HARV-SoC - A Fault-tolerant RISC-V Microcontroller for Dependable Applications

Université de Montpellier

Douglas A. Santos, Wesley Grignani, Carolina Imianosky, and Luigi Dilillo

3rd TWIN-RELECT Training School on
Fault-Tolerant AI Systems

14/01/2026

What is RISC-V

RISC-V International is a global nonprofit association based in Switzerland. Founded in **2015** as the RISC-V Foundation with 29 members, RISC-V is now a truly global organization with 2k+ members in more than **70 countries**.

- supports free and open RISC instruction set architecture and extensions
- new level of free, extensible software and hardware freedom on architecture

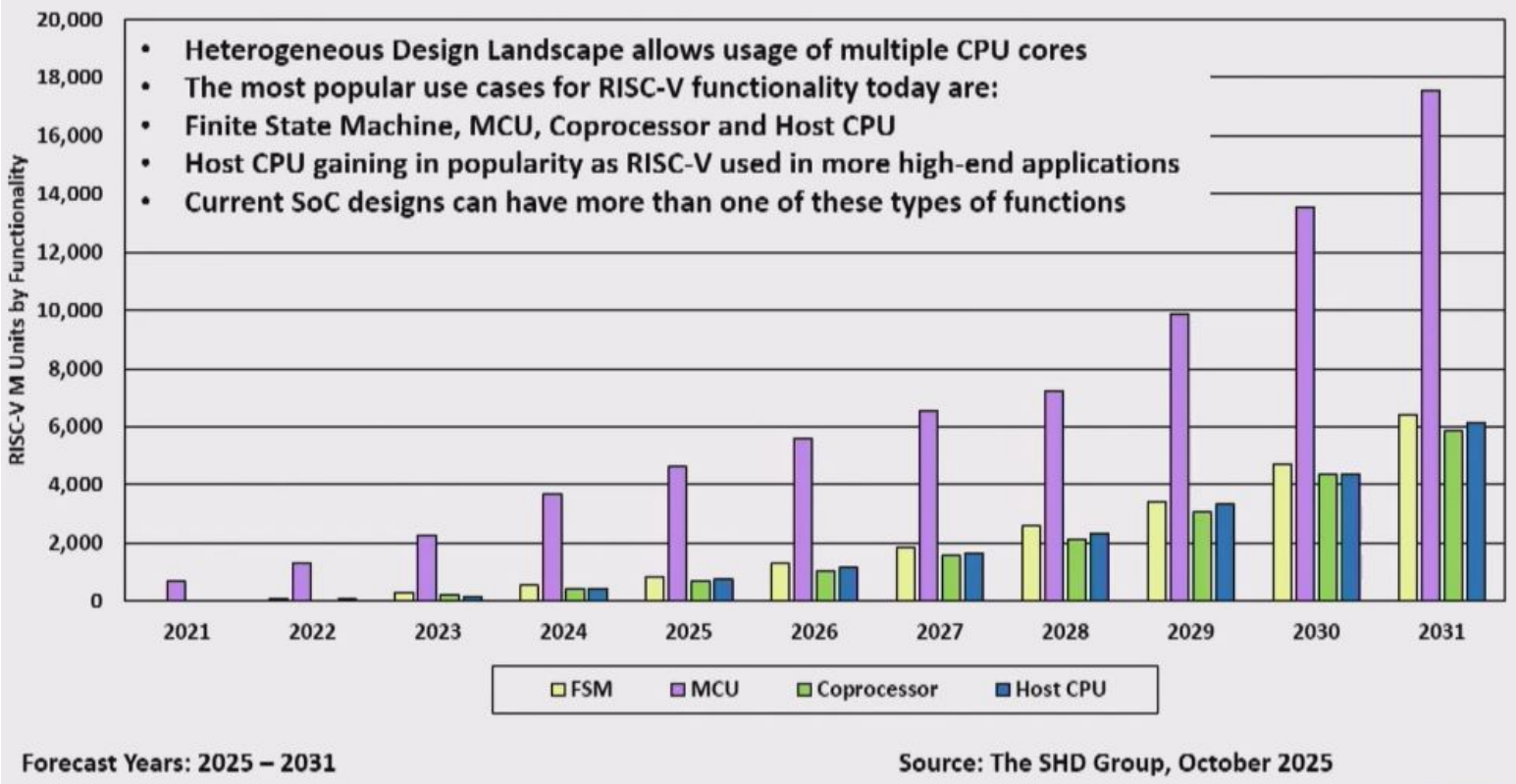
Source: RISC-V ORG

RISC-V: Disruptive Technology

Barriers	Legacy ISA	RISC-V ISA
Complexity	1500+ base instructions Incremental ISA	47 base instructions Modular ISA
Design freedom	$$$ - Limited	Free - Unlimited
License and Royalty fees	$$$	Free
Design ecosystem	Moderate	Growing rapidly. Numerous extensions, open & proprietary cores
Software ecosystem	Extensive	Growing rapidly

Source: RISC-V ORG

Rapid RISC-V Growth



Considered #2 architecture, behind ARM (source: The SHD Group)

RISC-V Adoption

Data Center
Cloud, HPC

Top providers like Amazon and Alibaba are designing their own chips.
New features specific to HPC such as Vector and larger Virtual address space

Telecom & Communications

Rapid evolution in 5G, handsets, and base stations grows with each generation of hardware and increased capability

Automotive

Transforming from autonomous vehicles to infotainment to safety, the whole vehicle relies on innovative electronics.

Consumer and IoT devices

Incredible innovation is driving volume with billions of connected devices in the next 5-10 years.

AI / ML

Artificial intelligence is incorporated across many areas including Industrial IoT, manufacturing, and financial

Edge Computing

A distributed, open architecture decentralizes processing power, reduces latency, and supports IoT performance in low bandwidth environments.

RISC-V adoption spans industries

Source: RISC-V ORG

HARV - Hardened RISC-V

Motivation

Harsh environments impose challenging design decisions

- Temperature variations
- Mechanical stress
- Ionizing radiation

Radiation harsh environments (e.g. avionics, space)

- Single-Event Effect (SEE)
- Total Ionizing Dose (TID)
- Displacement Damage (DD)

Motivation

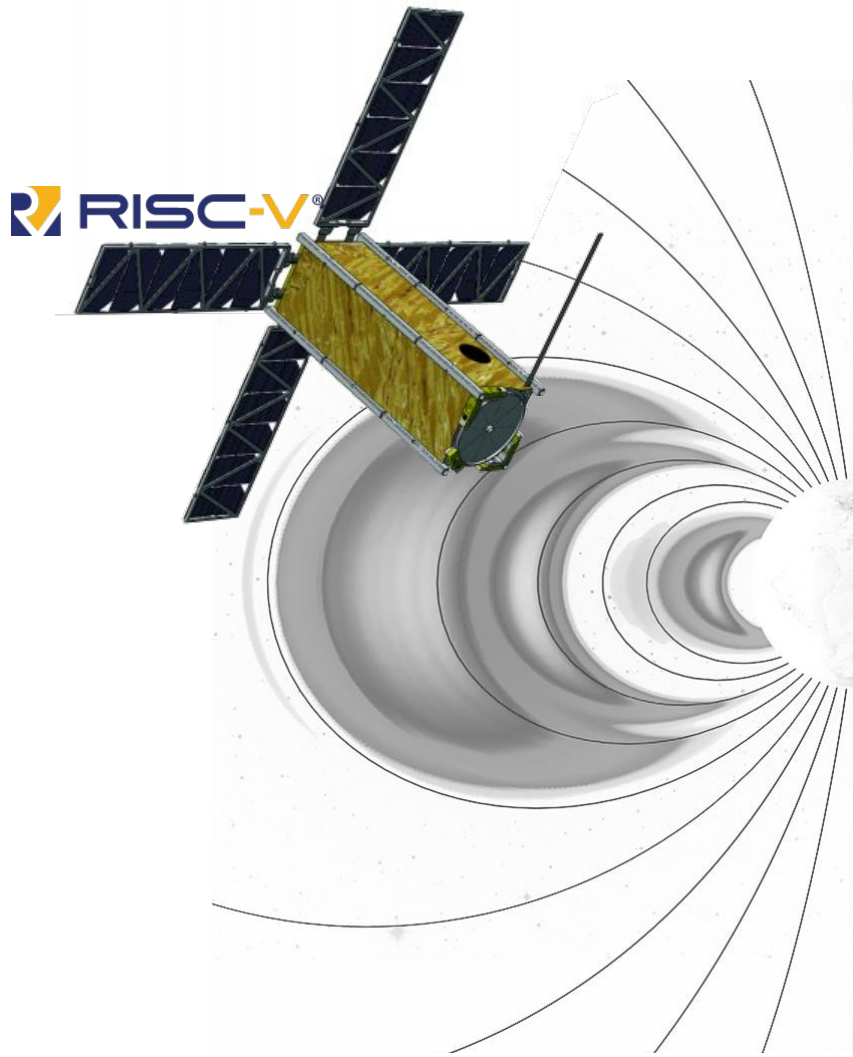
Challenge for electronics

- Failures in the functioning of such devices.

Reliable systems commonly use Systems-on-Chip as main processing core unit

Single-Event Effects (SEEs) in Systems-on-Chip

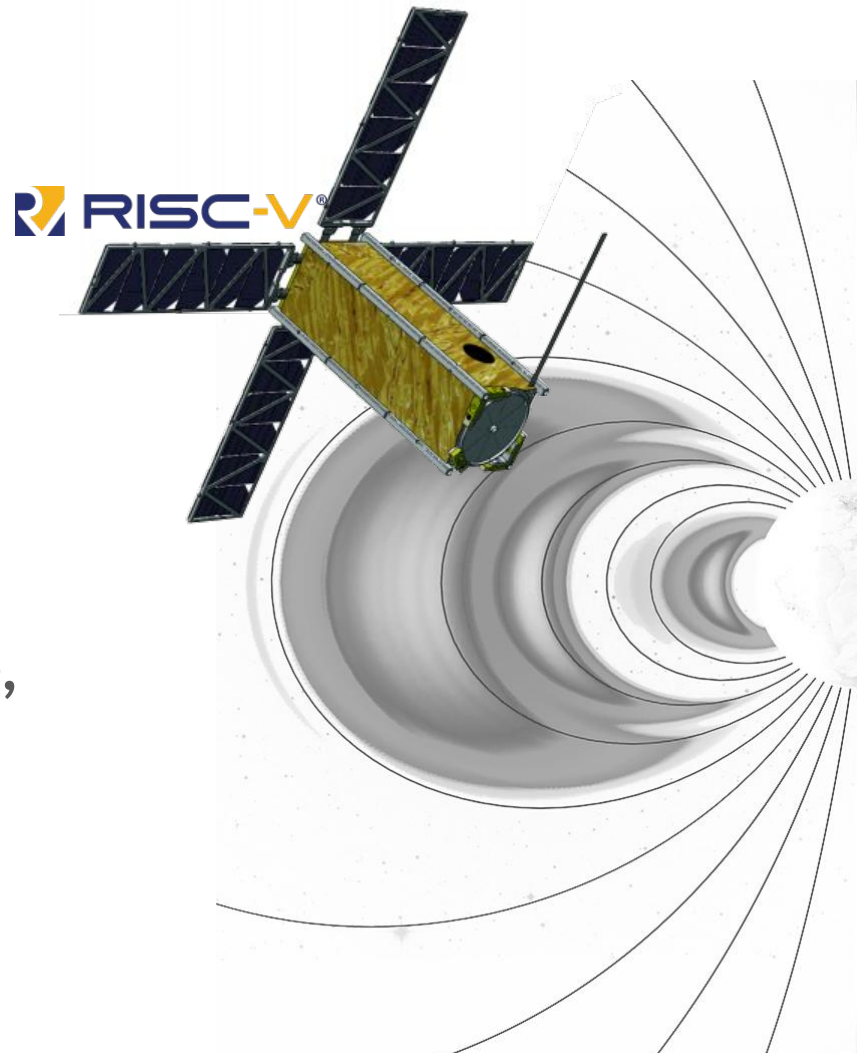
- System and code memories upsets
- System registers upsets
- Communication bus corruption
- Timeouts, loops, hangs



