

Fast SEU Detection and Recovery in FPGA-Based AI Accelerators

Eleonora Vacca, Giorgio Cora, Corrado De Sio, Luca Sterpone
Politecnico di Torino, Italy



Politecnico
di Torino



AI in Space - A Cool but Overlooked Challenge



- AI is rapidly being integrated into space applications, bringing enhanced autonomy and decision-making capabilities.
-  **But What About Reliability?**
 - Reliability concerns in AI for space are often overlooked.
 - The focus remains on performance, while traditional approaches like TMR (Triple Modular Redundancy) persist without exploring new methodologies.
-  **The Rise of RISC-V and Accelerators**
 - AI accelerators are increasingly coupled with RISC-V cores.
 - The scientific community is rapidly adopting new ISA extensions and core implementations
-  **The Missing Piece: Reliability in Heterogeneous Computing**
 - The focus on performance is critical, but we must rethink traditional methods and develop novel reliability-driven approaches for AI in space.

Proposed Approach



- What do we want?
 - **Fast error detection** in AI Inference
 - **Fast system recovery**
 - **Minimal area overhead**
 - **Minimal execution overhead**
- How?

Proposed Approach



- What do we want?
 - **Fast error detection** in AI Inference
 - **Fast system recovery**
 - **Minimal area overhead**
 - **Minimal execution overhead**
- How?
 - A runtime self-test **SEU-induced error detection** mechanism on the AI accelerator

Proposed Approach

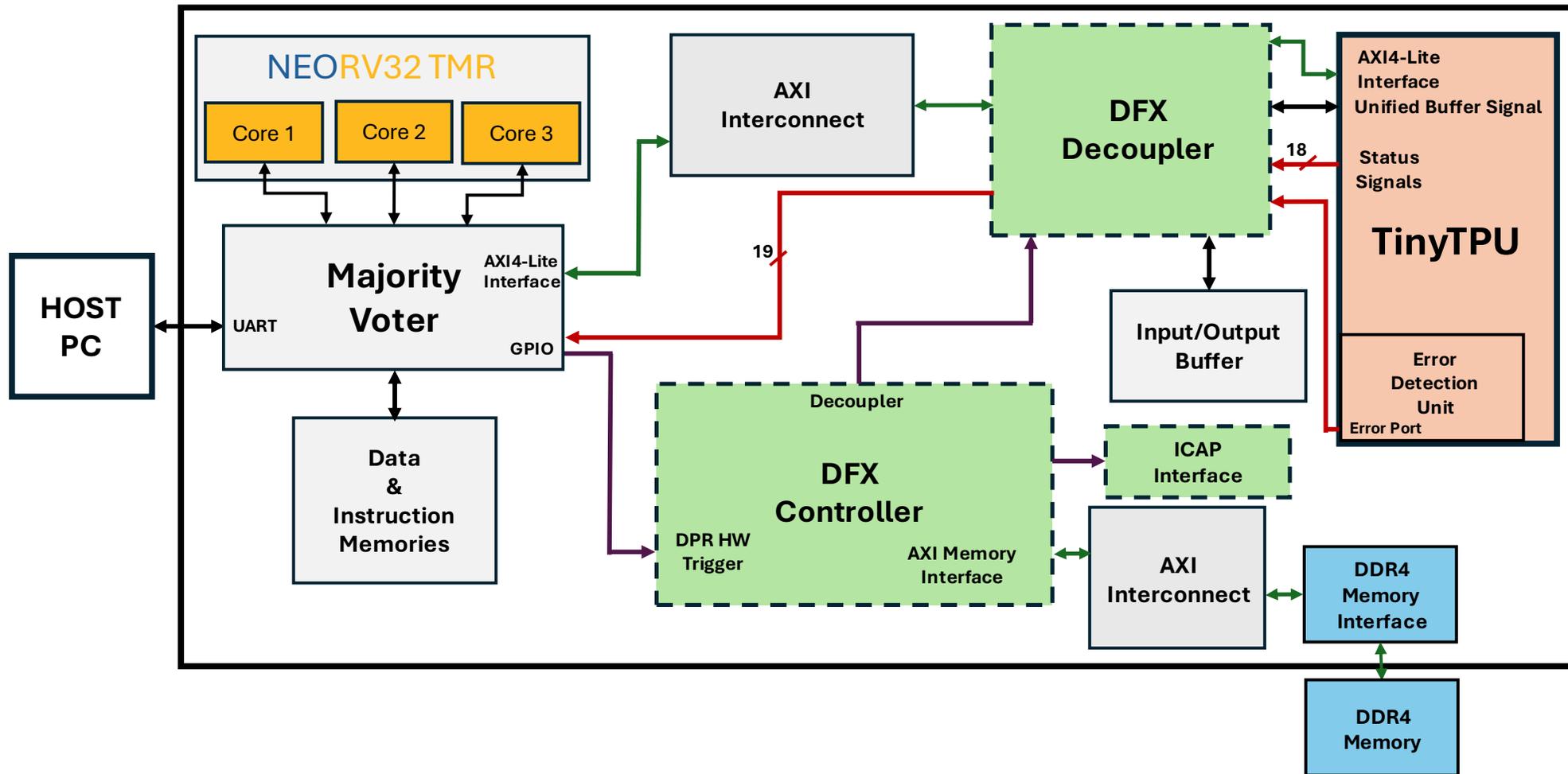


- Goals:
 - Fast error detection in AI Inference
 - Fast system recovery
 - Minimal area overhead
 - Minimal execution overhead
- How?
 - A runtime self-test **SEU-induced error detection** mechanism on the AI accelerator
 - **Dynamic partial reconfiguration to enable a fast and efficient error correction mechanism**

