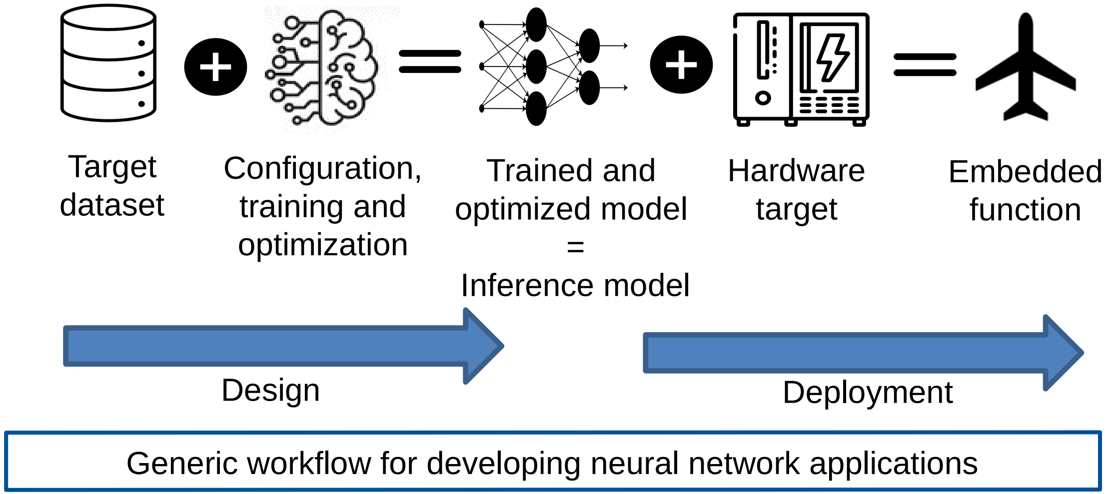
Certifiable and efficient implementation of machine learning algorithms on avionics systems

ANITI Days – November 17 2023

Iryna De Albuquerque Silva, Claire Pagetti, Thomas Carle, Adrien Gauffriau

Work scope

- Implementation of **off-line trained feed-forward deep neural networks** in avionics systems;



Certification requirements (subset of DO-178C):

- Ensure traceability (formal description of the function + semantics preservation);
- Compute tight WCET (restrictions on software and hardware)

Embedded targets:

- Attain good performance in single-core platforms

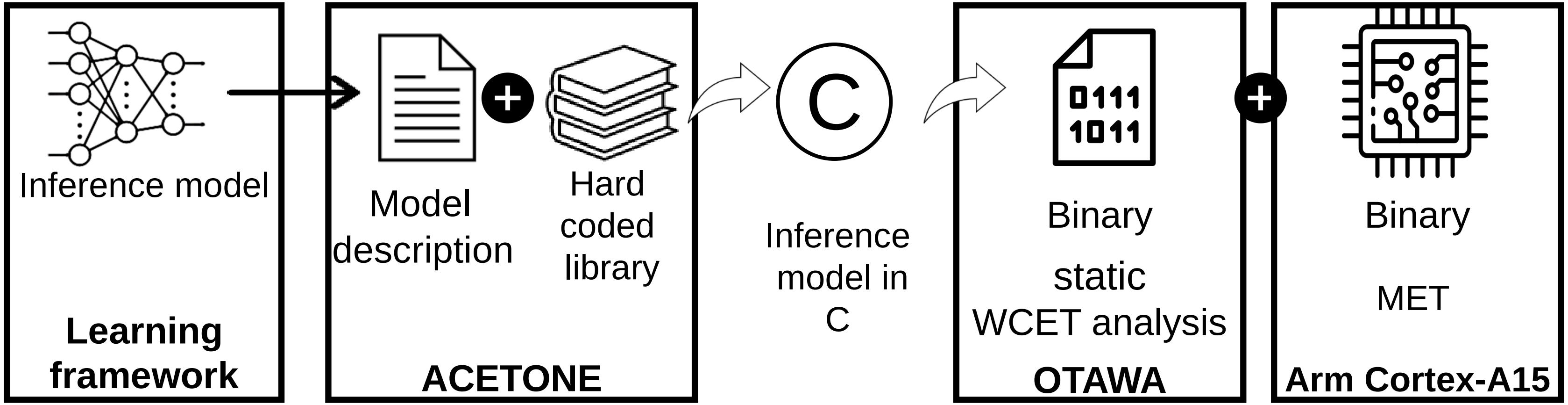
- Bridge the gap between ML and avionics domains