Motivation

Recently, RISC-V is being evaluated for critical applications, notably for harsh environments, such as Avionics and Space

Open-source soft-core processors are readily available, enabling customization

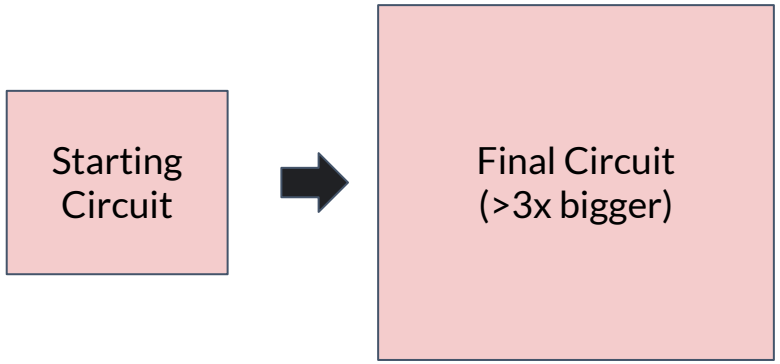


Design Strategy for HARV

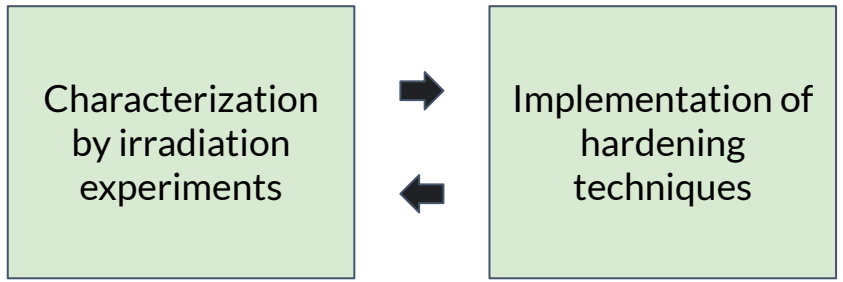
Proposal: Develop a processor with low resource usage that complies with radiation constraints

Result: Agile, Flexible, not resource greedy, proved radiation-hardened

Standard Approach



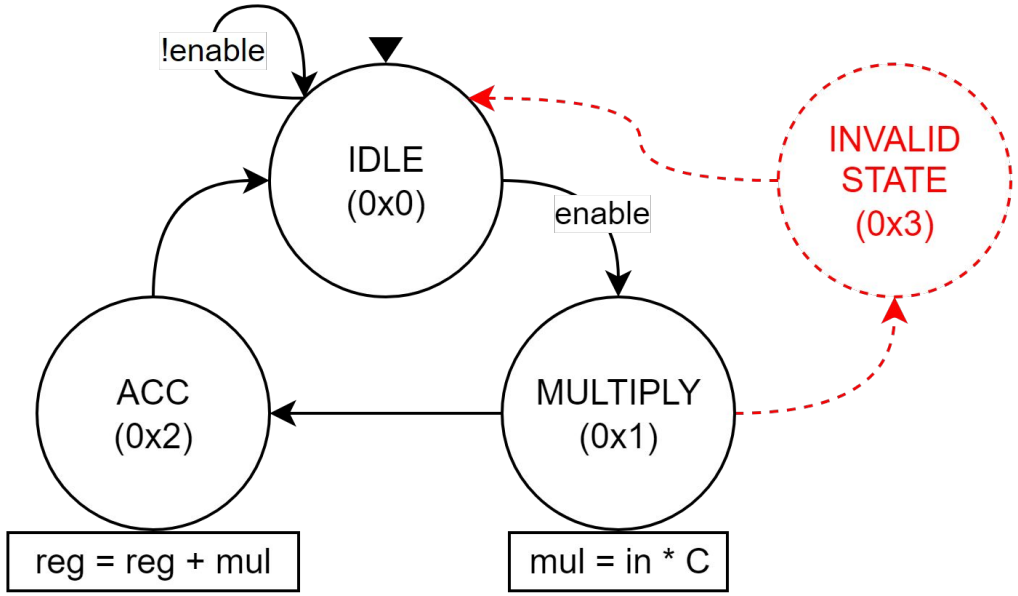
HARV Approach



System Observability

Observability is tooling or a technical solution that allows teams to actively debug their system. Observability is based on **exploring properties and patterns not defined in advance**.*

- ➔ consists of collecting and reporting data about the system's execution
- ➔ improves understanding possible unexpected system behavior
- ➔ allows for improved reliability in some system applications

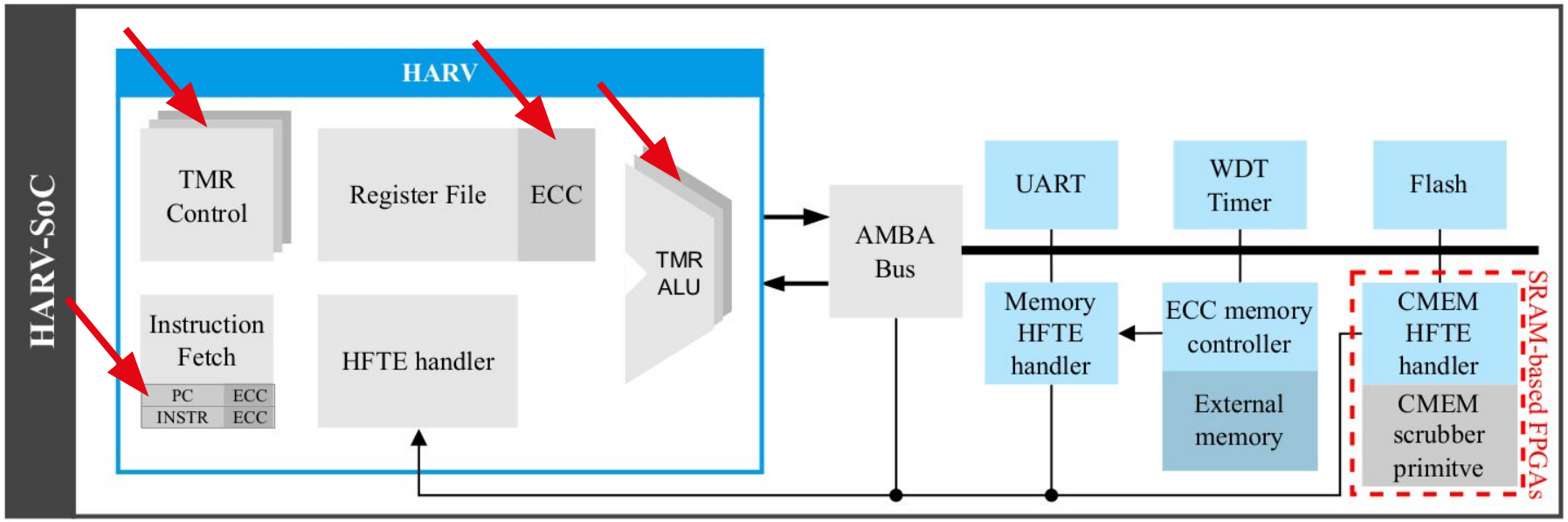


* definition by Google Cloud

HARV-SoC Overview

HARV-SoC - Hardened RISC-V System-on-Chip

- **Soft-core** SoC developed for FPGAs, with support for different technologies
- Reliable SoC with enhanced **fault observability**
- Based on the **RV32I** specification (supports **RV32E**), and extended with CSR, multiplication/division, compressed, and vector instructions
- Implements hardening in the processor core microarchitecture



Introduction

Radiation Effects in Electronics

Several different classified effects

- Total Ionizing Dose (TID)
- Displacement Damage (DD)
- Single-Event Effects (SEEs)

Effect in electronics depend on the type of particle

- Electrons and protons → Total Dose
- Heavy ions and protons → Single-Event Effects (SEEs)

Permanent, intermittent, or temporary faults

Radiation Effects in Electronics

Single-Event Effects

- Single-Event Upset (SEU)
  - Corruption of a stored memory element due to a particle strike
- Single-Event Transient (SET)
  - Associated with analog circuits and combinational logic in digital circuits
- Single-Event Functional Interruption (SEFI)
  - Leads to temporary non-functionality
- Single-Event Latch-Up (SEL)
  - Abnormal high-current state in a device

Reliable Systems

Fault tolerance techniques

Correct or mask errors

Can be implemented at logical, architectural, and application levels

Examples:

- Information redundancy
- Spatial redundancy
- Temporal redundancy
- Software redundancy
- Mixed redundancy

RISC-V

ISA proposed by Waterman, Lee, Patterson, Asanovic (2011)

Basic Instruction-sets

- RV32I
- RV32E
- RV64I

Extension modules:

- RV32M, RV32F, RV32D, RV32C, RV32V, ...

RISC → Simple instructions

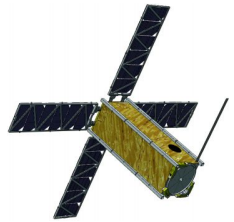
- No complex memory access

Proposal

Design of a RISC-V-based processor and System-on-Chip for utilization in harsh environments

- ➔ Provide reliability in the processor's architectural level
- ➔ Understand the behavior of the SoC in radiation environments
- ➔ Enable the applications to deal with radiation-induced faults

HARV-SoC Design Timeline



Basic processor
First version of the hardened processor with a single-cycle architecture

SoC design
Extension to a System-on-Chip

Error Observability
Improvement of analysis of the processor for irradiation experiments

Enhanced Observability
Improved error reporting and analysis of processor in irradiation experiments

Operating System
FreeRTOS support

Proton Irradiation

Mixed-Field Experiment

Vector Extension

ROBUSTA-3A
Flight as a payload for the processor characterization

Compressed Extension

Platform
Interface and documentation for engineers

2019 2020 2021 2022 2023 2024 2025

Simulation
Characterization of the processor with fault injection simulation

Neutron Irradiation
First experiments of the processor in irradiation environment

Neutron Irradiation
Experiment with the designed preliminary observability

Neutron Irradiation
Evaluation of software recovery and SRAM-based programmable devices

Electronics

DFTS

RADECS

TNS

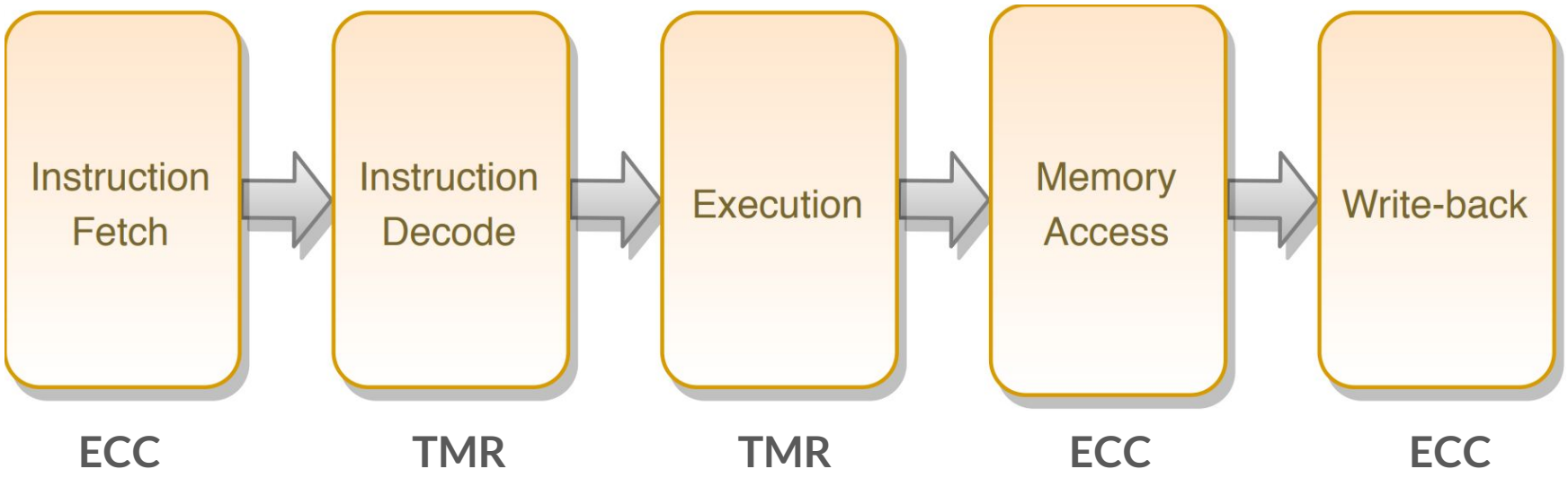
DFTS

Performance
Improvement of processor performance for demanding applications

HARV Hardening

RISC-V simple implementation is specified in 5 main stages (figure)