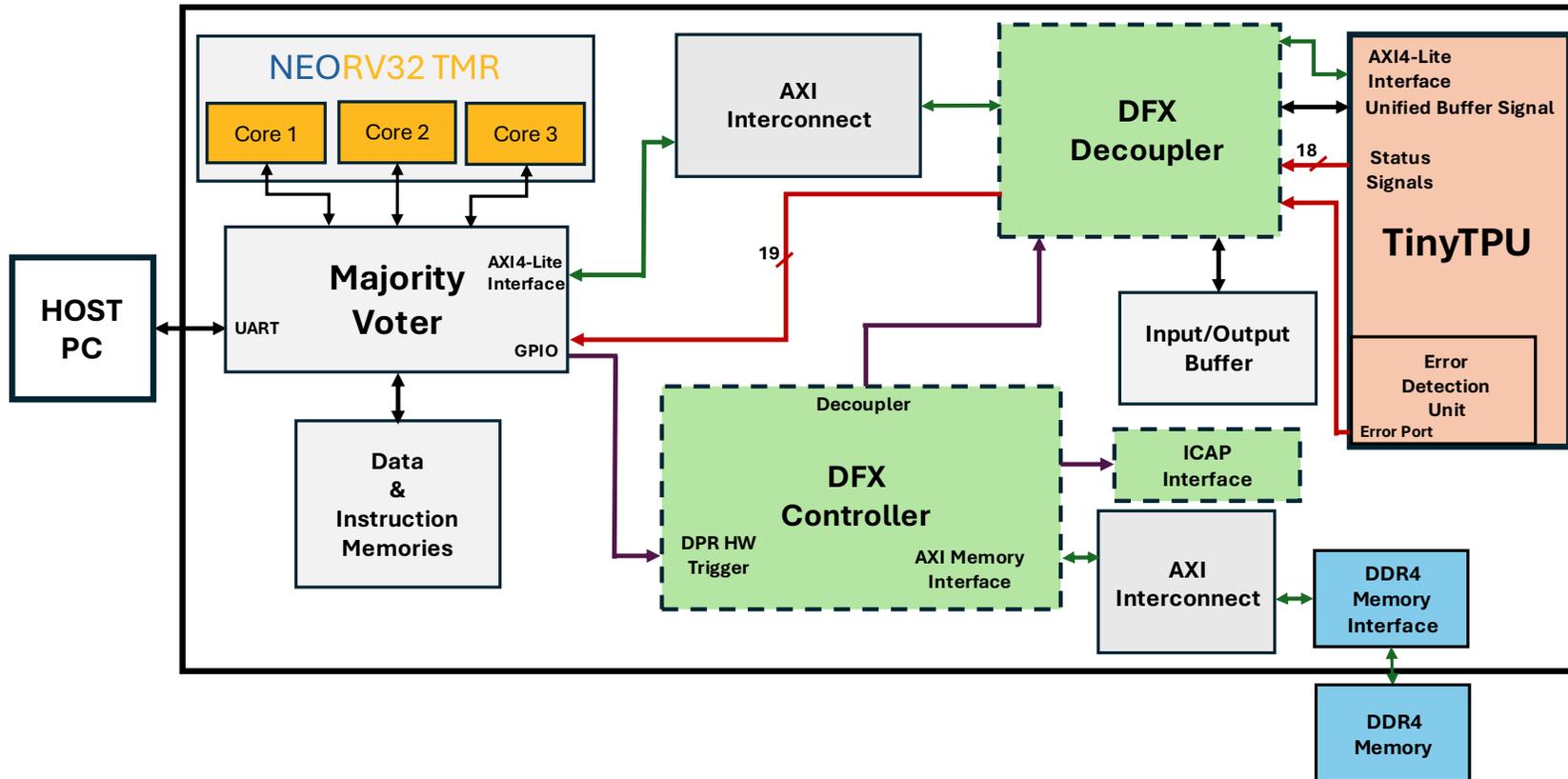
Proposed Approach



- Goals:
 - Fast error detection in AI Inference
 - Fast system recovery
 - Minimal area overhead
 - Minimal execution overhead
- How?
 - A runtime self-test **SEU-induced error detection** mechanism on the AI accelerator
 - **Dynamic partial reconfiguration to enable a fast and efficient error correction mechanism**
 - A RISC-V core monitoring the system
- The RePAIR (Reconfigurable Platform for AI Resilience within RISC-V Ecosystem) platform



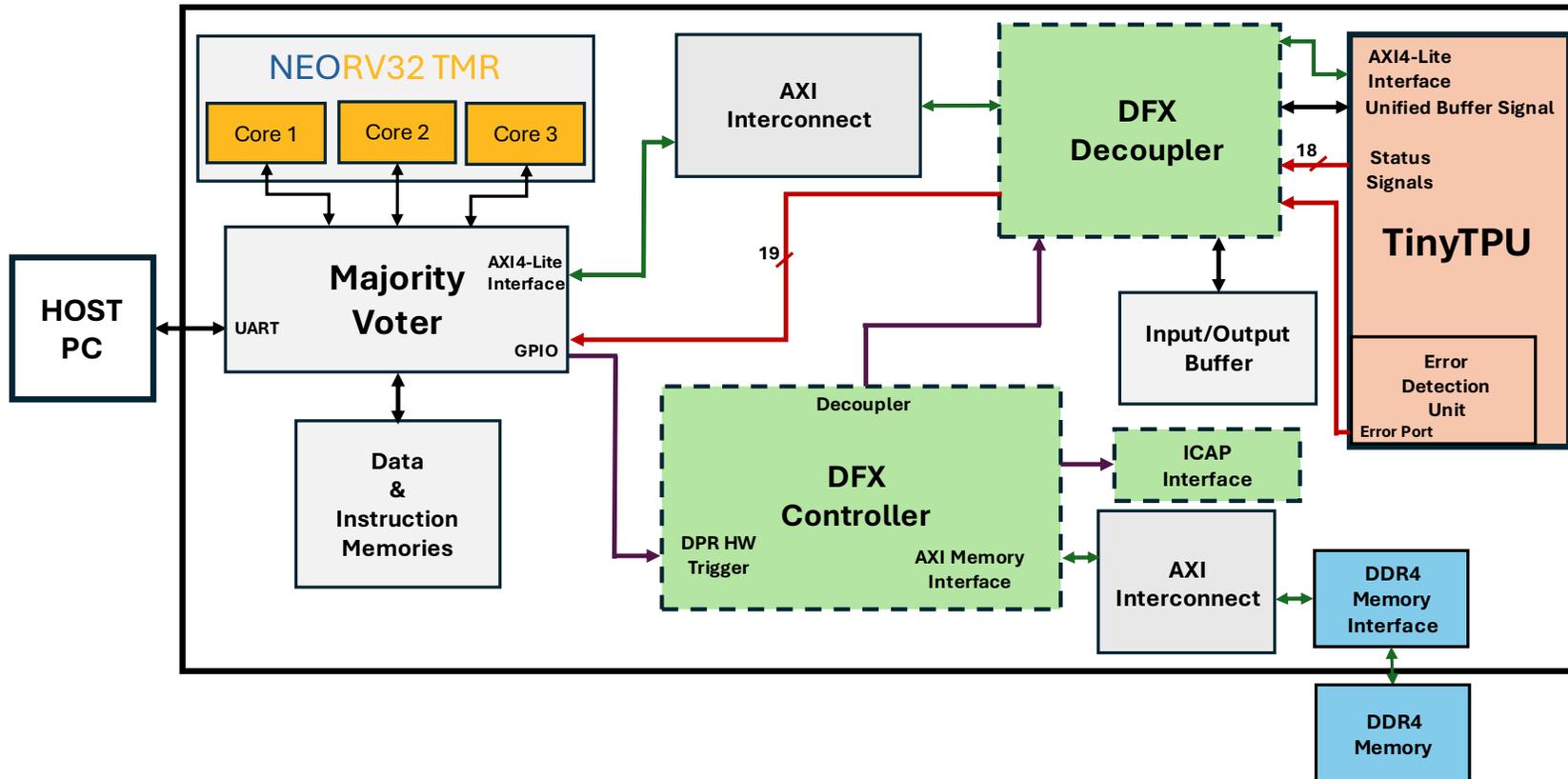
RePAIR – The RISC-V core



NEORV32:

- Tiny, Highly reconfigurable, and modular 32-bit VHDL-based architecture implementing RV32I ISA.
- AXI4-LITE Interface for communication with the TPU and DDR memory.
- UART Communication with the Host
- GPIO Interfaces for Error Detection and Partial Reconfiguration management.
- TMR for Improved Reliability.

