# Towards Offloading C/C++ Kernels and ONNX Models to CGRAs through MLIR

Nelson Neto[1,2], José Pedro Ferreira[1,2], Pedro Gonçalo Correia[1,2], Juan Gallego[3], Alfonso Rodríguez[3], Andrés Otero[3], Nuno Paulino[1,2], João Bispo[1,2]

EDGE A-IQ READY

Chips JU

Co-funded by the European Union

## Motivation

Offloading edge AI applications to CGRAs is convenient because:

- Increasing edge AI applications requires efficient execution platforms.
- GPUs and FPGAs present scalability and energy-consumption challenges.
- CGRAs offer promising performance-energy balance for edge AI.

ONNX

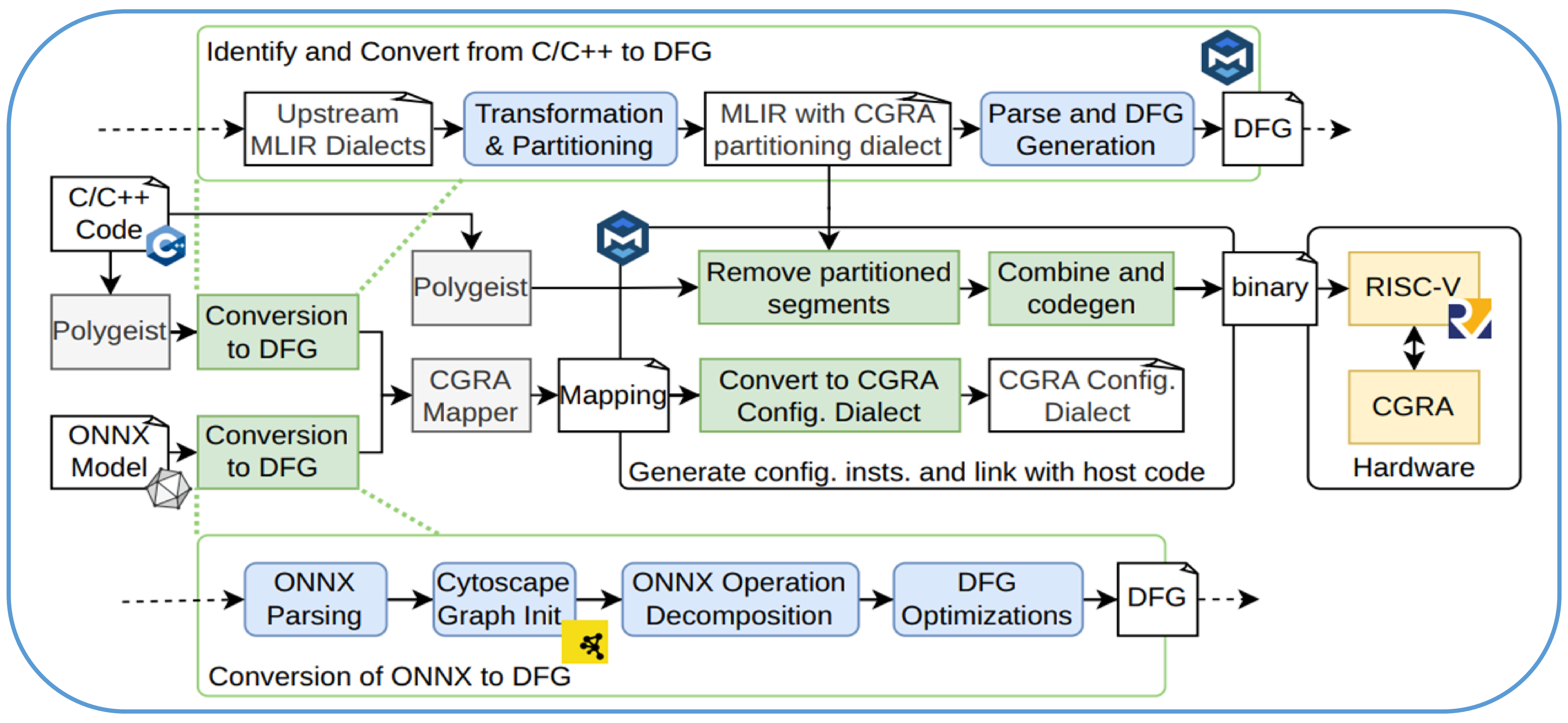
## C/C++ Path: MLIR Dialect for DFG Extraction

- Polygeist tool converts source code to MLIR.
- MLIR-based extraction of vectorized computation kernels as Data Flow Graphs (DFGs).
- Custom MLIR dialect identifies operations compatible with available CGRA processing elements for acceleration.