Fault tolerance implementation in the **organization level**



HARV Hardening

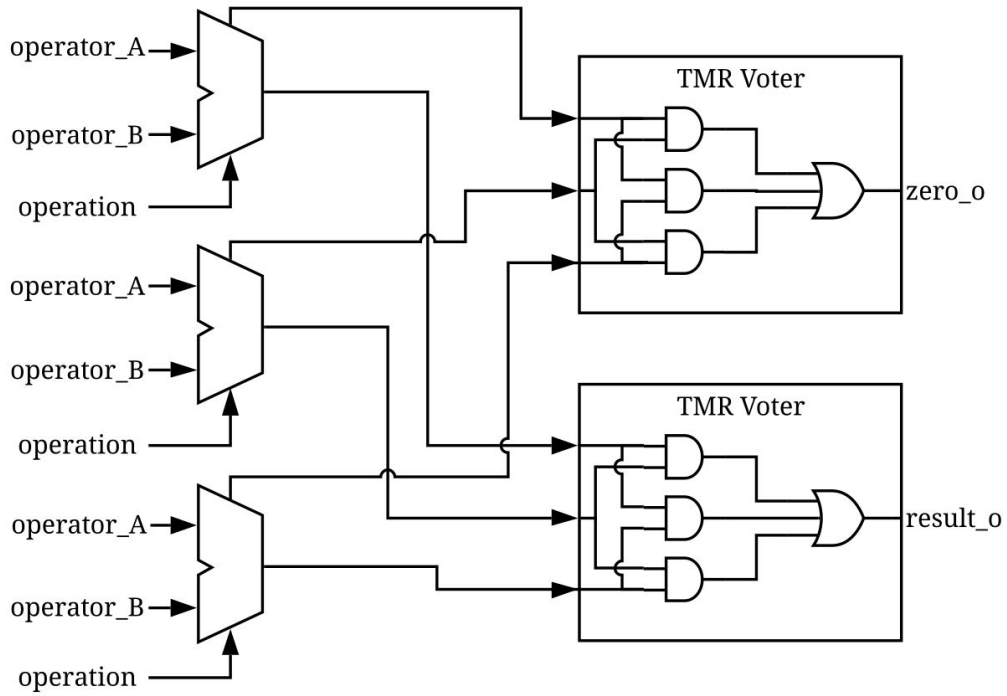
TMR: Triplication of the module

↓

Bitwise voting of the most common result

Implemented in combinational structures

- Control
- Arithmetic-Logic Unit (ALU)



HARV Hardening

Hamming code-based Error Correcting Code (ECC)

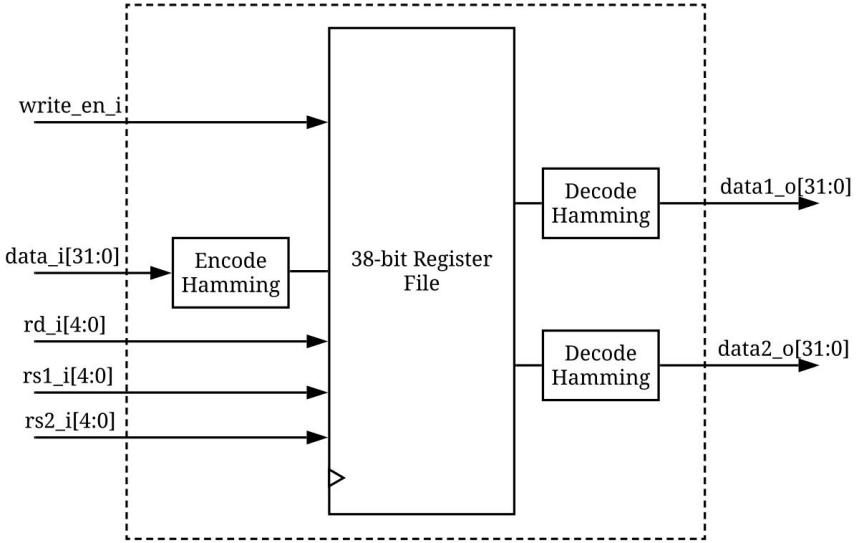
- Add 7 redundant bits to correct 1-bit upset and detect a double-bit upset

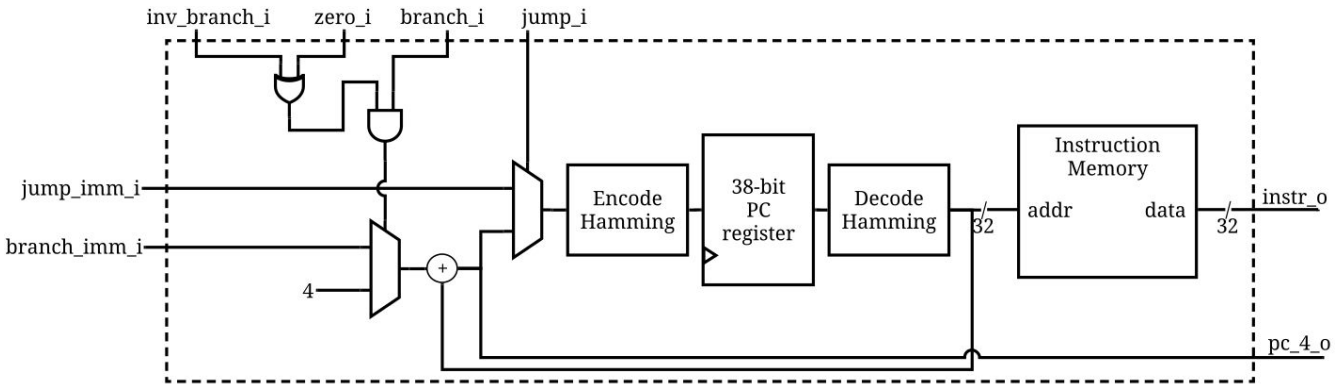
Require the implementation of two blocks

- ECC encoder → generates ECC data
- ECC decoder → corrects upset

Implemented in critical registers

- Program Counter (PC)
- Register file





Registers Hardened with TMR

- **mtvec**

  Stores the function address that will handle the traps. Bit upsets in this register will cause the program to jump to an incorrect address, which affects the execution flow of the program.

- **mie**

  Stores the enabled interrupt flags. Bit upsets may cause applications that rely on interrupts for their execution, such as operating systems, in which an interrupt disabling will result in an application failure.

- **hardening**

  It is a read-write-enabled custom CSR that stores the enabled hardening of the HARV. It stores the correction-enable flags for each hardening component. Therefore, bit upsets in this register affect the correction of errors, which may result in unwanted execution experiments.

Enhanced Observability

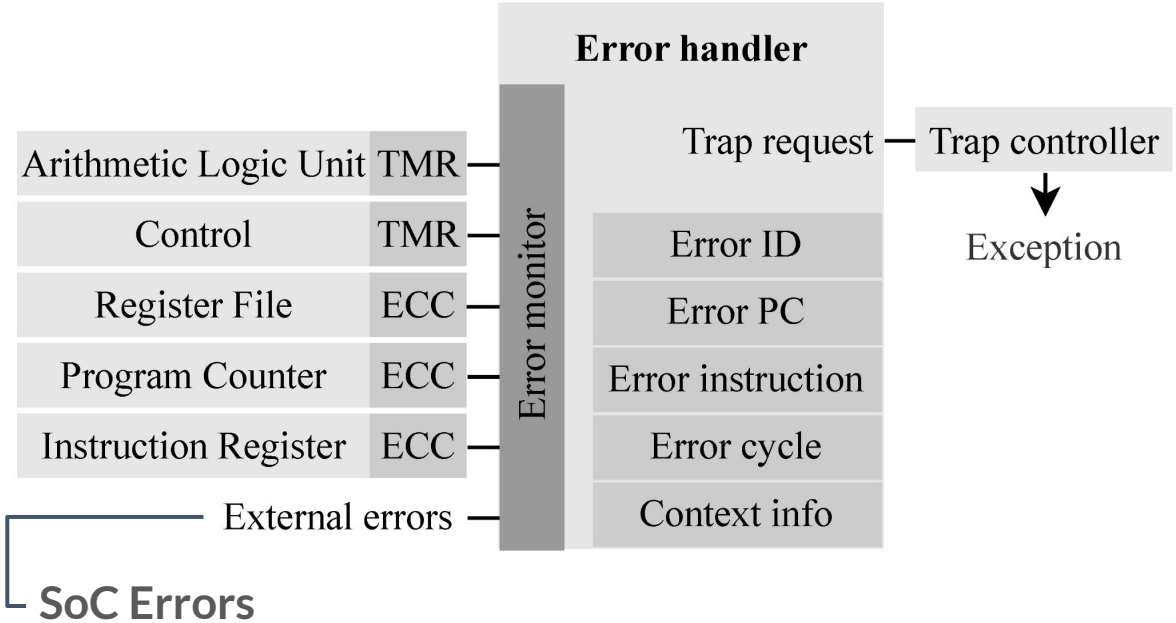
HARV-SoC

Combination of TMR and ECC for most critical structures

Added Observation of the hardened structures of the processor and SoC

- Provides detailed information about the processor context and errors

HARV Fault Tolerance Exceptions (HFTE)



SoC Errors

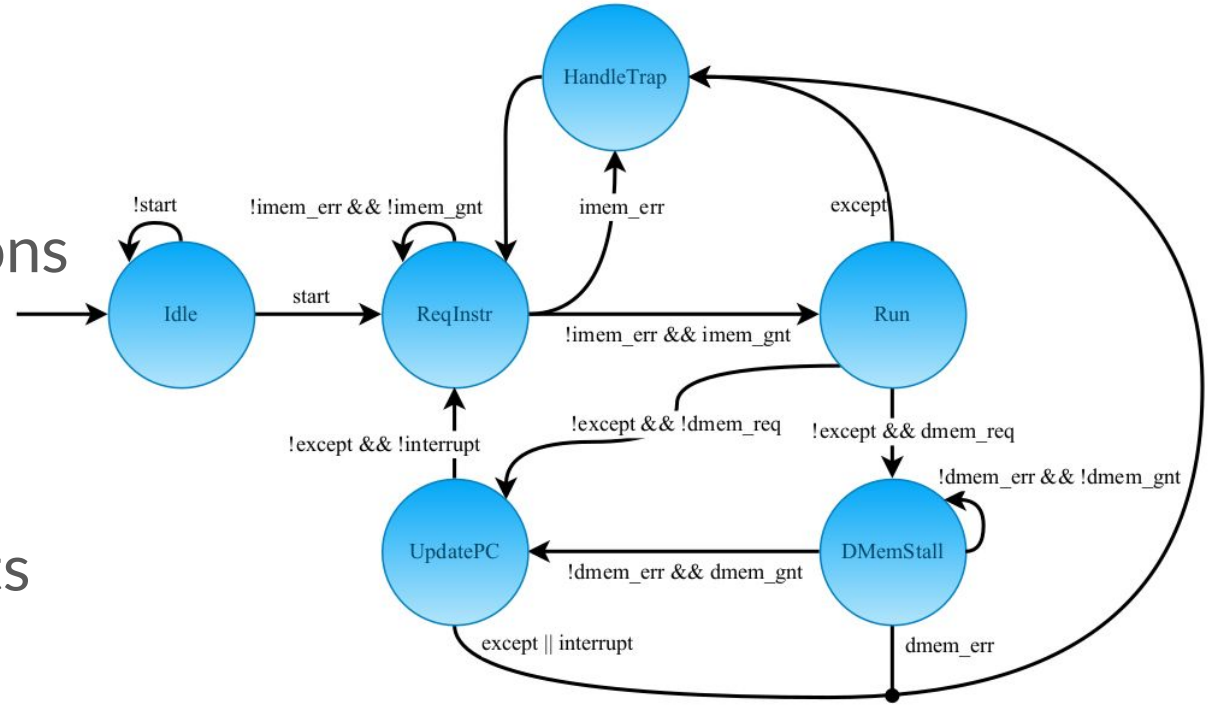
- Upsets in the external memory (SECDED)
- Peripheral access timeout
- Watchdog timer

HARV Processor core

Processor core controller

- Multi-cycle implementation
- Supports RV32I and CSR instructions
- AMBA bus access

State machine handles traps and events from the radiation-induced errors



HARV-SoC: Detailed Error Information

Several information are reported through memory-mapped registers

- Type of error
- Loaded instruction
- Encoded data (upset included)
- ALU output
- Detection cycle
- Application context

Table 6.2 – Memory-mapped registers with HARV HFTE information.