Contribution: Development of ACETONE

- ACETONE : Avionics C code generator for Neural NEtworks
  - Generated code: preserves the semantics and is predictable

Real use cases:
- ACAS-Xu
- Lenet
- Alexnet
- AudioNet

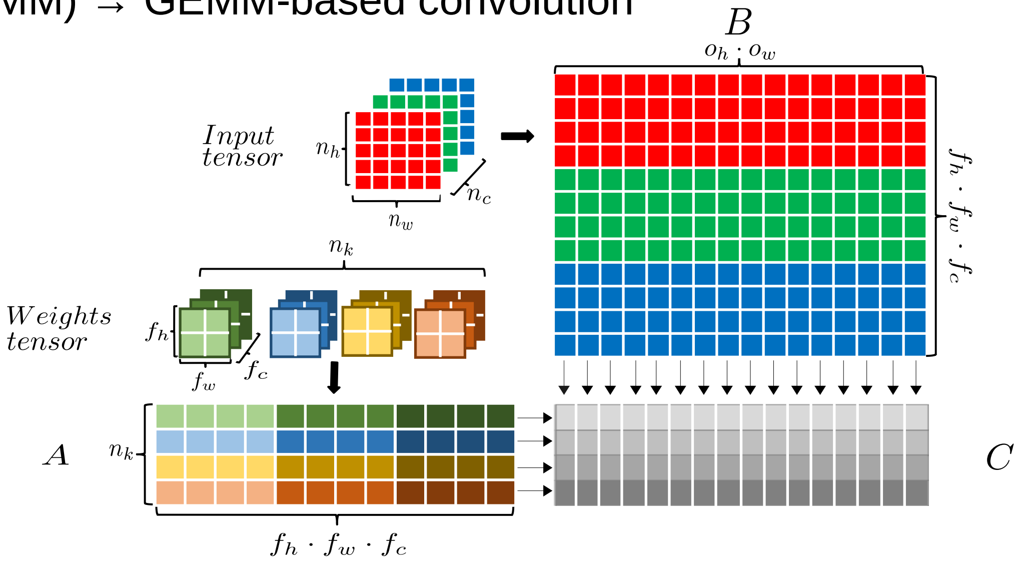
Evaluated w.r.t SOTA:
- Keras2C
- uTVM



  - Compatible with avionics requirements but convolutional layers were not really efficient...

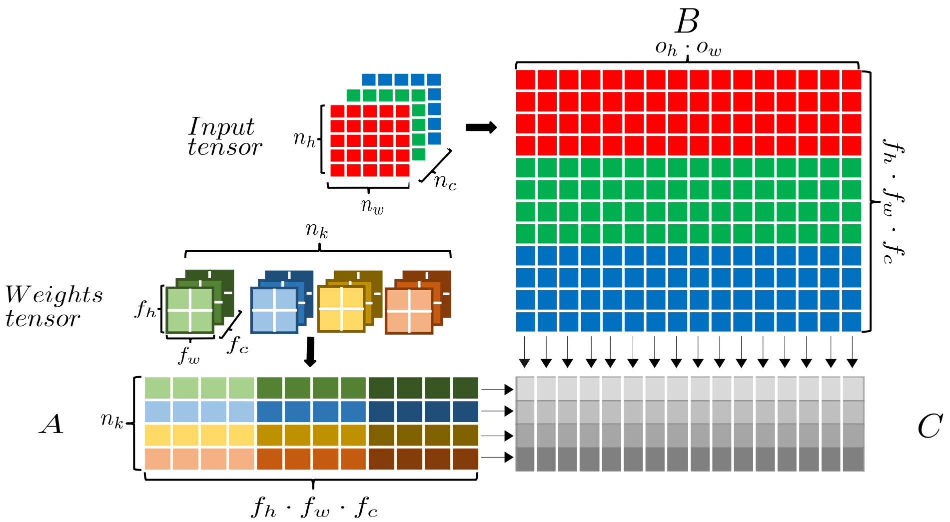
Improved implementation of convolutional layers