RePAIR – The AI Accelerator

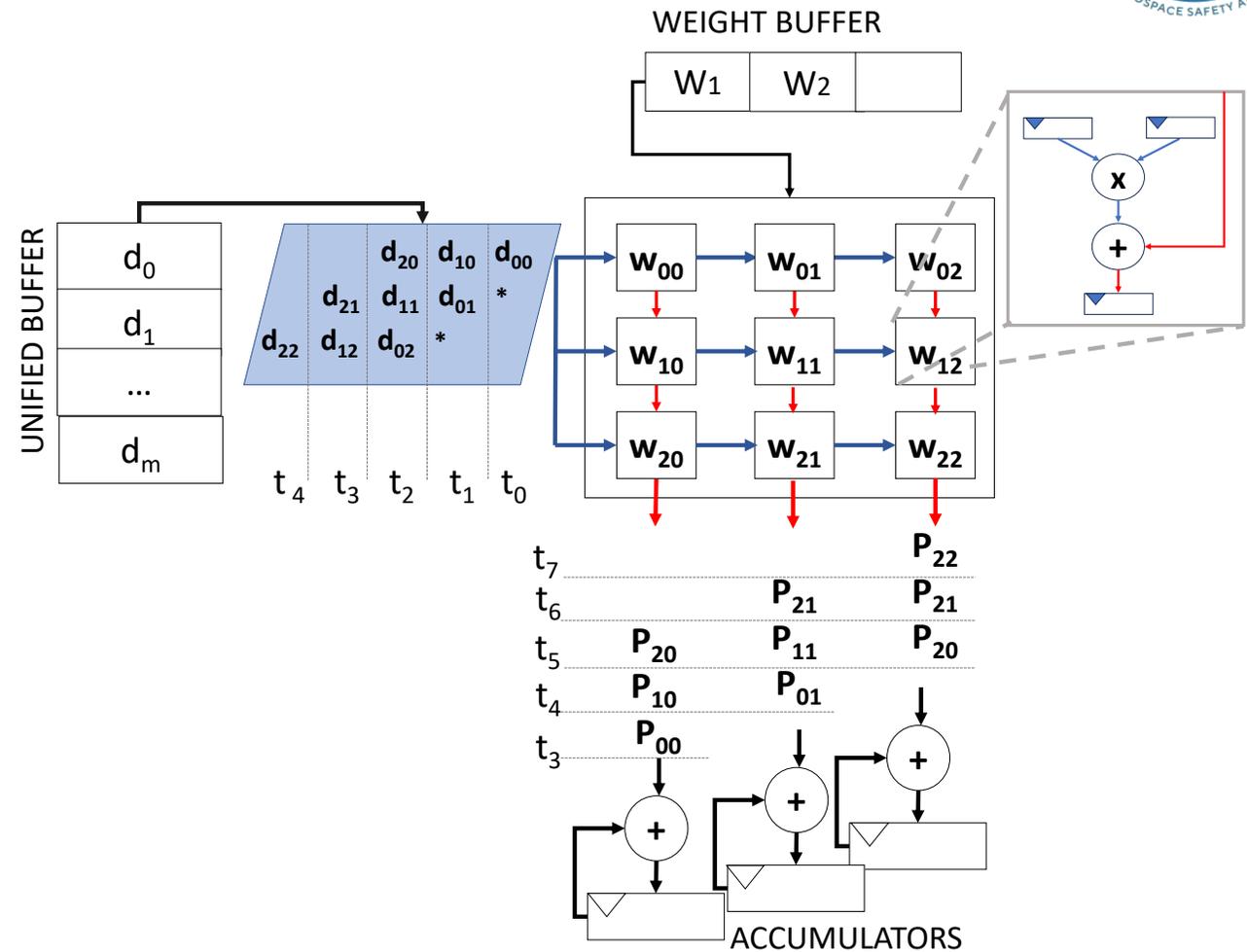


TinyTPU:

- Configurable Systolic Array size, from 6x6 to 14x14 MAC units.
- Custom 80-bits CISC ISA.
- Designed for DNN execution.
- Support for ReLU and Sigmoid activation.
- Custom ISA Extension to Support Error Detection capabilities.
- Minimal hardware and execution time overhead.

