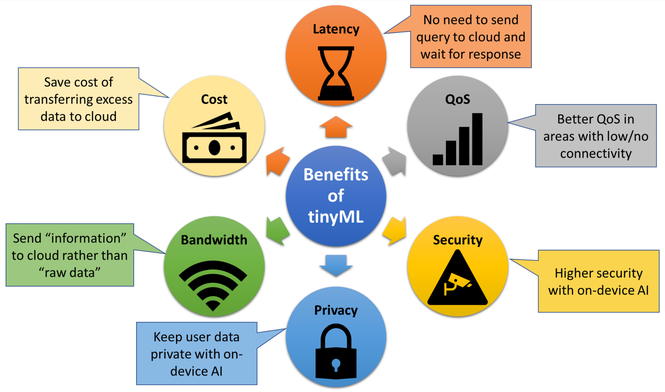
Co-designing the hardware, ISA and software for RISC-V based tinyML processor

Vaibhav Verma, Mircea R. Stan
University of Virginia

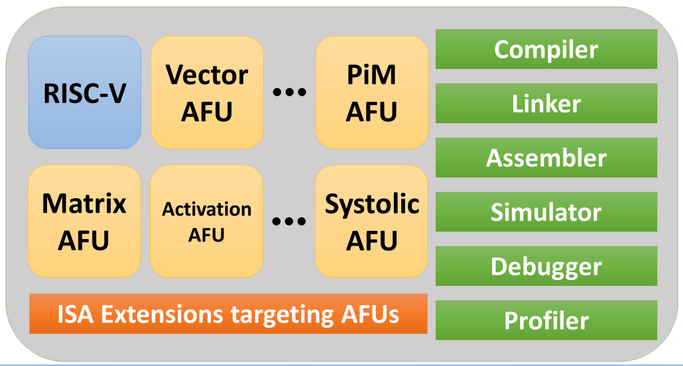
1. Why tinyML?



2. But tinyML is different from cloud ML!

- Smaller neural network models
- Smaller batch sizes (≈ 1)
- Edge devices are power, cost, area and size limited
- Edge devices need to support both AI and non-AI applications
- Edge processors lack support for Keras, PyTorch, MXNet etc.

3. Proposed Solution – AI-RISC



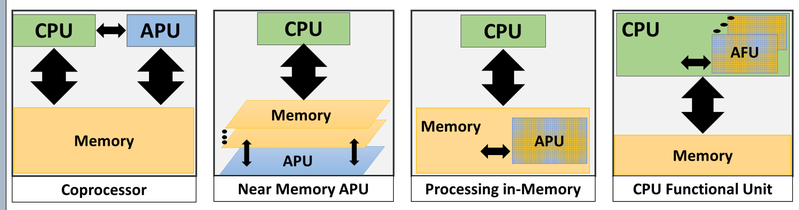
Custom RISC-V processor with ISA extensions targeting AI applications

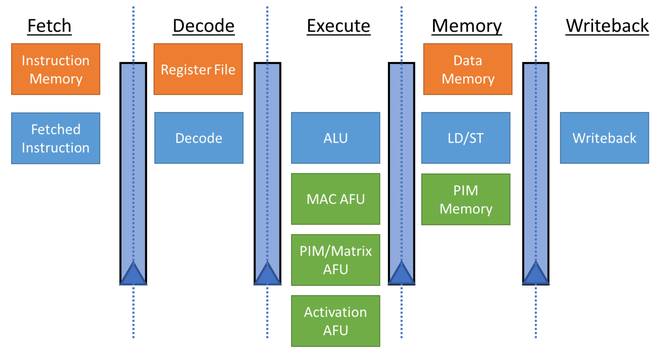
Tightly integrated AI accelerators for fine-grained offloading of AI tasks