- **Idea:** reduce convolutional layers *execution time* by implementing it as a matrix multiplication (GEMM) → GEMM-based convolution

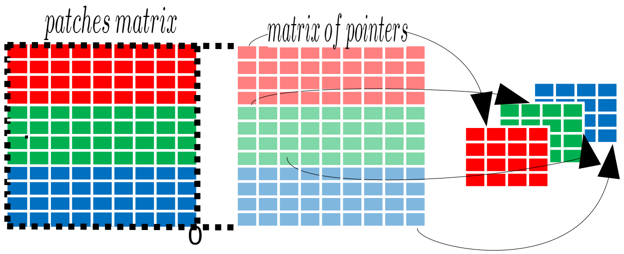


Improved implementation of convolutional layers

- **Idea:** reduce convolutional layers *execution time* by implementing it as a matrix multiplication (GEMM) → GEMM-based convolution



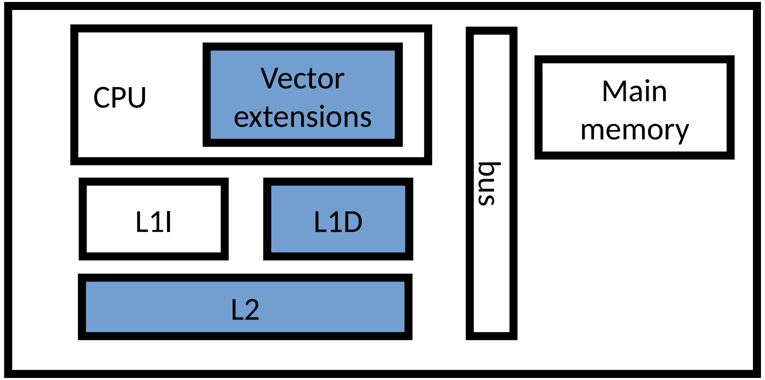
$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$$
$$\mathbf{C} = \mathbf{A} \cdot \mathbf{B}^\intercal$$
$$\mathbf{C} = \mathbf{A}^\intercal \cdot \mathbf{B}$$
$$\mathbf{C} = \mathbf{A}^\intercal \cdot \mathbf{B}^\intercal$$



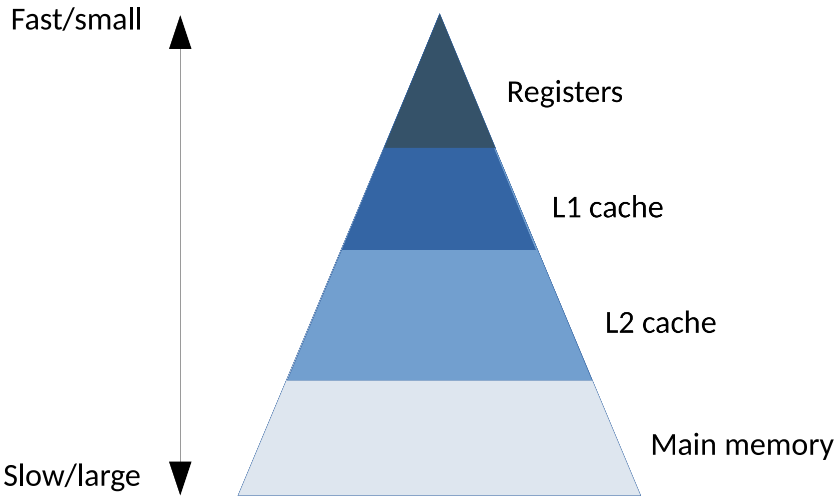
- **Contribution:** compliant C code for several variants (transposed matrices, indirect access)
- **Result:** MET reduced by 50% on average

