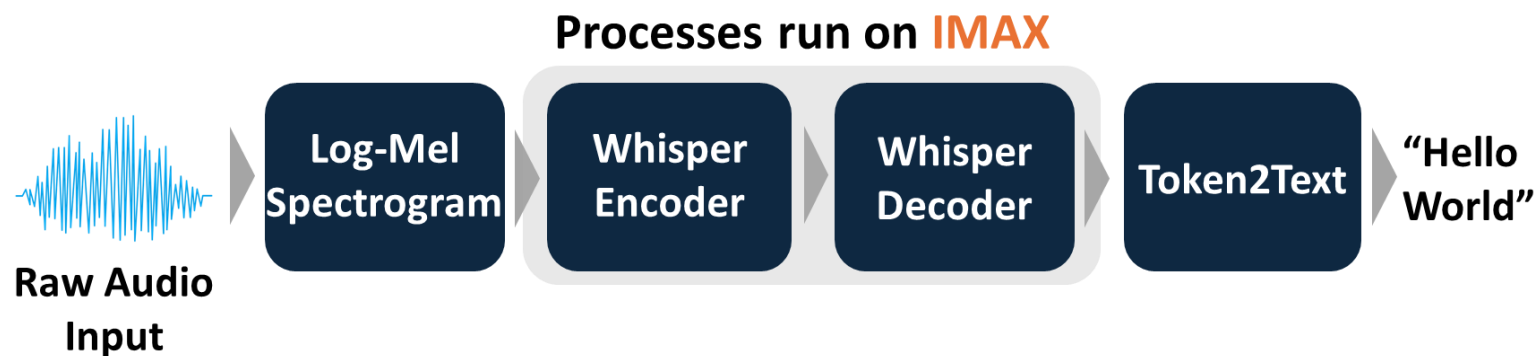


Energy-Efficient Hardware Acceleration of Whisper ASR on a CGLA

Takuto Ando, Yu Eto, Ayumu Takeuchi, Yasuhiko Nakashima
Nara Institute of Science and Technology (NAIST)



Highlights

World's First

Successfully demonstrated **Whisper ASR*** kernels on a CGRA.

Superior Energy Efficiency (PDP)

Outperforms existing GPUs (ASIC Prediction)

- **1.90x** vs. Edge GPU (Jetson AGX Orin)
- **9.83x** vs. High-end GPU (RTX 4090)

Conclusion

CGLA shows potential as a **sustainable ASR platform for the edge.**

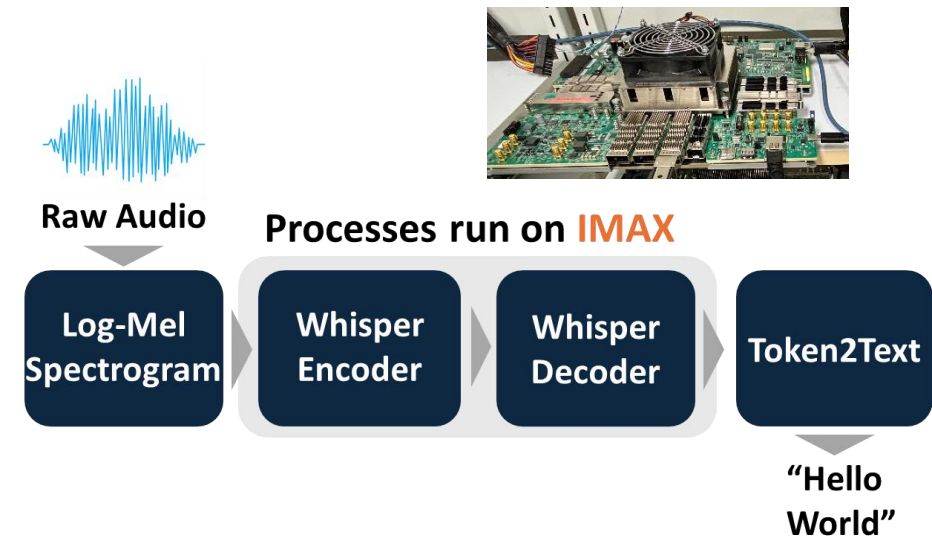


Table of contents

01

Background & Objective

Defining the challenge and our research goal.

02

Proposed Method (HW/SW Co-Design)

Our accelerator architecture and optimization strategy.

03

Experimental Results

Our evaluation setup and key performance findings.

04

Discussion & Conclusion

Analyzing the results and summarizing our takeaways.

AI's Energy Problem & Our Solution

Problem: AI relies on power-hungry GPGPUs.

GPGPU: High Flexibility 😄 , Low Efficiency 😞

Specialized ASIC: High Efficiency 😄 , Low Flexibility 😞



Solution: CGRA - Bridging the Gap

We focus on CGRAs (Coarse-Grained Reconfigurable Arrays) as an architecture to solve the trade-off.

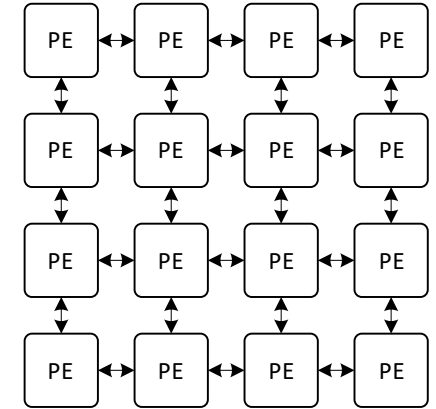
- ▲ **Efficiency:** They achieve ASIC-like efficiency by directly mapping dataflows.
- ▲ **Flexibility:** They retain flexibility by dynamically reconfiguring the datapath.

Our CGLA “IMAX” ...What’s this? 🤔

Conventional CGRA

Concept: Reconfigurable at the PEs level on a 2D array

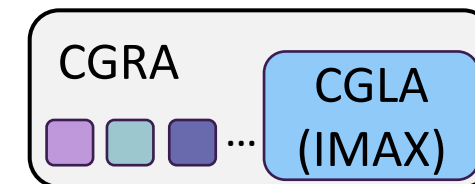
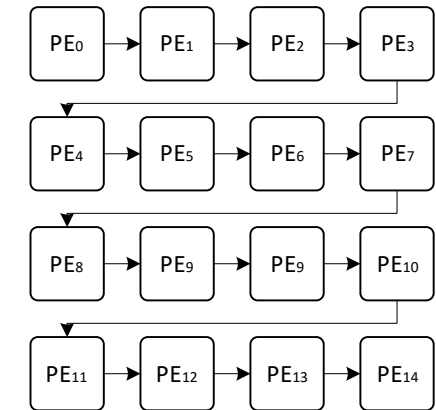
Challenge: Complex routing algorithms lead to long compilation times



IMAX Architecture (CGLA: Coarse-Grained Linear Array)

Concept: PEs are arranged in a linear structure, fast compilation

Design: Interleaves 64 PEs with 512KB of local cache memory to minimize slow main memory access



➡ The Next Step: Deployment to **ASR** (Automatic Speech Recognition)

ASR Challenges

What is **ASR** (Automatic Speech Recognition)?

