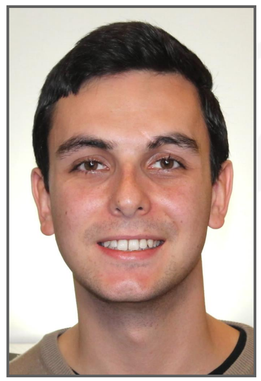
A Parameterizable CGRA-based Engine for Machine Learning Acceleration On-board Satellites

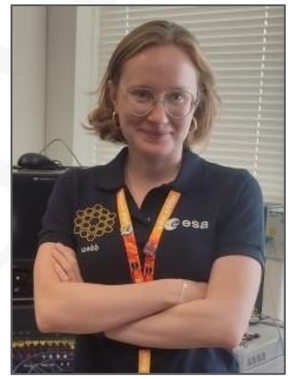
L. Zulberti, **M. Monopoli**, **G. Mystkowska**, S. Moranti, P. Nannipieri, L. Fanucci

6th SEFUW: SpacE FPGA Users Workshop



Matteo Monopoli
matteo.monopoli@phd.unipi.it
PhD Student
University of Pisa
Dept. Information Eng.

Gabriela Mystkowska
gabriela.mystkowska@phd.unipi.it
PhD Student
University of Pisa
Dept. Information Eng.



26th March 2025

Outline

1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing Work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

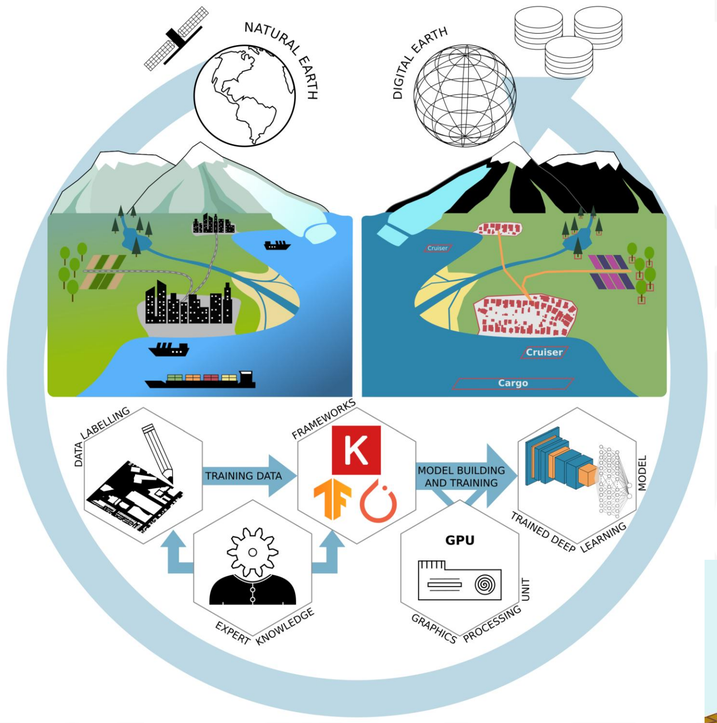
Outline

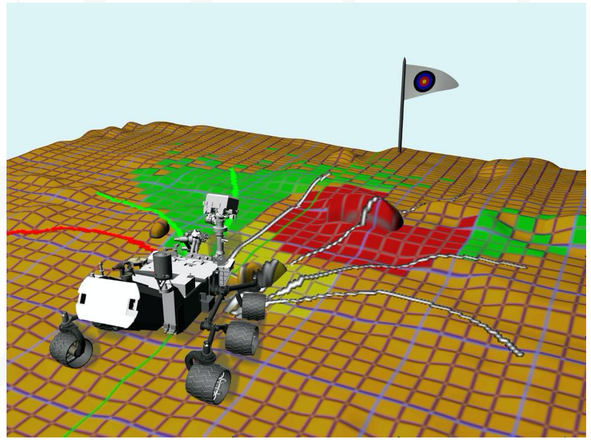
1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing Work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

Background – Motivation

AI Applications in orbit:

- Remote Sensing
  - Object detection
  - Weather forecast
  - Earth observation
- Autonomous spacecrafts
  - Vehicle docking
  - Probes
  - Landers, rovers
  - Deep space missions
  - Fault detection and isolation, recovery
- Data privacy increase
  - Downlink requirements reduction





Background – Hardware for AI Acceleration On-board

- High Computational Efficiency
  - Inference
  - Real-time processing in resource-constrained environments
  - Tailored for parallel processing, and NN operations
- Energy efficient
- Compact and Integrated Design:
  - Small form factors
- Customizability and Flexibility:
  - Reconfigurable to meet specific AI requirements
  - Adaptable to evolving AI models and algorithms without requiring new hardware
- Reliability
  - Must be able to withstand exposure to radiation
  - Protection against Single Event Effects (SEEs) through radiation-hardening or fault mitigation