Architecture-aware GEMM implementation

- **Idea:** take into account hardware specifics (number of vector registers and size of caches)

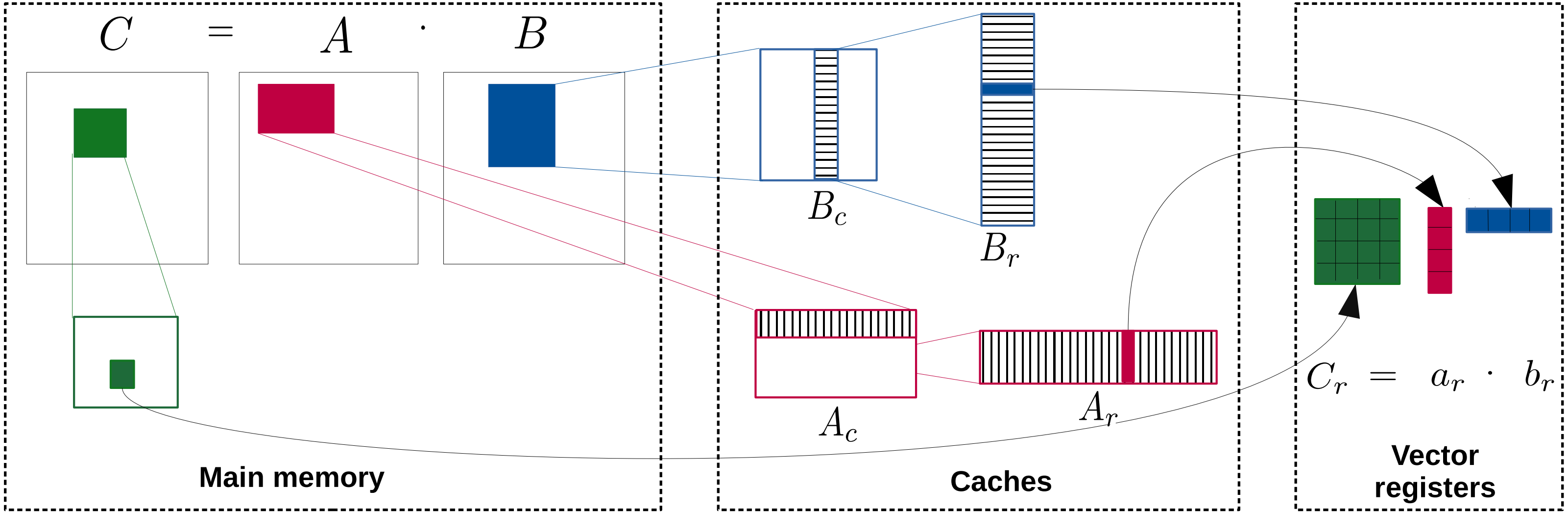


Simplified representation of a SoC



Efficient blocked GEMM implementation

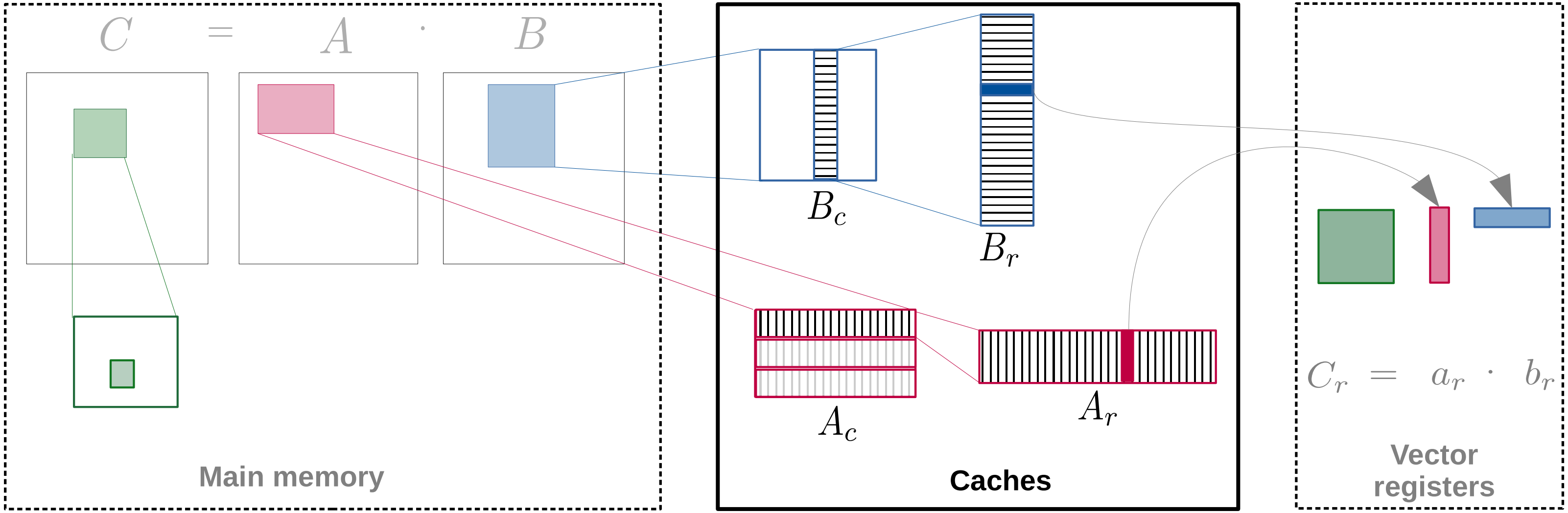
- **Idea:** take into account hardware specifics (number of vector registers and size of caches)