Technology that converts human speech into text.



Modern ASR models are **highly accurate** but **computationally massive**.



Prior ASR Accelerators

Focused on specialized hardware (ASICs/FPGAs) [1-2].

When new models (like Whisper) emerged, this hardware became **obsolete**.

Our Objective

- ▲ Implement Whisper's core kernels on our flexible and efficient CGLA (IMAX).
- ▲ **Demonstrate CGLA's superior energy efficiency** compared to GPUs.

[1] H. Hu, J. Li, C. Wu, X. Li, and Y. Chen, "Design and Implementation of Intelligent Speech Recognition System Based on FPGA," 2022.

[2] S. D. Yamini, G. S. Mirishkar, A. K. Vuppala, and S. Purini, "Hard-ware Accelerator for Transformer based End-to-End Automatic Speech Recognition System," 2023.

Table of contents

01

Background & Objective

Defining the challenge and our research goal.

02

Proposed Method (HW/SW Co-Design)

Our accelerator architecture and optimization strategy.

03

Experimental Results

Our evaluation setup and key performance findings.

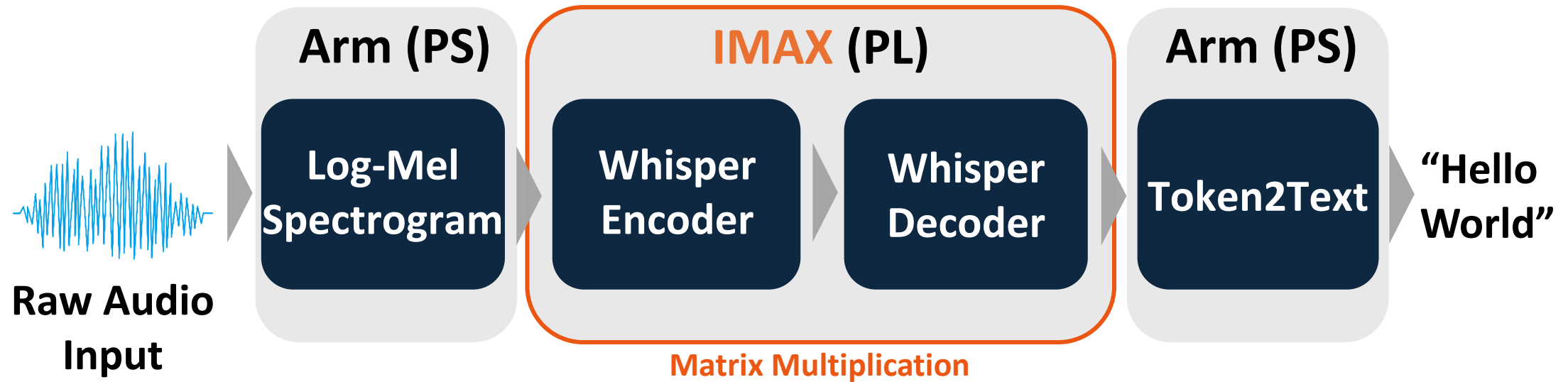
04

Discussion & Conclusion

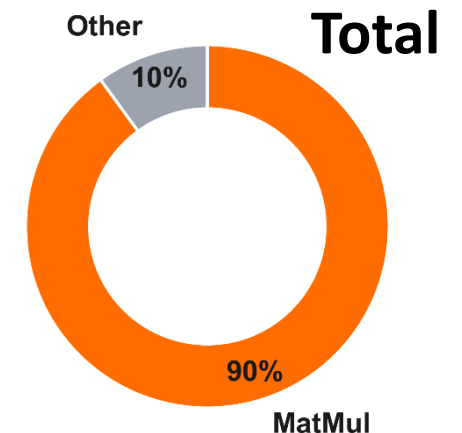
Analyzing the results and summarizing our takeaways.

Whisper ASR System Overview

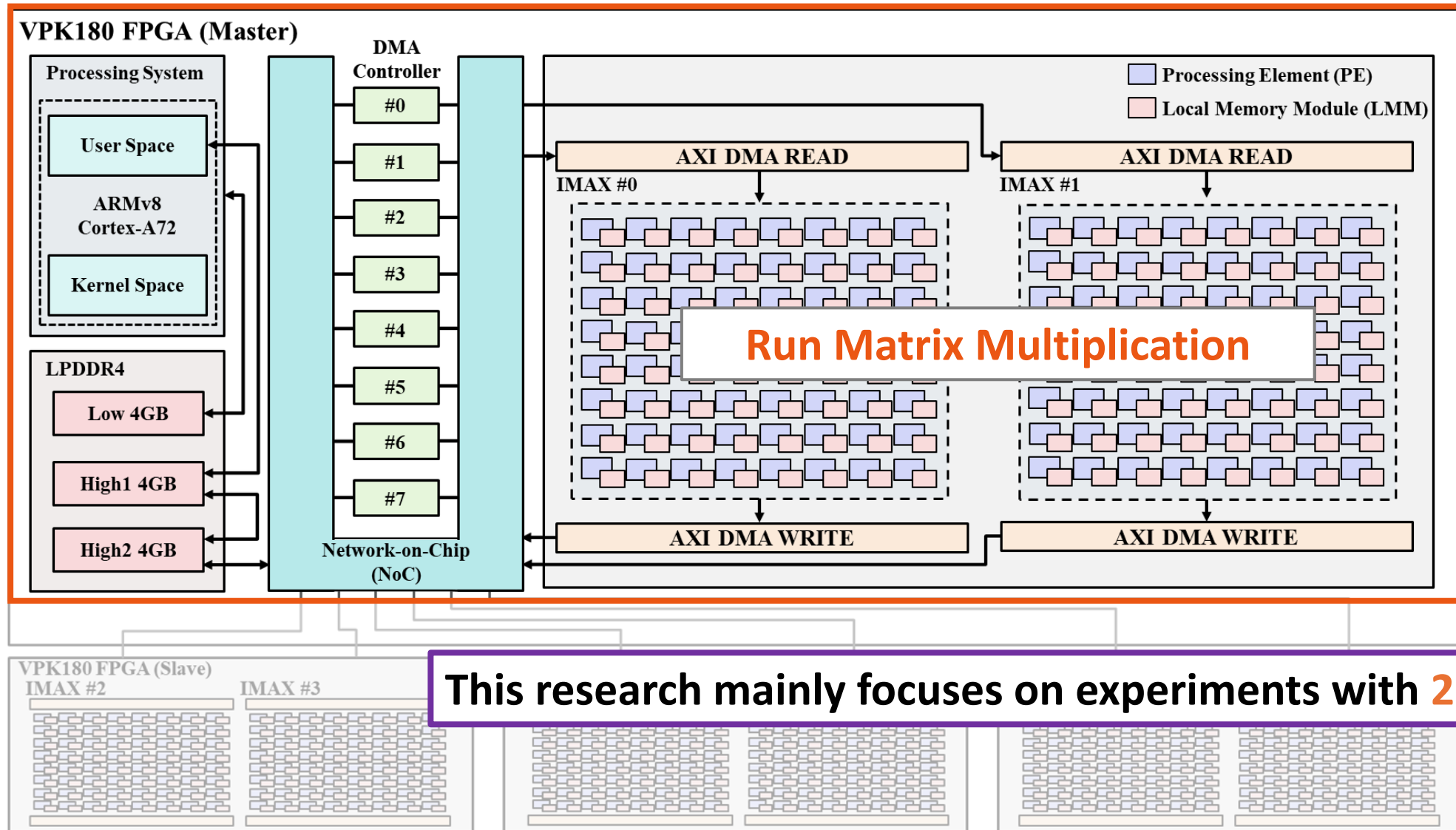
Standard Whisper ASR Process:



Most processing time is spent on **Matrix Multiplication**



IMAX Configuration



Dataflow within a Single PE Unit

1. Input Registers:

Hold data for the current cycle (n).

2. Pipelined Execution:

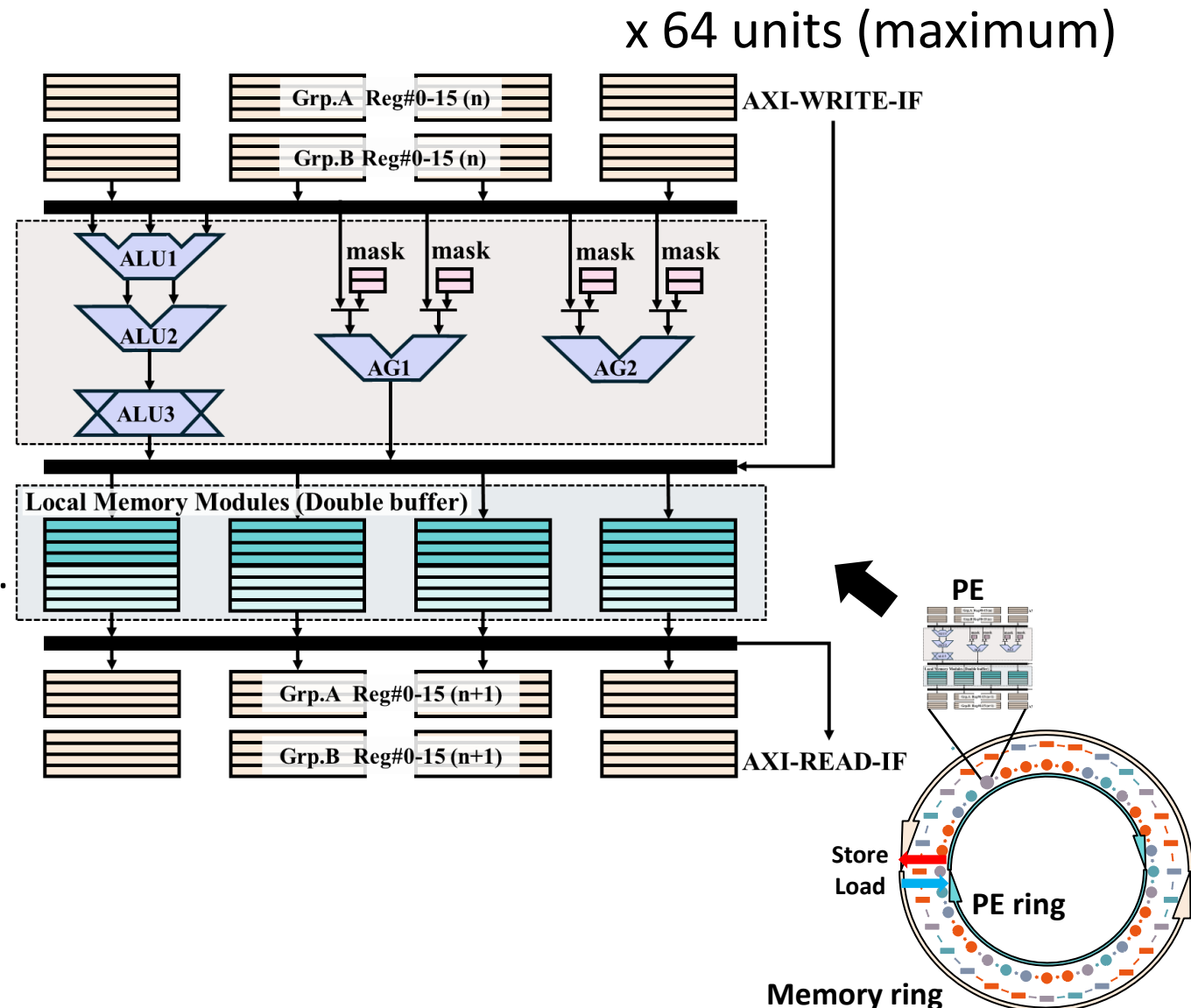
Data flows through a 3-stage ALU and AGs(address generation).

3. LMM (Load and Store)

Load or store data in 512KB of local memory, overlapping computation with data transfers.

4. Output to Next Stage:

Results are forwarded to the next PE's registers (n+1).



Mat Mul (A x B) Kernel Implementation

10

F16 kernel

1. Load:

FP16(float-16) data (A, B) .

2. Convert:

FP16 → FP32 conversion using LUT.

3. Compute (Float):

Performs the **multiply-accumulate operation in FP32**.

4. Store:

Sum the accumulation results and store them.

Q8_0 kernel

1. Load:

I8(Int-8) quantized data (A,B).

2. Compute (Int):

SIMD **8-bit Integer Dot Product** (I8 x I8).