## ONNX Path: Operation Decomposition

- Direct ONNX parsing into structured JSON objects and Cytoscape graphs.
- Decomposition of high-level ONNX operations (like MatMul) into lower-level arithmetic/memory operations.
- Generation of detailed DFGs, suitable for mapping onto CGRA.

## Architecture Diagram




## Key Contributions

- Unified MLIR-based compilation workflow targeting both C/C++ kernels and ONNX AI models for CGRA acceleration.
- Preserves high-level graph semantics throughout compilation, enhancing opportunities for optimization and debugging.
- Seamless integration of CGRA configuration instructions directly into RISC-V executable binaries, removing the need for external configuration memory.
- Streamlines the deployment process and accelerates the practical adoption of edge AI applications.

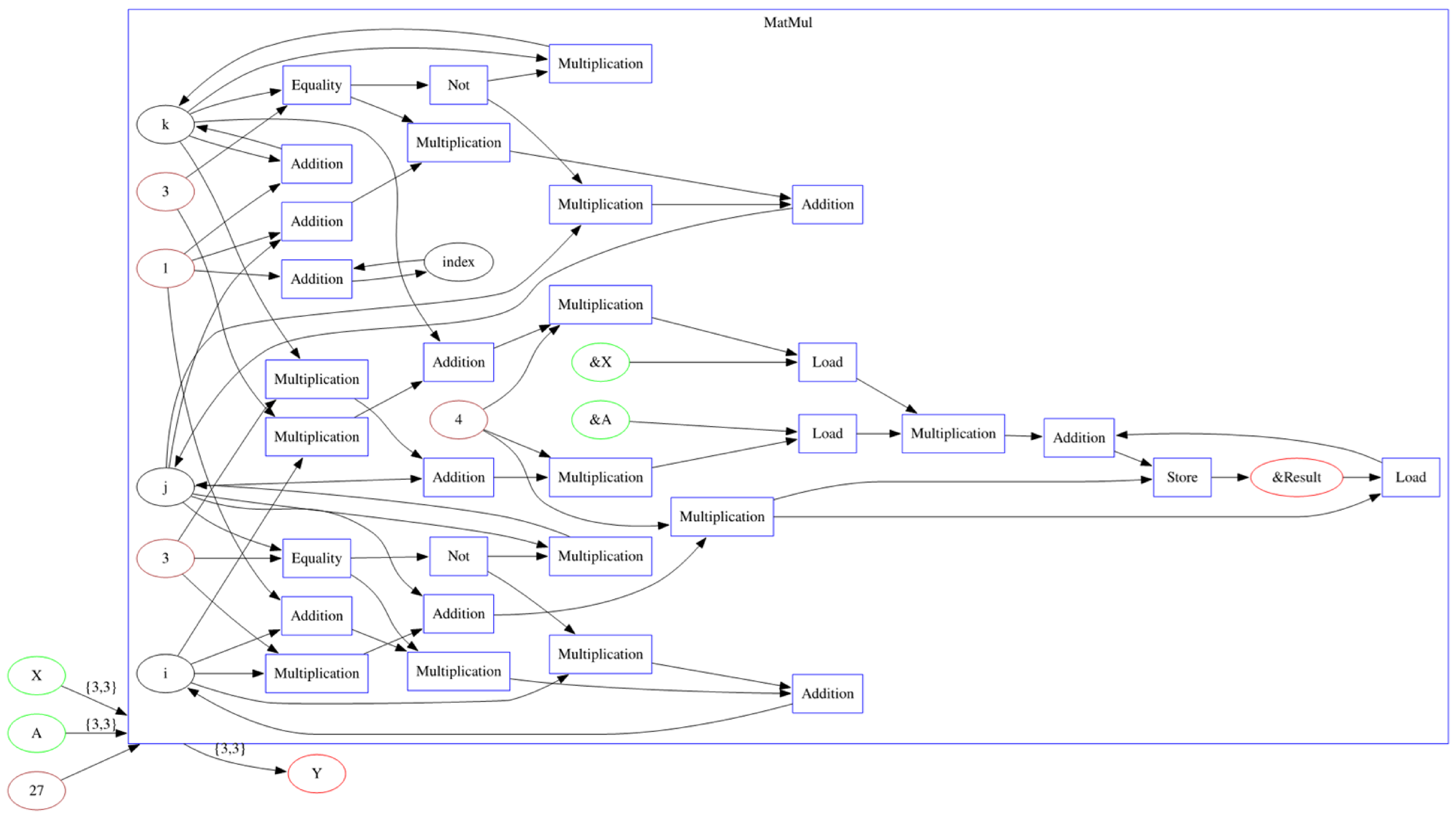
## Results - Data Flow Graphs suitable for CGRA Mapping

```
func.func @mac(%arg0: memref<?xi32>, %arg1: memref<?xi32>) -> i32 {
  %c0_i32 = arith.constant 0 : i32
  %0 = affine.for %arg2 = 0 to 100 iter_args(%arg3 = %c0_i32) -> (i32) {
    %1 = affine.load %arg0[%arg2] : memref<?xi32>
    %2 = affine.load %arg1[%arg2] : memref<?xi32>
    %3 = arith.muli %1, %2 : i32
    %4 = arith.addi %arg3, %3 : i32
    affine.yield %4 : i32
  }
  return %0 : i32
}
```