  → blocked matrix multiplication



- **Contribution:** vectorized implementation without compiler optimizations
- **Result:** MET reduced by 98% on average

Efficient and predictable blocked GEMM implementation

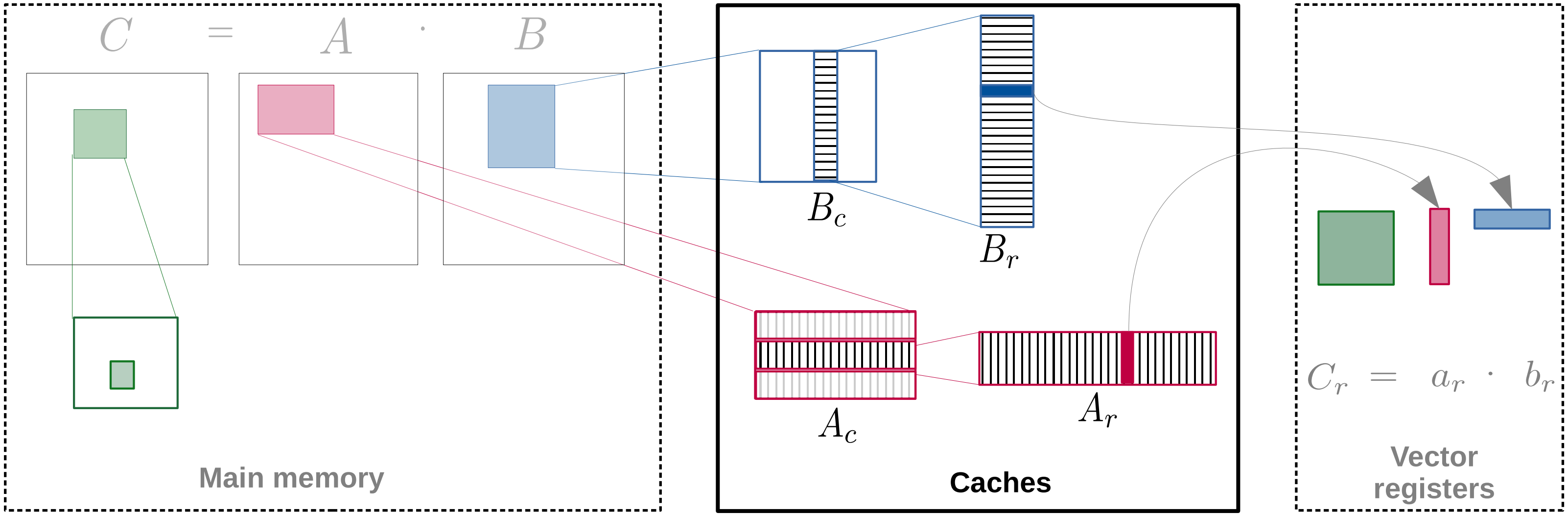
- **Idea:** bound cache misses and tighten the WCET estimation