Register	Address	Information
EH_EVENT_ID	0x01000000	[0] PC SBU [1] PC DBU [2] IR SBU [3] IR DBU [4] register file SBU [5] register file DBU [6] register file SBU [7] register file DBU [8] control TMR error [9] ALU TMR error [10] external event (SoC) [11] peripheral access timeout
EH_PC	0x01000004	PC when error detected
EH_INSTR	0x01000008	instruction when error detected
EH_ENC_DATA	0x0100000C	encoded data according to error
EH_ENC_DATA_ECC	0x01000010	encoded data's ECC
EH_ALU_RESULT	0x01000014	ALU result
EH_MCYCLE	0x01000018	MCYCLE when error detected
EH_MCYCLEH	0x0100001C	MCYCLEH when error detected
EH_JALLOG_PTR	0x01000020	pointer to next register in jal logger
EH_JALLOG_SIZE	0x01000024	size of jal logger
EH_JALLOG_BASE	0x01000800	jal logger base

HARV-SoC: Detailed Error Information

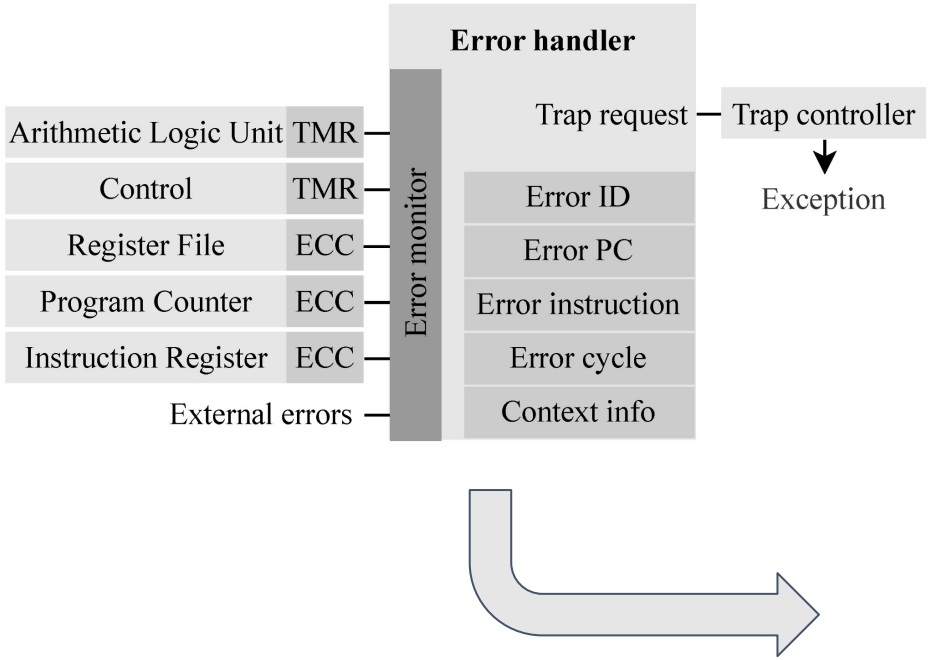
Abstraction for SoC errors

- Memory errors
- Peripheral errors
- Device errors

Table 6.3 – Memory-mapped registers in bus with HARV-SoC memory HFTE information.

Register	Address	Information
SOC_EH_EVENT_ID	0x70800000	[0] memory SBU [1] memory DBU [2] stuck-at error
MEM_EH_MEM_ADDR	0x70800004	address with error
MEM_EH_MEM_ECC_ADDR	0x70800014	ECC address
MEM_EH_ENC_DATA	0x70800008	encoded data
MEM_EH_ENC_DATA_ECC	0x7080000C	encoded data's ECC
MEM_EH_PREV_MEM_ADDR	0x70800018	previous address with error
MEM_EH_ENABLE	0x70800010	HFTE detection enable

Enhanced Observability



Error reporting

Performed using an exception service routine

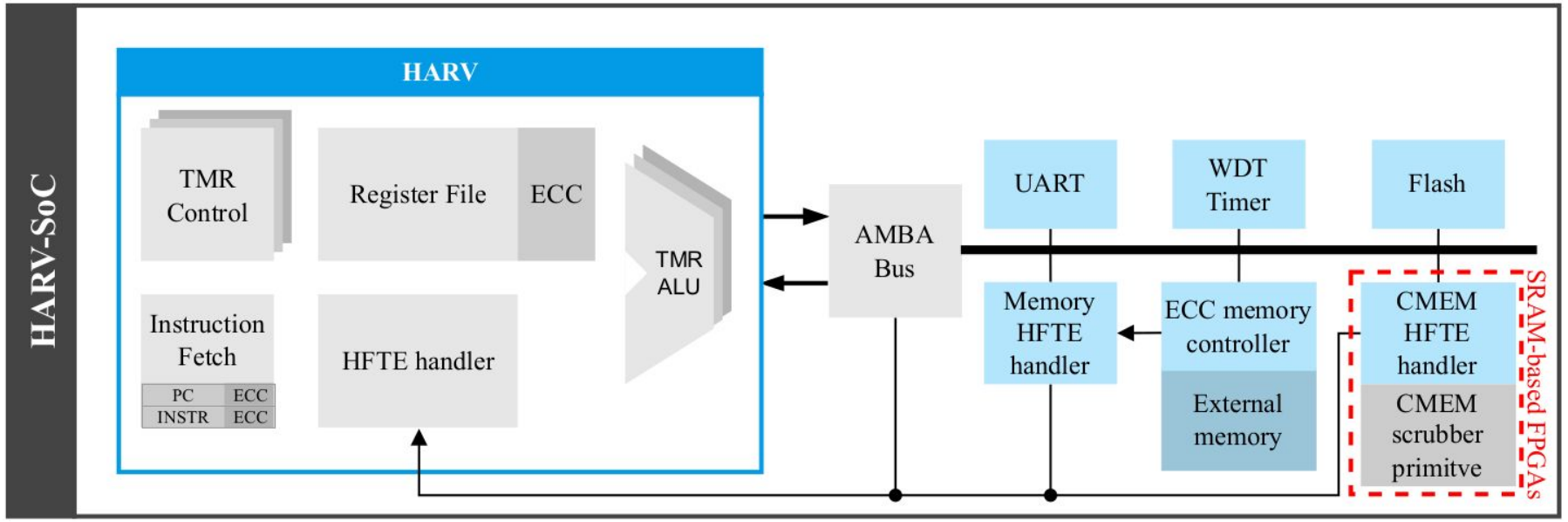
Applications may perform actions according to the error severity

- Correct SBU in the memory (rewrite word)
- **Use software recovery techniques**
- Cancel task execution
- Force soft reset
- …

HARV-SoC

Fault-tolerant RISC-V-based System-on-Chip

- Soft-core SoC developed by the group
- Reliable SoC with enhanced fault observability

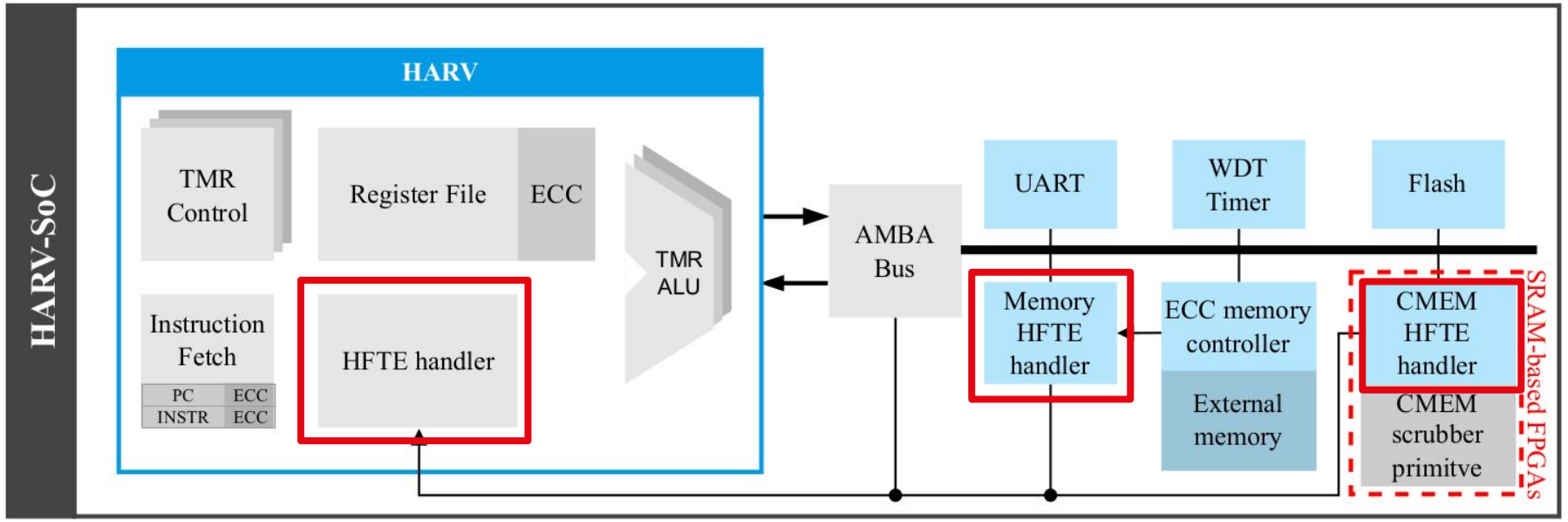


HARV-SoC

Separated error handlers for the processor and SoC

- Error handler
- SoC error handler

Simplifies error handler for processor



Resource Usage

Resource usage comparison of different configurations of HARV-SoC in the flash-based FPGA: Microchip's SmartFusion2 M2S010

- Depending on the FPGA and hardening parameters, resource usage varies from **10% to ~60%**

Error Handler	Hardening	4LUT	DFF	LEs [1]	LEs Usage
n.a.	disabled	4344	2150	4649	38.47 %
disabled	disabled	5922	2799	6471	53.55 %
disabled	enabled	8382	3131	8957	74.12 %
enabled	enabled	9730	4471	10596	87.69 %