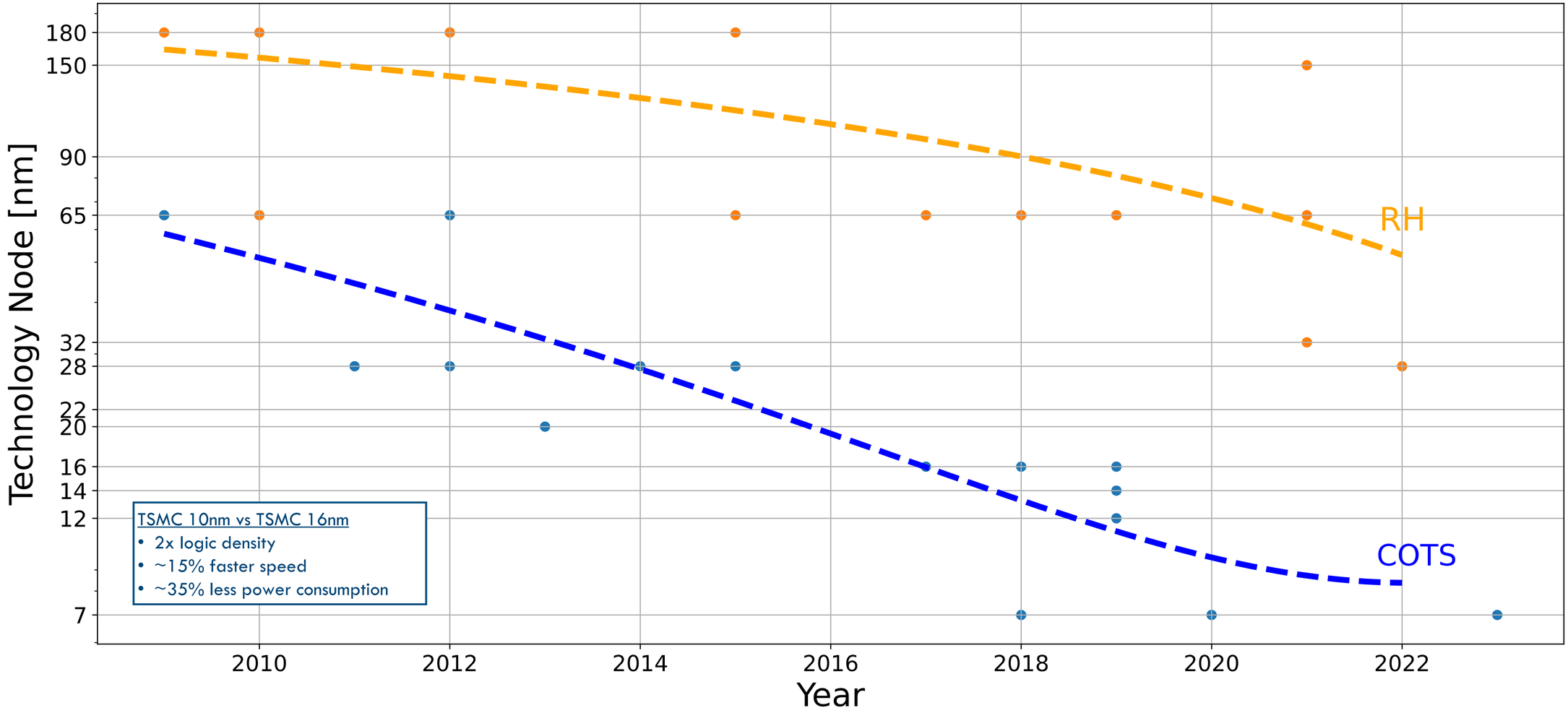
Outline

1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing Work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

State-of-the-Art

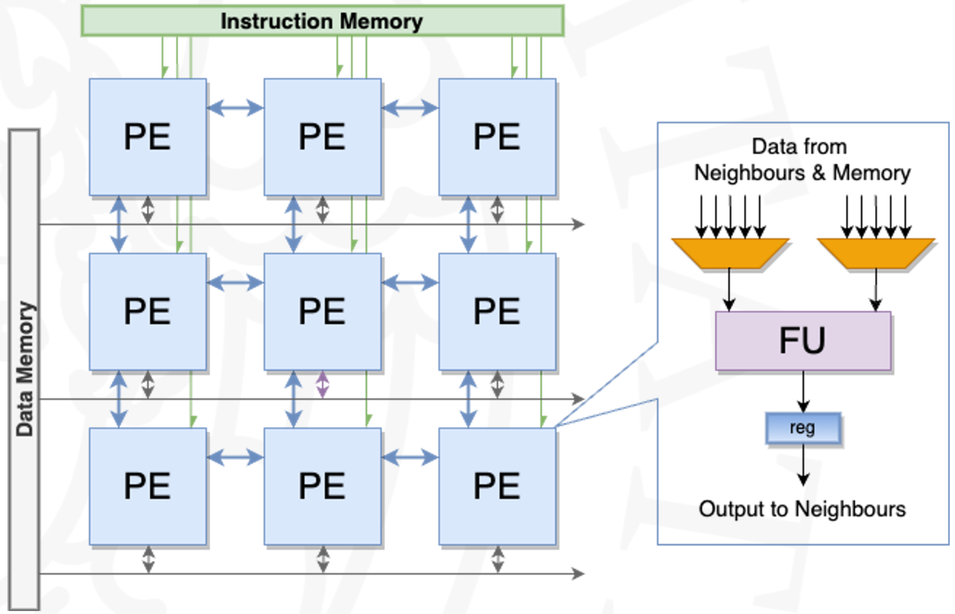
	Architecture	Chip	Pros	Cons
NOT RAD-HARD	GPU	NVIDIA Xavier, NVIDIA Jetson Orion, AMD Ryzen	COTS, High performance (up to 300 TOPS and 5 TOPS/W)	Not rad-hard, only short missions, only LEO
NOT RAD-HARD	VPU	Intel Myriad X		
NOT RAD-HARD	TPU	Google Coral		
RAD-HARD	FPGA	Xilinx Zynq UltraScale+, Microchip PolarFire	Flight heritage, Rad-hard, Flexible, Reprogrammable	General purpose, Relatively high energy consumption and low performance
RAD-HARD	CPU	Gaisler GR740		
RAD-HARD	Spatial Architecture	Xilinx Versal	Rad-tolerant, High performance (up to 430 TOPS)	Low interpretability (“black box”) Low predictability in time critical tasks, Relatively high power consumption (35W)
RAD-HARD	Systolic Array	HPDP	High flexibility High reconfigurability Low power consumption	Relatively old technology (STM 65 nm), Proprietary tools