End-to-end hardware/ISA/software co-design solution

Support for AI and non-AI applications on the same processor

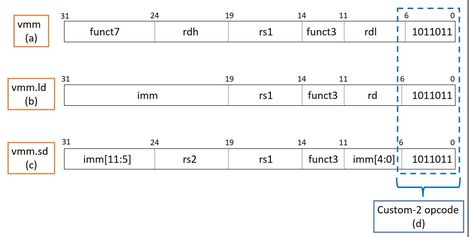
4. Tightly Integrated AI Functional Units (AFU)



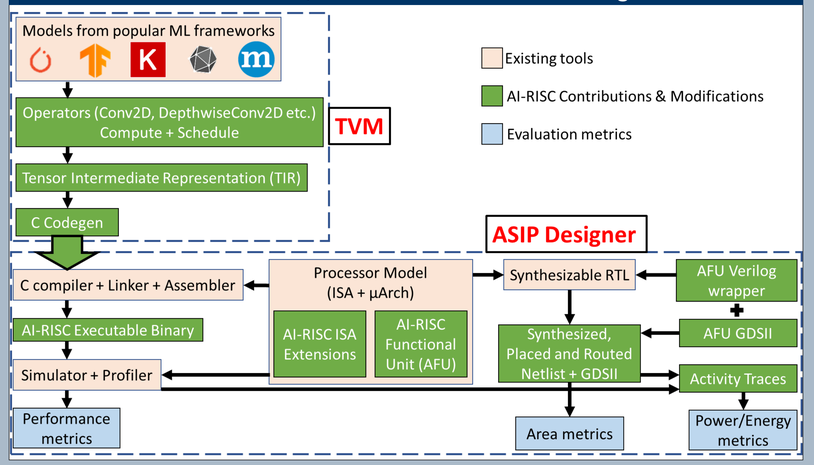


5. RISC-V AI ISA extensions

- MAC
- Packed SIMD MAC
- Post Increment LD/ST
- Hardware Loops
- GEMM, GEMV, GEVV – mxk * kxn | m, n ∈ {1,2,4,8}
- PIM VMM – 1xm * mxn | m, n ∈ {2,4,8,16}
- Activation Functions

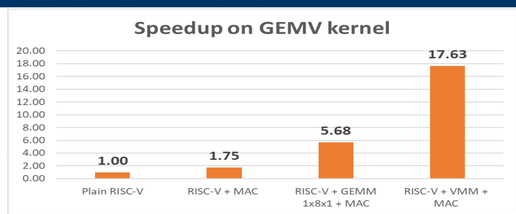


6. Hardware, ISA and Software Co-design

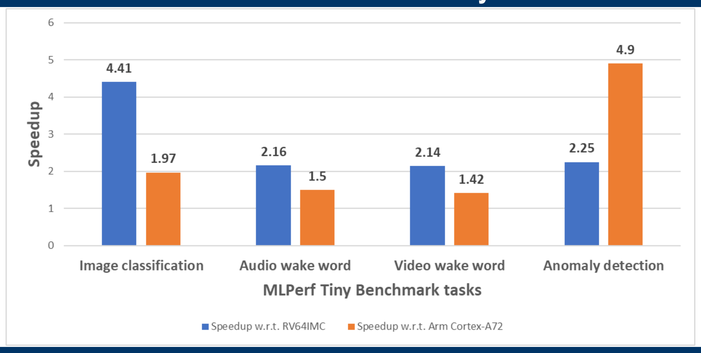


7. Results – GEMV Kernel

- A Matrix → 8x8
- B Vector → 1x8
- Input datatype → int8
- Output datatype → int16



8. Results – MLPerf Tiny



9. Design-Space Exploration

Performance Improvement

Area Overhead

Performance / Area FoM

10. Summary

- **Hardware** – Tightly integrated AI Functional Units (AFU)
- **ISA** – Novel ISA extensions to RISC-V
- **Software** – Complete SDK generation with support for PyTorch, TensorFlow etc.
- **System** – Scalable, flexible and support for both AI and non-AI applications
- **System-design** – Agile co-design of hardware, ISA, software and applications

vv8dn@virginia.edu