SoftEx: a Low Power and Flexible Softmax Accelerator with Fast Approximate Exponentiation

Andrea Belano*‡, Yvan Tortorella*, Angelo Garofalo*†, Luca Benini*†, Davide Rossi*, Francesco Conti*

*University of Bologna, Bologna, Italy †ETH Zurich, Zurich, Switzerland ‡University of Pavia, Pavia, Italy

MOTIVATION

TRANSFORMER-BASED MODELS

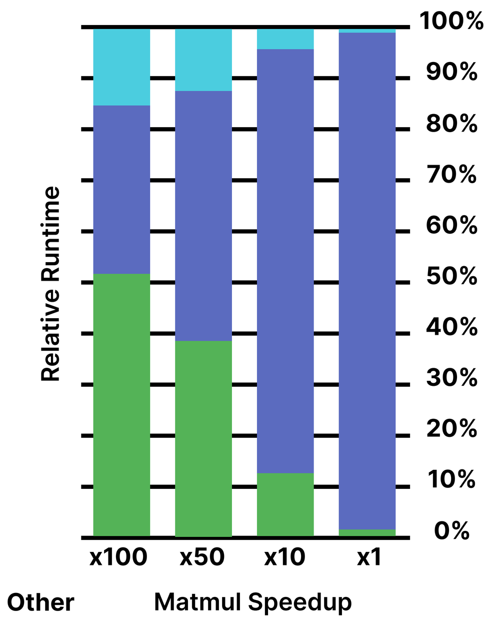
Transformers drive **AI evolution** in perception and generation but at a cost, relying on multi-head self-attention (MHSA), which incorporates **softmax**, and additional projections.

$$Attention(Q,K,V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

WHAT'S THE DEAL WITH SOFTMAX?

The **softmax** function is a **bottleneck**: it's used repeatedly per layer, relies on expensive **exponential** calculations, and its **non point-to-point** nature **complicates acceleration**.

$$\mathrm{Softmax}(x)_i = \frac{e^{x_i - \max(x)}}{\sum_{j=1}^{N} e^{x_j - \max(x)}}$$



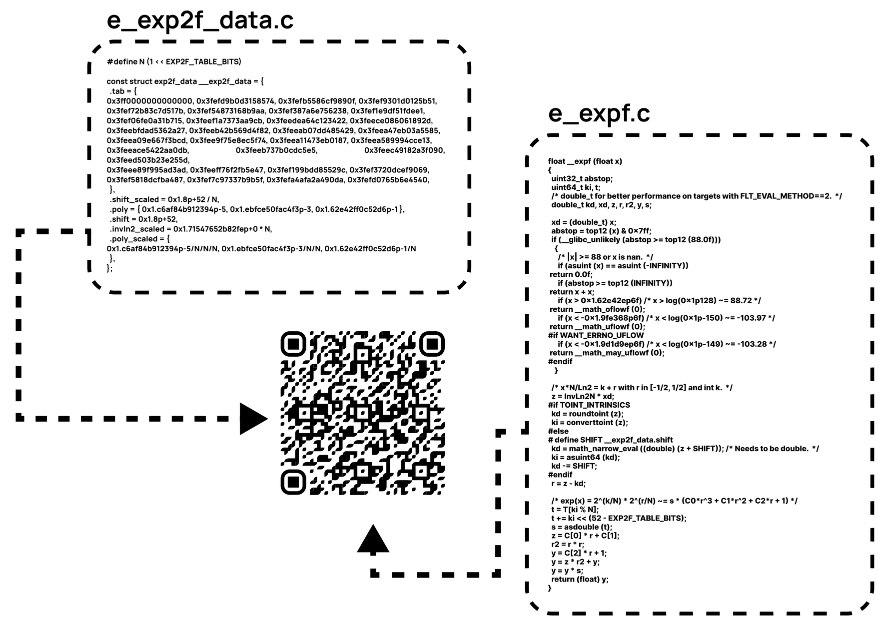
THE PROBLEM WITH EXP

Accurate implementations of exp (e.g., glibc's) are **extremely complex**.
Clearly not suitable for low-power applications, let alone hardware accelerators.
We need an alternative.
What about less computationally-intensive approximations?

CORDIC
Good accuracy and efficient.
Slow convergence.

