HetCIM: Balancing Computations in In-Memory non-volatile heterogenous systems

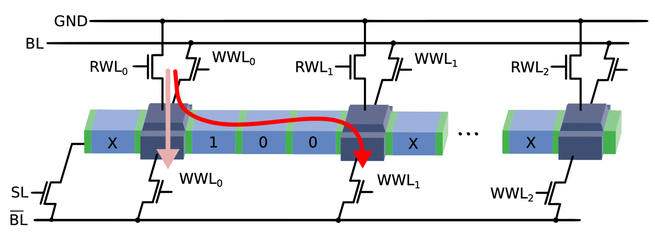
Contact: Asif Ali Khan (asif_ali.khan@tu-dresden.de), Hamid Farzaneh (hamid.farzaneh@tu-dresden.de),
Jeronimo Castrillon (jeronimo.castrillon@tu-dresden.de)

Motivation

- Data movement accounts for more than 60% of total system energy[1]
- Emerging nonvolatile memories (NVMs) allow in-place computations
- Resistive NVMs perform matrix-matrix multiplications in constant time
- Magnetic memories perform bulk bitwise logic with unparallel speed
- Compute-in-memory (CIM) paradigms have demonstrated order-of-magnitude performance and energy benefits
- Programmability of these systems is still a **serious challenge**

CIM using racetrack memory (RTM)

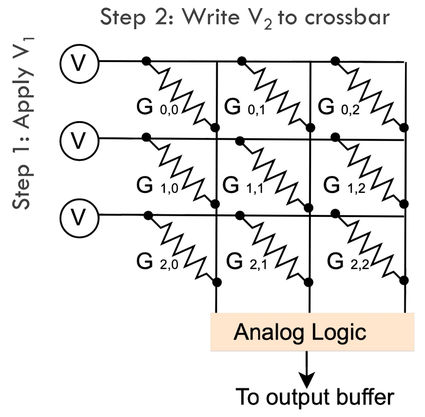
Transverse read (TR) with TR distance equal to 4



Step 1: Perform transverse read (TR)

Step 2: Infer the number of 1s/0s from the TR resistance state

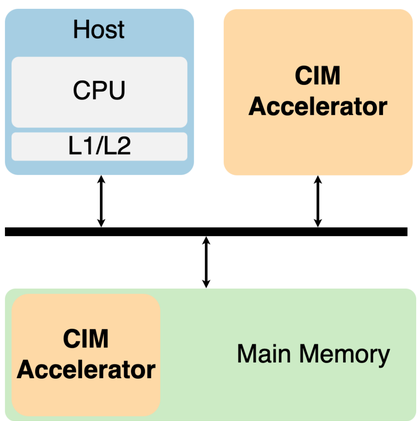
CIM using phase change memory (PCM)



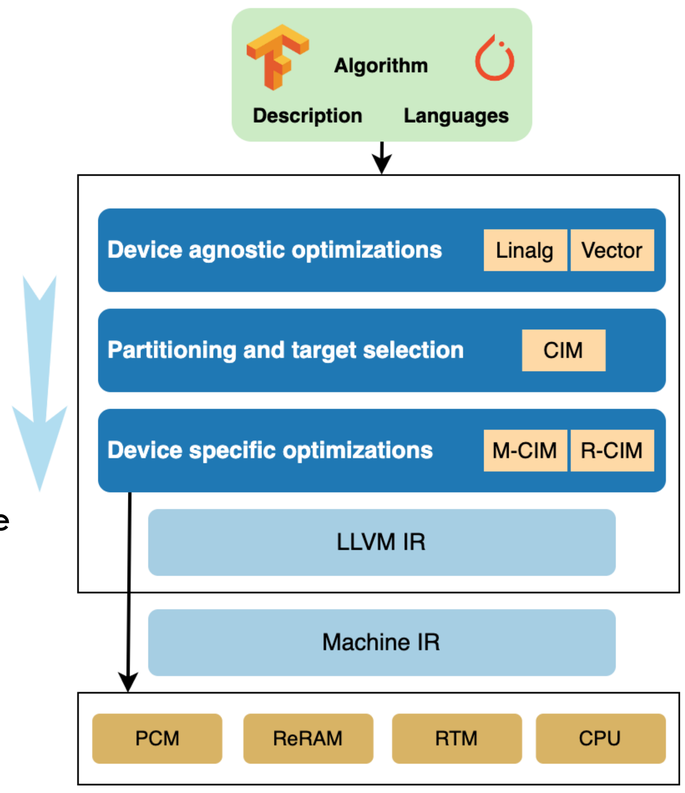
Step 3: Measure output current

Design space exploration and software stack for heterogeneous CIM systems

- We will explore the design space of HetCIM systems by evaluating different architectural decisions
- We will develop a unified simulation infrastructure for HetCIM, based on the one used in OCC[3].
- We will develop APIs for the underlying memristive and RTM CIM devices



