Paper 795

MINT-FX: Microscaling Integer Accelerator Template with Flexible Nonlinear Function Execution

Lionnus Kesting*, Arpan Suravi Prasad*, Francesco Conti†, Luca Benini *†
*IIS, ETH Zurich, Switzerland; †DEI, University of Bologna, Italy
Presenter: Lionnus Kesting, ETH Zurich, Switzerland

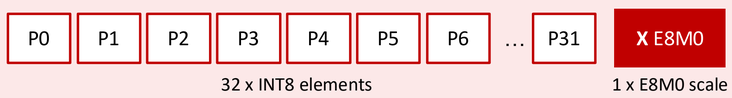
Motivation

Edge devices such as smart glasses run transformer inference on tight compute and energy budgets



Image: Ray-Ban Meta Wayfarer, meta.com

- MXINT8 enables accurate and efficient matrix multiplication using shared block scaling



- Accelerated GEMMs turn **nonlinear operations** such as GELU, softmax, and LayerNorm into the **next bottleneck**

MXINT8 GEMM

Existing nonlinear hardware struggles with two limitations

1. Separate units
Adds data movement and synchronization overhead

2. Fixed-function units
Lack flexibility across transformer operators

Our goal: Integrate nonlinear computation using Piecewise Polynomial Approximation (PwPA) into an MXINT8 accelerator

Experimental Procedures

Design space exploration

MAC array	Lane ratio L/W	PwPA segments S
1K, 2K, and 4K	1, ½, ¼, and ⅛	4, 8, 16, 32, and 64

Software accuracy evaluation

- PyTorch model emulating MXINT8, BF16 accumulation and BF16 PwPA
- 6 ViT models on ImageNet-1k, 2 language models on QNLI
- Compared with FP32 checkpoints, no fine-tuning

Hardware implementation

- GlobalFoundries 12 nm, 1 GHz
- 1K, 2K, 4K MAC arrays with 4, 8, 16, 32, 64 nonlinear lanes
- RTL simulation for kernel cycles
- Post-layout simulations for energy

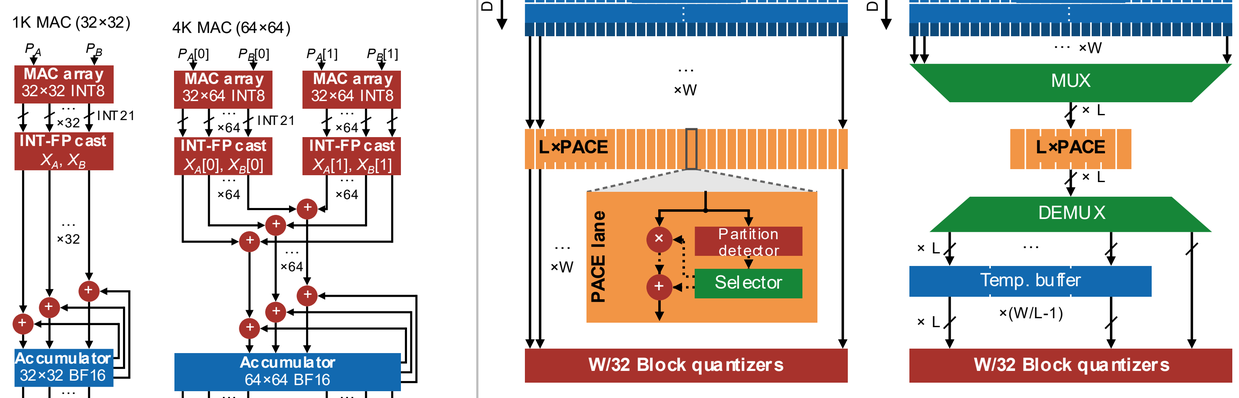
Accuracy vs. PwPA segments

Area, cycles, energy

Design point selection
End-to-end latency and energy vs. area overhead per array

Architecture

MINT-FX combines an MXINT8 MAC array with programmable BF16 nonlinear lanes in a unified accelerator



MXINT8 GEMM engine

- 1K, 2K, and 4K integer MACs
- E8M0 scales applied in the INT→BF16 cast
- BF16 accumulation
- MXINT8 requantized outputs

Integrated PACE lanes

- Runtime-loaded breakpoints and coefficients in one shared memory
- One PACE lane per MAC output lane (L=W), or shared via mux (L<W)
- Placed between accumulators and block quantizers, operating on BF16 intermediate values