LUT-based methods
Perform no computation at all besides interpolation.
Costly in terms of area.
Work best with limited ranges.

Polynomial approximations
We can build optimal approximations using Chebyshev's polynomials.
For a good result we must limit the input range.

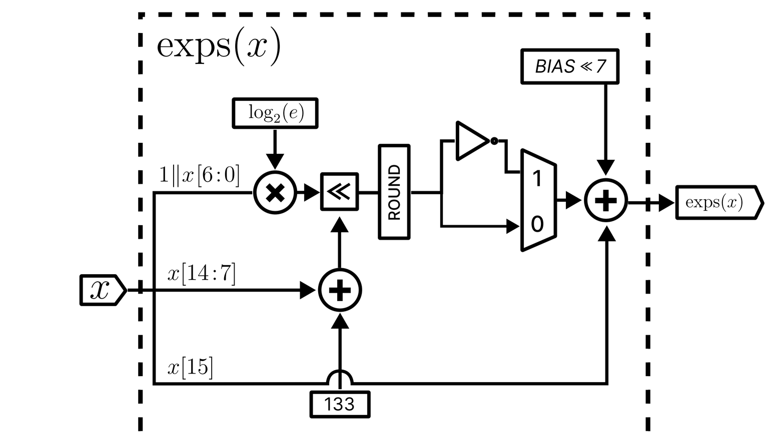


SCHRAUDOLPH'S METHOD

Approximate the base-2 exp of any fixed point number by **exploiting IEEE 754 floating point representation**:

- **Convert** the number to **fixed-point format**.
- Multiply x by a scaling constant to **align its integer part** with IEEE 754 **exponent encoding**.
- **Add the** floating-point **bias** into the integer part of the scaled value.
- **Reinterpret the bits as a float**, yielding an approximate exp(x).

$$\exp(x) = 2^{\frac{x}{\log 2}} = 2^{x'} \approx 2^{\mathrm{int}(x')}\cdot\left(1+\mathrm{frac}(x')\right) \doteq \mathrm{exps}(x)$$



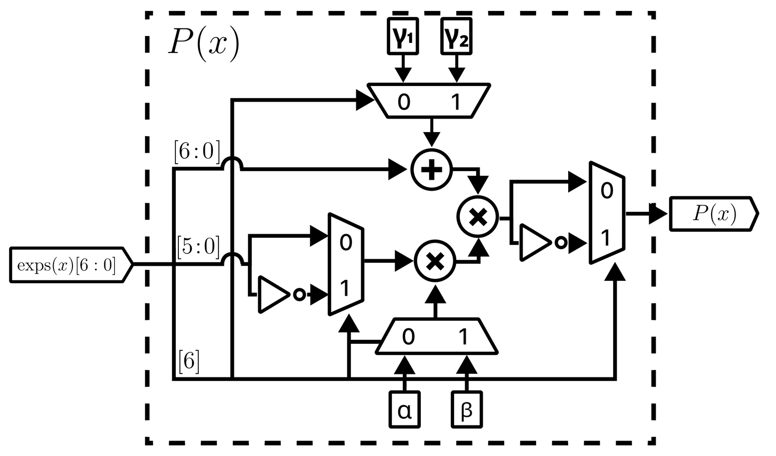
OUR EXP APPROXIMATION

We propose to **replace the fractional part** of x with a **piecewise polynomial** that approximates the function:

For $x \in [0,0.5]$ we sum a **straight-line tangent** to $2x-1$ in 0 with a **parabola** centered in 0.
For $x \in [0.5,1]$ as before, but the functions are **centered in 1**.
By **approximating 1-x with ~x** we get:

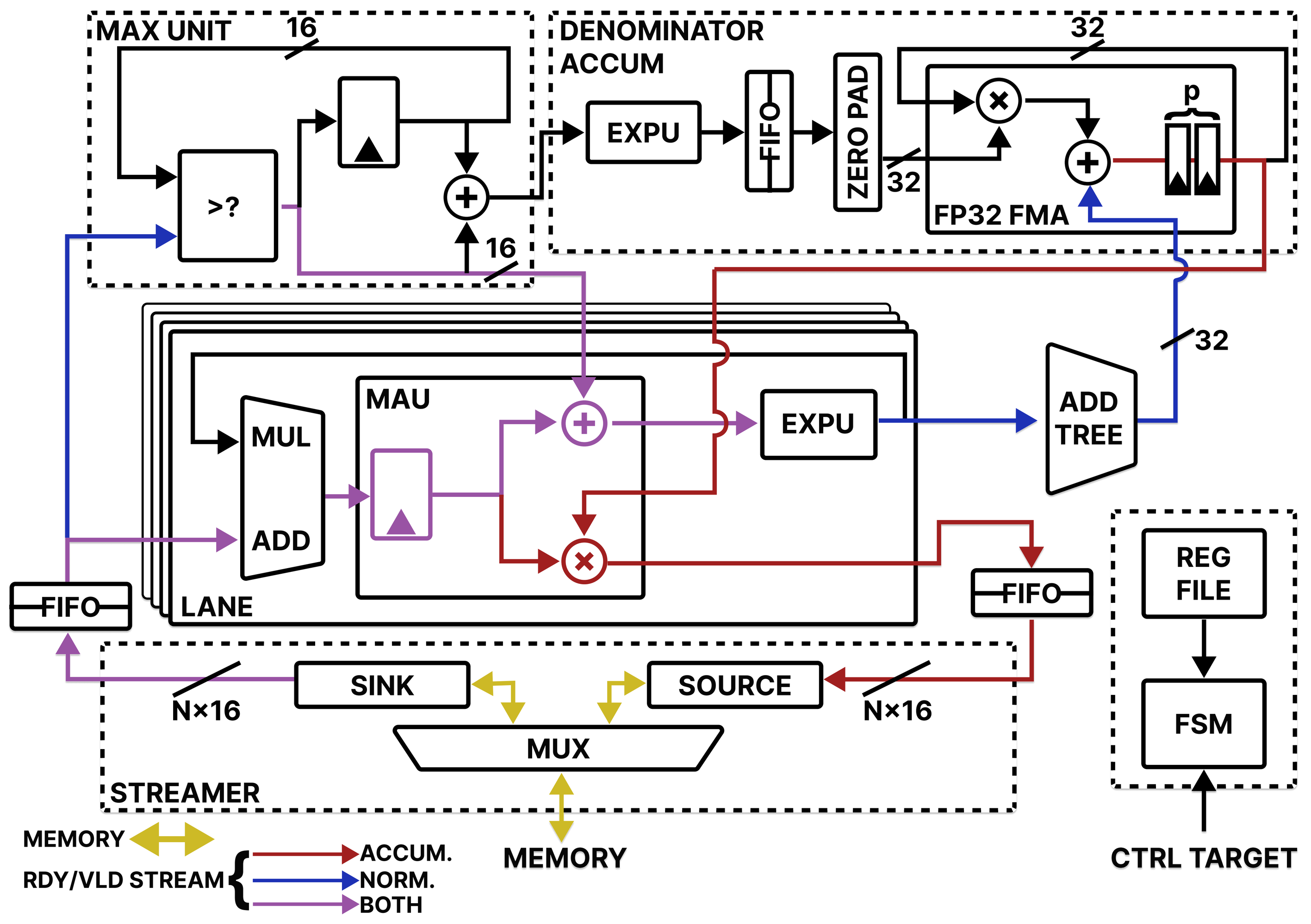
$$P(x) \doteq \alpha x\,(x+\gamma_1), \qquad x\in[0,0.5)$$
$$P(x) \doteq \mathrm{not}\big(\beta\,\mathrm{not}(x)\cdot(x+\gamma_2)\big), \qquad x\in[0.5,1)$$

Very **low bitwidth**: **4 bits** for α and β, **8 bits** for γ_1 and γ_2.
Mean relative error of **0.14%** and a **maximum relative error** of **0.78%**, 13× and 3.7× lower compared to Schraudolph's method.