3. Accumulate (Int):

Accumulates the parallel dot product results as 32-bit integers.

4. Decode (Dequantize):

Converts the final integer sum into an FP32 using the loaded scale and a LUT.

IMAX enhancements

Key Optimization Techniques

▲ In-line Conversion:

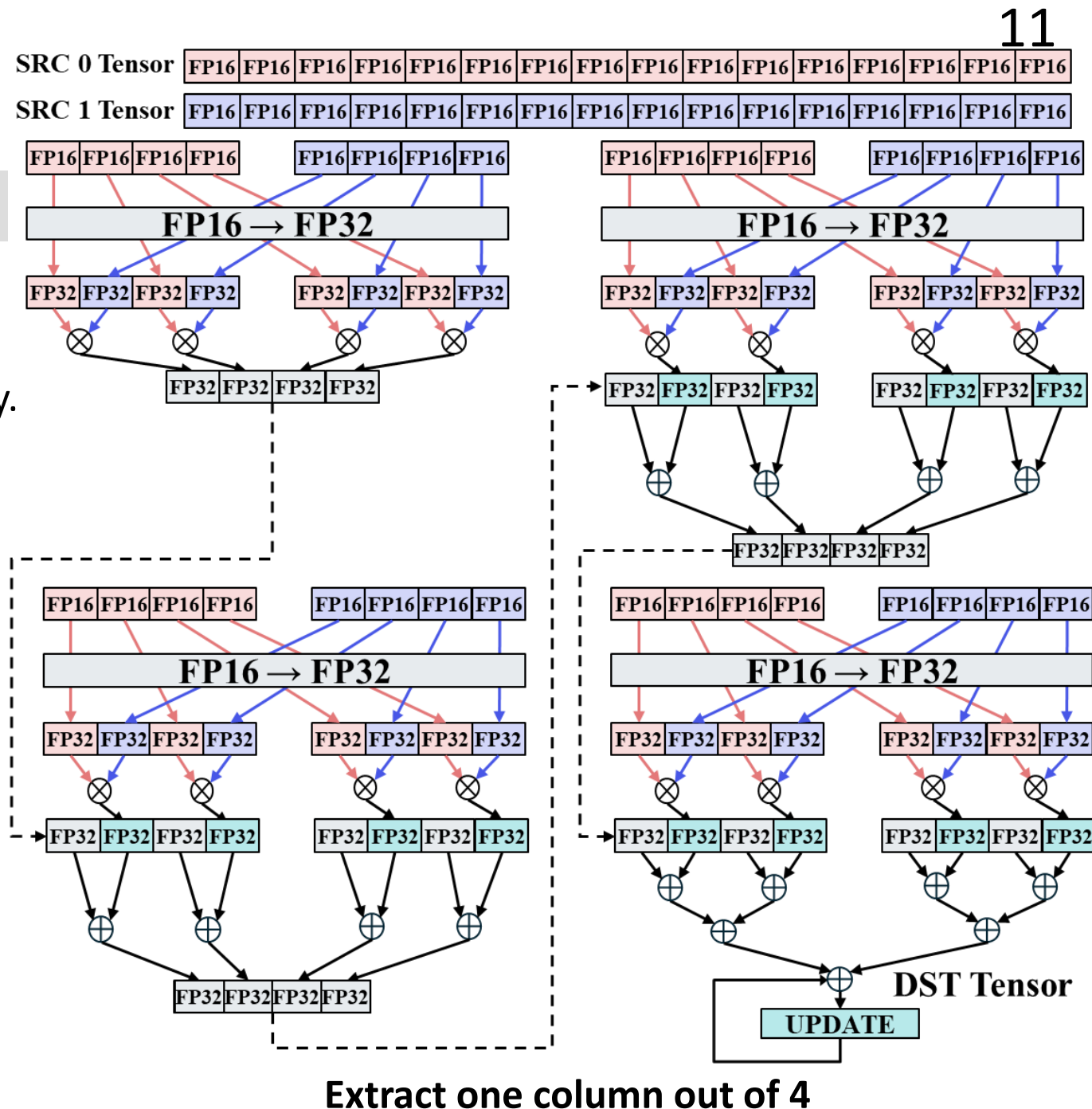
PEs perform FP16→FP32 conversion on-the-fly.

▲ SIMD Execution:

The 64-bit datapath processes two 32-bit operations in parallel.

▲ Column-wise Multithreading:

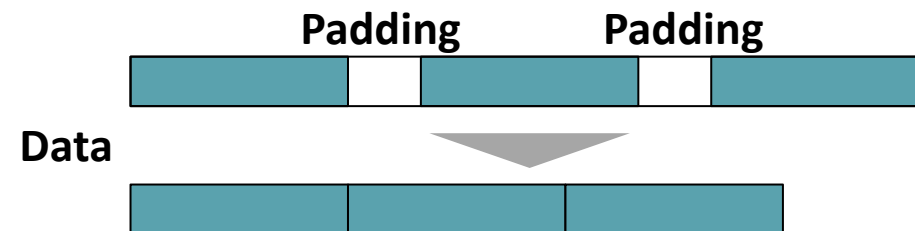
A single physical FPU executes 4 logical FMA operations through time-multiplexing.



Data Transfer & Memory

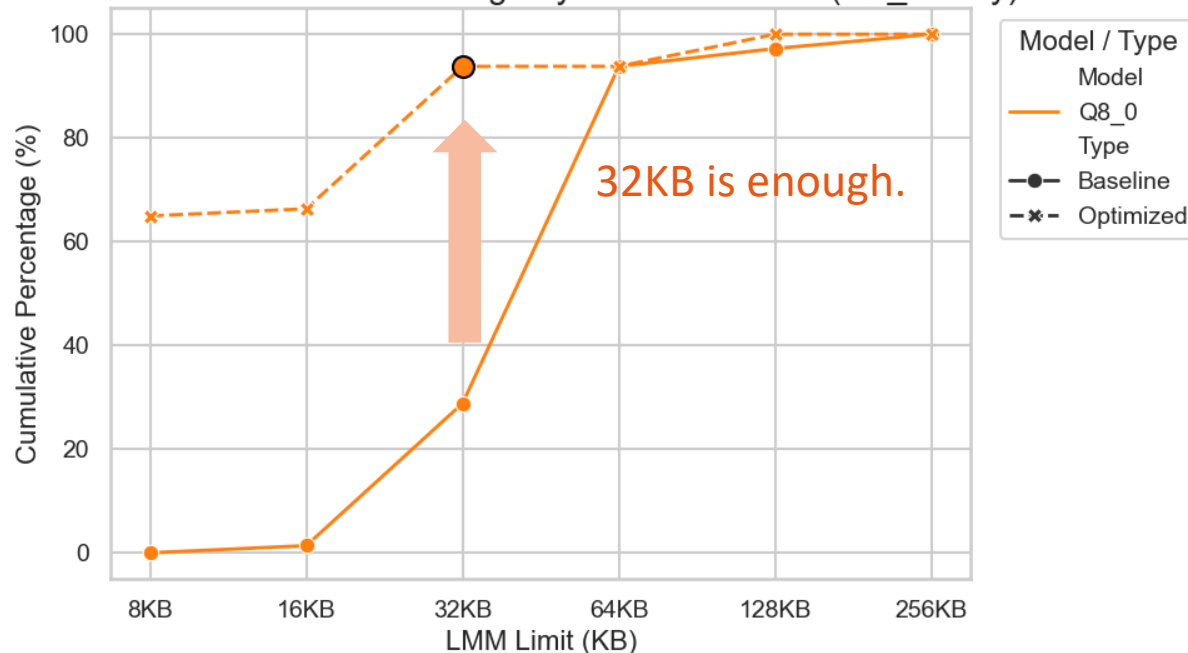
Challenge: Data tensors contain **significant padding** for alignment