- **Contribution:** analytical formulae to tune GEMM blocking parameters
- **Result:** cache misses reduced up to 60%

Efficient and predictable blocked GEMM implementation

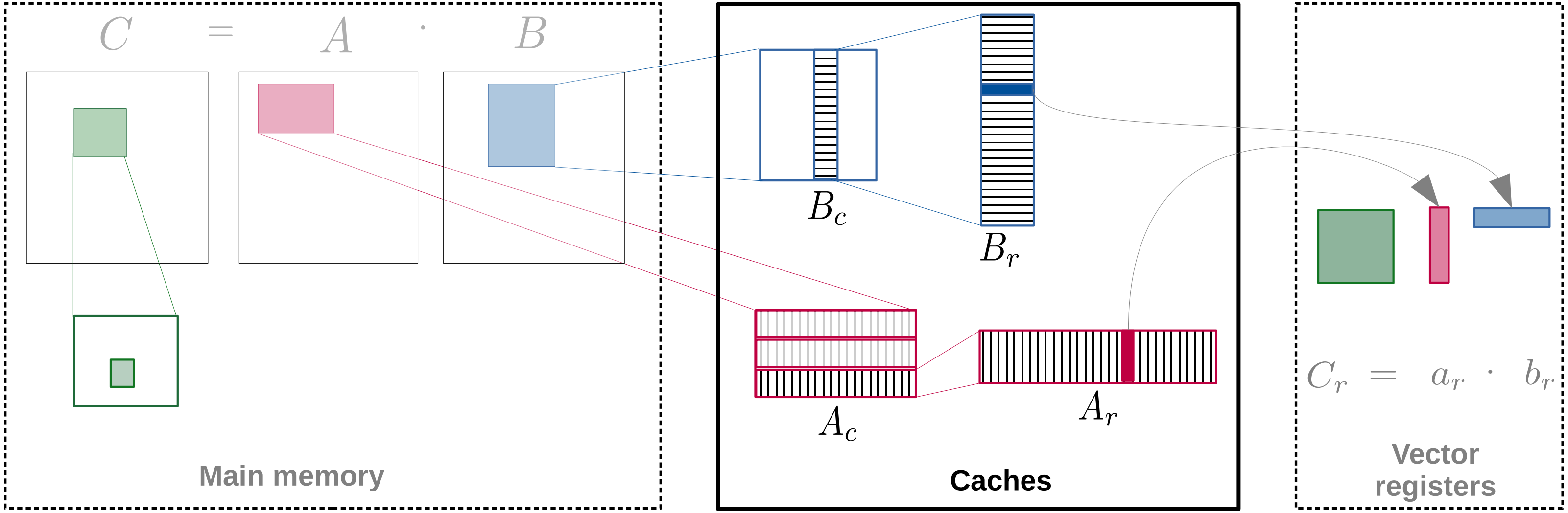
- **Idea:** bound cache misses and tighten the WCET estimation



- **Contribution:** analytical formulae to tune GEMM blocking parameters
- **Result:** cache misses reduced up to 60%

Efficient and predictable blocked GEMM implementation

- **Idea:** bound cache misses and tighten the WCET estimation



- **Contribution:** analytical formulae to tune GEMM blocking parameters
- **Result:** cache misses reduced up to 60%

Conclusions

Automatic generation of **functionally equivalent** and **time-predictable C code** from **feed-forward** neural networks;

Efficient implementation for a given target

Competitive with the state of the art with respect to the defined criteria (semantic preservation, WCET, measured execution time, memory layout)

Perspectives:

- Cover a wider range of inference models architectures;
- Extend automatic *optimized* code generation for different hardware targets.

Thank you for your attention.

Looking forward to your questions!

iryna.de_albuquerque_silva@onera.fr