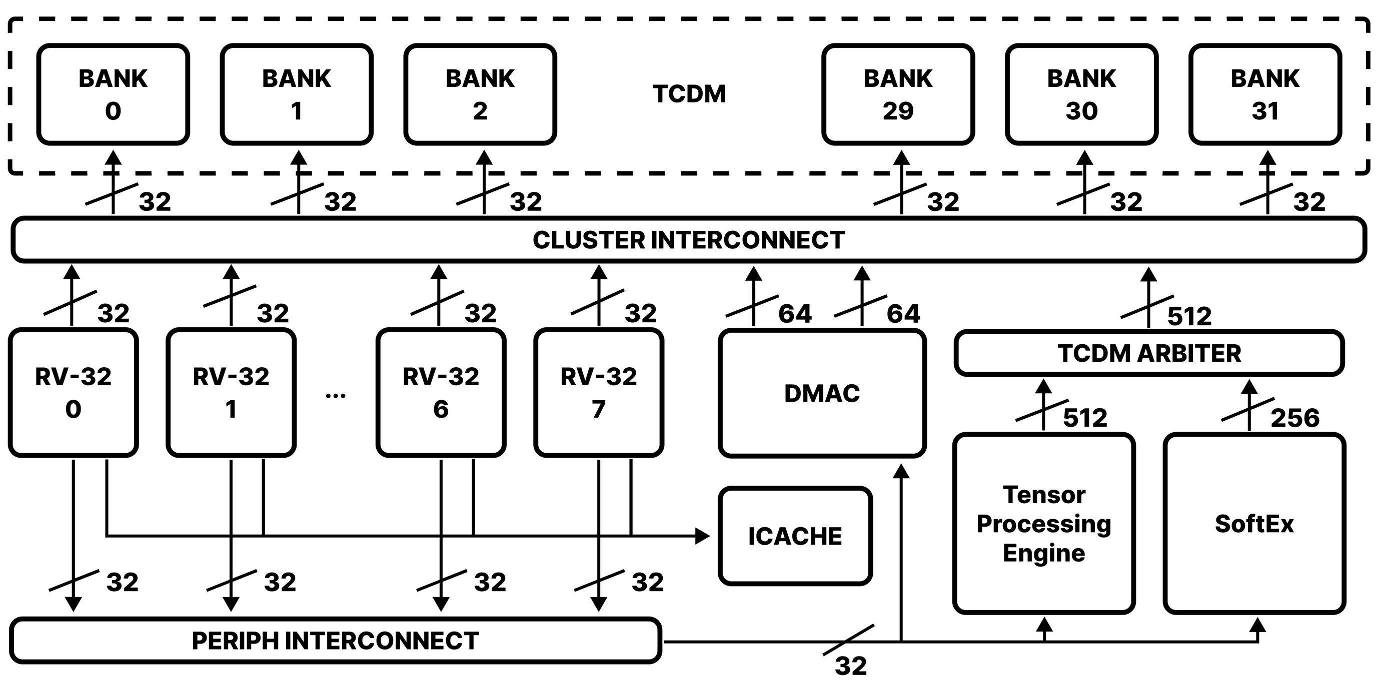
SOFTEX

We introduce **SoftEx**, a **parametric accelerator** designed for **softmax** computation on **BFloat16** vectors. It features N lanes, each containing a **Multiplication and Addition Unit** (MAU) and an **Exponential Unit** (EXPU) that employs our novel exponentiation algorithm. Additionally, it includes an **Accumulator** module with a single pipelined FP32 Fused Multiply-Add (FMA) unit. Softmax processing is divided into three stages: **Accumulation**, **Inversion**, and **Normalization**.



INTEGRATION IN A TRANSFORMER ACCELERATION CLUSTER

Integrated in a **Transformer acceleration cluster** featuring
8 RI5CY RISC-V cores with the Xpulp extension and private FPU
256KiB of **shared scratchpad memory** (TCDM) split among 32 banks
32 KiB of shared instruction cache
RedMulE **Tensor Processing Engine** in 24x8 computing element configuration
SoftEx softmax accelerator in 16 lanes configuration



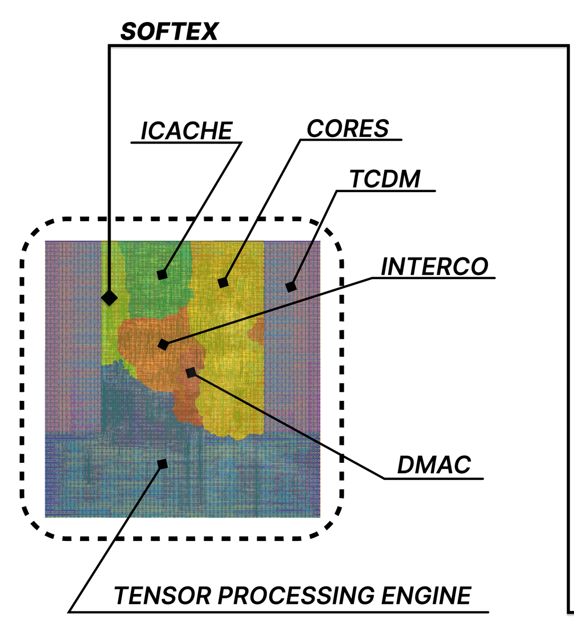
Cluster fully placed and routed in **12 nm technology**.