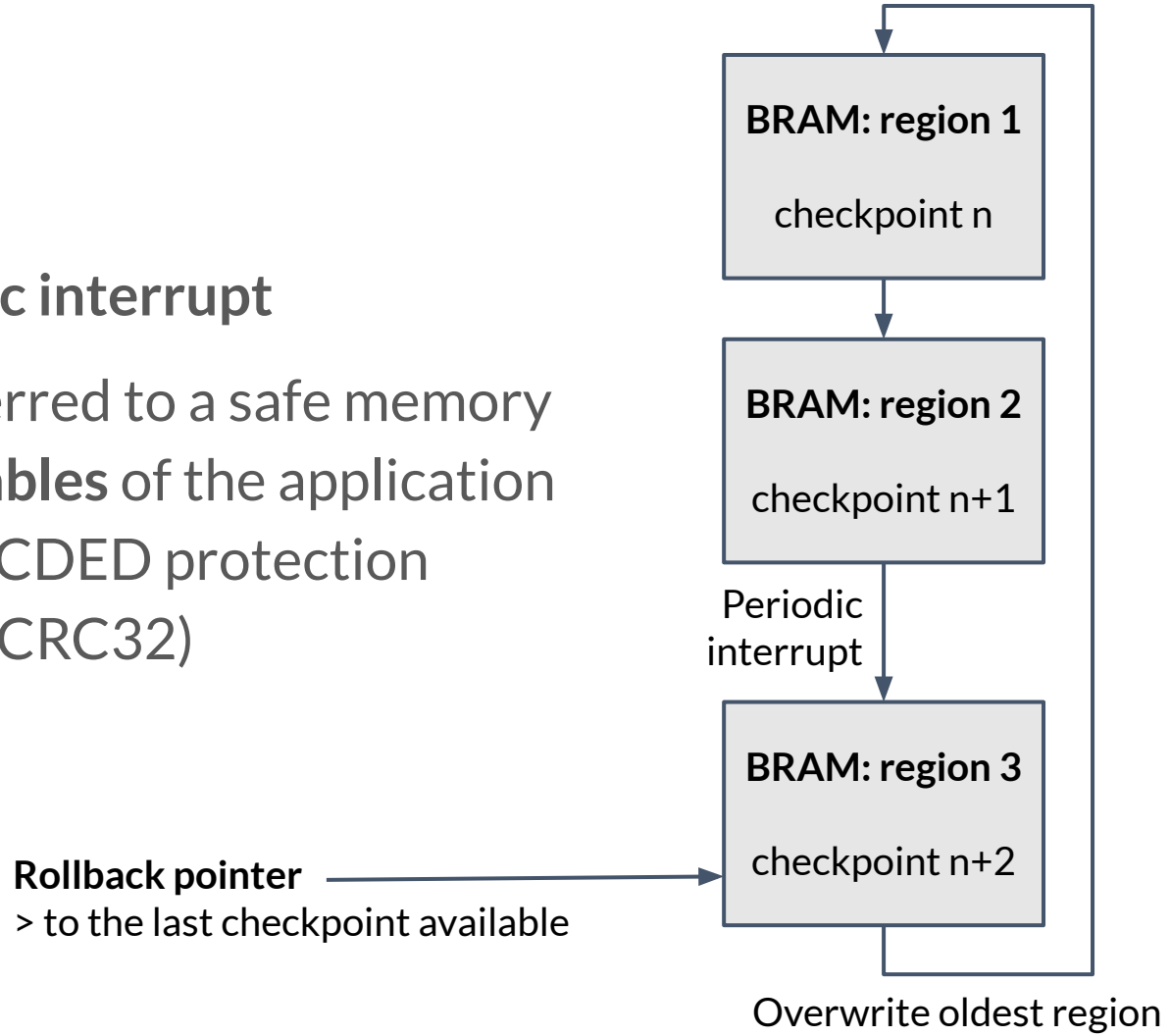
[1] The LE (Logic Element) report combines the 4-input Look-Up Tables (4LUTs) and DFFs (D-type Flip-Flops);

Example Implementation: Software Recovery

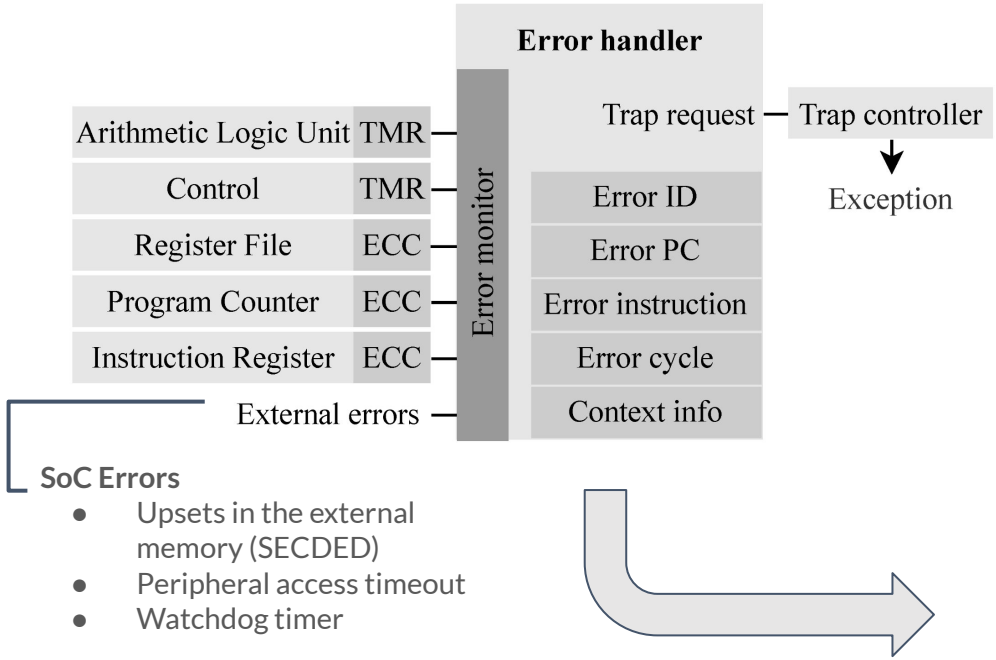
Checkpoint

Implemented in software through a **periodic interrupt**

- The context of the application is transferred to a safe memory
- Includes **register file** and the **local variables** of the application
- Saved in BRAM memory blocks with SECDED protection
- Includes an extra error detection code (CRC32)



Example Implementation: Software Recovery



Error reporting

Performed using an exception service routine

Applications may perform actions according to the error severity

- Correct SBU in the memory (rewrite word)
- **Use software recovery techniques**
- **Cancel task execution**
- Force soft reset
- …

Irradiation Experiments

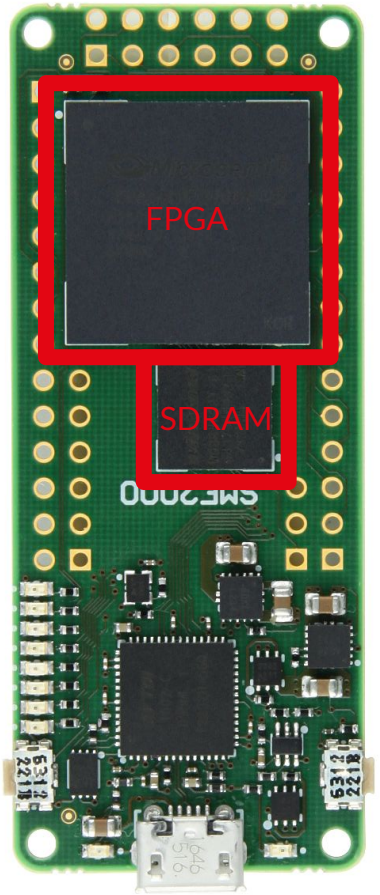
Experiment Setup

Microchip's Flash-based FPGA

- Flash-based Smartfusion2 M2S010
- Resilient against soft errors in the configuration memory
- SRAM-based block RAM
- D-type flip-flops

SMF2000 board

- Compact design → easy utilization in irradiation facilities
- External SDRAM component



Experiments

ChipIr Irradiation Facility

- Similar neutron spectrum to atmospheric environments
- Neutron flux of ~$5x10^6$ $n/cm^2/s$

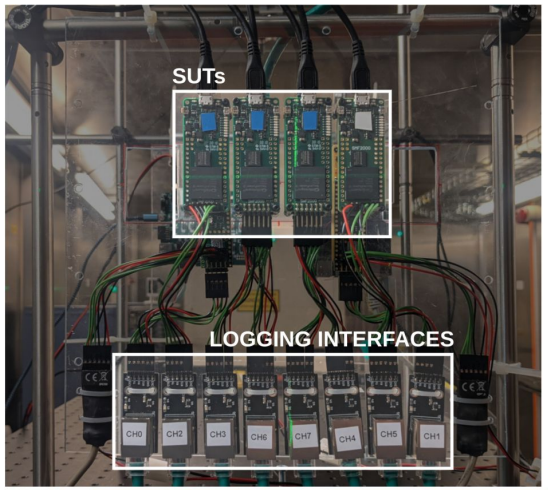
PARTREC Proton Irradiation Facility

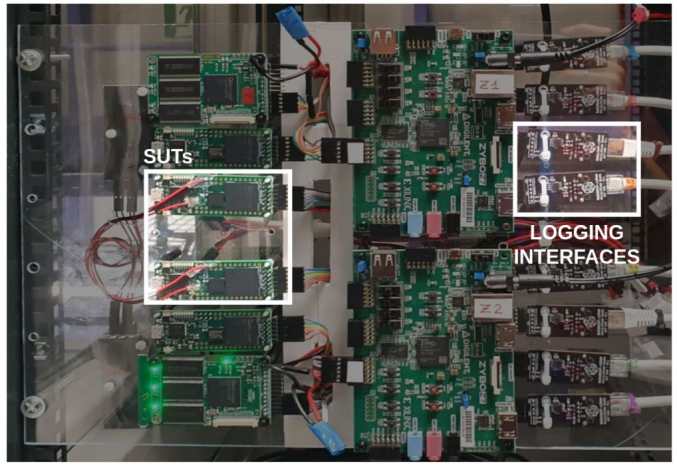
- Energy: 184 MeV
- Fluxes: up to 1×10^8 $p/cm^2/s$