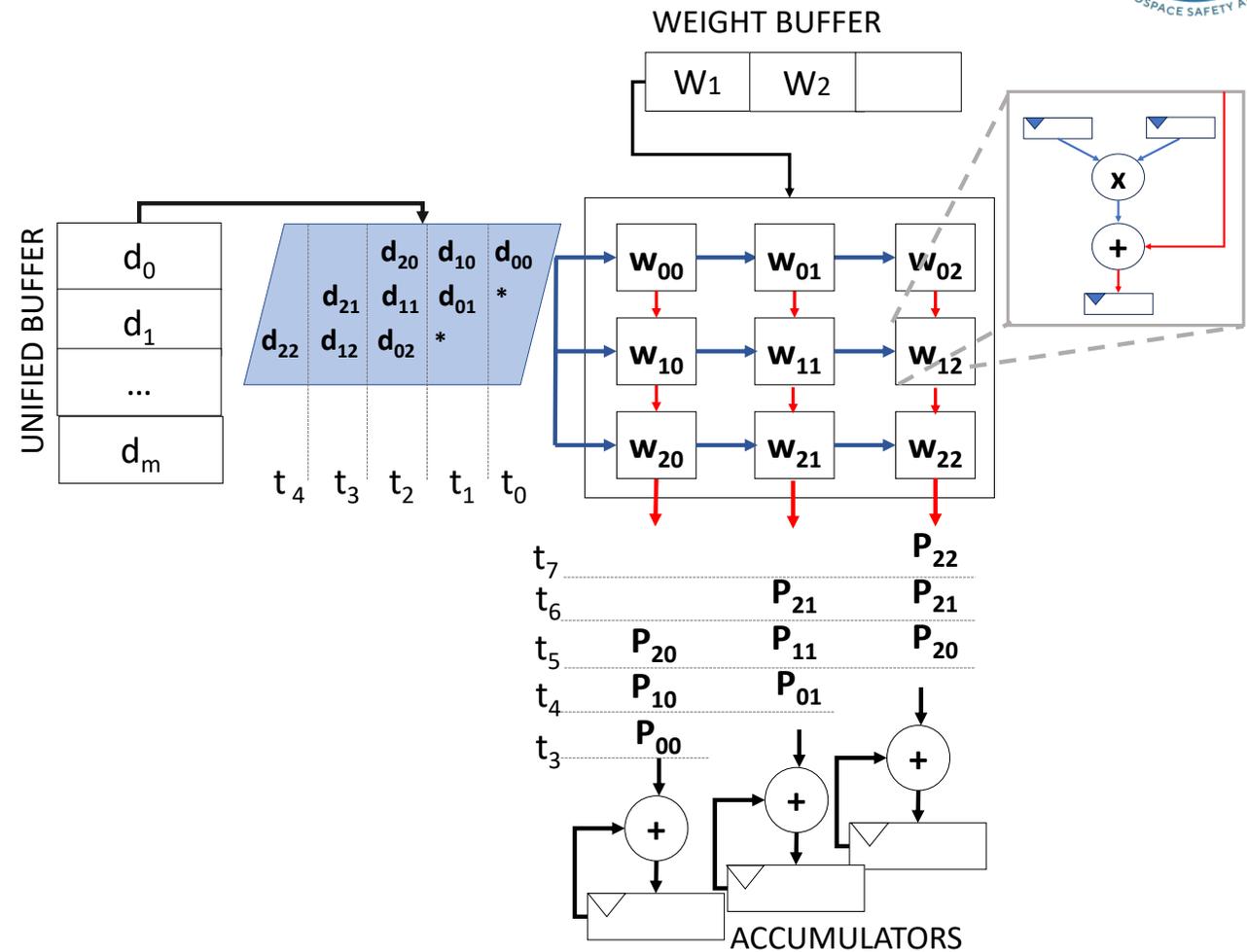
Systolic Arrays

- 2D Array of Processing Elements
- Fixed Interconnection path between PEs for fast data exchange and processing
- Neural Networks on SA are implemented as GEMM operations



Systolic Arrays – SOTA Fault Detection

- Algorithm Based Fault Tolerance
 - computing checksums on the matrices processed
 - **high area overhead**
 - $(2N + 1)$ adder for a SA $N \times N$
- Scan chain methods
 - Exploit the functional path between PEs to propagate test patterns
 - Requires modification on MAC units
 - Efficient in detection and diagnosis
 - **Not feasible for runtime execution during application workload**



Proposed Fault Detection

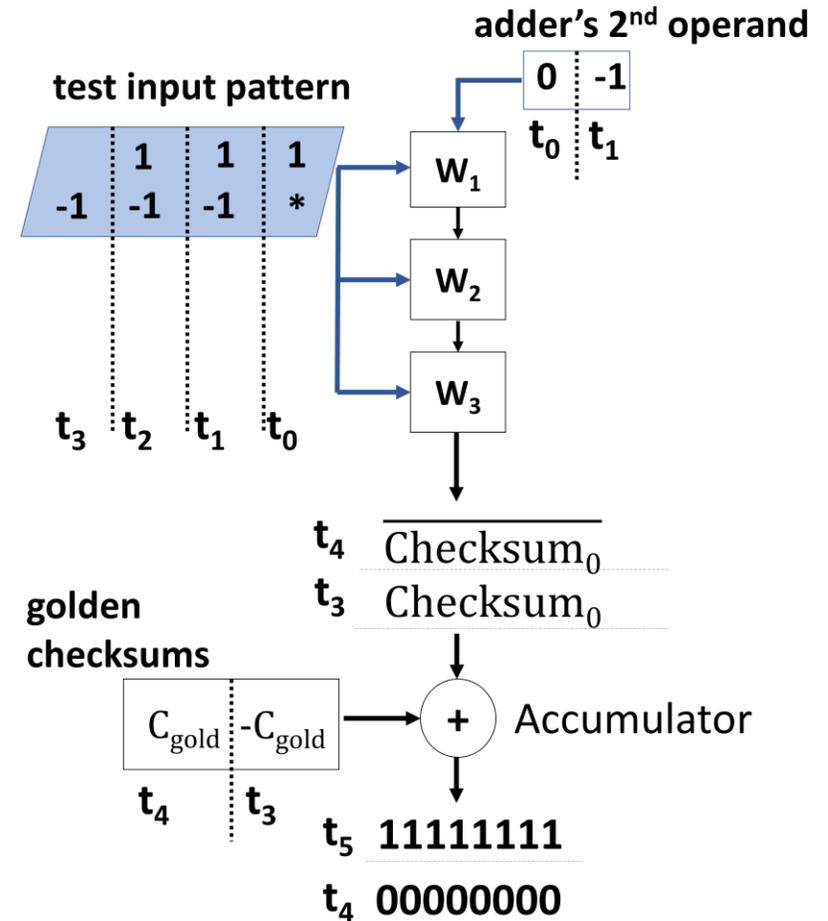


- Developing a novel runtime methodology for fault detection in Systolic Arrays named **RunSAFER**
- The method combines SCAN and ABFT with:
 - **Minimal hardware overhead**
 - **Reduced intrusiveness** on the application workload
 - **Fault detection during inference** execution
 - **Detection and diagnosis** of critical computational units of the Datapath:
 - Systolic Array core
 - Accumulators

E. Vacca et al., "RunSAFER: A Novel Runtime Fault Detection Approach for Systolic Array Accelerators," *2023 IEEE 41st International Conference on Computer Design (ICCD)*, Washington, DC, USA, 2023

Proposed Fault Detection

- The detection method consists of the following phase:
 - Exploit systolic core resources to compute checksums on the current workload data.
 - **Checksums' values** are computed in such a way that **complemented values** flow through all the Datapath resources
 - **Allowing for SEU-induced interconnection fault**
 - Diagnosis unit (XOR and OR gates) evaluates the checksums produced to detect faults