(a) MLIR code generated by Polygeist for a C/C++ multiply and accumulate kernel

```
func.func @mac(%arg0: memref<?xi32>, %arg1: memref<?xi32>) -> i32 attributes
    {llvm.linkage = #llvm.linkage<external>} {
  %c0 = arith.constant 0 : index
  %c0_i32 = arith.constant 0 : i32
  "cgra.deploy" {
    %0 = vector.transfer_read %arg0[%c0], %c0_i32 : memref<100xi32>, vector<100xi32>
    %1 = vector.transfer_read %arg1[%c0], %c0_i32 : memref<100xi32>, vector<100xi32>
    %2 = arith.muli %0, %1 : vector<100xi32>
    %3 = vector.reduction <add>, %2 : vector<100xi32> into i32
  }
  return %3 : i32
}
```

(b) code kernel transformed and identified with the partitioning dialect



@specs-feup/onnx-flow

## Ongoing Work

- Improving automatic identification and partitioning of CGRA-compatible computations.
- Implementing more powerful loop-level optimizations in the MLIR dialect.
- Extending ONNX support for diverse tensor operations.
- Refining validation methods.

## Conclusion

This workflow successfully unifies compilation of C/C++ kernels and ONNX models into a common graph format for CGRA acceleration. By preserving high-level semantics and embedding configuration directly in RISC-V binaries, it simplifies deployment and enhances edge AI performance, but can still be further optimized and extended.

Supported by Chips Joint Undertaking (Chips JU) and National Authorities under grant agreement n° 101096658 (A-IQ READY)

Supported by Fundação para a Ciência e Tecnologia (FCT) PhD scholarship 2024.05584.BDANA

Project PCI2022-135077-2 funded by MICIU/AEI /10.13039/501100011033 and by European Union NextGenerationEU/PRTR (nº 101096658)

1 U.PORTO FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

2 INESCTEC

3 UNIVERSIDAD POLITÉCNICA DE MADRID

REPÚBLICA PORTUGUESA | fct | MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES | Financiado por la Unión Europea NextGenerationEU | Plan de Recuperación, Transformación y Resiliencia | AGENCIA ESTATAL DE INVESTIGACIÓN | CEI UPM | POLITÉCNICA