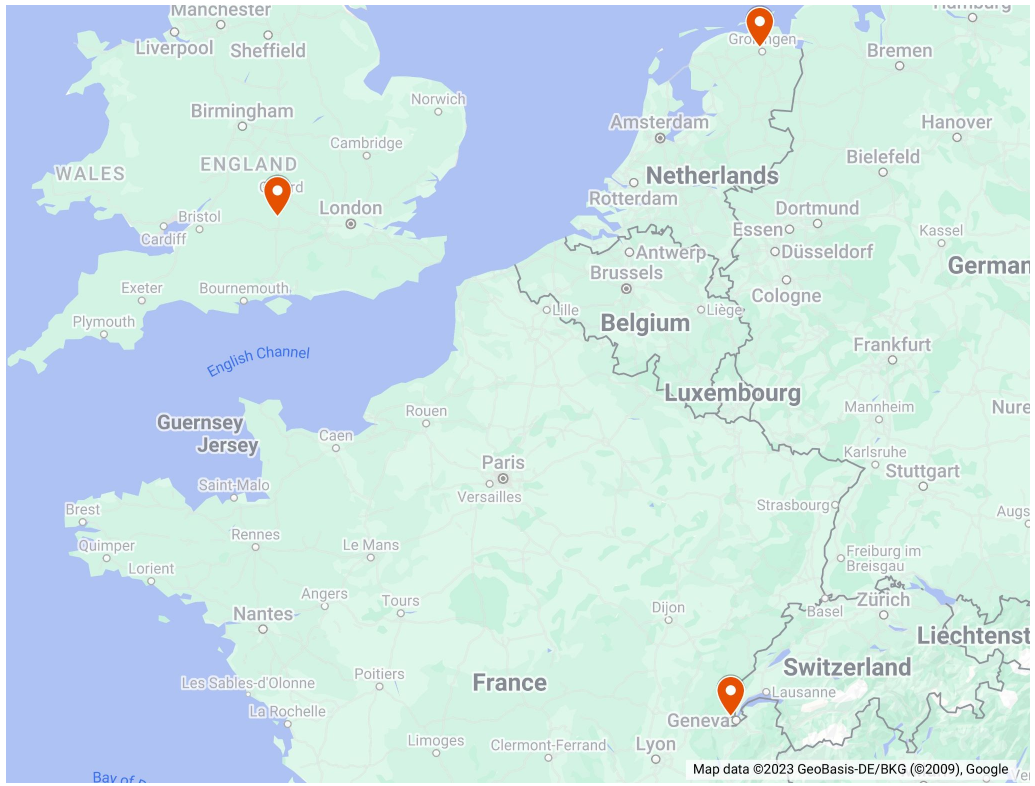
CERN CHARM

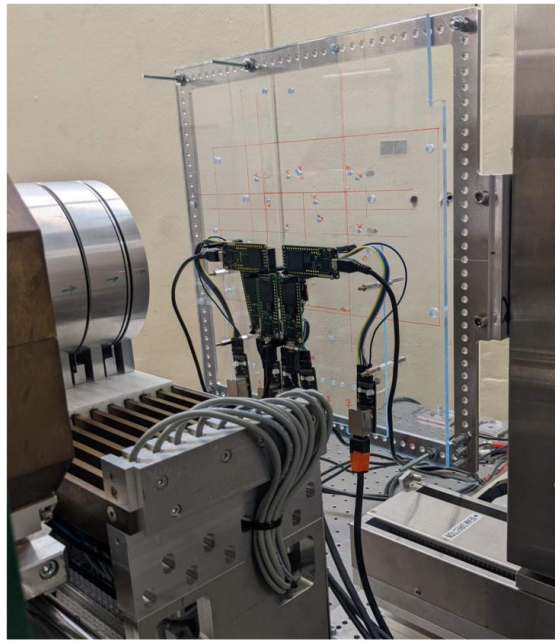
- Mixed-field irradiation
- Similar to space environment

Experiments









Experiment Setup

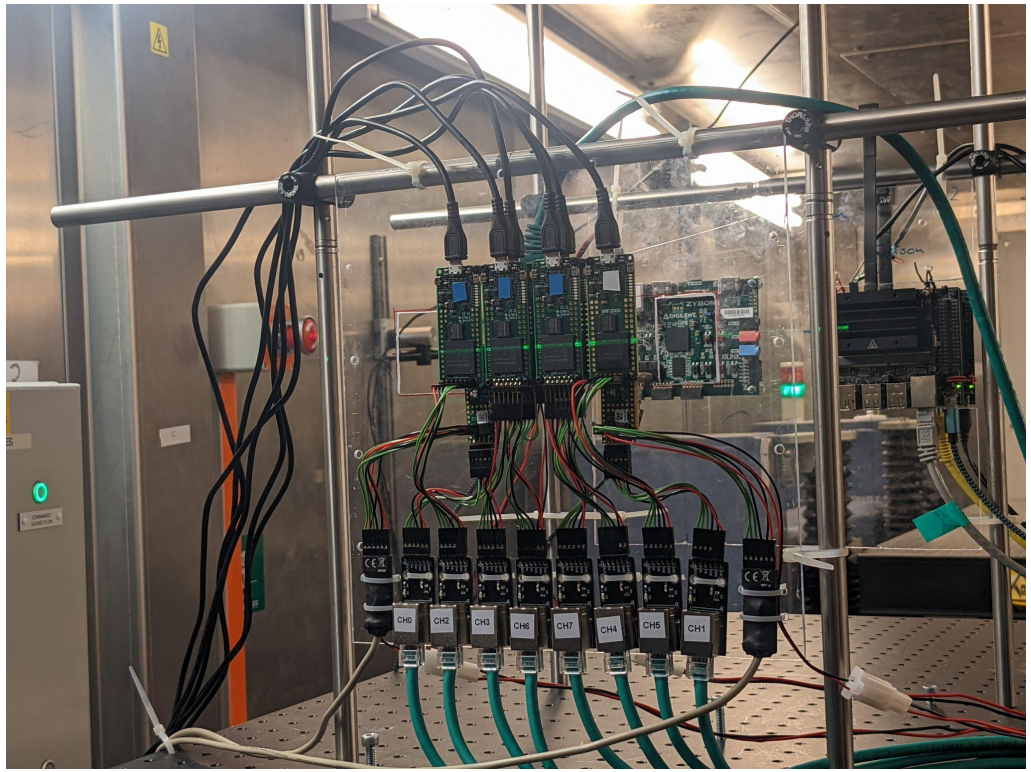
Four SMF2000 boards

Individual power supply with current monitoring

FTDI devices for logging the UART output

Coremark Benchmark

- List processing, matrix multiplication, state machines, and cyclic redundancy check
- Single-thread
- 200 iterations (~12 mins execution)
- Coremark debug flag enabled



Irradiation Experiments

Results: Neutrons

Neutron Experiment

Calculated device failure mean fluence to failure and cross-section

Similar numbers between boards

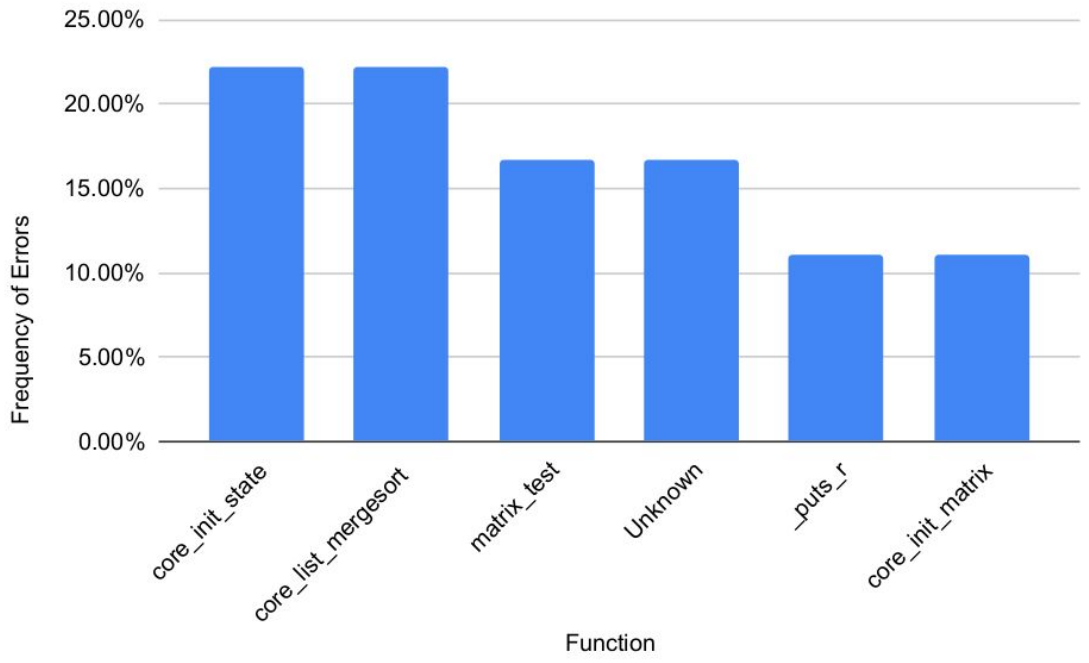
Board	MFTF [n/cm²]	Neutron Failure XS [cm²/device]
#B1	1.66×10^{10}	6.03×10^{-11}
#B2	1.85×10^{10}	5.42×10^{-11}
#B3	2.19×10^{10}	4.56×10^{-11}
#B4	2.15×10^{10}	4.66×10^{-11}
#B5	1.89×10^{10}	5.29×10^{-11}
#B6	2.60×10^{10}	3.84×10^{-11}

MFTF: Mean Fluence to Failure.

Neutron Experiment

Identified the execution context using the enhanced observability

- Most errors in core_init_state and core_list_mergesort coremark's functions
- Few errors during print operation



Irradiation Experiments

Results: Mixed-Field

Mixed-Field Experiment

Used only two boards due to cabling and space restrictions

Almost equal MFTF and cross-section

Board	HEH MFTF [H/cm²]	HEH Failure XS [cm²/device]
#B1	1.31×10^{11}	5.20×10^{-12}
#B2	1.31×10^{11}	5.20×10^{-12}

Mixed-Field Experiment

Improved classification of each type of error

- Calculated cross-section for each of them

Most were from the memory, followed by register file

Error	#Errors	Percentage	HEH XS [cm²/device]
Memory single-bit upset	71	52.21%	7.38×10^{-11}
Memory double-bit upset	35	25.74%	3.64×10^{-11}
Register file single-bit upset	21	15.44%	2.18×10^{-11}
Load access fault	6	4.41%	6.24×10^{-12}
Store access fault	2	1.47%	2.08×10^{-12}
Program counter double-bit upset	1	0.74%	1.04×10^{-12}

Mixed-Field Experiment

Enhanced enabled analysis of number of errors per instruction

- Mostly memory access due to memory being the most significant source of errors

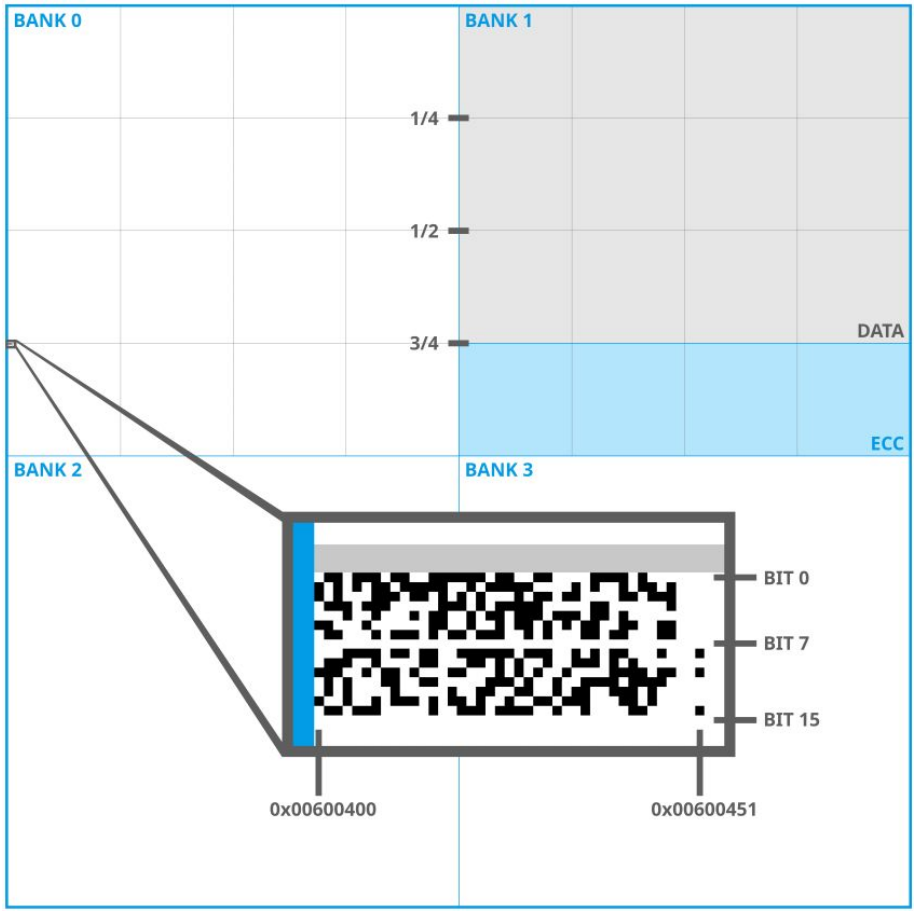
Class	Instruction	#Occurrences	Percentage	HEH XS [cm²/device]
Memory access	`lbu`	78	57.35%	8.11×10^{-11}
	`sw`	31	22.79%	3.22×10^{-11}
	`lw`	10	7.35%	1.04×10^{-11}
	`sb`	4	2.94%	4.16×10^{-12}
	`lh`	2	1.47%	2.08×10^{-12}
	`lb`	1	0.74%	1.04×10^{-12}
Flow control	`jalr`	4	2.94%	4.16×10^{-12}
Arithmetic	`addi`	3	2.21%	3.12×10^{-12}
	`add`	3	2.21%	3.12×10^{-12}