Two operating points:
- **0.80V** and **1.12 GHz** to maximize **throughput**
- **0.55V** and **460 MHz** to maximize energy **efficiency**

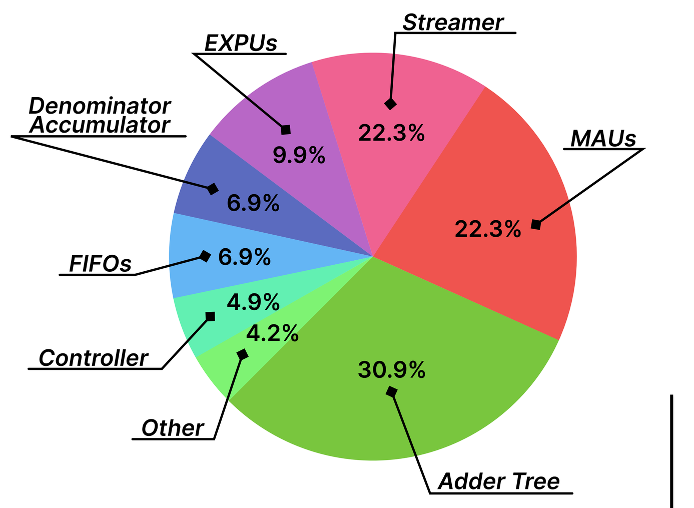
SoftEx **area** utilization: **0.033 mm²**
- **2.75%** of the **1.21 mm² cluster**
- 14% of the Tensor Processing Engine area

SoftEx's area dominated by the Adder Tree (30.9%) and the MAUs (22.3%).
Exponential Units only account for **9.9%** of the total