SotA – Technology Available over the Years



SotA – Resource Optimised CGRA Hardware Accelerator

- 2D array of PE interconnected through NoC
- Speeding up compute-intensive inner loops
  - Digital signal processing
  - AI/ML workloads
- Heterogeneous computing, high parallelism
- Highly predictable (timing model)
- Highly parametrized architecture
  - Optimised resource utilisation
  - Scalable and flexible
- Low reconfiguration time



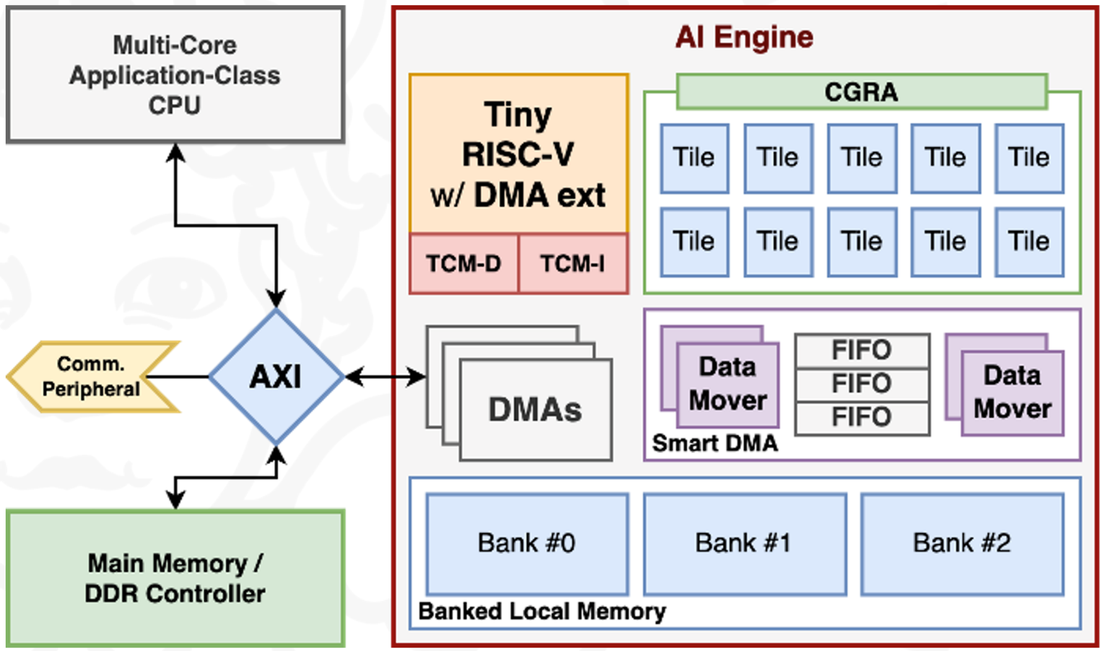
CGRA is	• Power efficient	compared to	FPGA
	• Power efficient • Higher performance		CPU
	• More flexible		Systolic Array
	• More flexible		ASIC

Outline

1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing Work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

UniPi – CGR-AI Engine

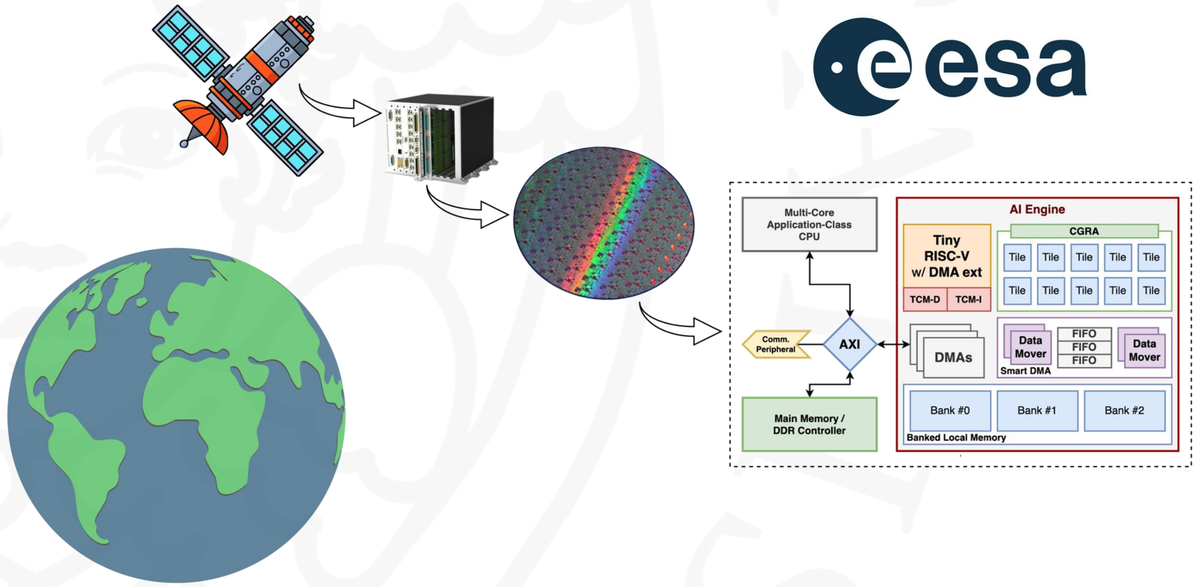
- CGRA-based accelerator for AI on-edge applications
  - Time-constrained applications (e.g., autonomous operations)
  - High-reliability applications
- Technology:
  - RHBD DARE65T std. cell
  - TSMC40LP std. cell
  - 7 nm radiation-hardened (in progress)
- Heritage of:
  - Three ongoing ESA supported projects
  - OPERAND project supported by Italian Ministry of Education and Research