Detected Block Error

Detected a block error in the SDRAM memory

- Did not affect the execution
- ECC part of the memory

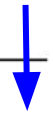
Reported by using several traps



Error Propagation

Error propagation reduced in 2.7x

Configuration	Error	#Errors	#Prop. Errors	Error Prop. HEH XS [cm²/device]
Baseline	Register file single-bit upset	13	3	3.12×10^{-12}
	PC double-bit upset	1	1	1.04×10^{-12}
	Load access fault	4	4	4.16×10^{-12}
Hardened	Store access fault	2	2	2.08×10^{-12}
	Memory double-bit upset	1	1	1.04×10^{-12}



2.7x reduction

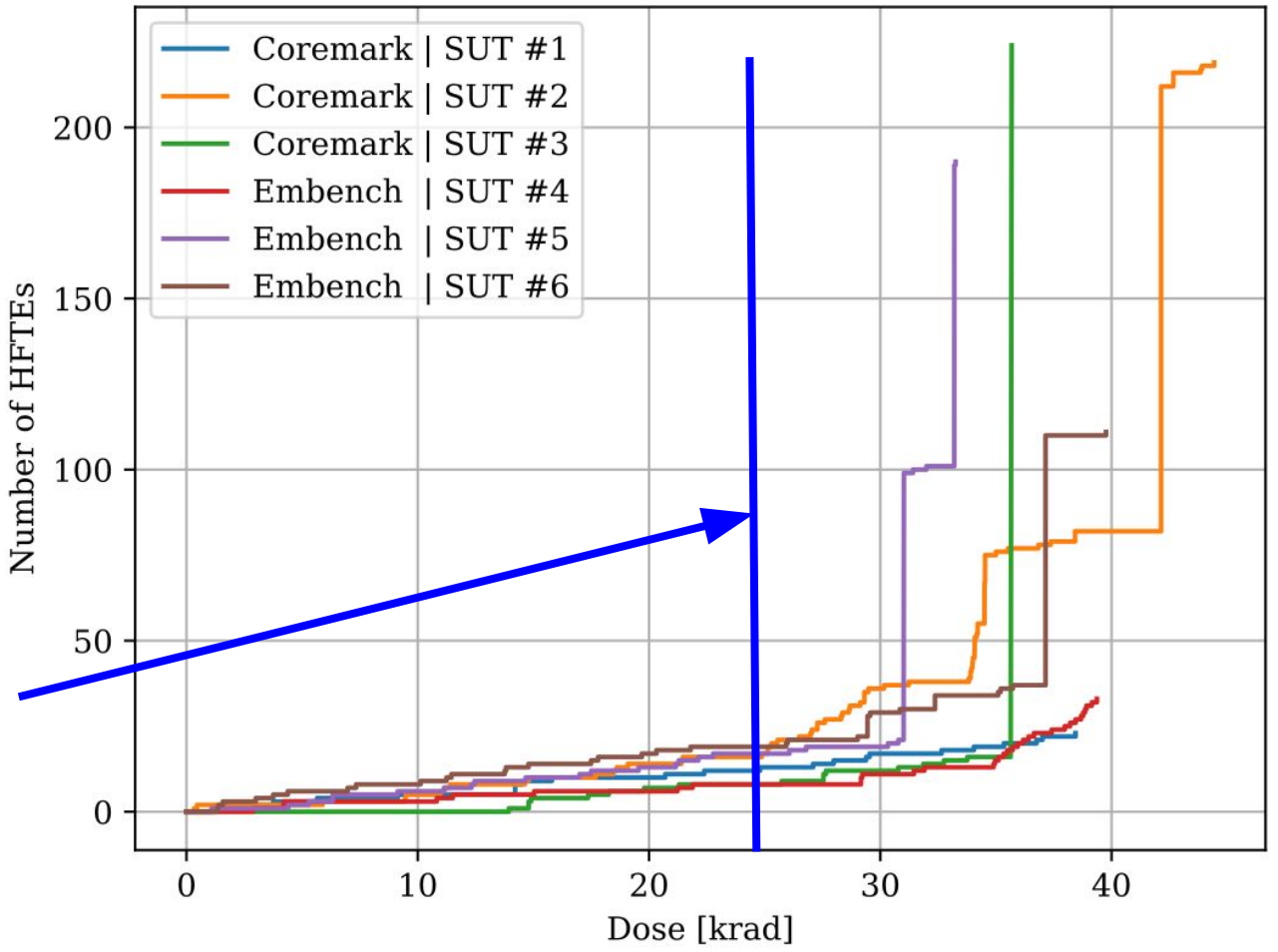
Irradiation Experiments

Results: Protons

Proton Experiment

Number of reported HFTEs increased significantly after 25 kRAD because of dose effect

Threshold for SEE analysis defined as **25 krad**



Proton Experiment

Calculated cross-section for each defined HFTE

- Mostly from the memory, followed by the register file

HFTE Type	#HFTE[1]	HFTE XS[1] [cm²/device]
Memory single-bit upset	44	1.88×10^{-11}
Register file single-bit upset	17	7.25×10^{-12}
Timeout load access fault	9	3.84×10^{-12}
Memory double-bit upset	2	8.53×10^{-13}
Program counter double-bit upset	1	4.27×10^{-13}
Instruction register double-bit upset	1	4.27×10^{-13}
Program counter single-bit upset	1	4.27×10^{-13}
Load access fault	1	4.27×10^{-13}
Total	76	3.24×10^{-11}

[1] Analysis considering before device failure at 25 *krad*.

Proton Experiment

Error propagation reduced in 4.5x

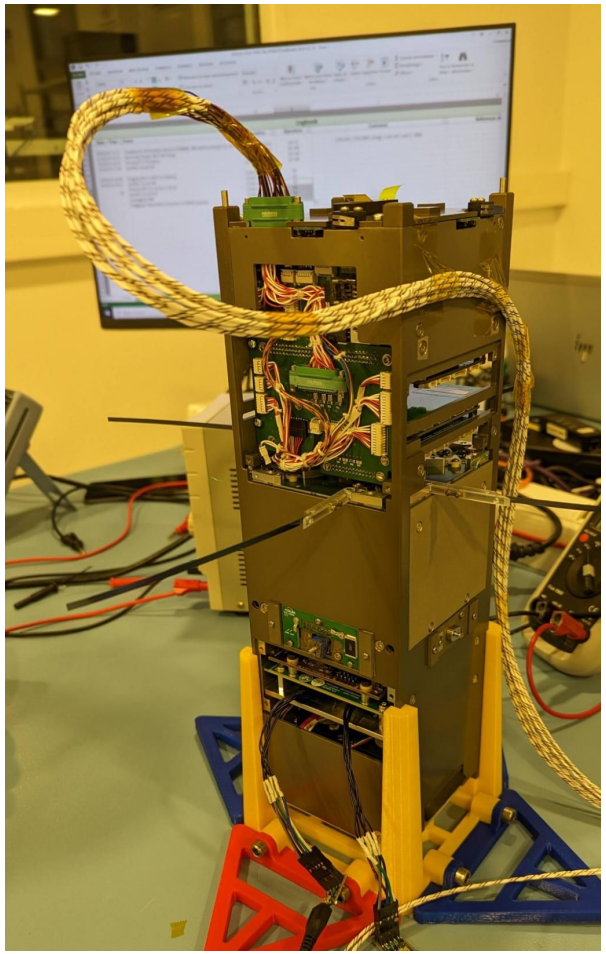
Table 4.13 – Error propagation per HFTEs for each HARV-SoC configuration.

Configuration	HFTE Type	#HFTE[1]	#Prop. HFTEs[1]	HFTE Prop. XS[1] [cm²/device]
Baseline	Memory single-bit upset	21	0	-
	Register file single-bit upset	6	1	6.99×10^{-13}
	Timeout load access fault	6	4	2.80×10^{-12}
Hardened	Memory single-bit upset	23	0	-
	Register file single-bit upset	11	0	-
	Timeout load access fault	3	1	6.28×10^{-13}
	Memory double-bit upset	2	0	-
	Program counter single-bit upset	1	0	-
	Instruction register double-bit upset	1	0	-
	Program counter double-bit upset	1	0	-
	Load access fault	1	0	-

[1] Analysis considering before device failure at 25 *krad*

4.5x reduction

ROBUSTA-3A - HARV payload



ROBUSTA-3A - CSUM - Centre spatial de l'Université de Montpellier

- 3U cubesat
- Launched in July 9th, 2024
- LEO orbit at 600km
- Expected to last at least 2 years

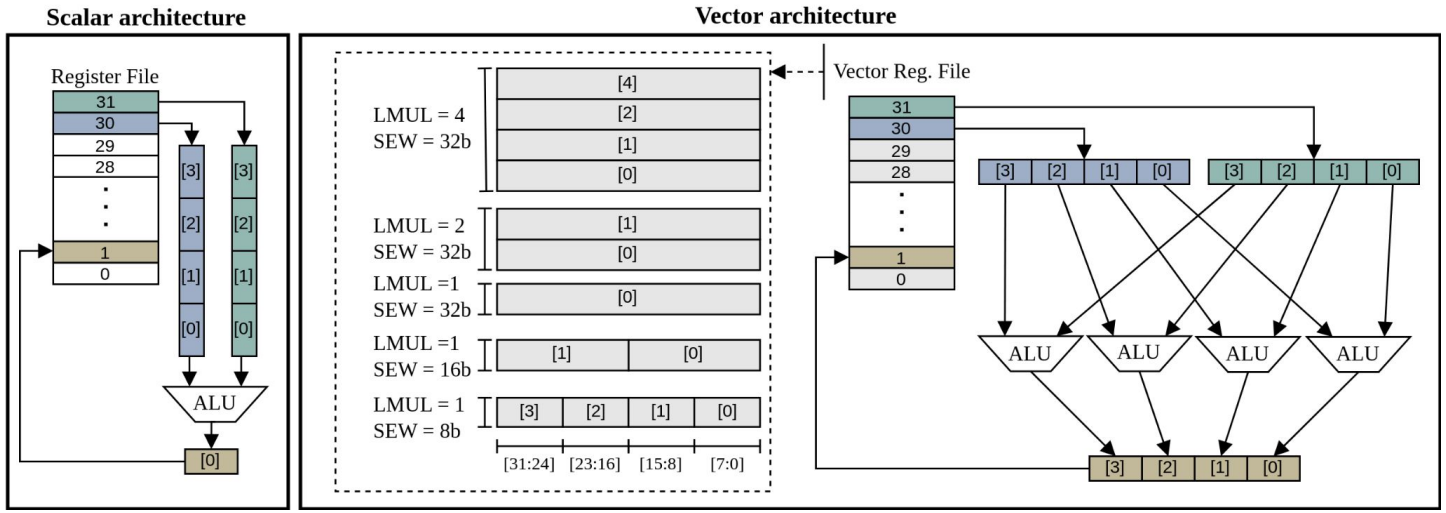
HARV payload

- Developed for characterizing the HARV-SoC
- >588 hours of experiment so far

Artificial Intelligence Acceleration

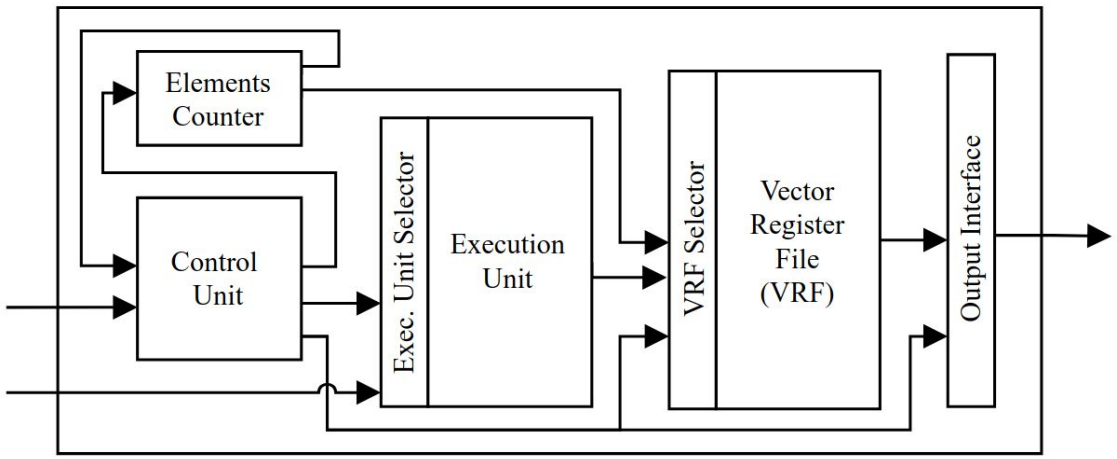
RISC-V Vector Extension

- Explore Data-level Parallelism (DLP)
  - One instruction operates in multiple elements
- Partitioning and grouping the Vector Register File
  - Single Element Width (SEW), Vector Length (VL), Length Multiplier (LMUL)
- Improve processing latency and power consumption
- Speed up onboard applications
  - Edge AI, TinyML