Proposed Fault Detection - Implementation

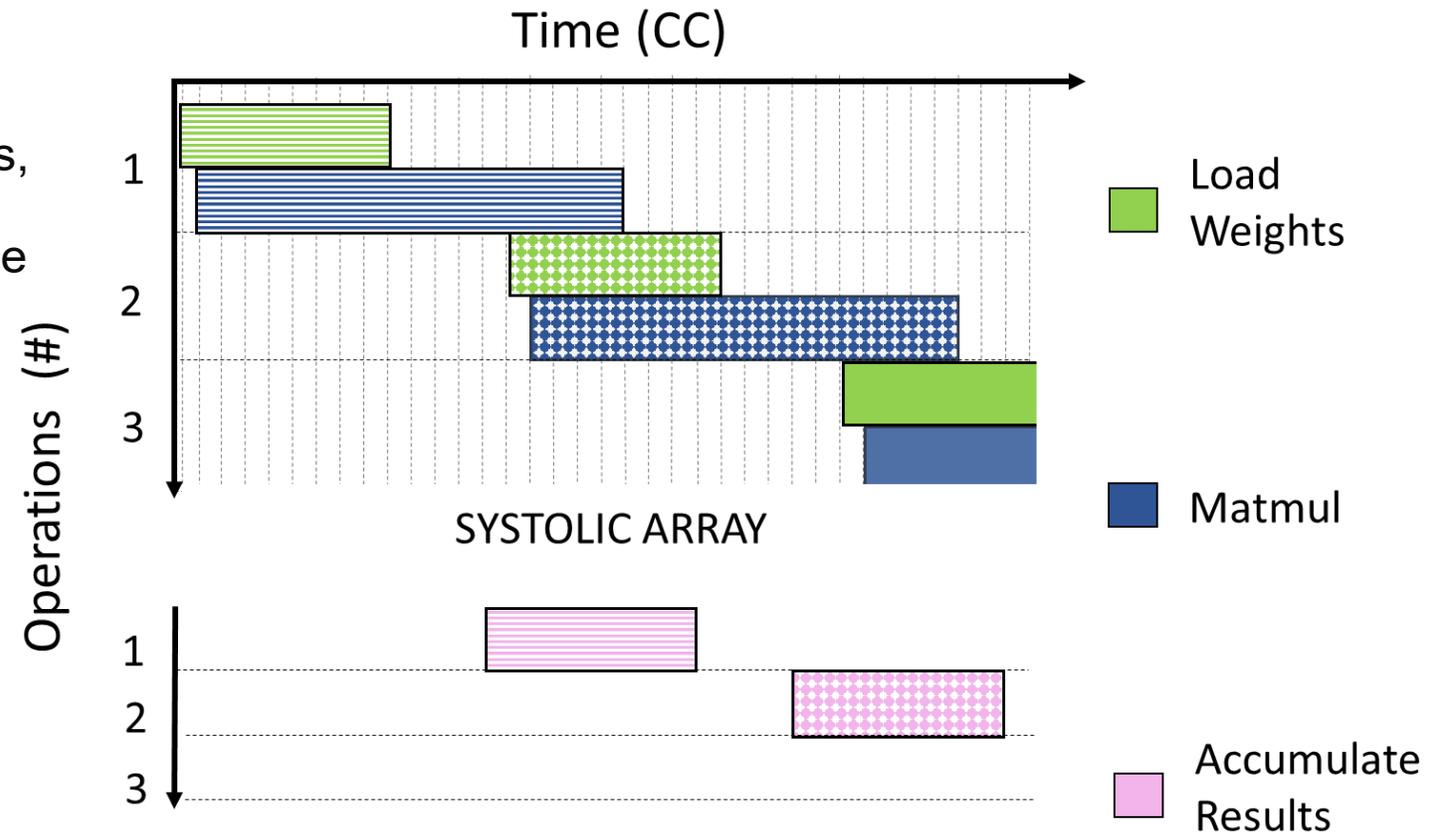


- The fault detection method has been integrated in the ISA of open-source TPU core
- The matmul instruction has been augmented to support the self-testing mode (*tmatmul*)
- Each tmatmul induces a penalty of 3 clock cycle
 - Due to additional processing of test vectors appended to the main computation
- Datapath modifications have been implemented **to introduce no hardware overhead** for the golden checksum computation
 - Use of the available Accumulators through the implementation of asymmetric SIMD