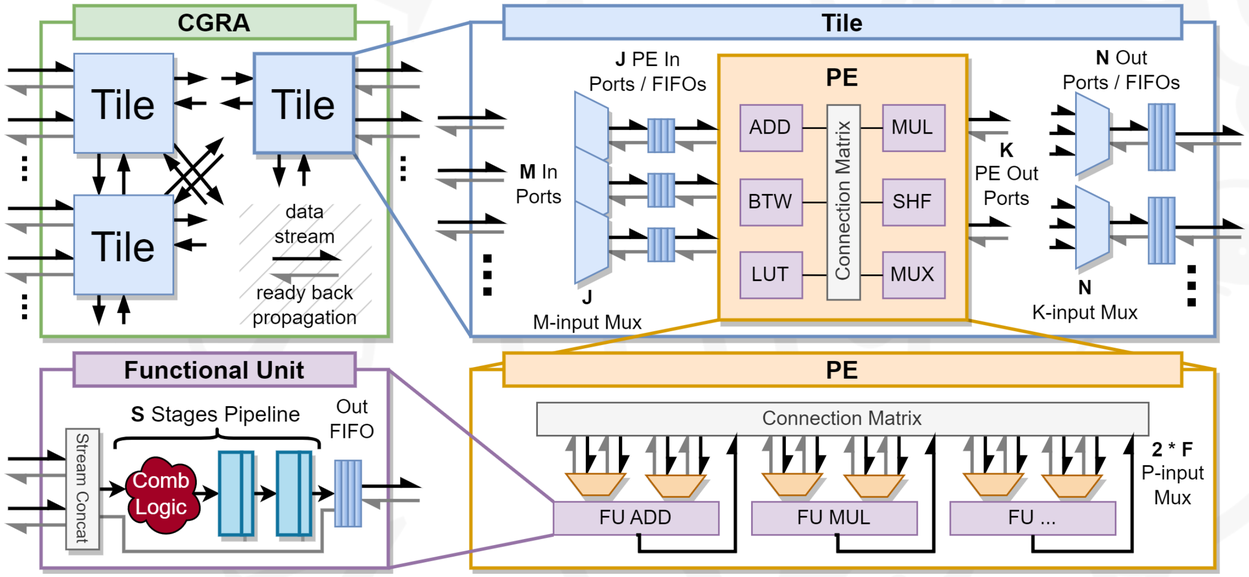
OSIP I-2021-03237 Innovative Coarse-Grained Reconfigurable Array Platform for Computing Artificial Intelligence On-Board
OSIP I-2022-04765 Risc-V Based SoC Featuring A Soft-GPU Hardware Accelerator for Artificial Intelligence On-Board
OSIP I-2023-09415 UDSM AI-engine For Reliable, Energy-efficient Next-generation Satellites

ESA OSIP - UDSM AI-Engine for Next-Gen. Satellites

- AI Engine IP
  - CGRA core
  - Simplified programming environment
    - CGRA configurations
    - Scheduling firmware
- What we have
  - CGRA core
  - Software-controllable SmartDMAs
  - Tiny RISC-V with DMA extension for data management
- What we need to carry out
  - Reliability analysis
  - SoC integration in UDSM 7nm
  - Programming environment
  - Benchmarking with multiple CNNs