Solution: Pre-Transfer **Padding Removal**



The Host CPU **strips all padding** before DMA transfer

Cumulative Model Percentage by LMM Size Limit (Q8_0 Only)



Cumulative Model Percentage by LMM Size Limit (F16 Only)

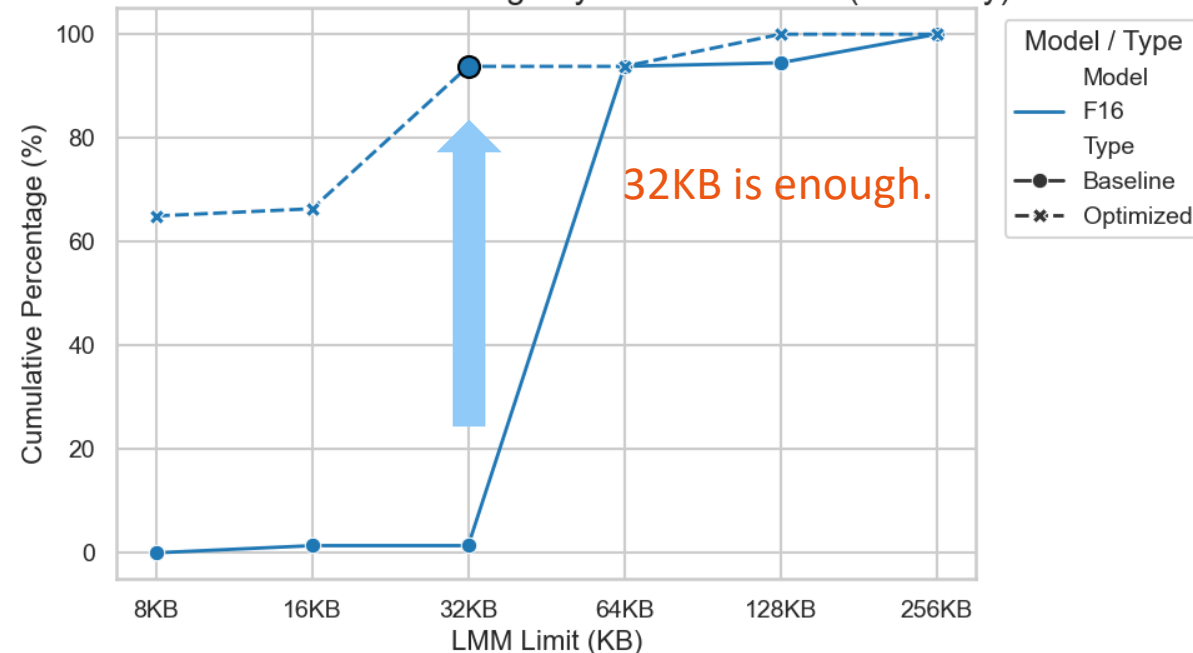


Table of contents

01

Background & Objective

Defining the challenge and our research goal.

02

Proposed Method (HW/SW Co-Design)

Our accelerator architecture and optimization strategy.

03

Experimental Results

Our evaluation setup and key performance findings.

04

Discussion & Conclusion

Analyzing the results and summarizing our takeaways.

Experimental Setup

Workload: Transcribed the jfk.wav using the whisper.cpp framework with **Whisper-tiny.en**.

Objective: To evaluate the performance of our **Whisper ASR** on the IMAX.



jfk.wav



VPK180 FPGA (145 MHz)



And so my fellow Americans ask not what your country
ask what you can do for your country.

Output result

Two key metrics for **the end-to-end transcription task**:

- ▲ **Performance:** The ***E2E latency** required to transcribe the audio file, measured in sec.
- ▲ **Energy Efficiency:** The **Power-Delay Product** (PDP), calculated as $Latency (s) \times Power (W)$.

*E2E Latency: End-to-End Latency

Experimental Setup



VPK180

TABLE III: Physical specifications and performance comparisons of various devices

Device	CPU	Cores	Chip area (mm ²)	Process node	Operating frequency	Memory	Power ^c (W)
ARM Cortex-A72 (on Versal)	-	2	-	7 nm	1400 MHz	8 GB DDR4	0.6485
IMAX3 (Xilinx VPK180)	ARM Cortex-A72	64 ^a	-	7 nm	145 MHz	8 + 4 GB DDR4 ^b	180
IMAX3 (28nm)	-	64 ^a	14.6	28 nm	840 MHz	-	0.647 or 1.32
Jetson AGX Orin 32GB	Arm® Cortex-A78AE	1792	200	8 nm	930 MHz	32 GB DDR5	15 ^d
NVIDIA RTX 4090	Xeon W5-2455X	16384	608	5 nm	2520 MHz	24 GB DDR6	450

^a The number of cores for IMAX3 refers to the number of PEs per lane. ^b 8 GB DDR4 for OS buffer and 4 GB DDR4 for DMA buffer.

^c IMAX3 (28 nm) is an estimated value, references for other devices are from Cortex-A72 [27], Jetson AGX Orin 32GB [28], NVIDIA RTX 4090 [29].

^d This device has multiple power consumption modes (15–45 W), and we executed in the lowest power consumption mode.

IMAX Platform:

Performance and power were projected for a 28nm ASIC implementation, allowing for a realistic 6x frequency increase to 840 MHz(FPGA:145MHz).