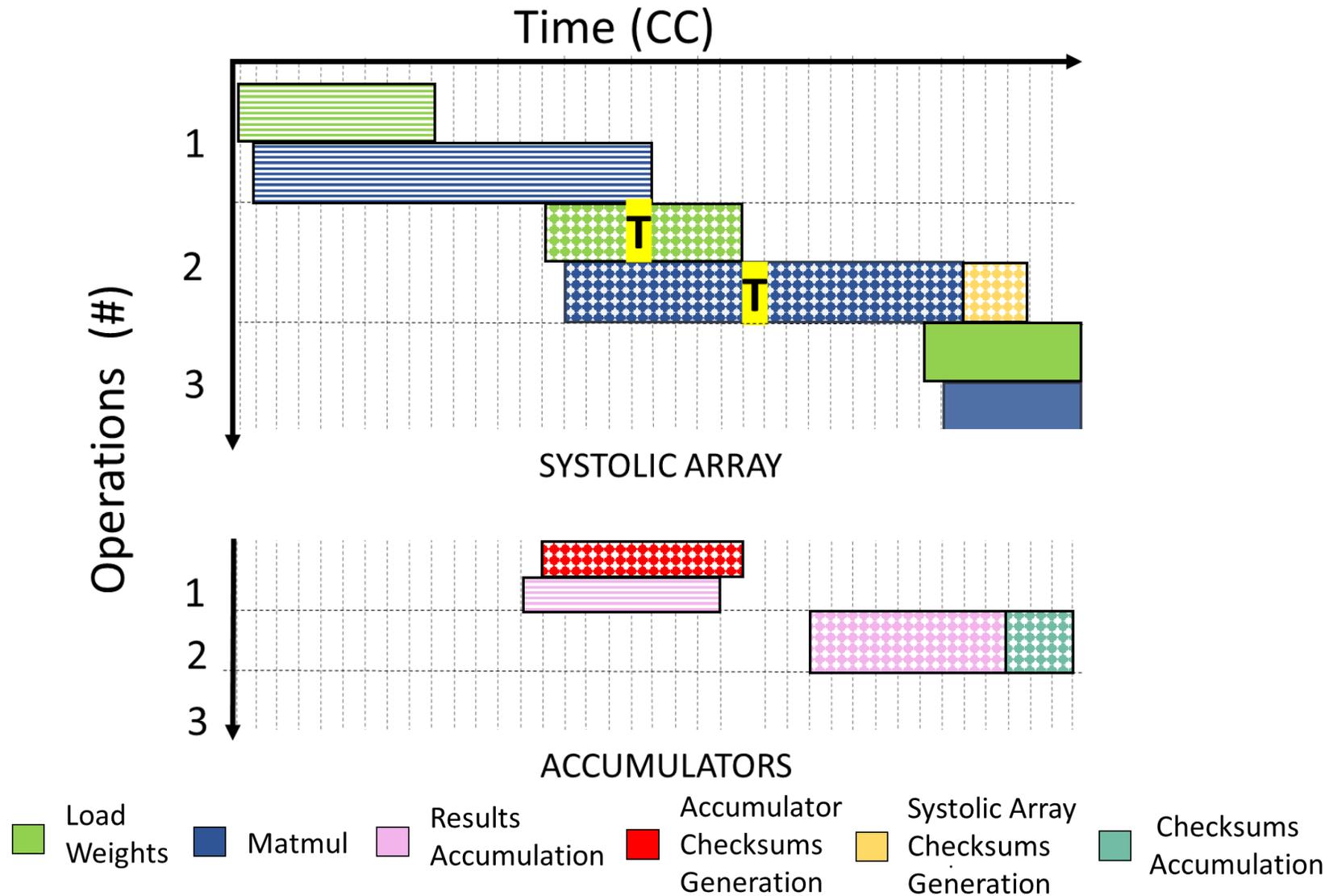
Original Pipeline



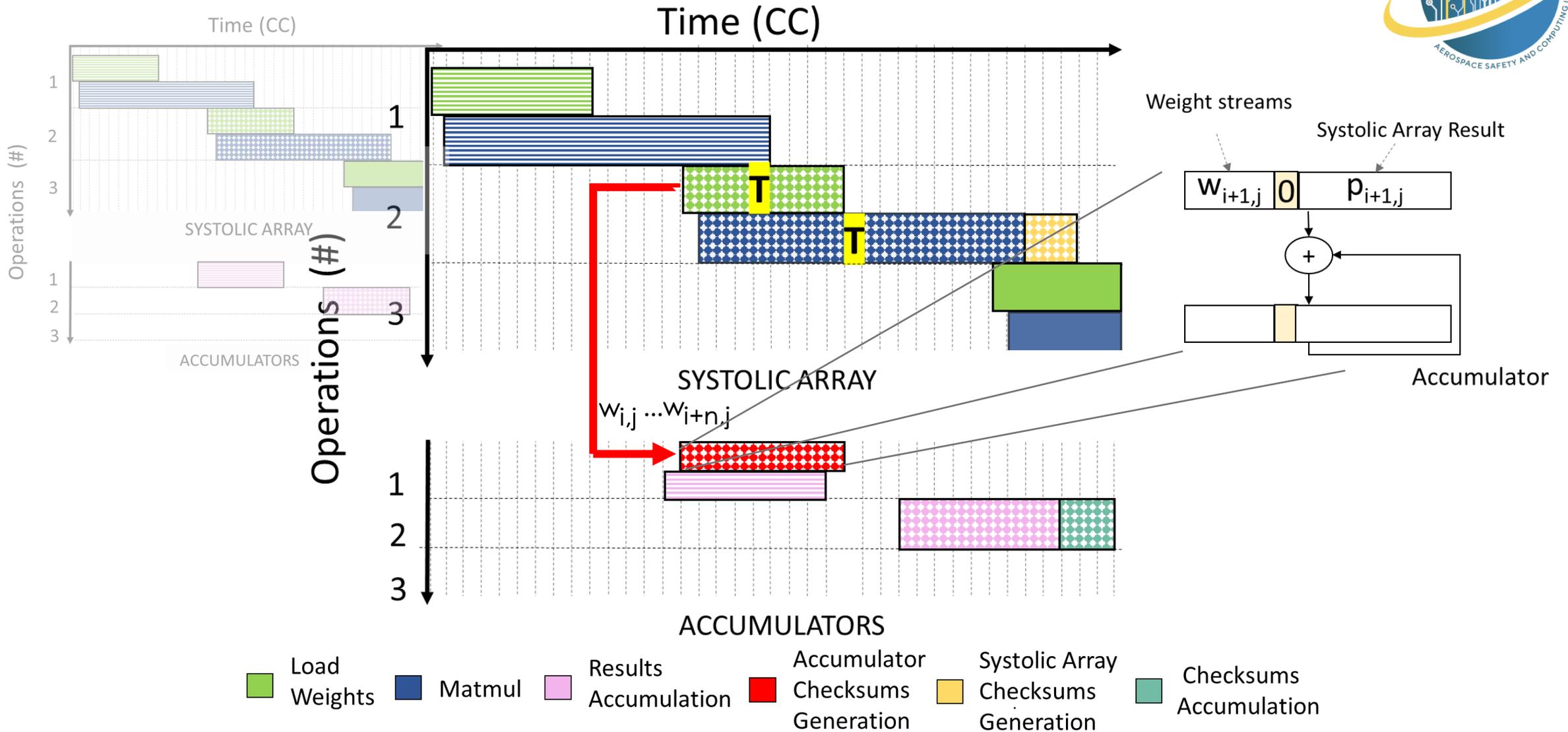
- Every matrix multiplication operation starts with a *load weights* instruction, followed by a *matmul* instruction.
- Once the results are generated, they are sequentially processed by the Accumulators, vector by vector. Meanwhile, a new set of load weights and matmul instructions can be sent to the Systolic core.