HW Development – CGRA Architecture

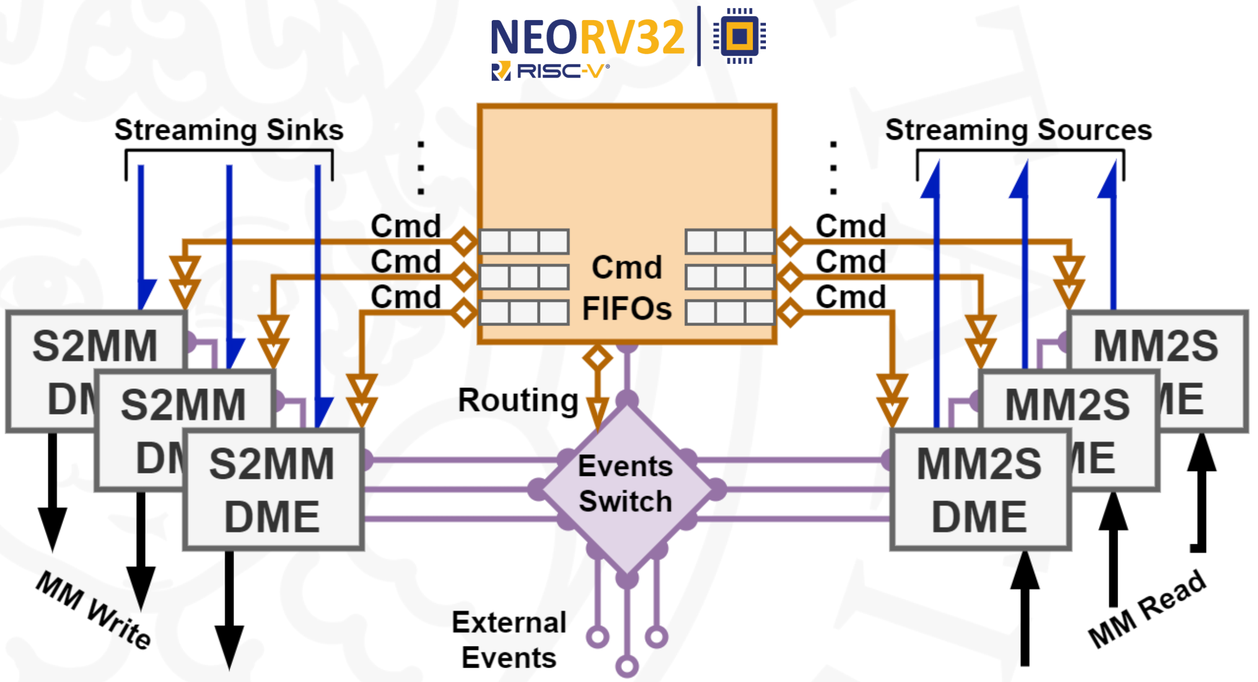


- Four hierarchy levels:
  - Functional Units: *ADD*, *MUL*, *SHF*, *BTW*, *LUT*, *MUX*
  - Processing Elements
  - Tiles
  - CGRA
- High parametrization and scalability
  - Number of rows and columns
  - Implemented FUs and connection matrix
  - FIFOs depth and pipeline stages
- Design optimized for ML execution
  - MACC operations
  - Non-linear activation functions

L. Zulberti, M. Monopoli, P. Nannipieri and L. Fanucci, "Architectural Implications for Inference of Graph Neural Networks on CGRA-based Accelerators", doi: 10.1109/PRIME55000.2022.9816810.
L. Zulberti, M. Monopoli, P. Nannipieri, L. Fanucci and S. Moranti, "Highly Parameterised CGRA Architecture for Design Space Exploration of Machine Learning Applications Onboard Satellites", doi: 10.23919/EDHPC59100.2023.10396632.
L. Zulberti, M. Monopoli, P. Nannipieri, S. Moranti, G. Thys and L. Fanucci, "Efficient Coarse-Grained Reconfigurable Array architecture for machine learning applications in space using DARE65T library platform", doi: 10.1016/j.micpro.2025.105142

HW Development – SmartDMA Architecture

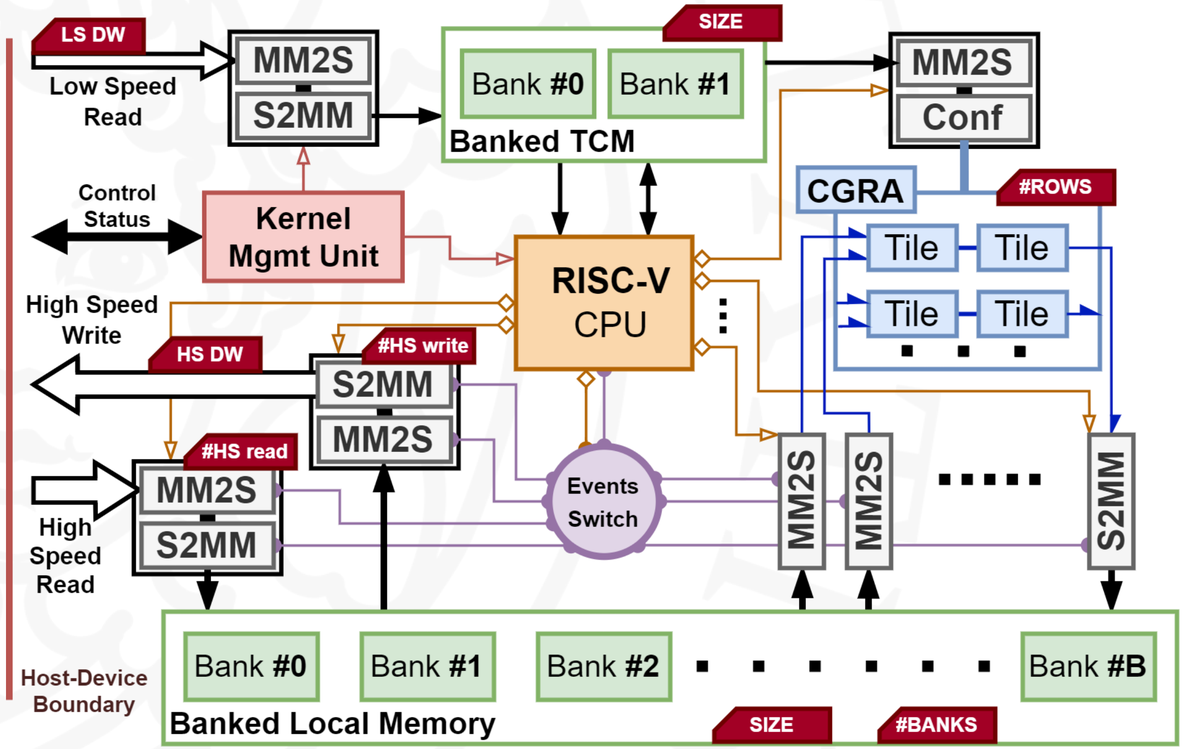
- RISC-V CPU
  - NeoRV32 CPU
  - *Xdma* custom extension
- Data Mover Engines
  - Support for 2D/3D data movement
  - Unaligned access
- Event Switch
  - Configurable DME dependencies
  - Data is loaded to the CGRA after transfer to local memory is finished