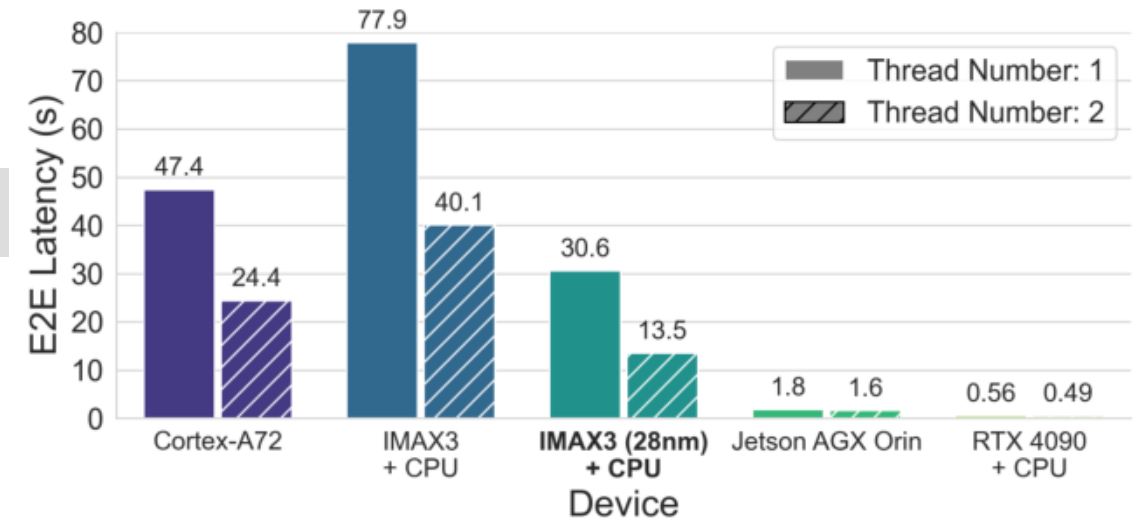
E2E Latency

GPU are Fastest:

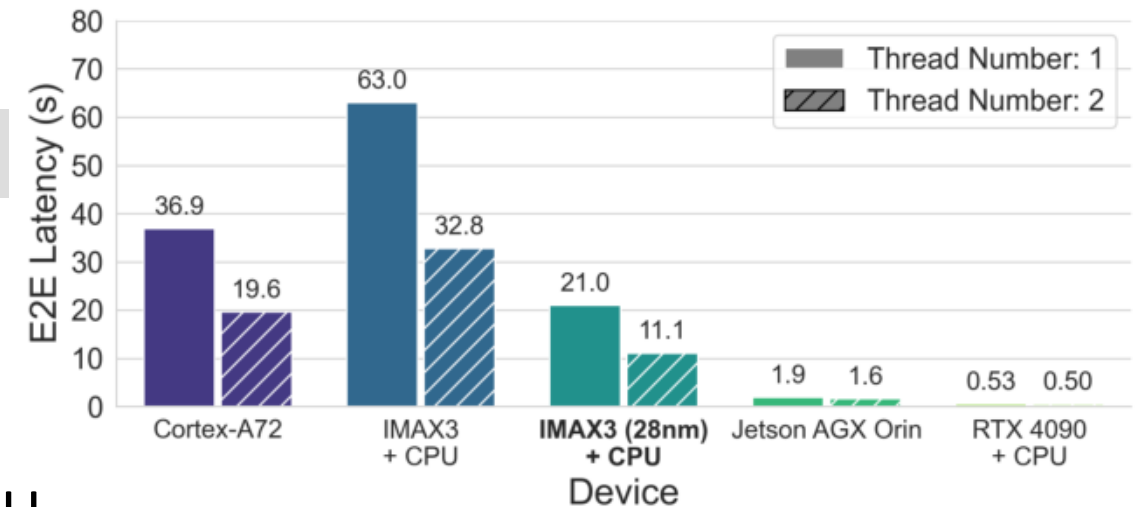
- RTX 4090: 0.5s
- Jetson AGX Orin: 1.6s
- Their parallelism and high-power deliver the lowest latency.

IMAX Shows Significant Speedup:

- IMAX (ASIC): 11s
- CPU (ARM Cortex-A72): 20s
- IMAX provides a **1.8x speedup** over the CPU, demonstrating effective acceleration.



(a) FP16 model inference



(b) Q8_0 model inference

E2E latency comparison by device

Energy Efficiency (PDP)

$$PDP = \text{Latency (s)} \times \text{Power Consumption (W)}$$

**lower is better*

GPUs Show Poor Energy Efficiency

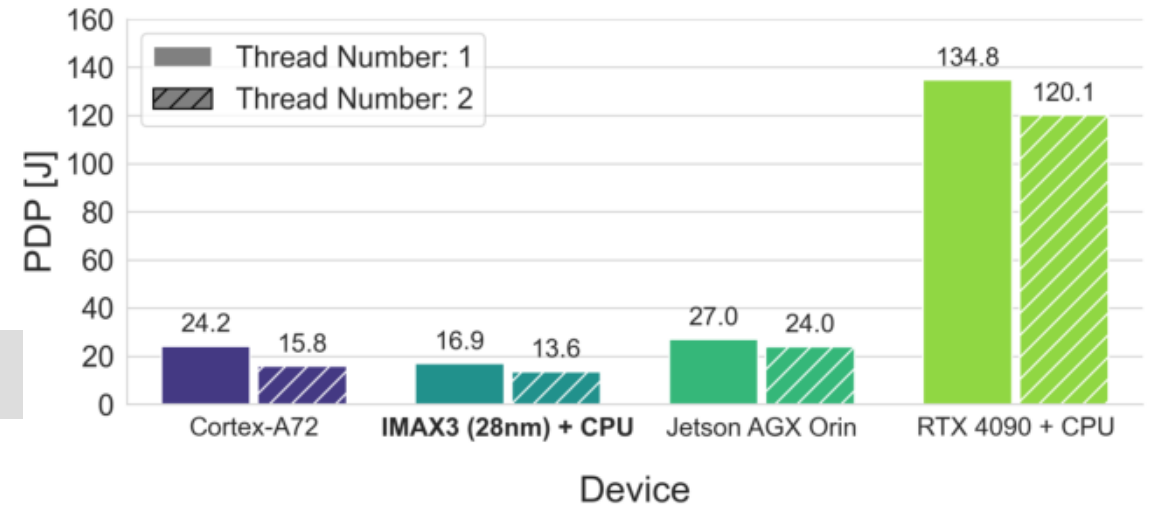
Despite their speed,

GPUs consume far more total energy.