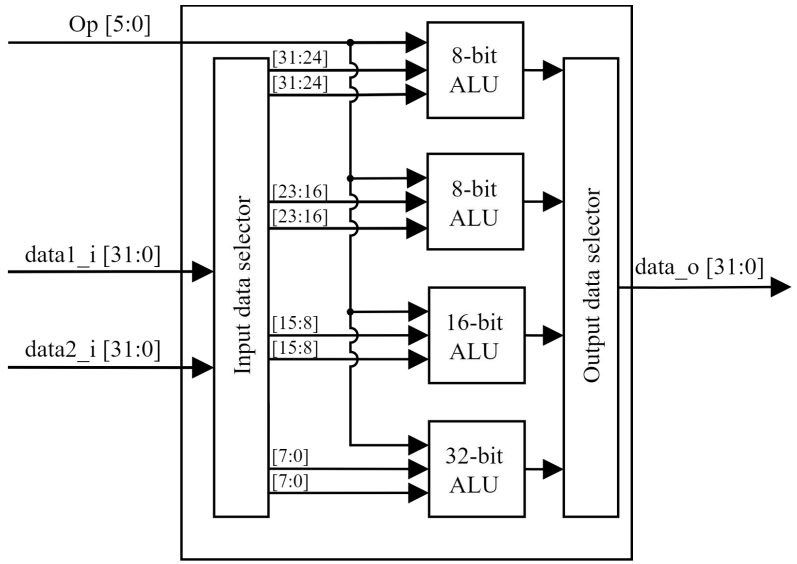


HARV Vector Extension Unit (VEU)

- Supports a subset of the RISC-V Zve32x
  - Restricts the element width to 32 bits and supports only integer data
  - Vector configuration, unit-stride load and store instructions, integer arithmetic operations



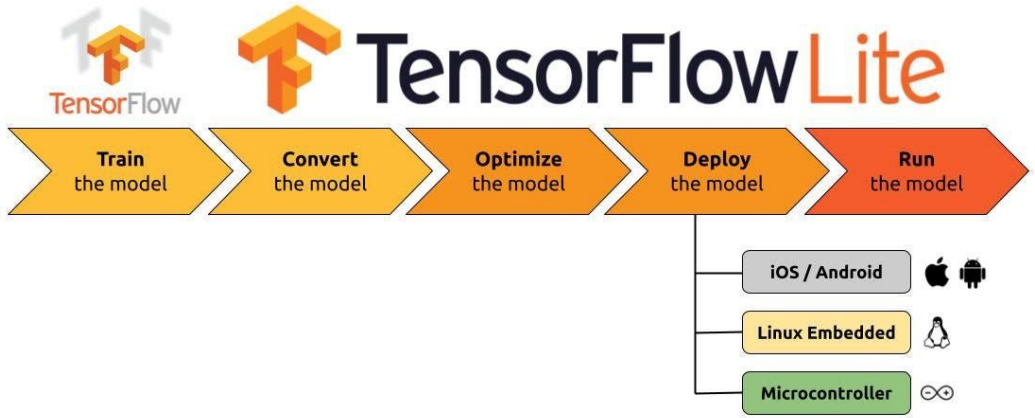
VEU Architecture overview



Execution unit overview

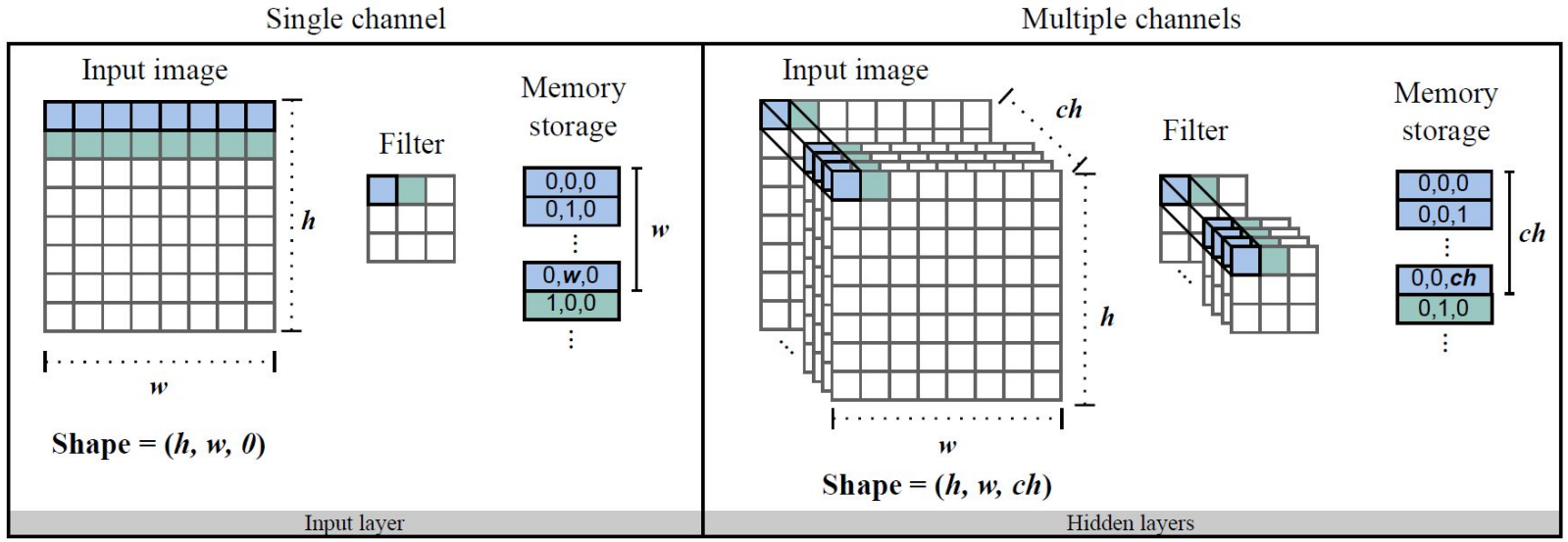
Artificial Intelligence and TensorFlow Lite Micro

- Artificial Intelligence enables local processing, real-time control, and autonomous decisions on embedded devices
- TensorFlow Lite Micro (TFLM)
  - Lightweight, open-source inference framework for devices with constrained resources
  - Run neural network inferences directly on the MCU



Artificial Intelligence and TensorFlow Lite Micro

- We customized TFLM for the HARV-SoC to exploit **HARV-SoC's VEU**
  - Modifications on the convolution kernel for vectorization
  - Different strategies depending on the input channel depth



Test Experiment

- Convolutional Neural Network
  - MNIST Dataset
  - 3 conv layers, 2 dense, 12.5k params
  - 98% accuracy
- Fault injection simulations:
  - Select random register signals and time to inject an error
  - Comparison of baseline and hardened SoC, with scalar and vector convolution kernels

Results: Reliability

Reliability analysis of the vector extension by fault injection simulation

- **Correct**: Correct execution
- **Benchmark error**: Incorrect results
- **Hang**: Execution does not end or no results are produced
- **TFLM Halted**: One of the TFLM internal verifications failed
- **Exception**: Program disrupted

FAULT INJECTION EXECUTIONS.

Execution classification	Baseline		Hardened	
	Original	Vector	Original	Vector
Correct	25	49	116	124
Benchmark error	2	0	0	0
Hang	6	5	0	0
TFLM halted	57	56	0	0
Exception	38	18	12	4
Reliability analysis				
Simulated fluence*	1.2×10^{13}	9.6×10^{12}	2×10^{13}	1.3×10^{13}
Failure XS†	9×10^{-12}	8.3×10^{-12}	6×10^{-13}	3.1×10^{-13}
MWTF	1.25	1.62	10.67	32

*[n/cm^2]; †[cm^2/device]

Results: Performance

- Convolution layers represent more than 90% of inference time
- Vectorization reduces inference time by 1.58 times compared to scalar kernel

PERFORMANCE VARIATION OF THE INFERENCE WITH DIFFERENT CONVOLUTION VERSIONS.

Version	Conv1	Conv2	Conv3	Total Exec	Speedup
Original	69.48M	188.91M	39.84M	306.85M	–
Vector	50.60M	118.78M	20.76M	193.99M	1.58x

Results: Reliability

- Vectorized kernels lead to more correct executions
  - Eliminates benchmark errors and reduces exceptions
- Mean Work to Failure (MWTF) increased by approximately 30% (baseline) and 200% (hardened) with vectorized kernels
- Reliability improvement is due to reduced execution time, which limits fault exposure

FAULT INJECTION EXECUTIONS.

Execution classification	Baseline		Hardened	
	Original	Vector	Original	Vector
Correct	25	49	116	124
Benchmark error	2	0	0	0
Hang	6	5	0	0
TFLM halted	57	56	0	0
Exception	38	18	12	4
Reliability analysis				
Simulated fluence*	1.2×10^{13}	9.6×10^{12}	2×10^{13}	1.3×10^{13}
Failure XS†	9×10^{-12}	8.3×10^{-12}	6×10^{-13}	3.1×10^{-13}
MWTF	1.25	1.62	10.67	32

*[n/cm^2]; †[cm^2/device]

Summary and Conclusion

Summary and Conclusion

HARV processor core

- Fault tolerance at the microarchitecture level
- Reliability of the system in an abstract manner for programmers

HARV-SoC

- Error detecting and reporting
- Microcontroller capabilities

Performed fault-injection experiments to evaluate the processor

- Characterized with neutrons, mixed-field, and protons
- Analyzed the failures caused by the errors within the processor and its propagation

Summary and Conclusion

HFTE error handler

- Temporarily storing the information about the detected errors in the processor core and SoC structures
- Report radiation-induced errors through RISC-V exceptions

Identified the radiation-induced events and provided information that points to the compromised structure

Extended the analysis by testing operating systems and using software recovery techniques

Future Work

Improve HARV processor, platform

- Performance, capabilities, and tools for programmers

Caching: The implementation of caches for the processor is a very important step for the improvement of the processor's performance

Pipelining: Using pipelines in processor cores is already a standard of the industry since it increases the operating frequency and overall throughput of the processor

Debugging: Currently, the processor does not provide debugging capabilities, which affects the development flow for programmers

Multi-core: Systems with multi-core are important for applications that require performance, multitasking, and energy efficiency

Floating-point unit: representation is a must-have for a range of applications in various fields, and it needs to be efficient

HARV-SoC - A Fault-tolerant RISC-V Microcontroller for Dependable Applications

Université de Montpellier

Douglas A. Santos, Wesley Grignani, Carolina Imianosky, and Luigi Dilillo

3rd TWIN-RELECT Training School on
Fault-Tolerant AI Systems

14/01/2026