Nolting, S., & All the Awesome Contributors. (2024). The NEORV32 RISC-V Processor (v1.10.2). Zenodo. doi: 10.5281/zenodo.13127811.
L. Zulberti, A. Monorchio, M. Monopoli, G. Mystkowska, P. Nannipieri and L. Fanucci, "SmartDMA: Adaptable Memory Access Controller for CGRA-based Processing Systems", doi: 10.1109/DSD64264.2024.00048.

HW Development – CGR-AI Engine

- Scalable and parametrizable architecture:
  - CGRA size (rows, columns)
  - Size of banked local memories (for data and firmware)
  - Number of reading and writing DMEs
- External host loads firmware and kernel parameters
- RISC-V loads CGRA configuration from banked TCM
- RISC-V executes only data movement operations to instruct DMEs
- Event Switch module synchronizes high-speed DMEs communicating with the host with CGRA low-speed DMEs



Ongoing Work – Reliability Enhancement Approach

- Strict reliability requirements for space applications, especially when implementing on non-radiation-hardened technology
- Limited area and power consumption budget
- Evaluation of possible fault detection and mitigation strategies (TMR, EDAC)
  - Trade-offs between PPA metrics and robustness
  - Different level of granularity to consider (from register level to engine level)
- Tabular-based classification approach already developed by University of Pisa as result of OSIP activity

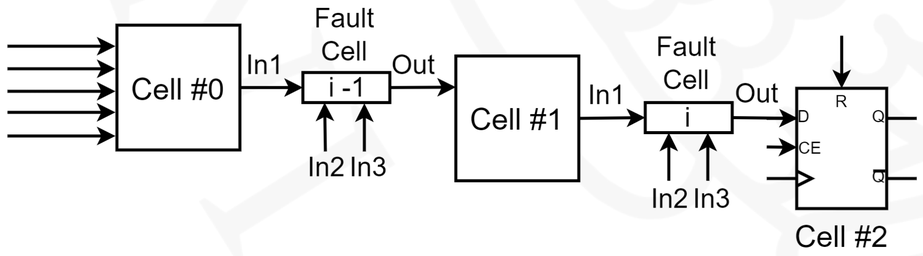
PA \ Criticality	Class 1	Class 2	Class 3	Class 4
Class 1	C2	C3	C4	C4
Class 2	C2	C3	C3	C4
Class 3	C1	C2	C3	C3
Class 4	C1	C1	C2	C2

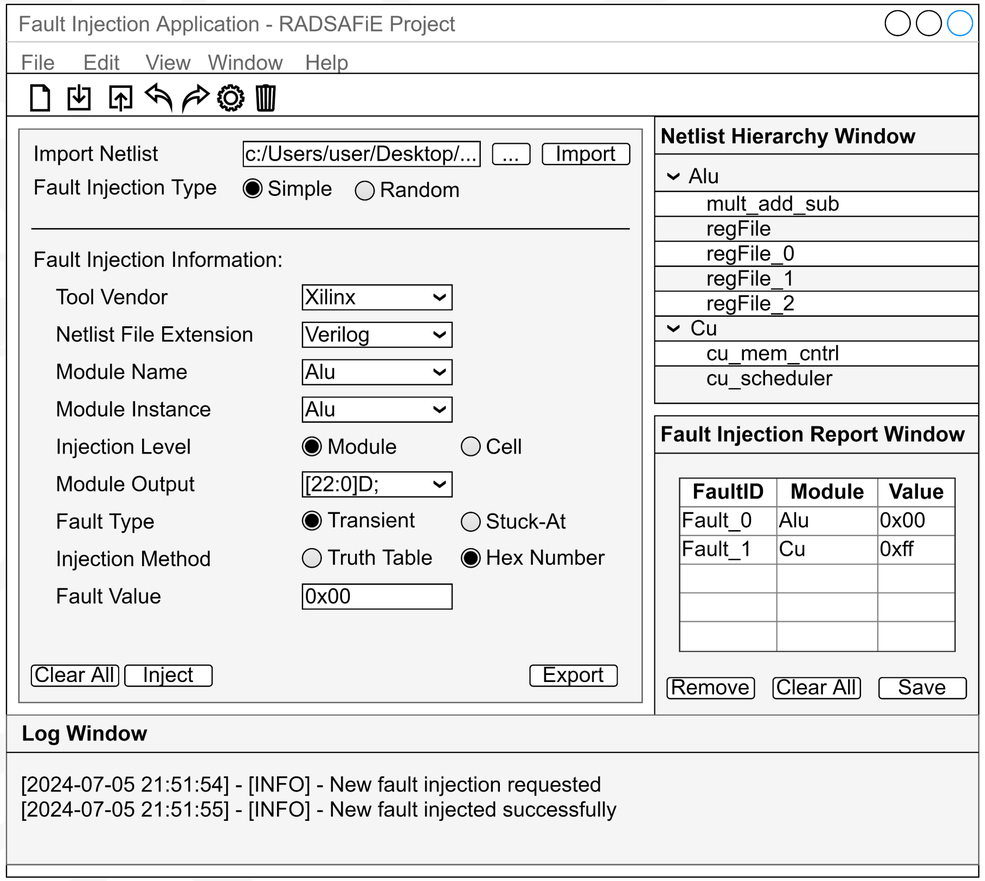
Class	Operational Element	Memory
C1	Partial Duplication w/ Self Check	Parity Bit
C2	Partial Distributed TMR	DED Hamming Code
C3	Block TMR w/ 1 Voter	SEC-DED Hamming Code
C4	Block TMR w/ 3 Voters	TMR