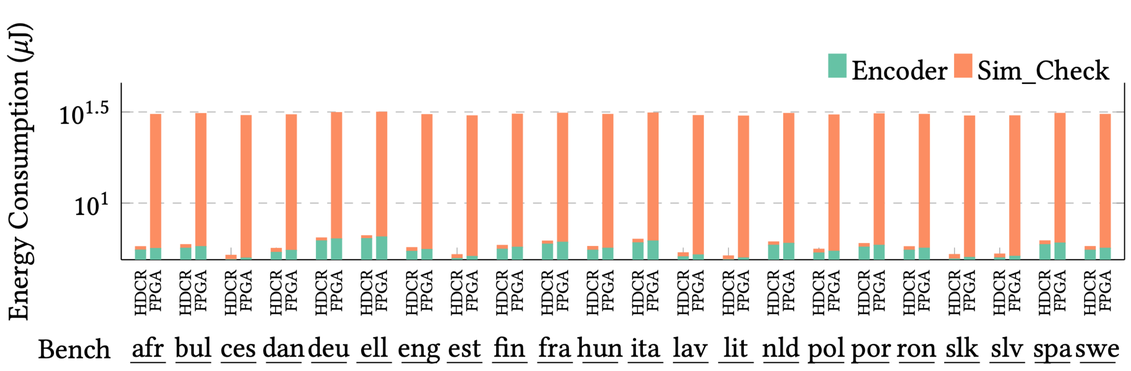
- Multi-level IR (MLIR) is a novel compiler technology inspired by LLVM that offers custom, reusable and extensible abstractions
- We will leverage MLIR to abstract from various CIM devices and develop a multi-level compilation infrastructure for HetCIM
- We will use the hierarchal analysis to find the best suited hardware target
- The progressive lowering enables multi-level, device-agnostic/aware optimizations



Use-case and preliminary work

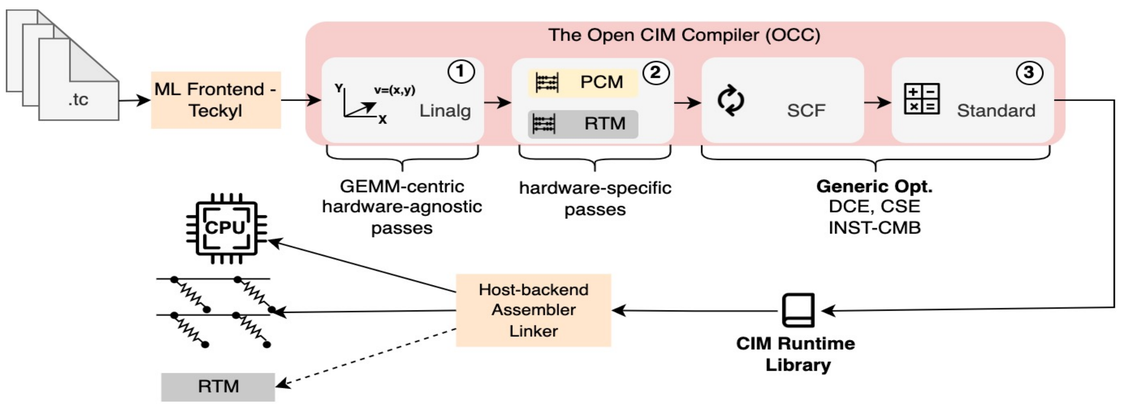
In-RTM hyper-dimensional computing (HDC)[2]

- HDC is a promising model for efficient inference
- It has challenging combination of bit-level and arithmetic operations
- The in-RTM HDC implementation is an order of magnitude faster and energy-efficient compared to the state-of-the-art FPGA implementation



The Open CIM Compiler (OCC)[3]

- An MLIR based end-to-end compilation flow for PCM-based CIM systems
- OCC transparently detects and offloads *GEMM* like kernels to the CIM accelerator
- It performs optimizations to improve performance and PCM lifetime



1. A. Boroumand, S. Ghose, Y. Kim, R. Ausavarungnirun, E. Shiu, R. Thakur, D. Kim, A. Kuusela, A. Knies, P. Ranganathan, O. Mutlu, Google Workloads for Consumer Devices: Mitigating Data Movement Bottlenecks, in: ASPLOS, 2018.
2. Khan et al. , Brain-inspired Cognition in Next Generation Racetrack Memories. ACM Trans. Embed. Comput. Syst. 2022.
3. Siemieniuk, L. Chelini, A. A. Khan, J. Castrillon, A. Drebes, H. Corporaal, T. Grosser, M. Kong, "OCC: An Automated End-to-End Machine Learning Optimizing Compiler for Computing-In-Memory", In IEEE TCAD, Jul 2021