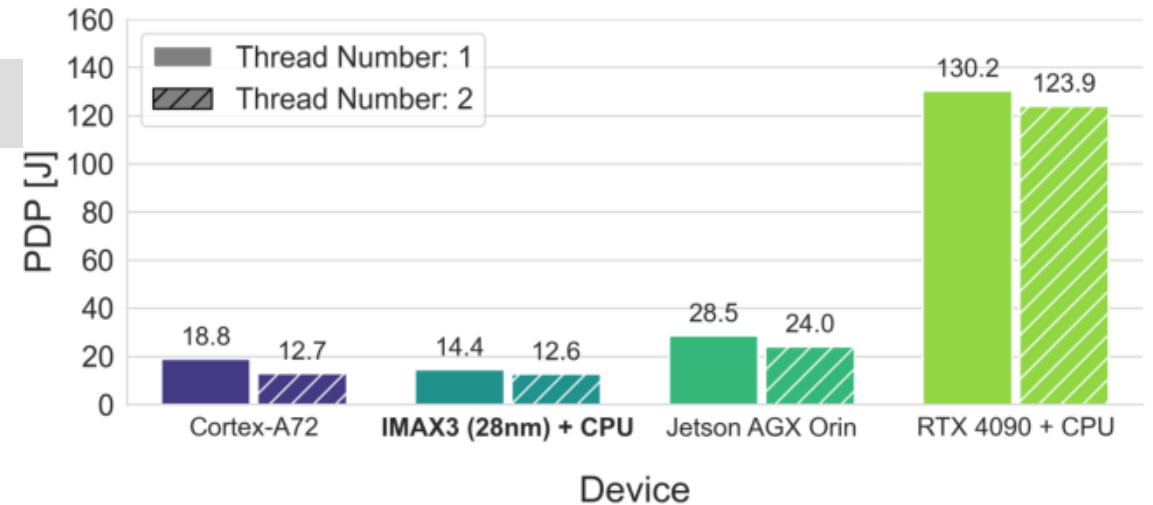
IMAX is superior in energy efficiency:

1.90x more efficient than Jetson AGX Orin

9.83x more efficient than RTX 4090



(a) FP16 model inference



(b) Q8_0 model inference

PDP performance comparison by device (lower is better)

Table of contents

01

Background & Objective

Defining the challenge and our research goal.

02

Proposed Method (HW/SW Co-Design)

Our accelerator architecture and optimization strategy.

03

Experimental Results

Our evaluation setup and key performance findings.

04

Discussion & Conclusion

Analyzing the results and summarizing our takeaways.

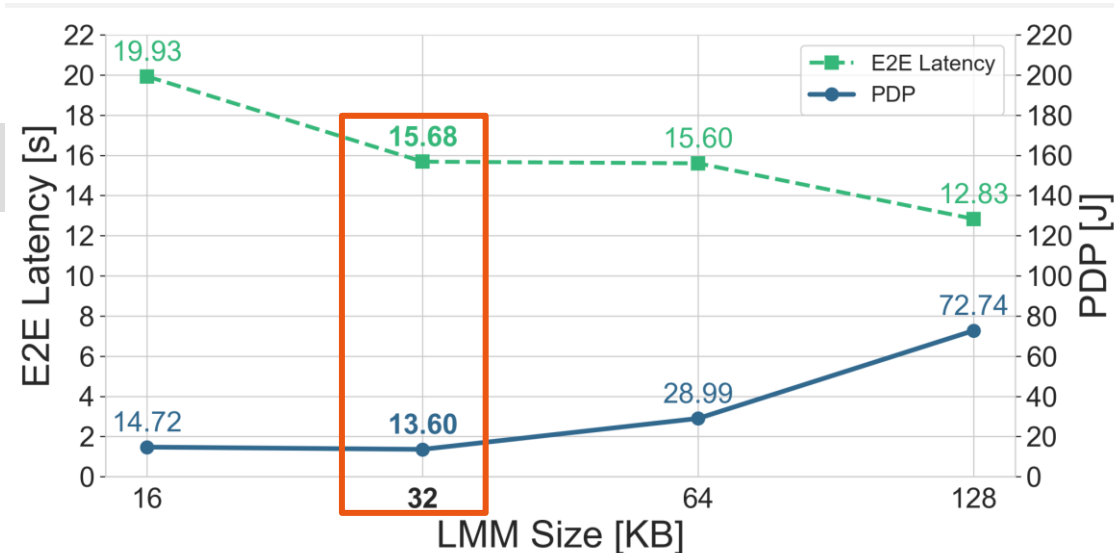
Impact of LMM Size (PDP)

Why did we choose 32KB LMM?

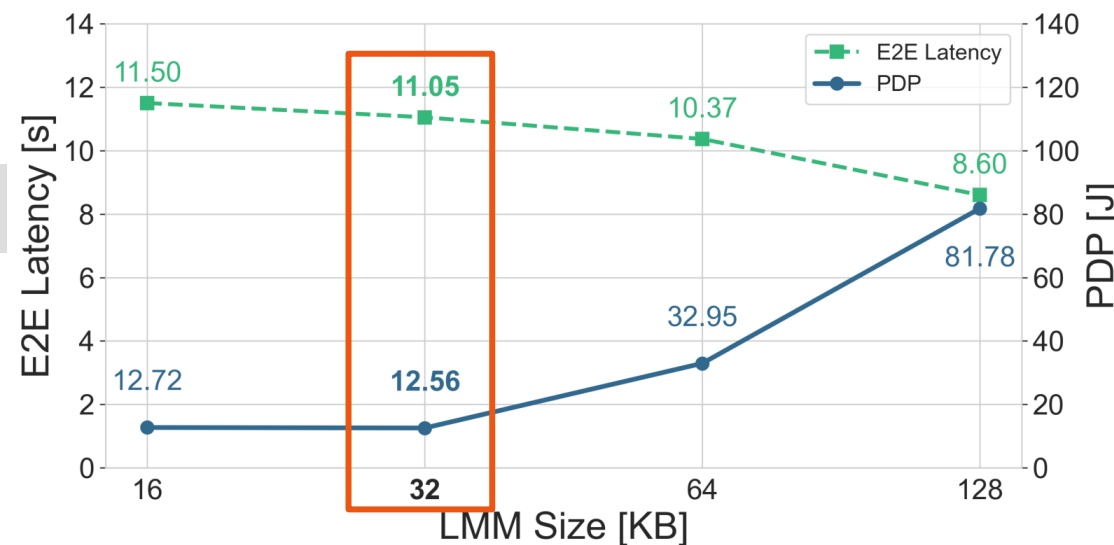
- ▲ **16 KB:** Too small. Data spills to CPU, drastically worsening PDP.
- ▲ **32 KB:** The optimal point with **minimal PDP**.
- ▲ **64~KB:** Static power consumption increases, worsening total PDP.