M. Monopoli, M. Biondi, P. Nannipieri, S. Moranti, C. Bernardeschi and L. Fanucci, "Enhancing a Soft GPU IP Reliability Against SEUs in Space: Modelling Approach and Criticality Analysis on a Radiation-Tolerant FPGA", doi: 10.36227/techrxiv.173273657.76155928/v1

Ongoing Work – Reliability Validation Approach

- Use of GUI fault injection application developed internally
- Netlist-based approach to emulate different types of fault on the FPGA implemented netlist
  - Stuck-at
  - Transient
- Addition of LUT-based fault injection cells
- Integration in a SystemVerilog simulation environment





M. Monopoli, M. Biondi, P. Nannipieri, S. Moranti and L. Fanucci, "RADSAFiE: A Netlist-Level Fault Injection User Interface Application for FPGA-Based Digital Systems", doi: 10.1109/ACCESS.2025.3539932

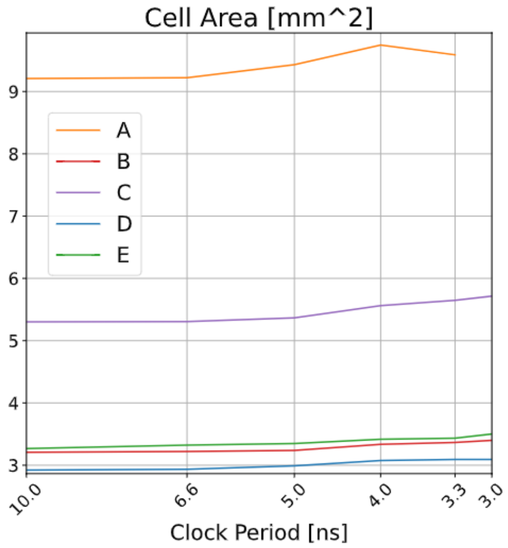
Outline

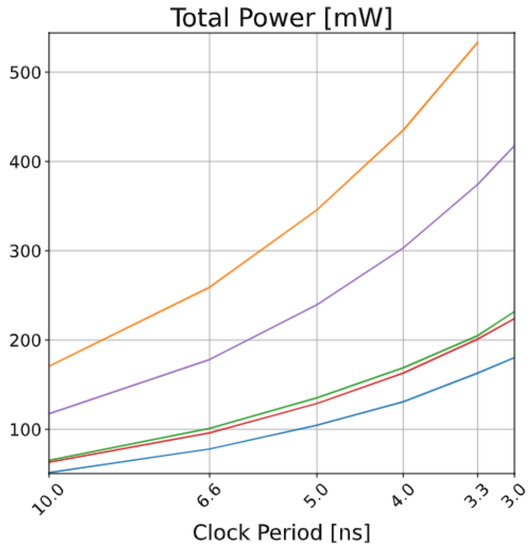
1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

CGR-AI on DARE65T RHBD Library Platform

- Optimized for std. cell implementation
- Synthesized on the 65nm RHBD DARE library platform
- Performed DSE activity with different:
  - Data width
  - CGRA rows
  - Local memory size
  - High-speed ports
- Tested against classic CNN kernels (e.g., conv2D, ReLU)
- Collected area and power consumption data across increasing clock frequencies
  - Great scalability
  - Less than 0.5 W @ 300 MHz on average