CLUSTER LAYOUT

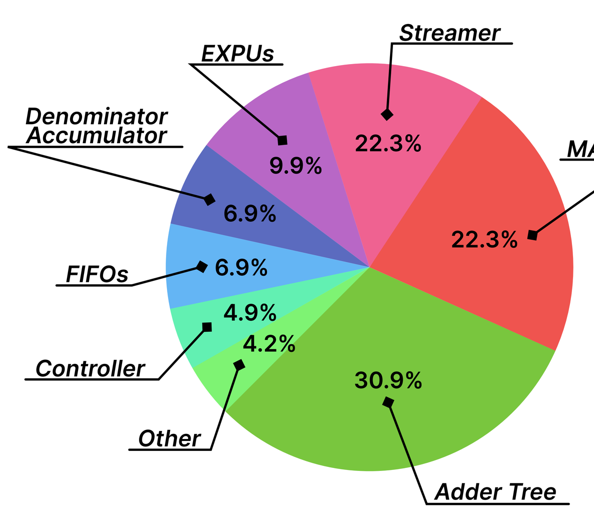


SOFTEX AREA BREAKDOWN



SOFTMAX LATENCY & ENERGY CONSUMPTION

SOFTEX POWER BREAKDOWN

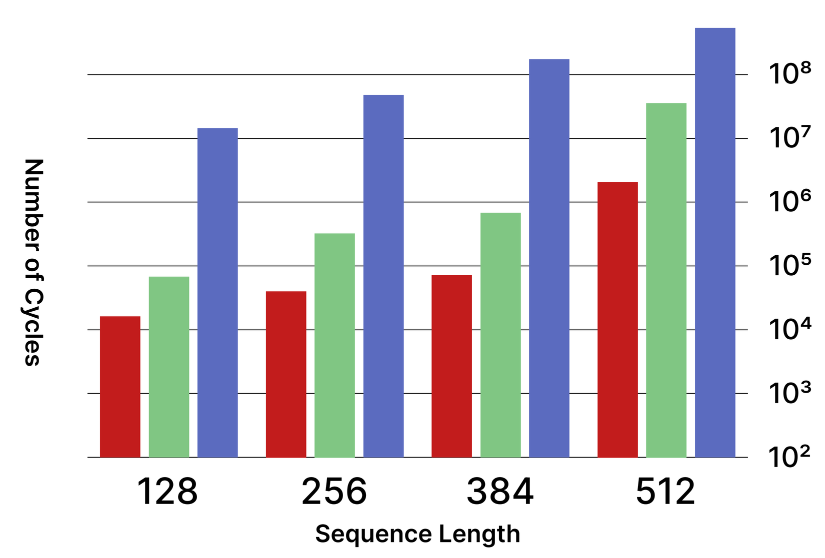


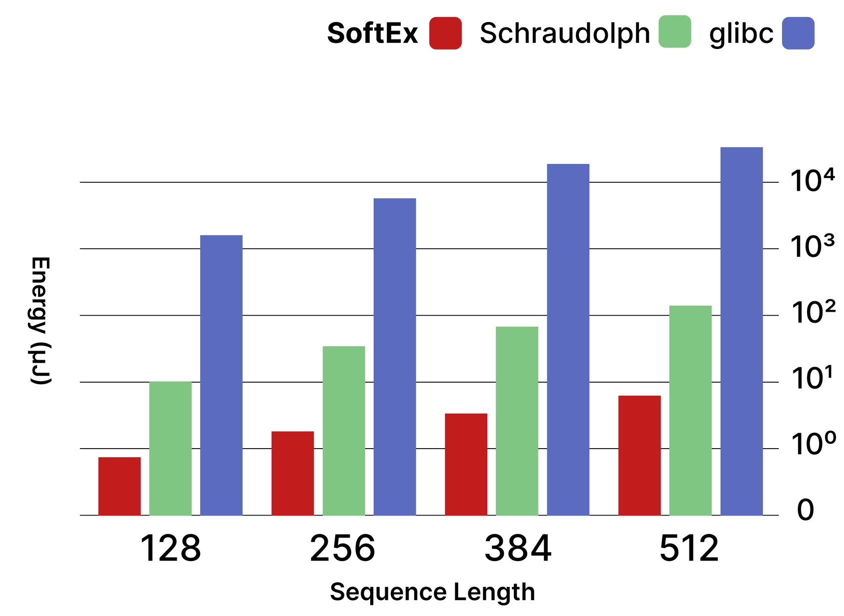
Cluster **power consumption** during softmax: **276 mW** (**49.3 mW** for **SoftEx**, with the rest dominated by the SRAM banks) at **0.80V** or **54.4 mW** at **0.55V** (**8.19 mW** for **SoftEx**).

Inside SoftEx, the MAUs dominate the power consumption (24.9%), followed by the adder tree (23.9%). **Minimal contribution** from the **exponential units** (**8.4%**).

SoftEx softmax performance benchmarked using activations from MobileBERT's attention layer with different sequence lengths.

Up to **10.8× faster softmax** and **26.8× lower energy** compared to the **8 RISC-V cores** employing Schraudolph's method.

