-
DCLS	280	54	125	118	0%	3.78%	-
$DCLS_{n=1}$	291	72	179	170	11.69%	16.23%	16.23%
$DCLS_{n=2}$	303	85	234	223	23.37%	29.45%	29.45%

- Latencies observed in the different sections of the safety mechanism.
- Overhead estimation based on AES algorithm execution during simulation.
- Single Bit-Flip simulation & Common Mode Failures for Lockstep modes.

System	Modules	Resources LuTs	Resources FFs
SoC & Accelerator Baseline		35898	32760
	Accelerator Baseline	11049	5868
	Triple Core Unit	10218	5748
	CV32E20 Core	3421	1916
SoC & Safe Accelerator		38068	33182
	Safe Accelerator	13129	6291
	Safe CPU Wrapper	12347	6158
Overhead		6.04%	1.28%

- FPGA resource utilization.
- System integration with X-HEEP platform (SoC).

This work has been supported by the Secretaría de Estado de Telecomunicaciones e Infraestructuras Digitales, Ministerio para la Transformación Digital y de la Función Pública, funded by NextGenerationEU, in the context of the Programme for the creation of Chip University-Business Chairs, Project TSI 069100-2023-0016, Chip Chair UPM-INDRA (CAUPIME)

Financiado por la Unión Europea NextGenerationEU · Gobierno de España · Ministerio para la Transformación Digital y de la Función Pública · Plan de Recuperación, Transformación y Resiliencia · España digital 2026 · PERTE Chip · Politécnica · Universidad Politécnica de Madrid