Point:

Simply increasing LMM(on-chip memory) is not always better.



(a) FP16 model inference



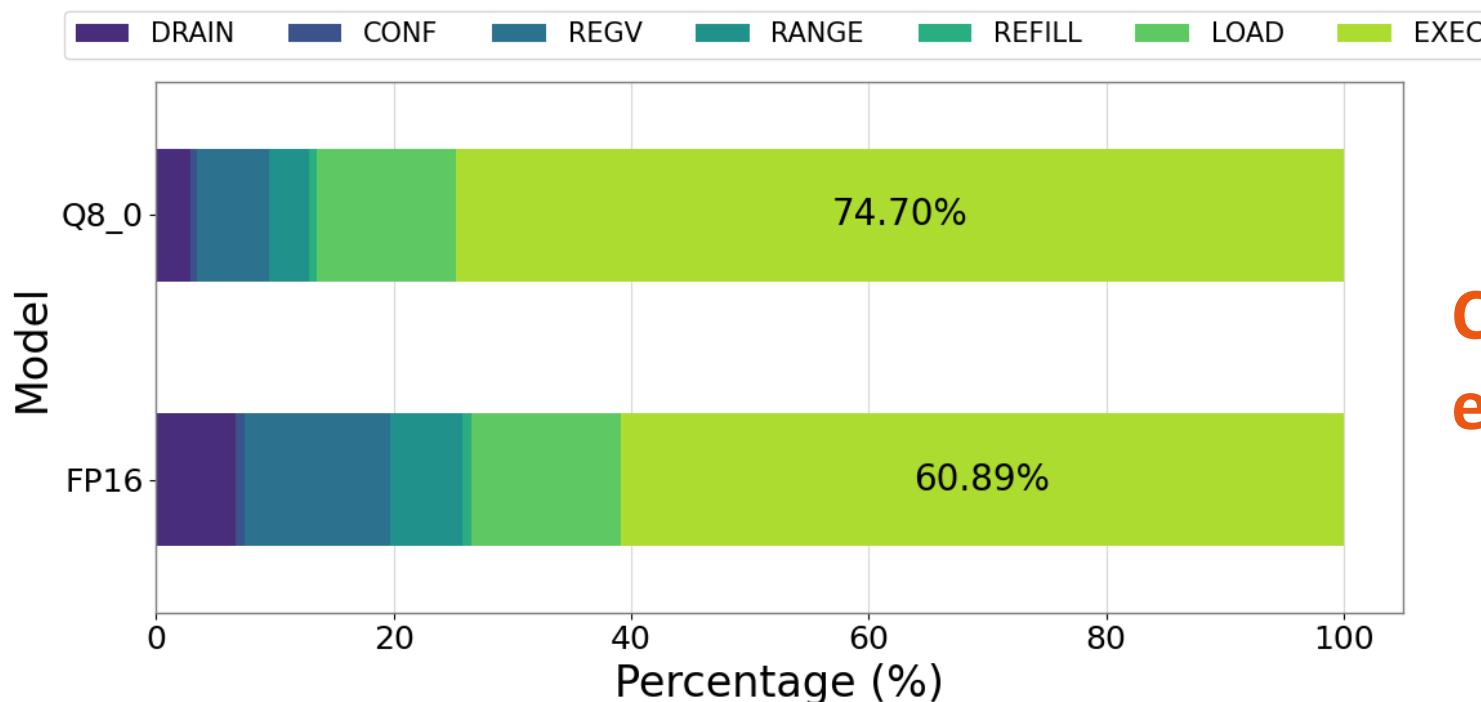
(b) Q8_0 model inference

E2E latency and PDP comparison by LMM size

Compute Efficiency & Scalability

Achieved Compute-Bound Status:

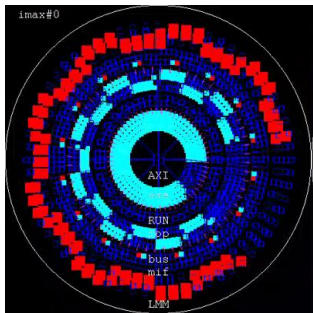
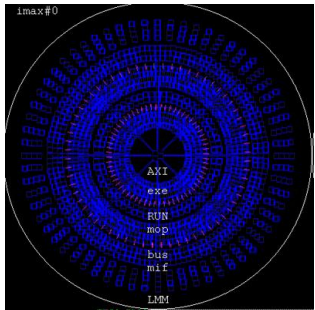
- AI workloads are often *memory-bound* (waiting for data).
- Our implementation: **74.70%** of time is "EXEC" (computation).



Computationally
efficient ! 🤖

Summary

<u>Background</u>	AI's high energy demand requires efficient hardware alternatives to power-hungry GPUs.
<u>Our Approach</u>	Accelerated Whisper ASR on the IMAX CGLA architecture using hardware/software co-design.
<u>Key Results</u>	Achieved superior energy efficiency (28nm ASIC): • 1.9x over Jetson AGX Orin (Edge GPU) • 9.8x over RTX 4090 (High-end GPU)
<u>Conclusion</u>	CGLA is a powerful and sustainable solution for Edge AI .
<u>Future Work</u>	Scale to larger Whisper models. Further enhance kernel optimizations.



IMAX execution visualization DEMO

Thanks!

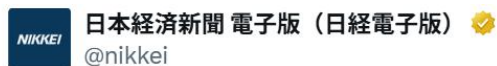


LENZO Page



Lab Page

Do you have any questions?



Show translation

元「プレステ」CPU開発者、NVIDIAの壁に挑む AI半導体会社設立



元ソニー「プレステ2」CPU開発者が挑むNVIDIAの壁 AI半導体会社設立

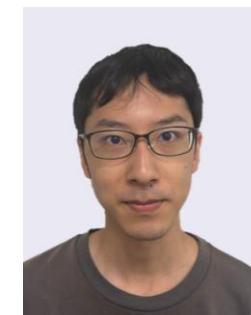
< We at **LENZO** are developing a new architecture "CGLA," looking towards the era of **AI** and **crypto assets**.



Takuto ANDO



Yu ETO



Ayumu TAKEUCHI



Yasuhiko NAKASHIMA