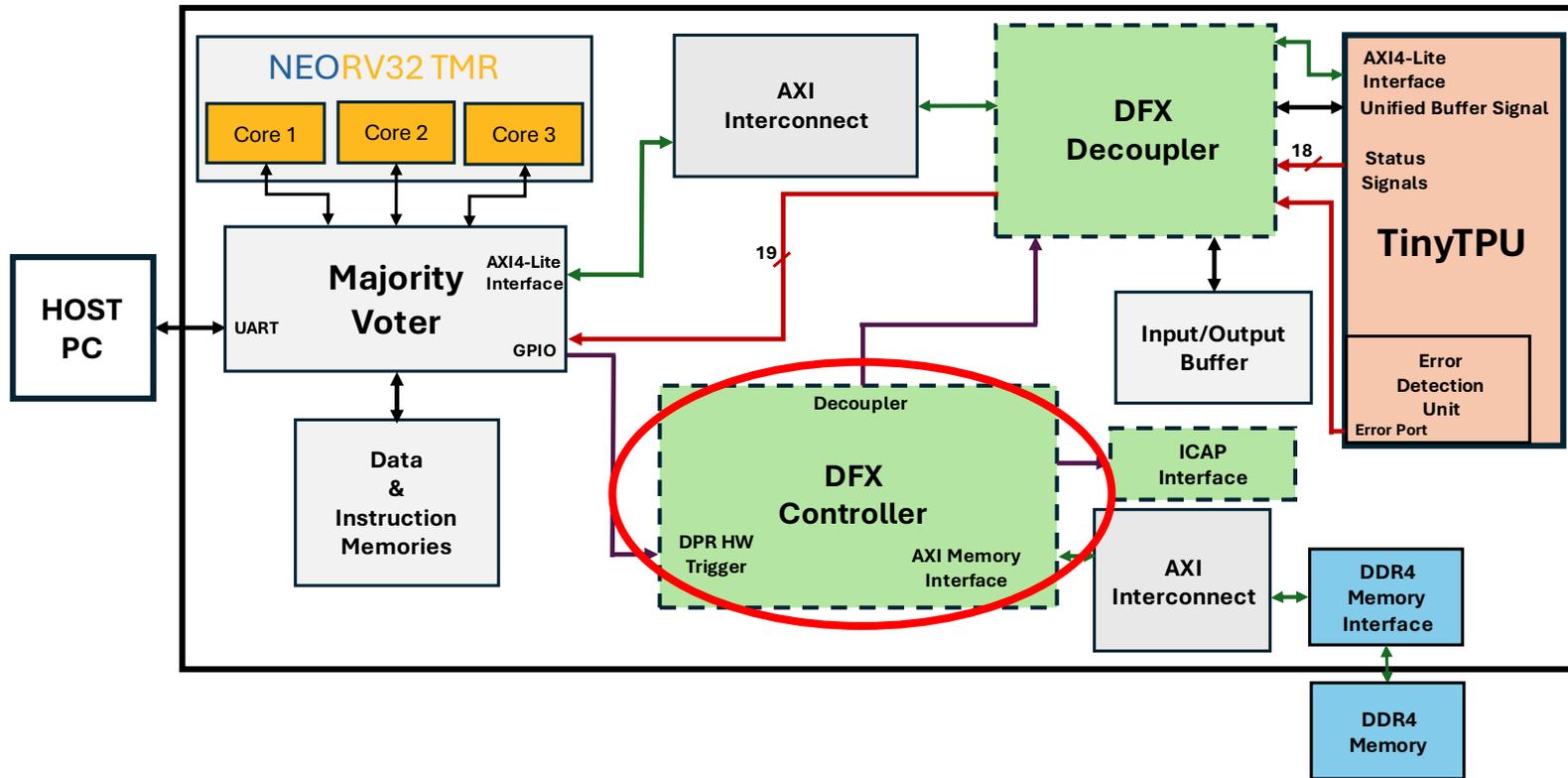
Modified Pipeline



Modified Pipeline



Partial Reconfiguration Support



Partial Reconfiguration:

- TPU raises an error signal mapped to RISC-V GPIO.
- RISC-V triggers the DFX Controller to perform DPR
- Recovery time in the range of tens of milliseconds, based on the TPU Size.
- Allows for execution resumption from the last correct state, saved in memory.
- Ensure minimal system downtime.



- **KCU105 Development Board**

Benchmarks CNNs:

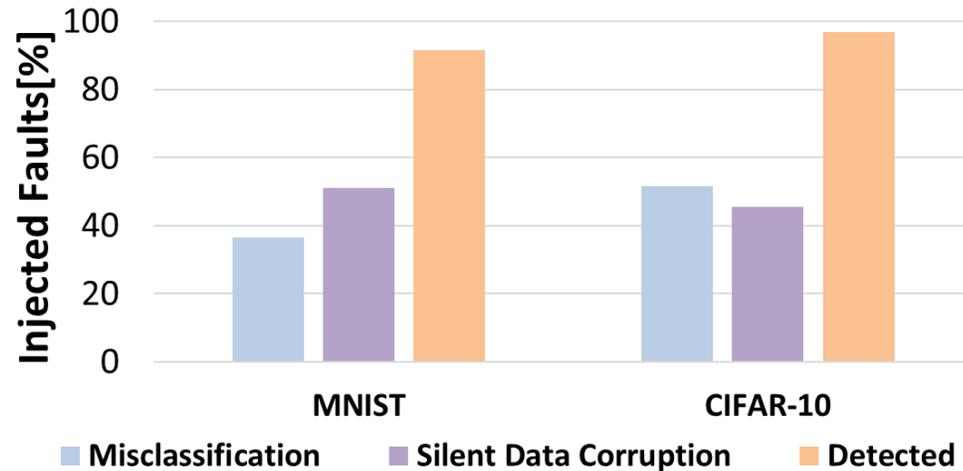
- CIFAR-10
- MNIST

Platform Modules	LUTs	FFs	BRAMs	DSPs
TinyTPU	4,294	7,211	181	210
TMR NEORV32	3,219	3,180	3	0
DPR Logic	1,185	989	0	0
Glue Logic Resources	13,874	17,670	95.5	3
Total [%]	9.31%	5.99%	46.58%	11.09%

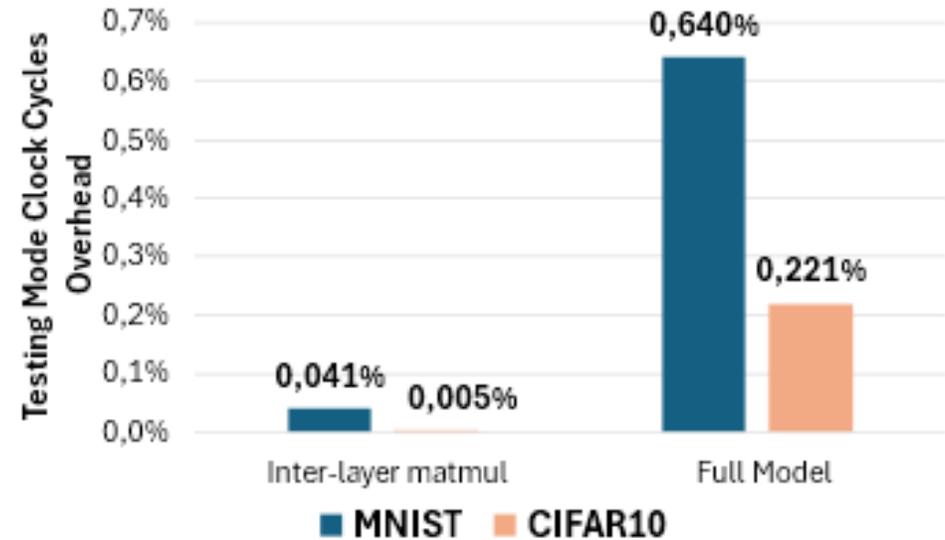
Experimental Results: Error Detection Mechanism Performances



- 5,000 SEU emulated through Fault Injection in CRAM, selectively targeting SA resources
 - For each fault both benchmarks are executed (to evaluate data masking effects)
- The detection mechanism provided 94% of detection
 - Detecting also faults masked by rounding and activation functions
- The overall time overhead is limited to a maximum of 0,64% clock cycles more in the worst-case scenario.
 - The resources overhead 0.31%.