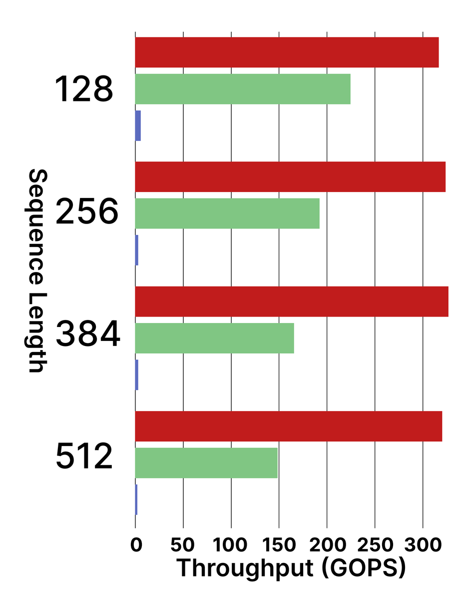




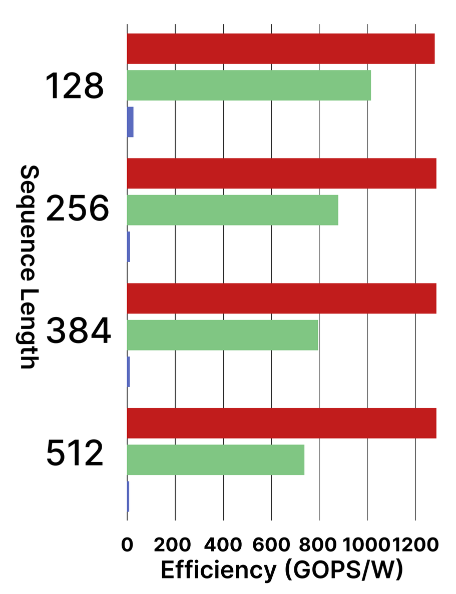
CLUSTER THROUGHPUT @ 0.80V & ENERGY EFFICIENCY @ 0.55V

Benchmarked on **MobileBERT's attention** layer.

With **SoftEx**, at **0.8V** the cluster achieves up to **324 GOPS** (**87%** of the **theoretical peak**). All **software** implementations result in a substantial **throughput loss** (**>2.17× slowdown** for larger sequence sizes), even when targeting the fastest and least precise exps.



In terms of **energy efficiency**, at **0.55V** the system achieves up to **1.30 TOPS/W**, improving this value by **20.5-75.4%** with respect to the **eight RISC-V cores** running Schraudolph exps.



SoftEx Schraudolph glibc

COMPARISON WITH THE STATE OF THE ART

	[Tambe et al.]	ITA [Wiese et al.]	[Keller et al.]	ViTA [Chen et al.]	[Dumoulin et al.]	[Agrawal et al.]	[NVIDIA H100 SXM5]	This Work
Data Format	FP8	INT8	INT8	INT8	INT8	FP16	BF16	BF16
Technology (nm)	12	22	5	28	28	7	4	12
Area (mm2)	4.60	0.991	0.153	2.00	1.48	19.6	814	1.21
Voltage (V)	0.62-1.0	0.65	0.46-1.05	1.05	-	0.55-0.75	-	0.55-0.8
Power (mW)	10-122	132	-	217	18.4	4400-13000	700000	110-581
Frequency (MHz)	77-717	425	152-1760	200	100	1000-1600	1830	460-1120
MAC Units	256	1024	512	512	256	4096	528	192
On-Chip SRAM (KiB)	256	1024	512	512	256	4096	528	192
Peak Throughput (GOPS)	367	870	1800	204	51.2	12800	989400	430
Peak Efficiency (TOPS/W)	3.0	5.49	39.1	0.943	2.78	1.8	1.41	1.61

Corresponding author: andrea.belano2@unibo.it

[Tambe et al.] T. Tambe, J. Zhang et al., "22.9 a 12nm 18.1tflops/W sparse transformer processor with entropy-based early exit, mixed-precision predication and fine-grained power management," in 2023 IEEE International Solid- State Circuits Conference (ISSCC)
[Wiese et al.] P. Wiese, G. ˙Islamo˘glu et al., "Toward Attention-based TinyML: A Heterogeneous Accelerated Architecture and Automated Deployment Flow," Aug. 2024. [Online]. Available: http://arxiv.org/abs/2408.02473
[Keller et al.] B. Keller, R. Venkatesan et al., "A 17–95.6 TOPS/W deep learning inference accelerator with per-vector scaled 4-bit quantization for transformers in 5nm," in 2022 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)
[Chen et al.] C. Chen, L. Li, and M. M. Sabry Aly, "ViTA: A Highly Efficient Dataflow and Architecture for Vision Transformers," in 2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)
[Dumoulin et al.] J. Dumoulin, P. Houshmand, V. Jain, and M. Verhelst, "Enabling Efficient Hardware Acceleration of Hybrid Vision Transformer (ViT) Networks at the Edge," in 2024 IEEE International Symposium on Circuits and Systems (ISCAS)
[Agrawal et al.] A. Agrawal, S. K. Lee et al., "9.1 A 7nm 4-Core AI Chip with 25.6TFLOPS Hybrid FP8 Training, 102.4TOPS INT4 Inference and Workload-Aware Throttling," in 2021 IEEE International Solid-State Circuits Conference (ISSCC)
[NVIDIA H100 SXM5] "NVIDIA H100 Tensor Core GPU Architecture Overview." [Online]. Available: https://resources.nvidia.com/en-us-tensor-core