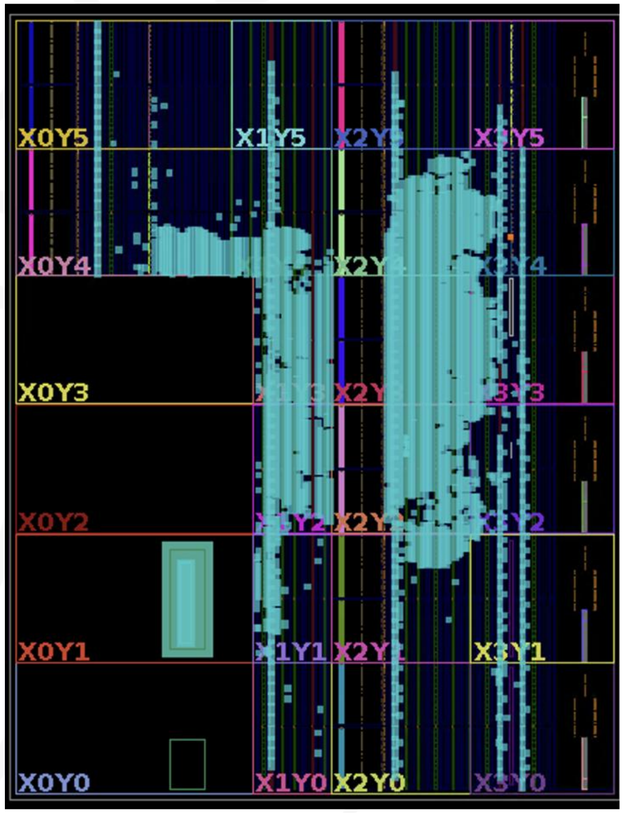
Config	Element Size	Vector Size	Rows	Memory (kB)	HS Ports
A	8-bit	32-bit	8	512	2R/2W
B	8-bit	64-bit	3	64	1R/1W
C	8-bit	64-bit	6	256	2R/2W
D	8-bit	128-bit	1	64	1R/1W
E	16-bit	64-bit	3	64	1R/1W





CGR-AI on Xilinx Zynq Ultrascale+ MPSoC

- System functionality confirmed in simulation
- FPGA overlay as intermediate milestone to demonstrate the readiness of the system
- Scale up directly based on the numbers obtained through std. cell synthesis
- Xilinx Zynq Ultrascale+ ZCU104 MPSoC
  - PS ARM core as external host
  - 600 MHz maximum operating frequency for the CGR-AI Engine
  - Contained resource utilization and power consumption, can still be optimized for FPGA implementation
- IP approach used for ease of portability



3x2 CGRA with 9MB of local memory

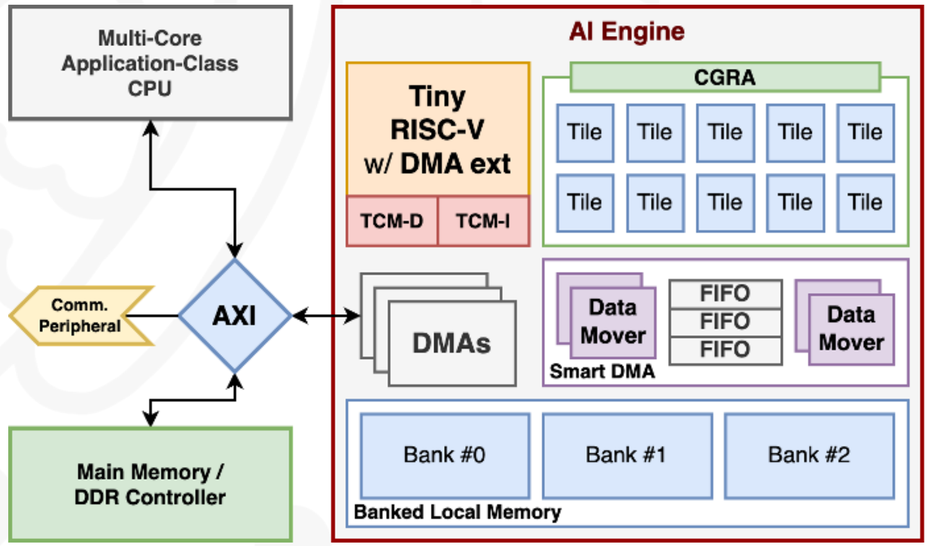
Outline

1. Background
   a. Motivation
   b. Hardware for AI Onboard
2. State-of-the-Art
3. CGRA-based HW Accelerator
   a. ESA OSIP Context
   b. CGR-AI Engine
   c. Ongoing work
4. Std. Cell Synthesis & FPGA Prototype
5. Conclusion

A Parameterizable CGRA-based Engine for Machine Learning Acceleration On-board Satellites

Matteo Monopoli and Gabriela Mystkowska
matteo.monopoli@phd.unipi.it, gabriela.mystkowska@phd.unipi.it

- CGRA-core for calculation acceleration
  - Heterogeneous computing
  - Scalable and flexible
  - Low reconfiguration time
- Developed systematic approach to increase reliability and fault injection application to validate results
- CGR-AI Engine synthesized on DARE65T RHBD
- FPGA overlay as intermediate milestone to demonstrate the readiness of the system



This study received funding from the European Union - Next-GenerationEU - National Recovery and Resilience Plan (NRRP) – MISSION 4 COMPONENT 2, INVESTMENT N. 1.1, CALL PRIN 2022 PNRR D.D. 1409 14-09-2022 – (OPERAND - a recOnfigurable Platform & framEwork foR Ai infereNce on the eDge) CUP N.I53D23006070001, from the Italian Ministry of Education and Research (MUR) within the framework of the FoReLab project (Departments of Excellence), from the European Space Agency (ESA) within the framework of the Open Space Innovation Platform (OSIP) co-funded research under the ESA contract n. 4000144254 and from the Space It Up project funded by the Italian Space Agency (ASI) and the Ministry of University and Research (MUR) under contract n. 2024-5-E.0 - CUP n. I53D24000060005.