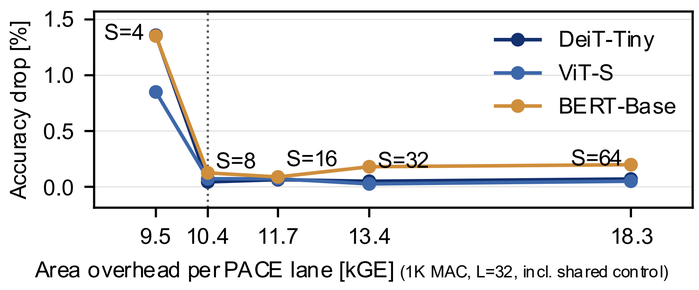
Flexible nonlinear execution

- Activations fused with GEMM outputs
- Controller performs softmax and LayerNorm using the PACE lanes

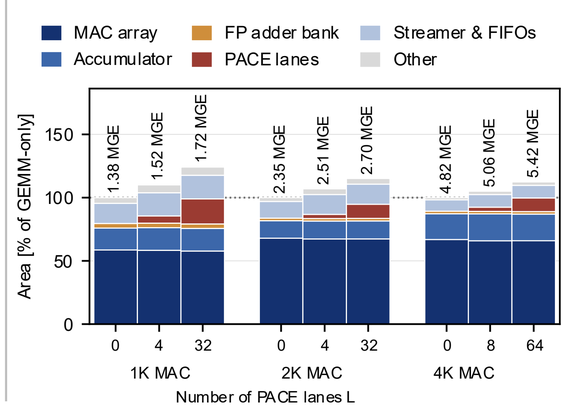
Experimental Results

1 Accuracy–area trade-off

Degree 1, 8 segments PwPA reaches the accuracy plateau with **≤0.7%** loss on all 8 models without fine-tuning

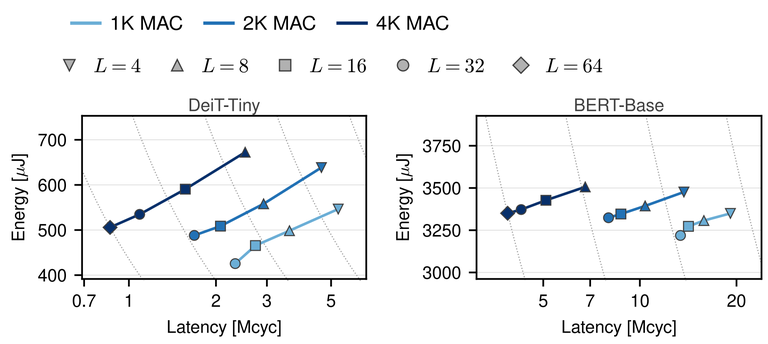


2 Hardware overhead



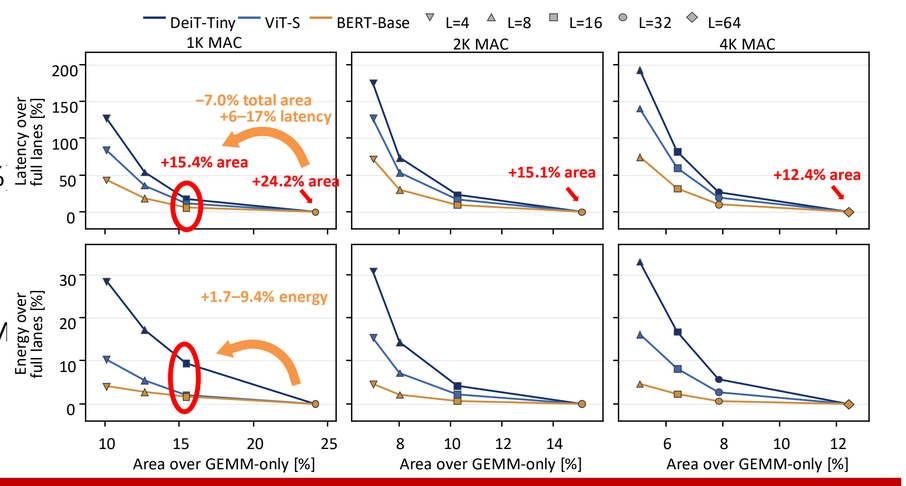
3 Latency and energy

- **Full lanes** give the lowest latency and energy on every array
- **1k MAC** gives lowest accelerator energy, **4k MAC** the lowest end-to-end latency



4 Area–latency design

- **1K MAC, half lanes: −7.0% total area** (overhead 24.2% → 15.4% for **+6–17%** latency and +1.7–9.4% energy
- **2K/4K MAC: ≤15.1%** area overhead over GEMM-only accelerator at full PwPA lanes



Key takeaways:
- One PACE lane per MAC output lane minimizes all network latency and energy on every array for as little as 12.4% area overhead over the GEMM-only accelerator
- Halving the nonlinear lanes makes sense on an area-constrained 1K MAC array only, with 15.4% area overhead over the GEMM-only accelerator

ETH zürich