Fault injection Results

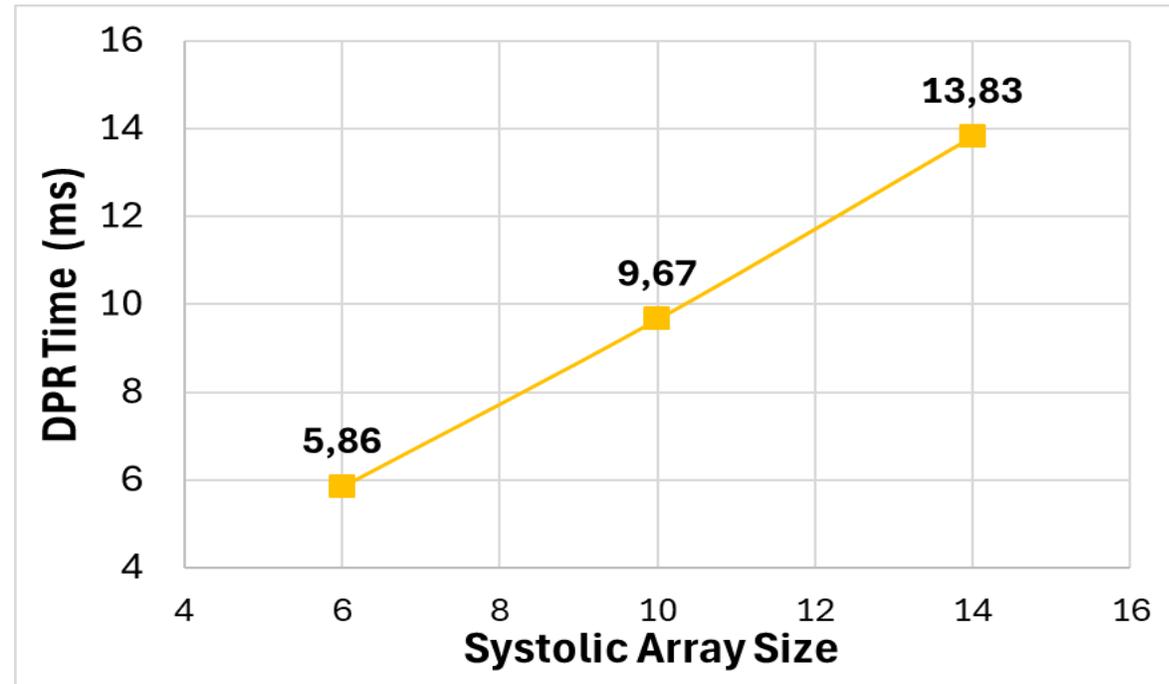


Error Detection Execution Time Overhead

Experimental Results: Recovery Time Overhead



- The DPR time scales linearly with the size of the PEs grid, from less than 6ms in the smallest case to around 14ms for the largest SA size.

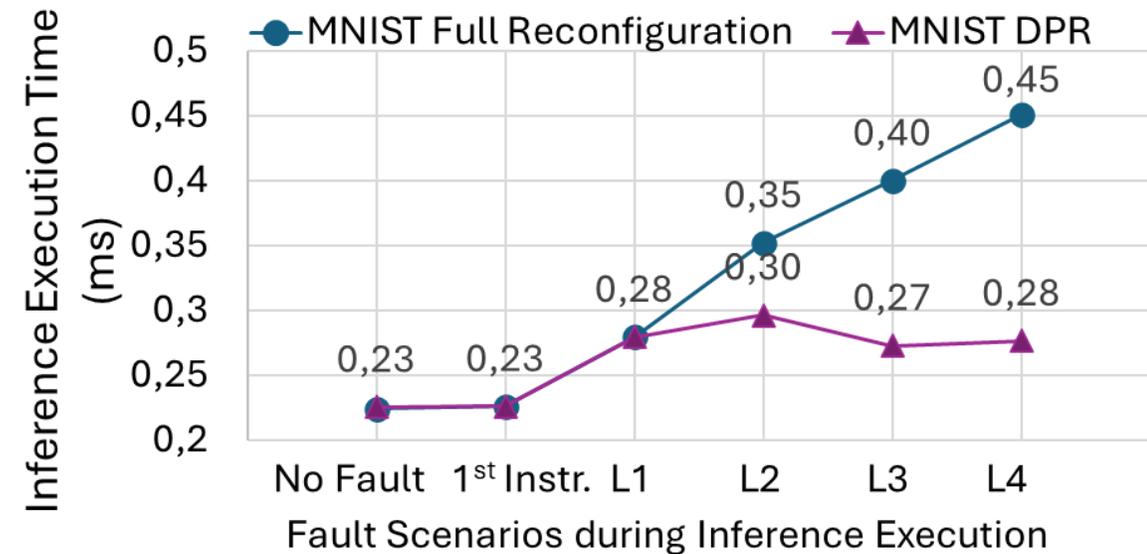
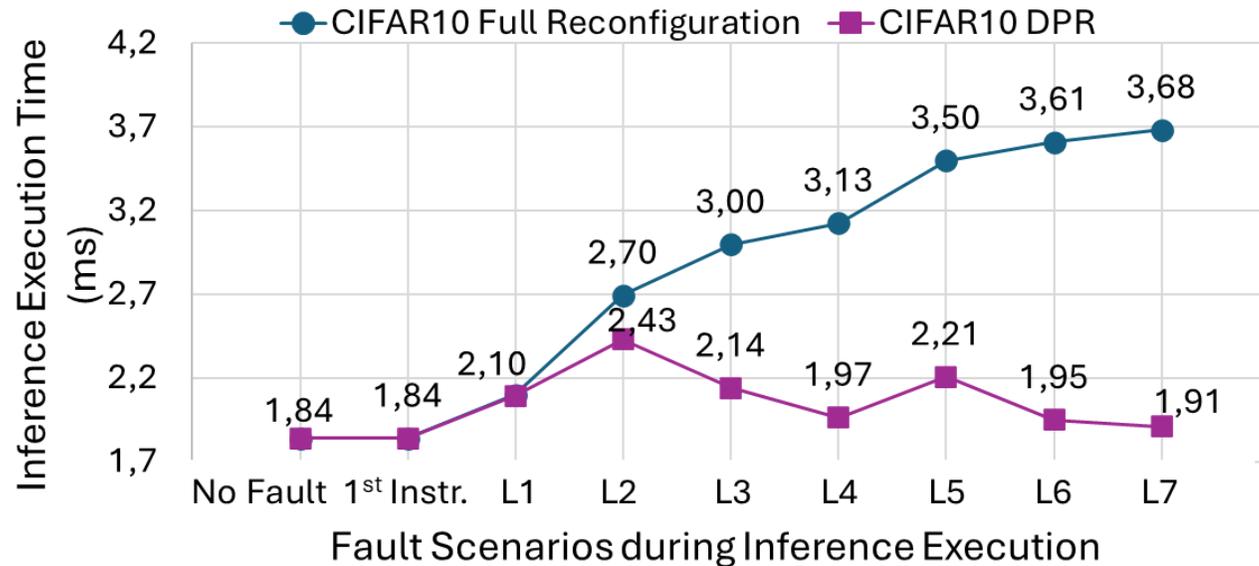


tinyTPU DPR Time

Experimental Results: Recovery Time Overhead



- The overall inference execution time is reduced in the DPR case, allowing operation recover from last correctly executed operation.



Conclusions



- A reliable platform for DNN execution in a safety-critical environment has been proposed.
- Error detection capabilities have been implemented into a Systolic Array.
- The Accelerator has been paired with the NEORV32 and Partial Reconfiguration to ensure error recovery and reduced system downtime.
- A fault injection campaign took place to validate the effectiveness of the proposed error detection mechanism.
- A detailed analysis of the efficiency of the proposed platform has been carried out.

Thank you for your attention!



Eleonora Vacca
Politecnico di Torino, Italy
Email: eleonora.vacca@polito.it
Link: <http://asaclab.polito.it/>

LinkedIn:

