ETH zürich

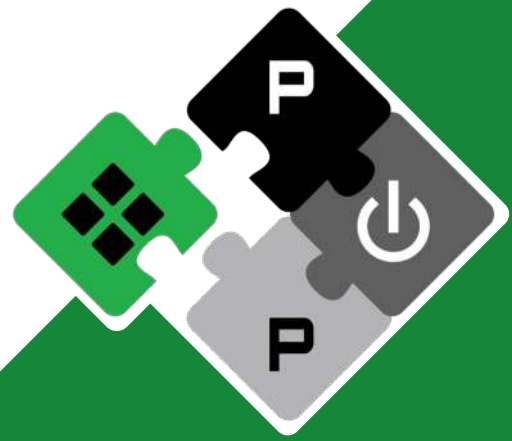
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Next Generation AI: Transformers and Hyper-Dimensional Computing

Integrated Systems Laboratory (ETH Zürich)

Gamze İslamoğlu gislamoglu@iis.ee.ethz.ch

Manuel Eggimann meggimann@iis.ee.ethz.ch



@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

PULP Platform

Open Source Hardware, the way it should be!

ChatGPT isn't so great!

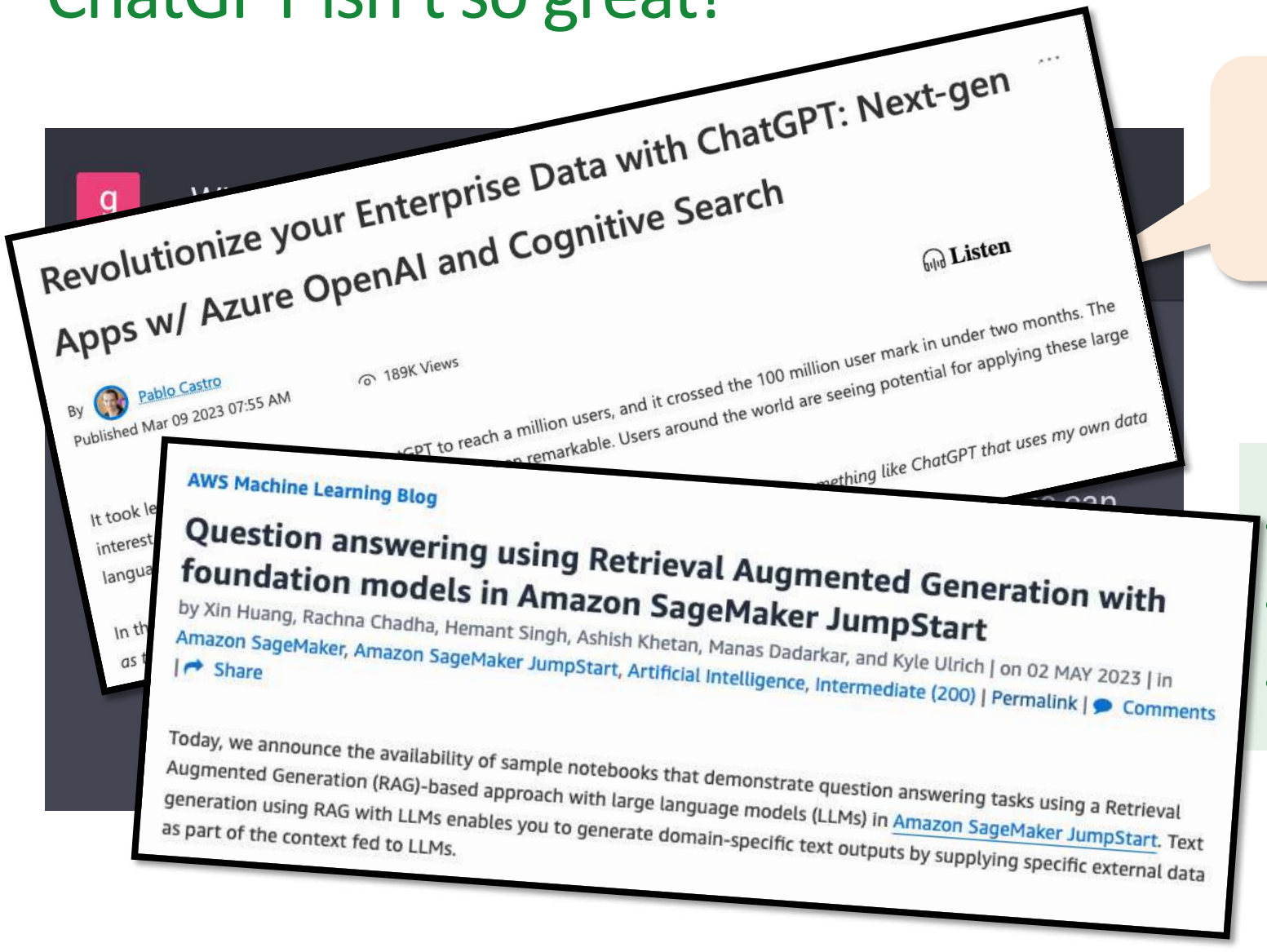
g Who is the leader of the PULP group?

As of my knowledge cutoff in September 2021, Professor Luca Benini was one of the key leaders of the PULP (Parallel Ultra Low Power) project. However, please note that leadership roles can change over time. To obtain the most accurate and up-to-date information on the current leader of the PULP group, I recommend visiting the official PULP project website or reaching out to the project directly for the latest information.

HELP!!!

- **Up-to-date** information
- Knowledge of **your** data
- More challenging **reasoning**

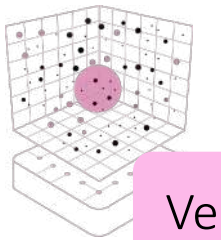
ChatGPT isn't so great!



HELP!!!

- **Up-to-date** information
- Knowledge of **your** data
- More challenging **reasoning**

Retrieval Augmented Generation (RAG)



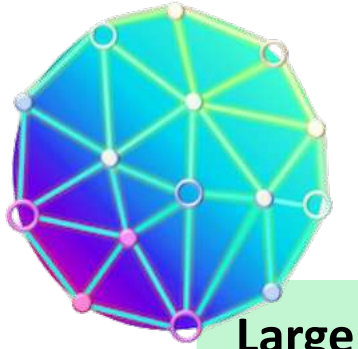
Vector search

Databases

Representation of question (embeddings)

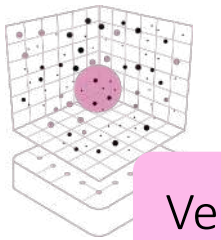
Relevant information for enhanced context

Question



Large Language Models

Retrieval Augmented Generation (RAG)



Vector search

Question

Representation of question (embeddings)

Ultra-efficient HW for next generation AI

Databases

Relevant information for enhanced context

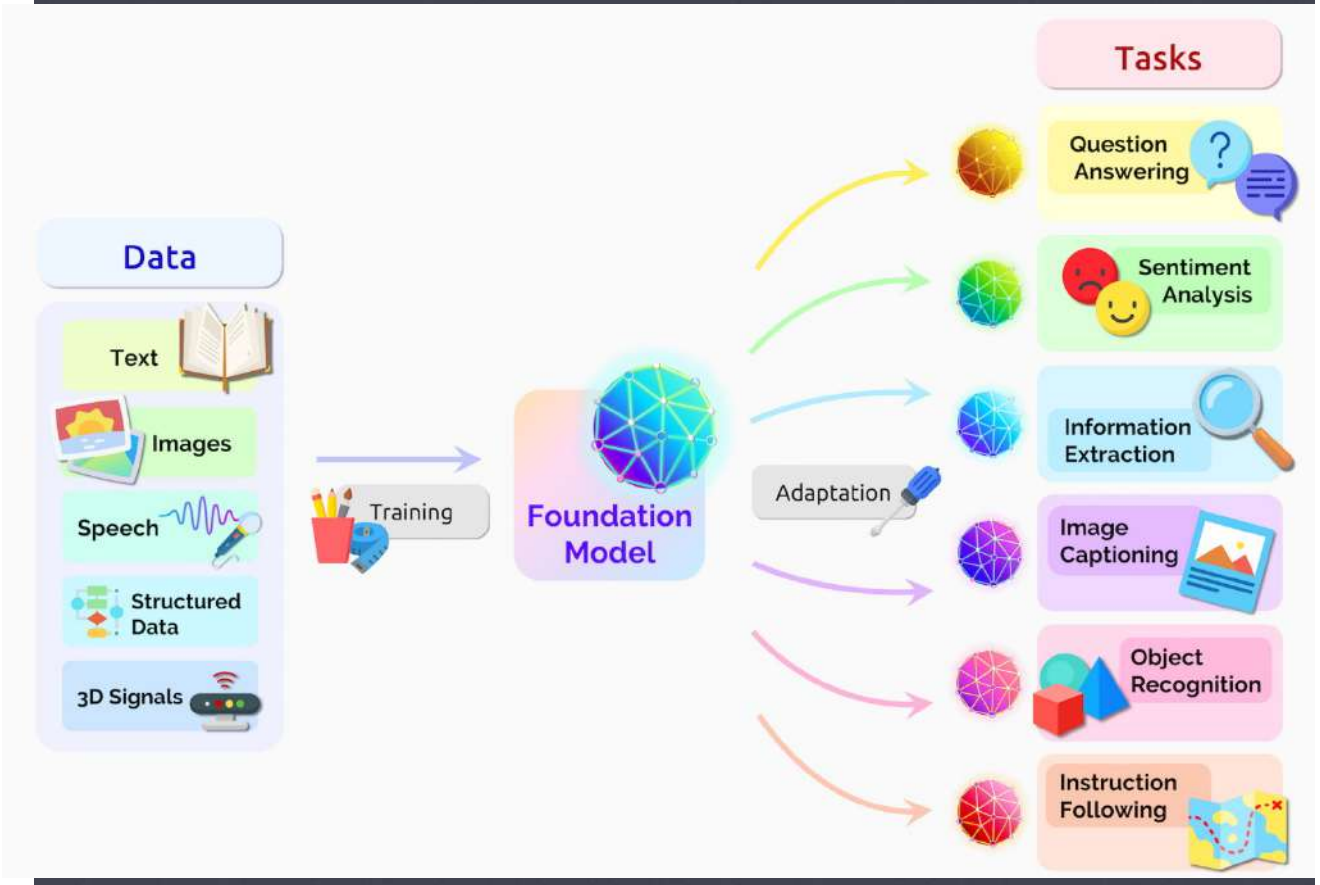
Large Language Models

Hyper-Dimensional Computing

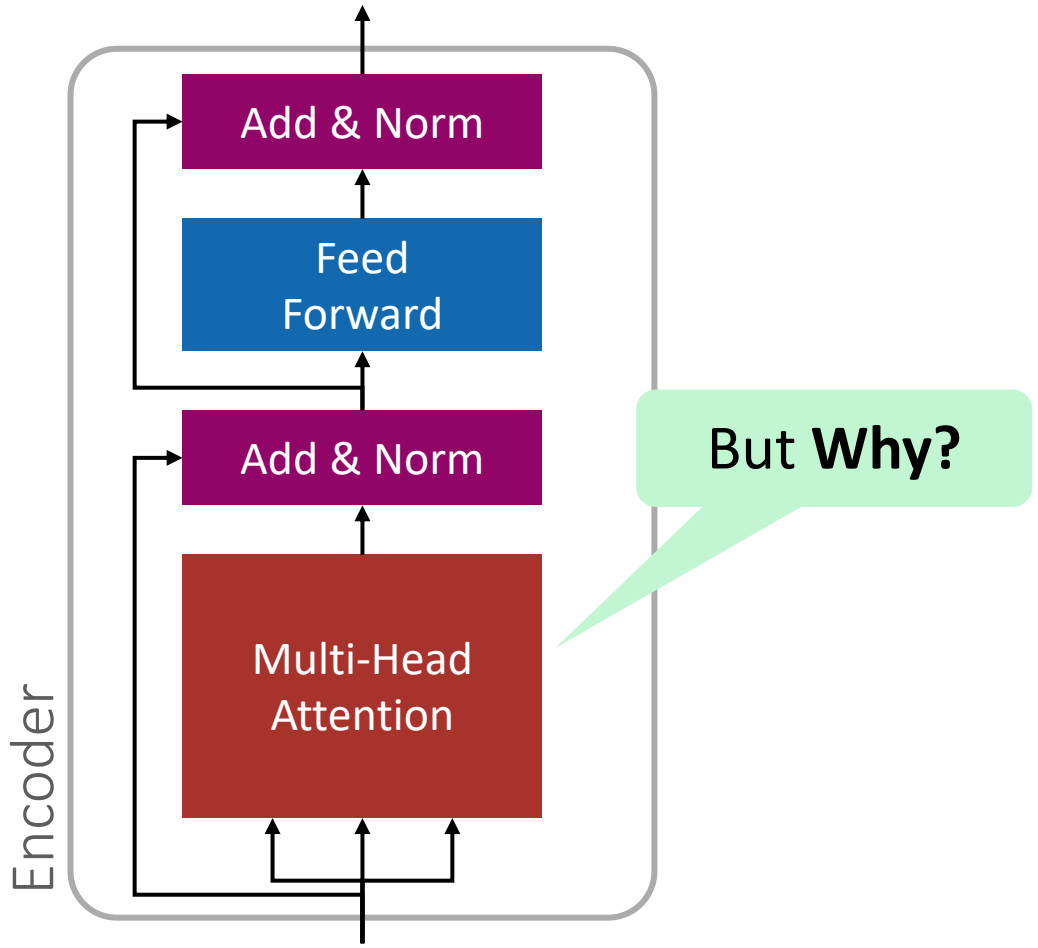
Transformers

Why Transformers?

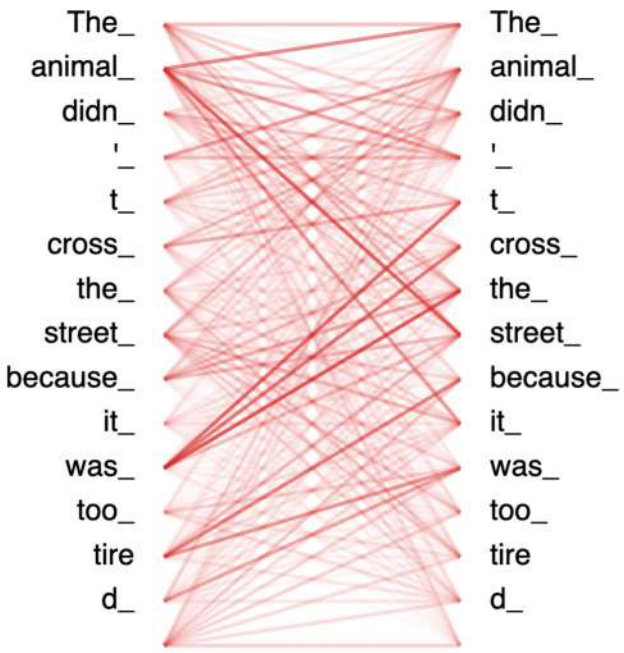
- **Versatility**
  - Natural language processing, computer vision, robotics, biology, …
- **Homogenization of models**
  - Transformers as *foundation models!*
- **Transfer learning**
  - Train on a large-scale dataset and fine-tune on specific tasks with smaller datasets.



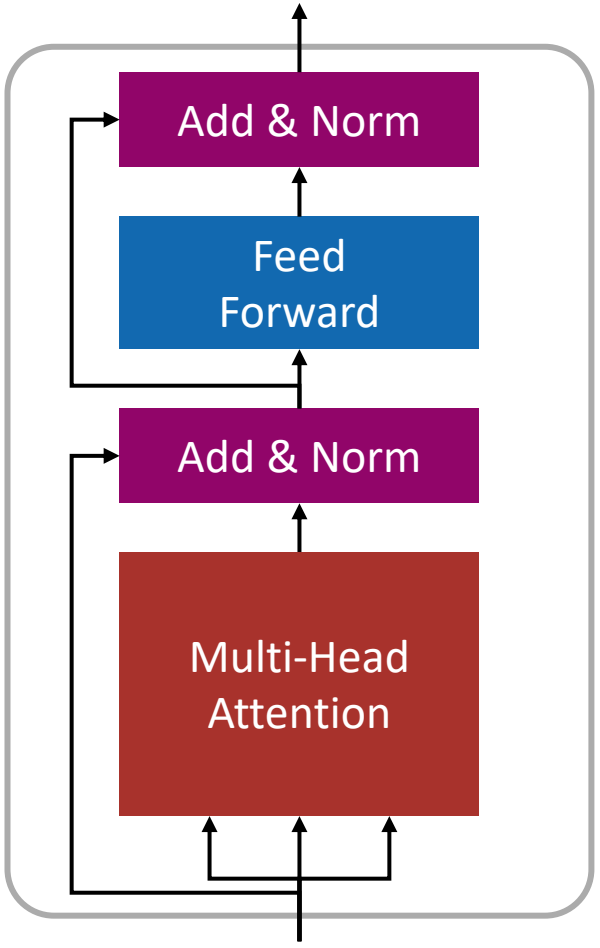
Attention is all you need!

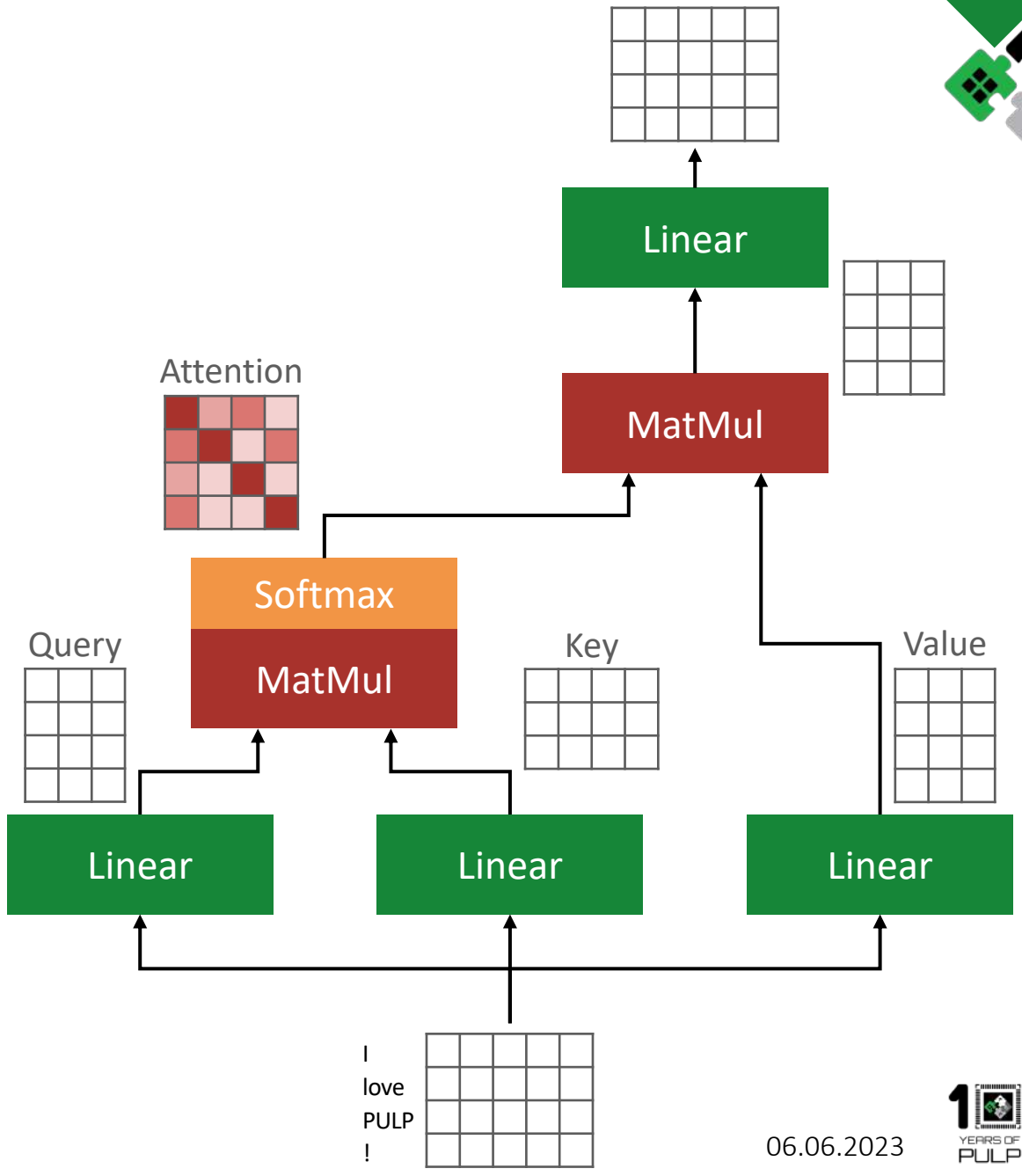


The animal didn't cross the street because it was too tired



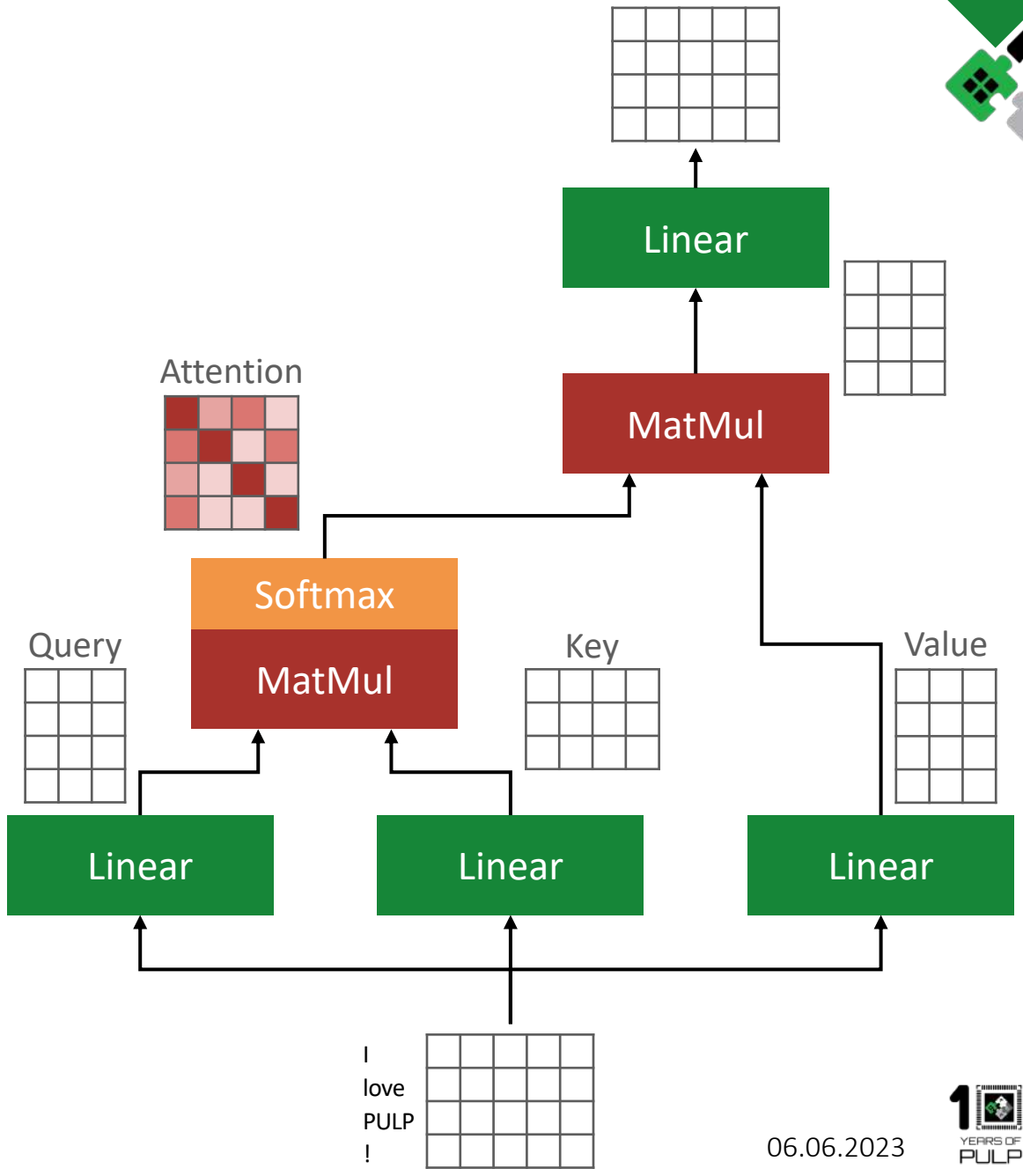
Attention but how?





Challenges in ***Attention***

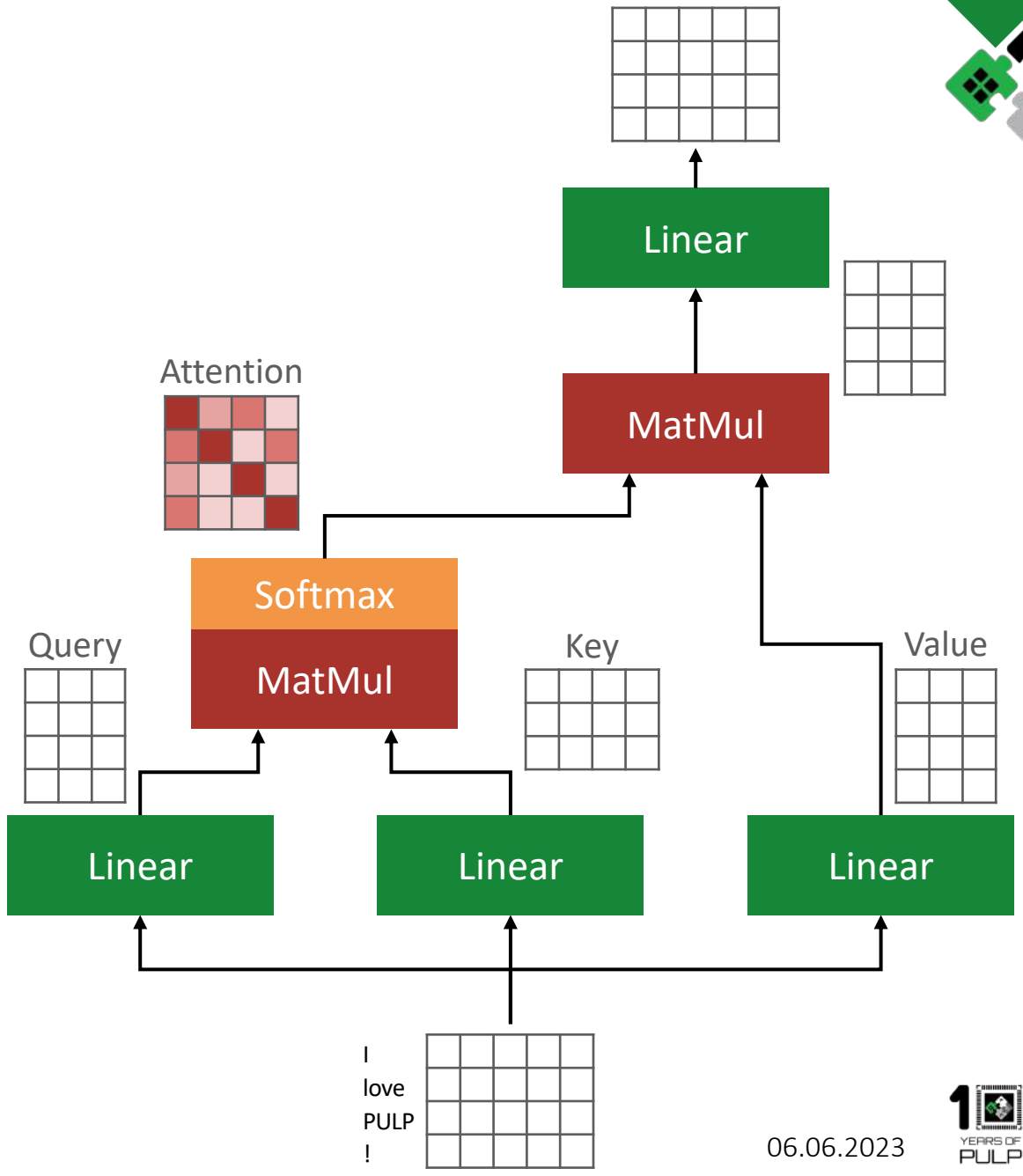
- **Attention matrix is a square matrix of order input length.**
  - ➢ Computational complexity
  - ➢ Memory requirements



Challenges in ***Attention***

- **Attention matrix is a square matrix of order input length.**
  - ➢ Computational complexity
  - ➢ Memory requirements

- **Every attention layer applies *Softmax* to attention matrix!**

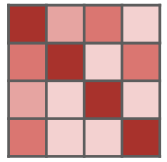


Challenges in ***Attention***

- **Attention matrix is a square matrix of order input length.**
  - ➢ Computational complexity
  - ➢ Memory requirements

- **Every attention layer applies *Softmax* to attention matrix!**
  - 3 passes over a row.
  - Quantization is problematic.

Attention



Softmax

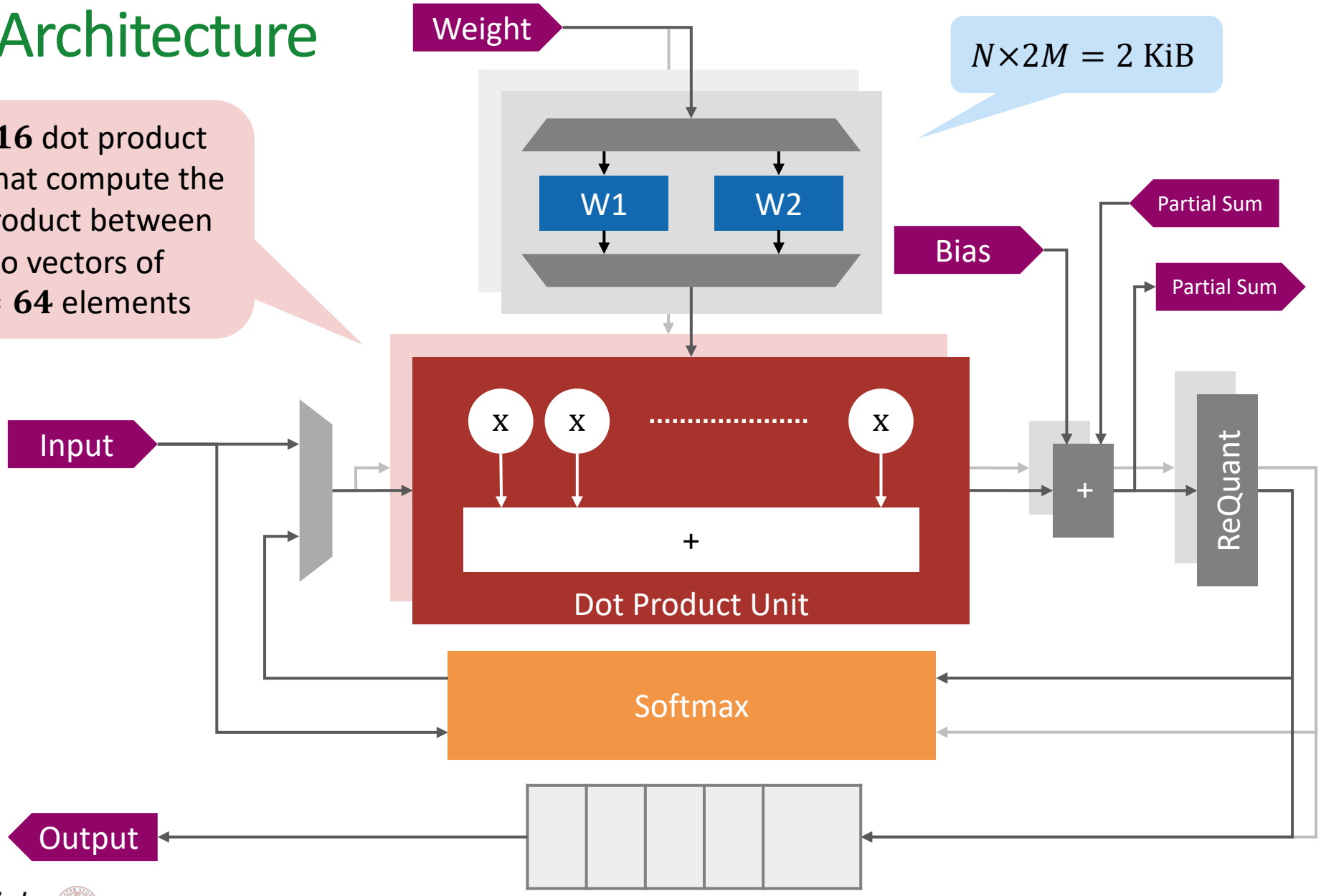
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{e^{x_i - \max(\mathbf{x})}}{\sum_j^n e^{x_j - \max(\mathbf{x})}}$$

ITA: **I**nteger **T**ransformer **A**ccelerator

- ***Attention* accelerator for transformers!**
- **INT8 quantized networks**
- **Output stationary - Local weight stationary**
  - Spatial input reuse
  - Spatial output partial sum reuse
- **Special *Softmax* unit!**

ITA - Architecture

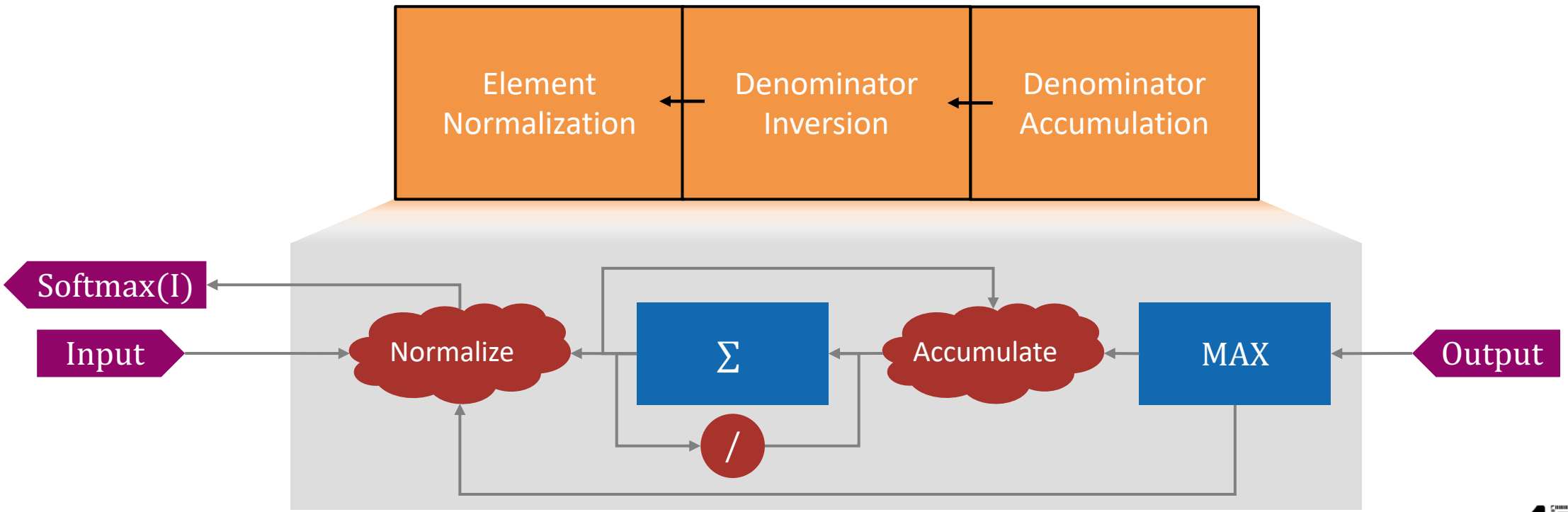
$N = \mathbf{16}$ dot product units that compute the dot product between two vectors of $M = \mathbf{64}$ elements



Hardware-friendly ***Softmax***

$$\text{Softmax}(\mathbf{x})_i = \frac{e^{x_i - \max(\mathbf{x})}}{\sum_j^n e^{x_j - \max(\mathbf{x})}}$$

$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x_q}))\gg 5}} 2^{(x_{qi} - \max(\mathbf{x_q}))\gg 5}$$



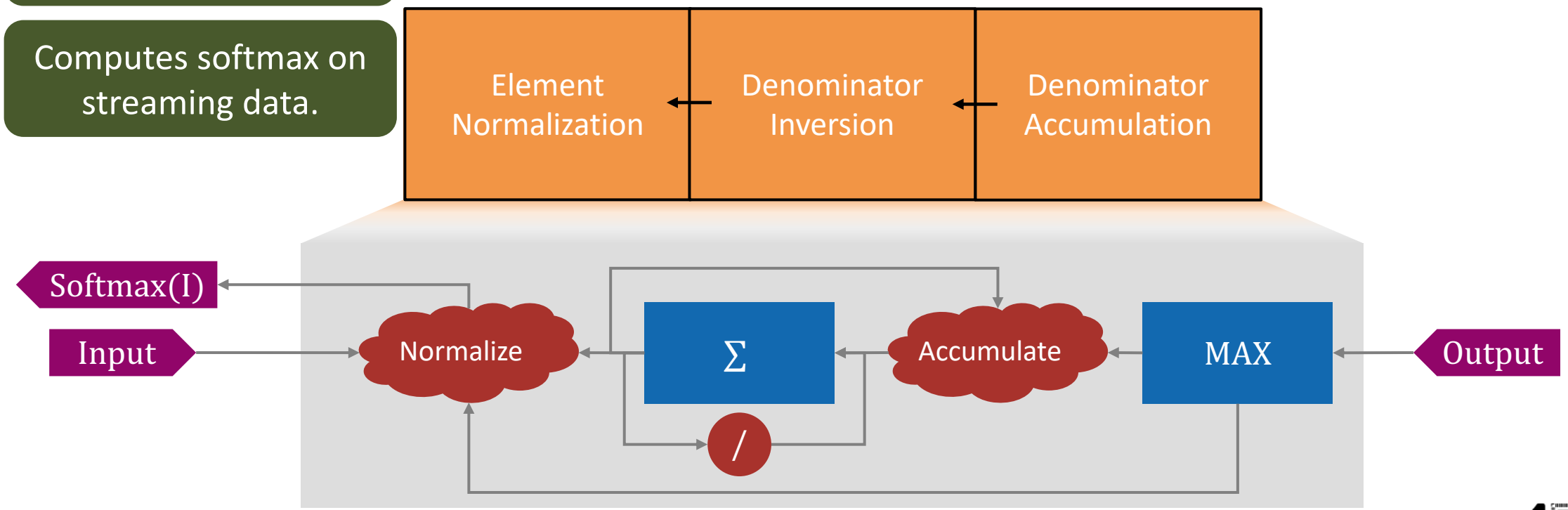
Hardware-friendly ***Softmax***

Directly operates on quantized values.

No exponentiation modules and multipliers.

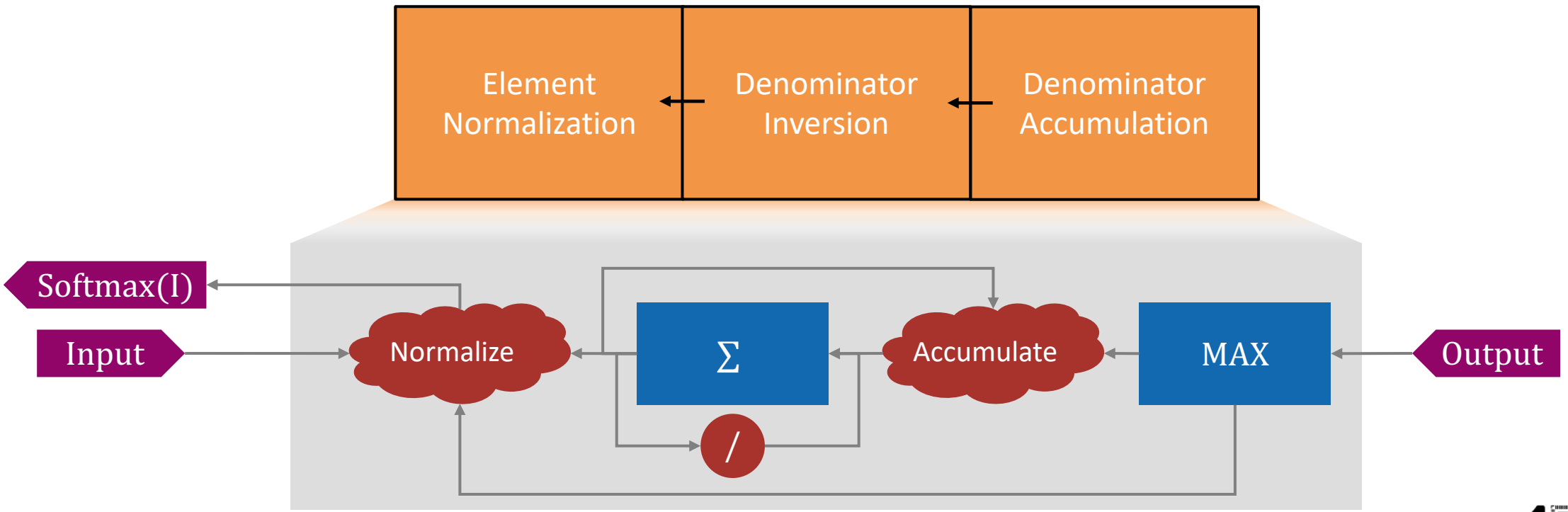
Computes softmax on streaming data.

$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q))\gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q))\gg 5}$$

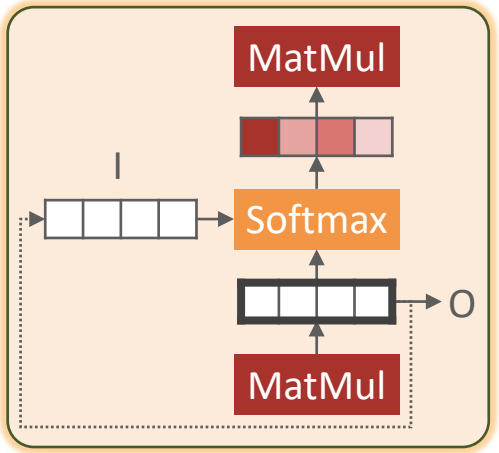


Hardware-friendly ***Softmax***

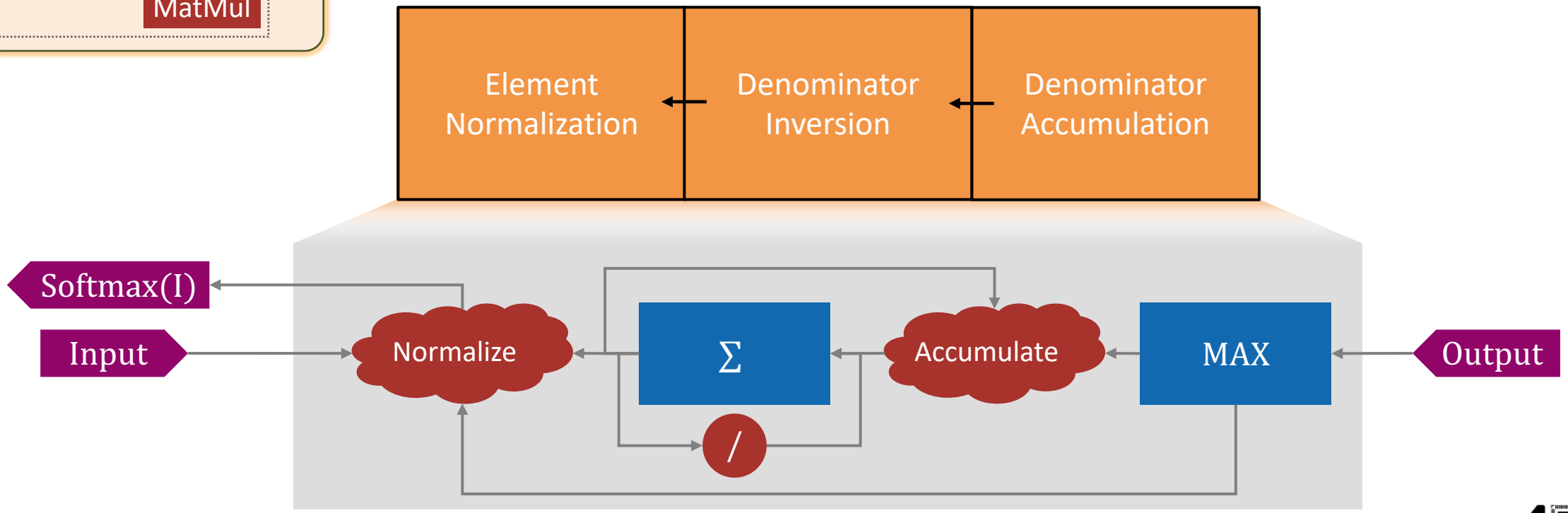
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



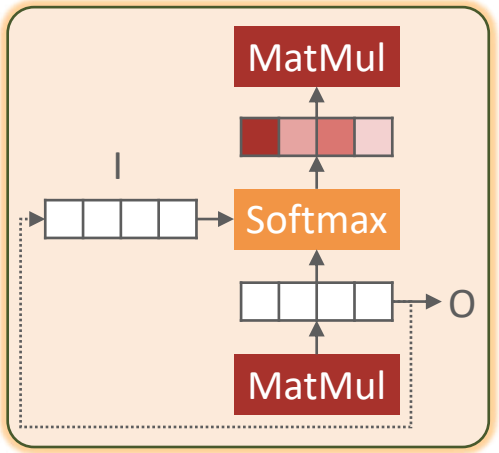
Hardware-friendly ***Softmax***



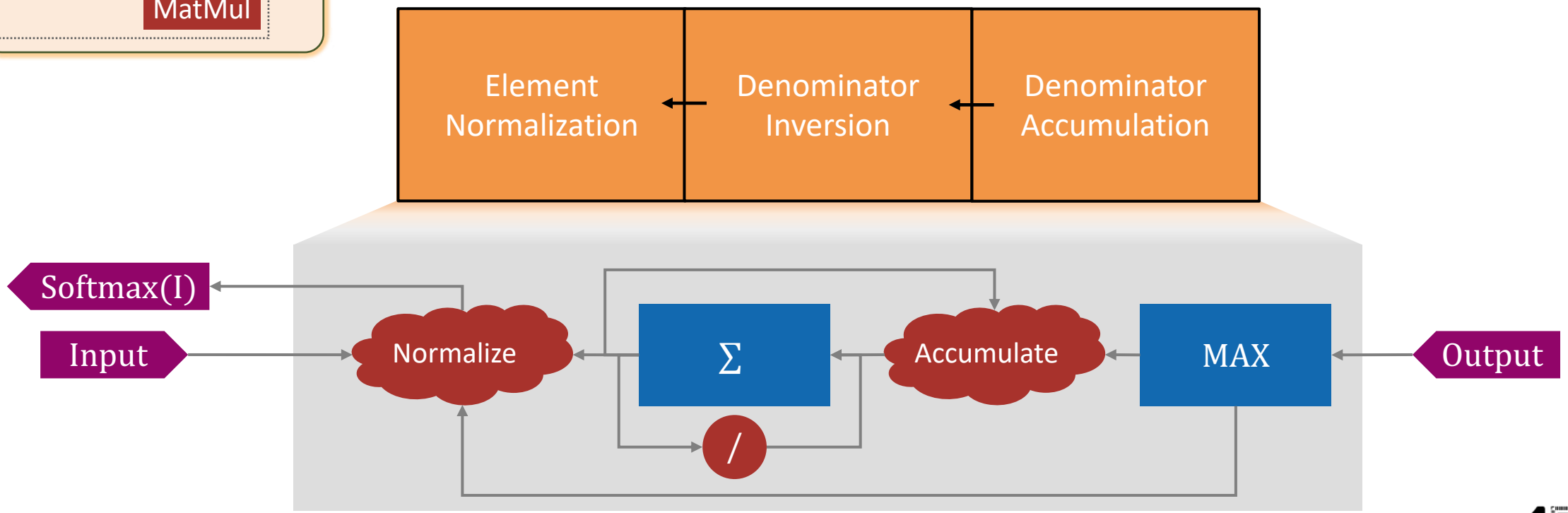
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



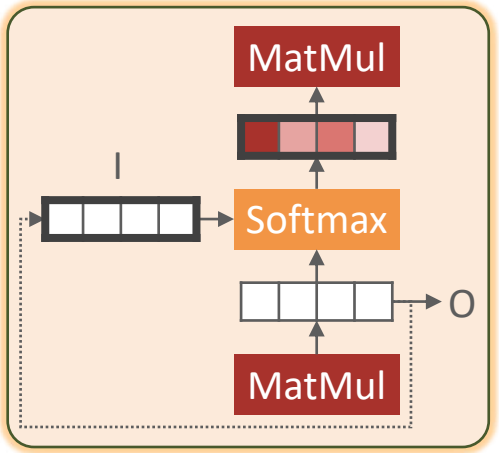
Hardware-friendly ***Softmax***



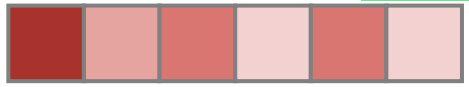
$$\mathrm{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$

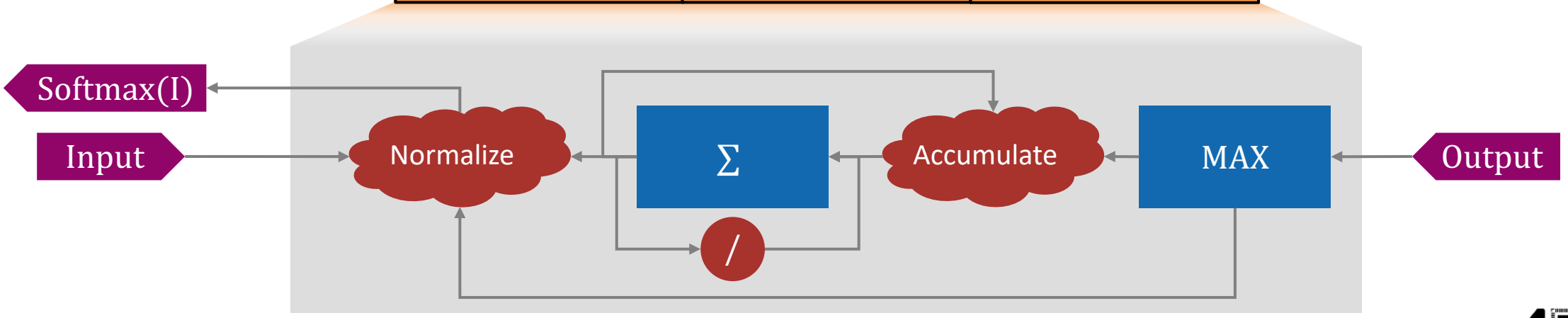


Hardware-friendly ***Softmax***



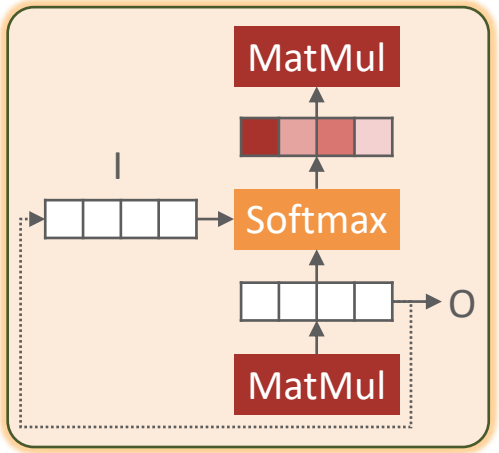
$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q)) \gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q)) \gg 5}$$



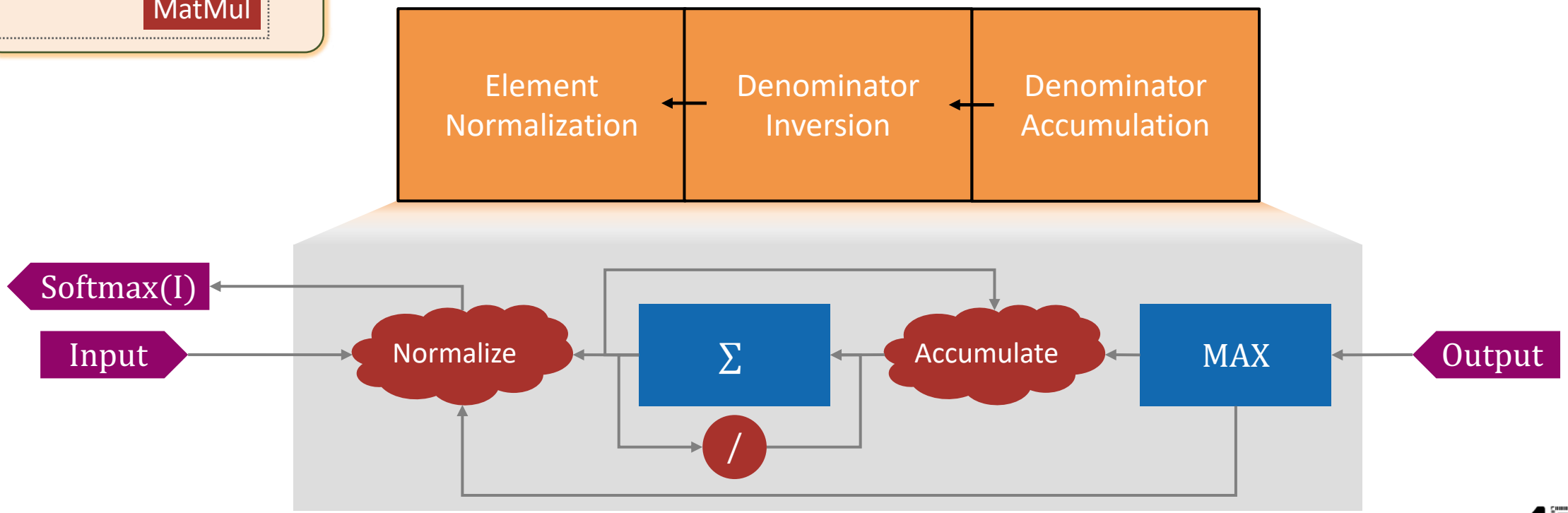


Hardware-friendly ***Softmax***

$MAE = 0.46\%$

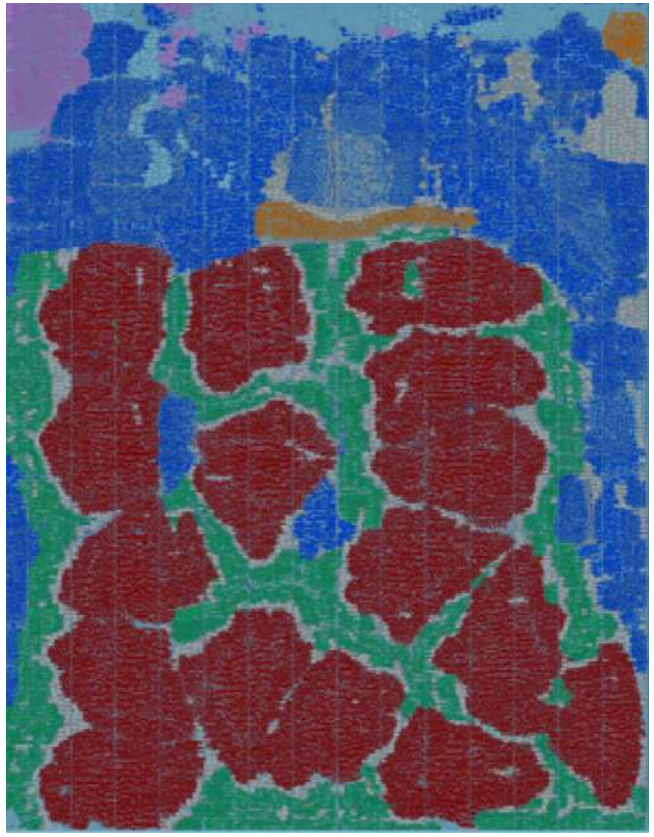


$$\text{Softmax}(\mathbf{x})_i = \frac{1}{\sum_j^n 2^{(x_{qj} - \max(\mathbf{x}_q))\gg 5}} 2^{(x_{qi} - \max(\mathbf{x}_q))\gg 5}$$



Physical Implementation

- **Implemented in GF22**
  - Target frequency of 500 MHz (SS/0.72V/125˚C)
- **Area of 0.17 mm²**
  - *Softmax* module has only **3.3%** area contribution, corresponding to 28.7 KGE.
- **Power of 60 mW** (TT/0.80V/25˚C)
  - *Softmax* module consumes **1.4%** of the power.



Comparison to State-of-the-Art

	Wang et al. *ISSCC'22*	**Keller et al.** *VLSI-TC'22*	**ITA**
Technology [nm]	28	5	22
Voltage [V]	0.56-1.1	0.46-1.05	0.8
Frequency [MHz]	50-510	152-1760	500
Data formats	INT8	INT8	INT8
Throughput [TOPS]	0.52-4.07*	1.8	1.02
Energy efficiency [TOPS/W]	1.91-27.56*	**39.1**	**51.1**
Area efficiency [TOPS/mm^2]	0.076-0.597*	**11.7**	5.93
Area efficiency [TOPS/MGE]	0.025-0.192*	0.121	**1.18**

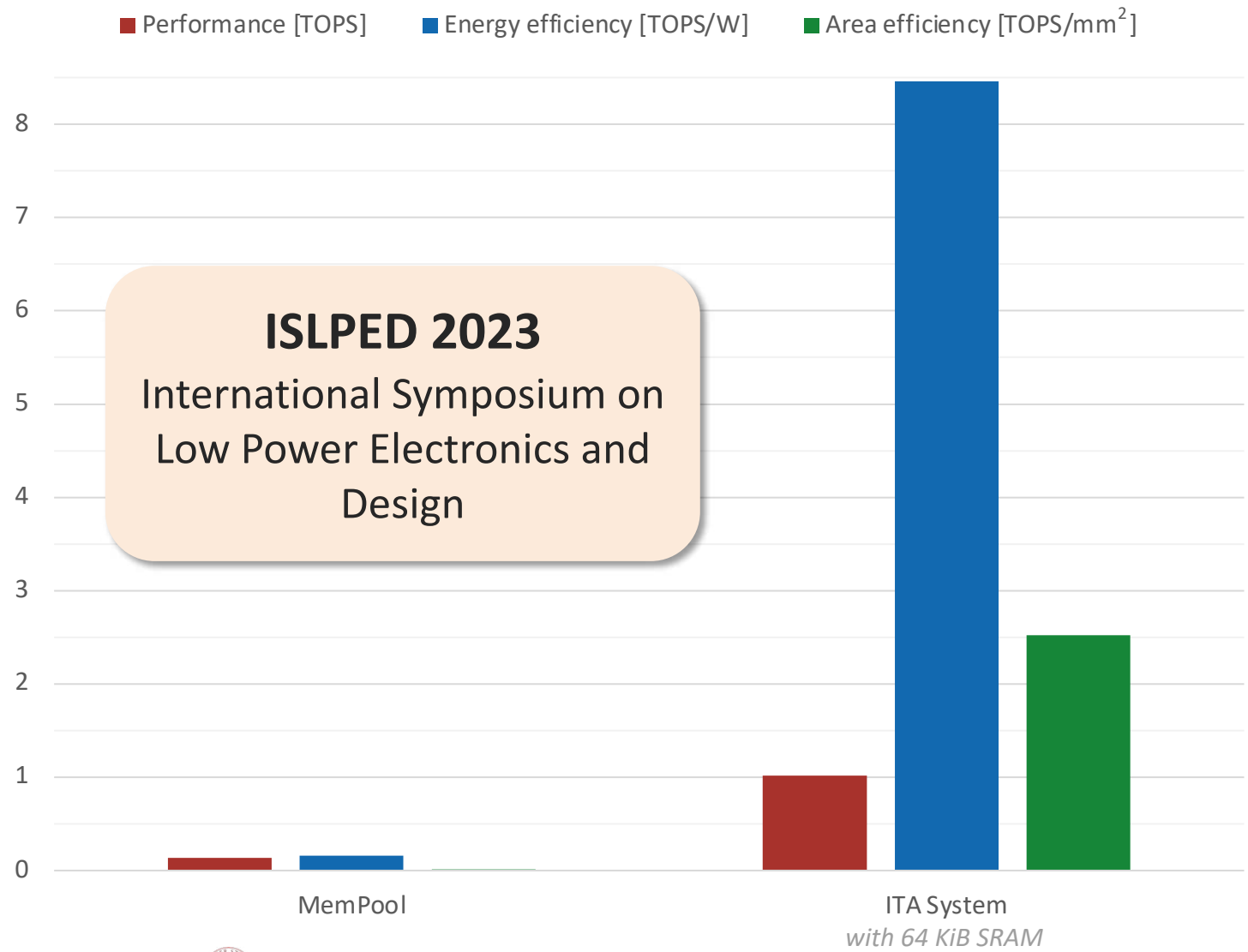
1.3x

6.1x

9.8x

*90% sparsity.

Comparison to a software baseline on MemPool



ISLPED 2023
International Symposium on Low Power Electronics and Design

Performance increase of **7.5x**

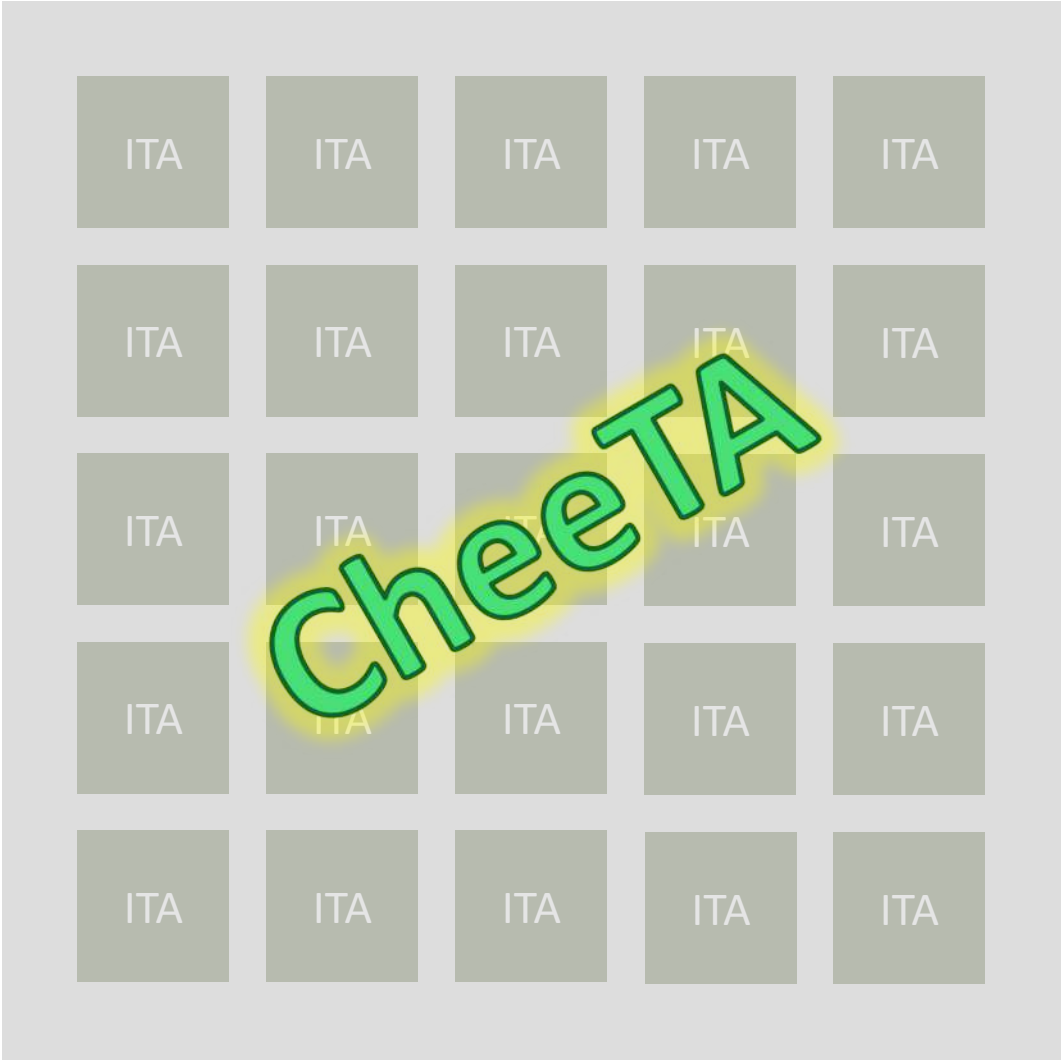
Energy Efficiency increase of **53x**

Area Efficiency increase of **220x**

Future of ITA: Scaling up further

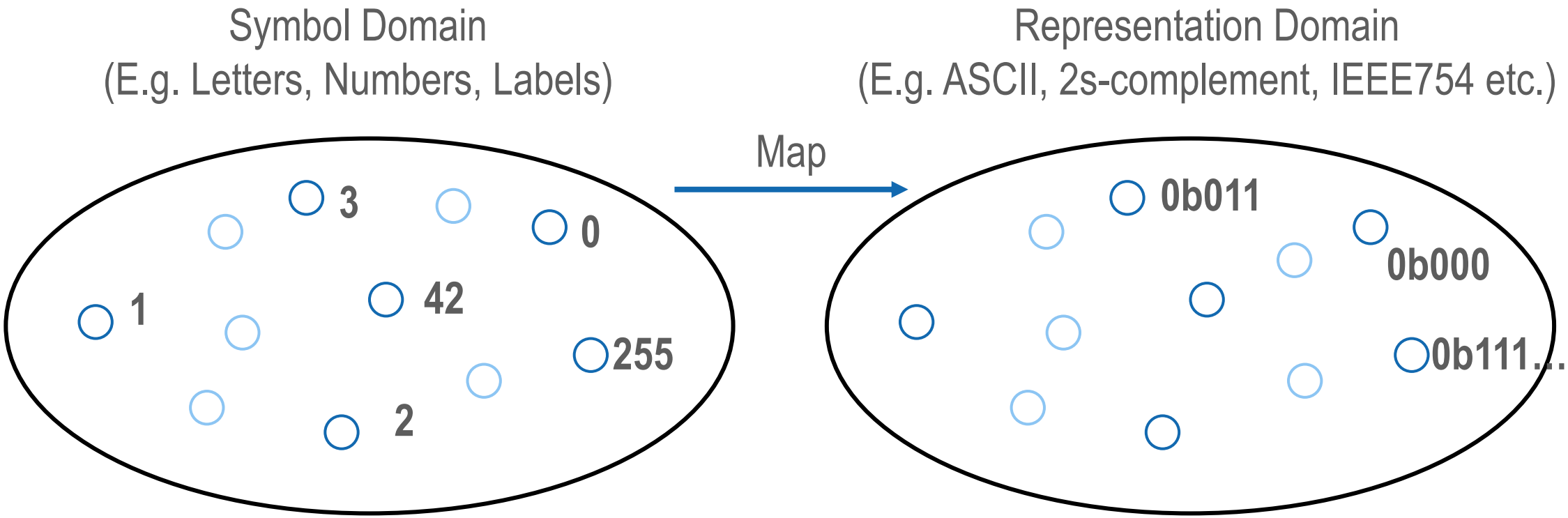
- Target workloads like GPT
- Floating-point capability

Accelerate **LLMs** and reach **100 TFLOPS or higher!**



A Primer on Hyper-Dimensional Computing

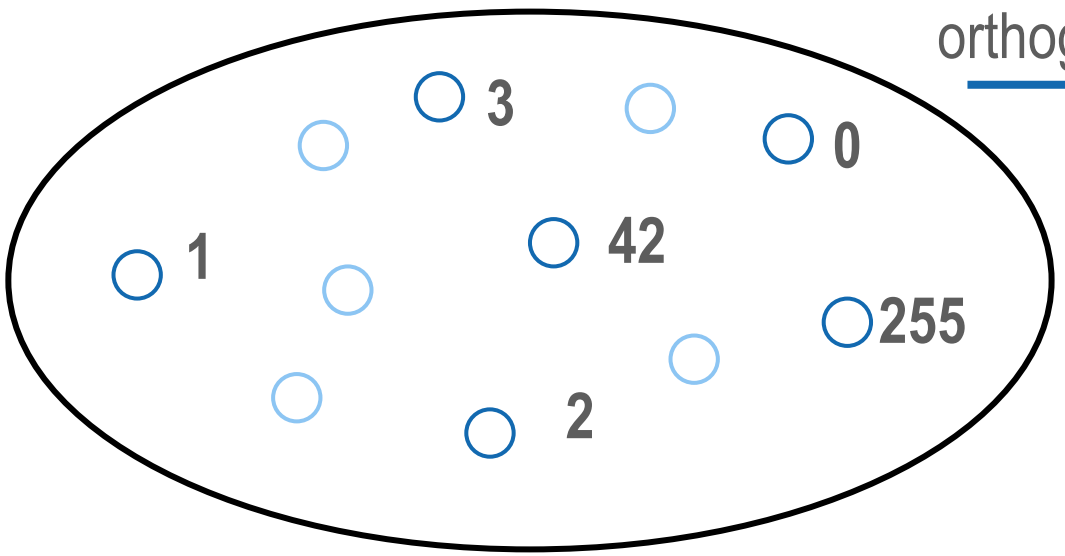
Mapping Symbols to Digital Representations (The conventional Way)



A Primer on Hyper-Dimensional Computing

Mapping Symbols to Digital Representations (The HDC Way)

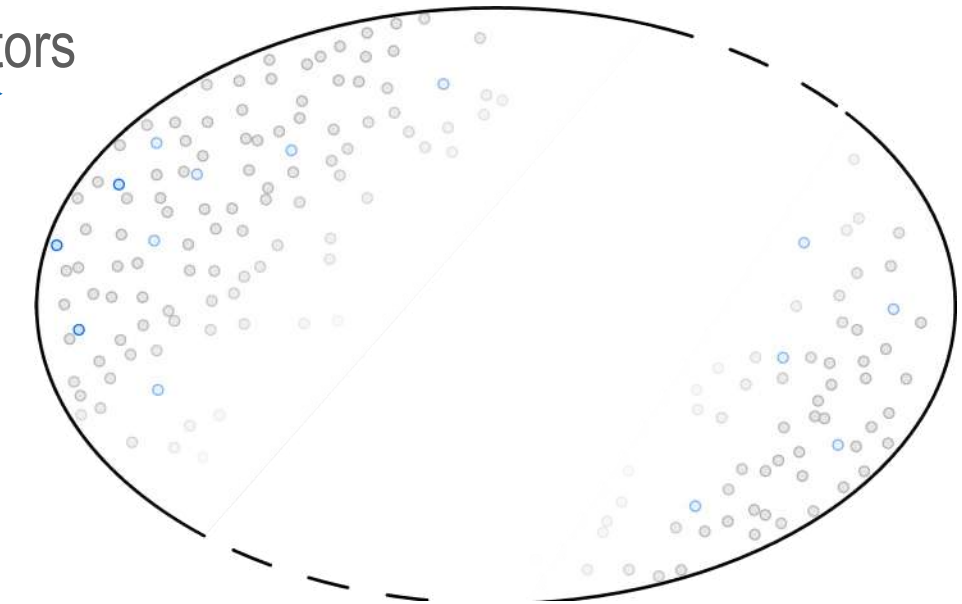
Symbol Domain
(E.g. Letters, Numbers, Labels)



Map to orthogonal Vectors

Representation Domain
(Binary Spatter Code, i.e. long Vectors of 0s and 1s)
D>1000

e.g. 0100110101100…101110001



A Primer on Hyper-Dimensional Computing

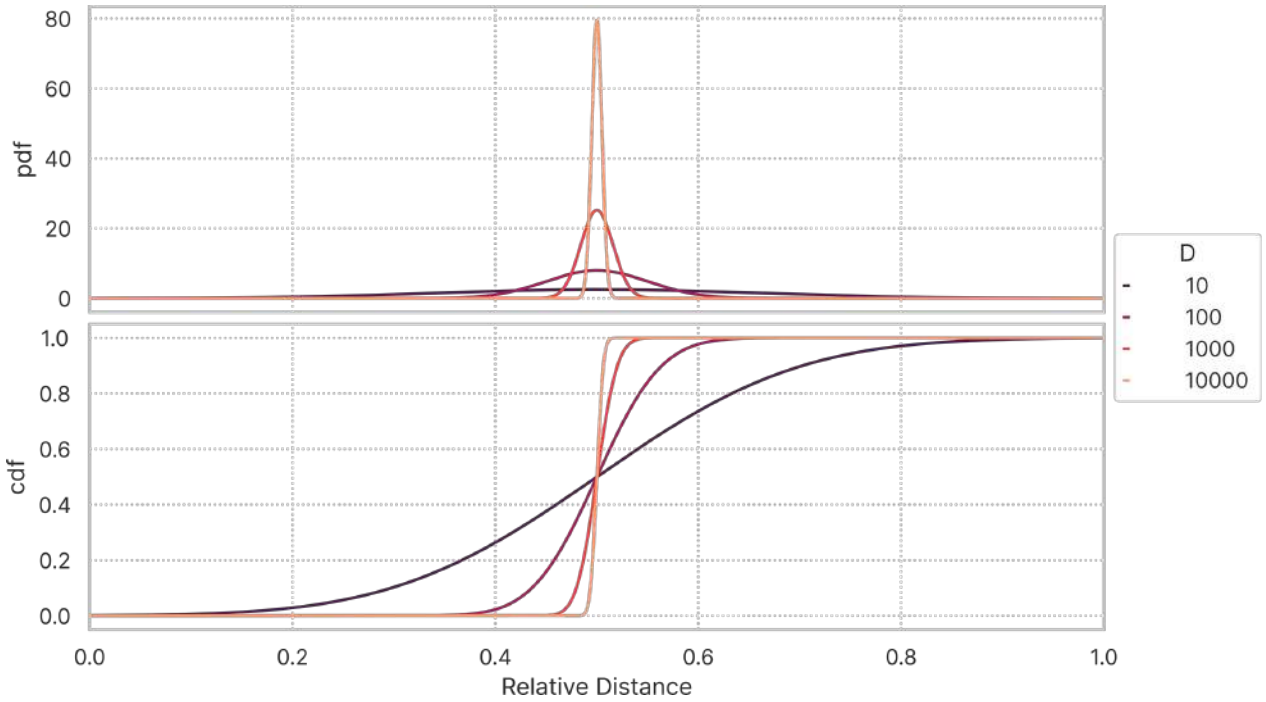
The properties of HD-Space:

Define Similarity Metric:

$$V_1 \approx V_2 \Leftrightarrow d_{\text{Hamming}}(V_1, V_2) \ll D/2$$

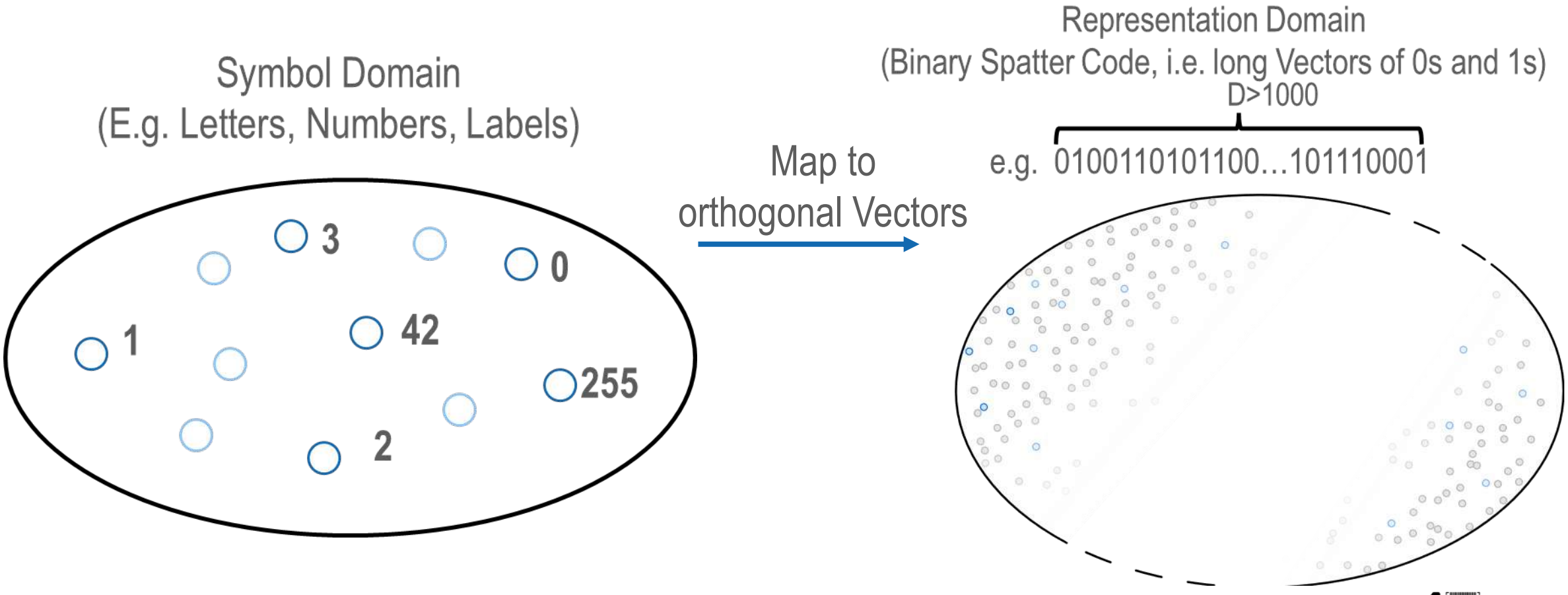
1. For large D, probability of two random vectors being similar to each other ≈ 0

→ random vectors are quasi-orthogonal



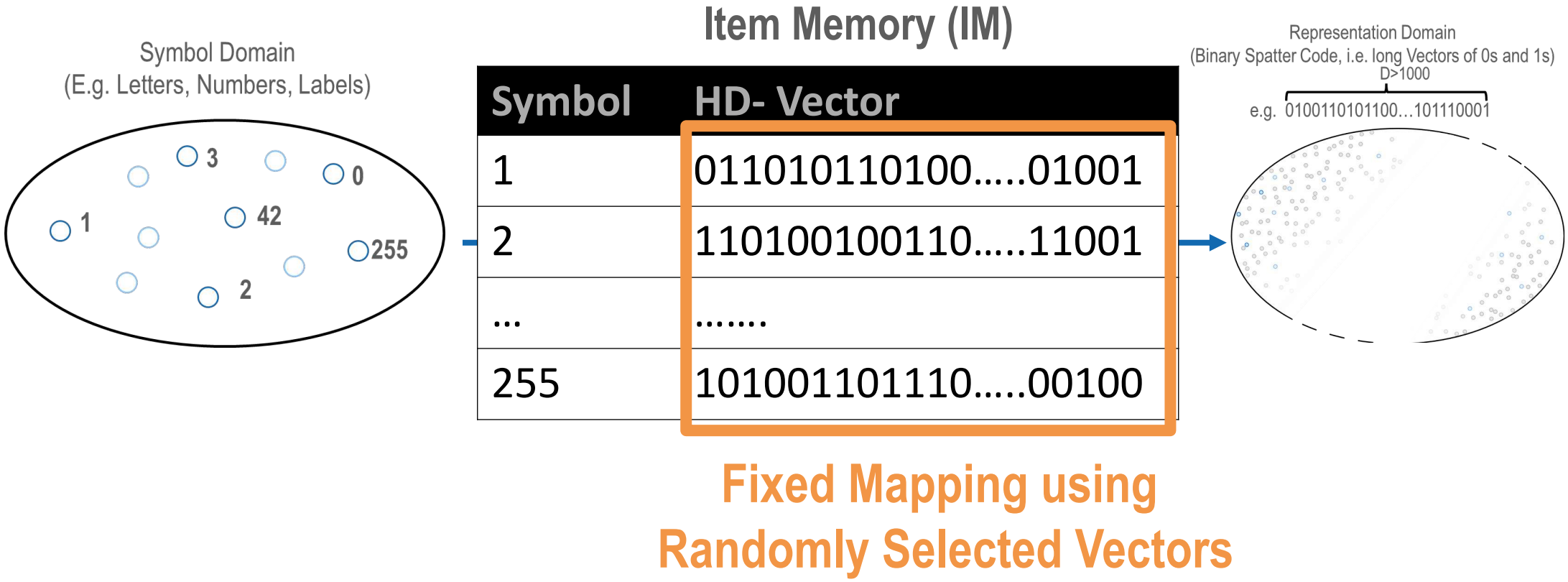
A Primer on Hyper-Dimensional Computing

Mapping Symbols to Digital Representations (The HDC Way)



A Primer on Hyper-Dimensional Computing

Mapping Symbols to Digital Representations (The HDC Way)



Fixed Mapping using Randomly Selected Vectors

A Primer on Hyper-Dimensional Computing

The properties of HD-Vectors:

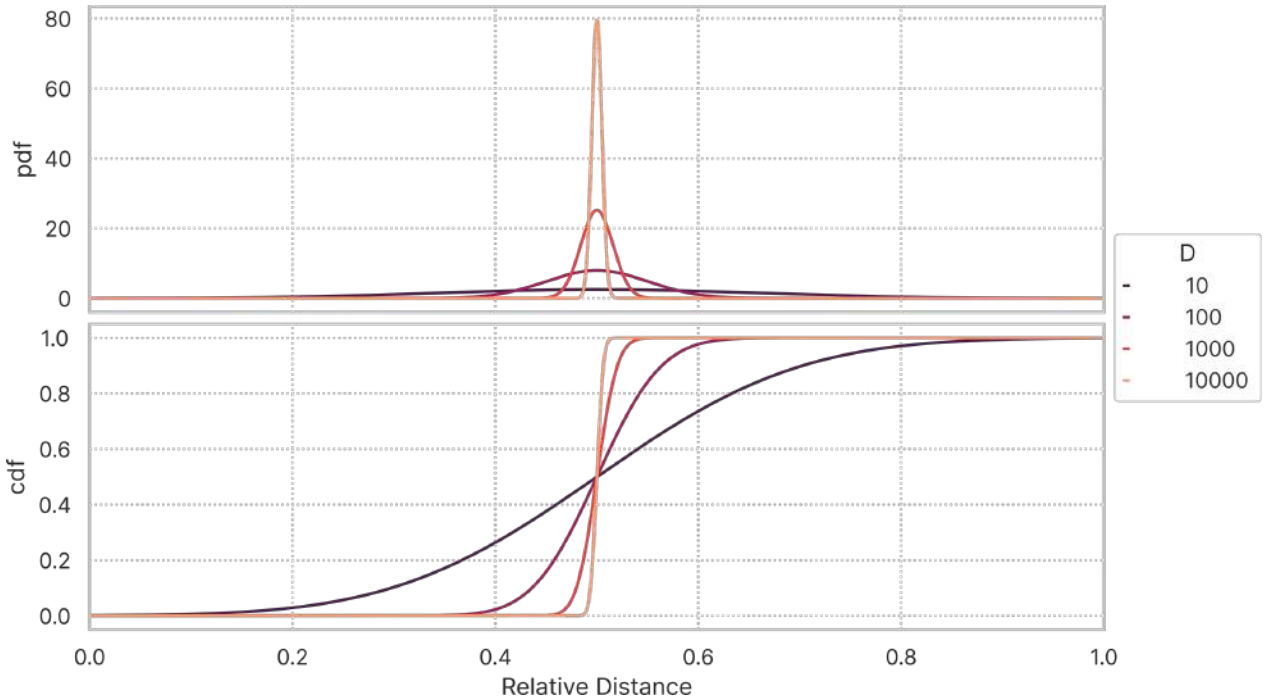
Similarity Metric:

$$V_1 \approx V_2 \Leftrightarrow d_{\text{Hamming}}(V_1, V_2) \ll D/2$$

1. For large D, probability of two random vectors being similar to each other ≈ 0

→ random vectors are quasi-orthogonal

2. A noisy Vector can be recovered with high probability by looking up most-similar vector in IM



A Primer on Hyper-Dimensional Computing

Item Memory (IM)
Implemented as Content-Addresseable-Memory (CAM)

01001101011000...101110001

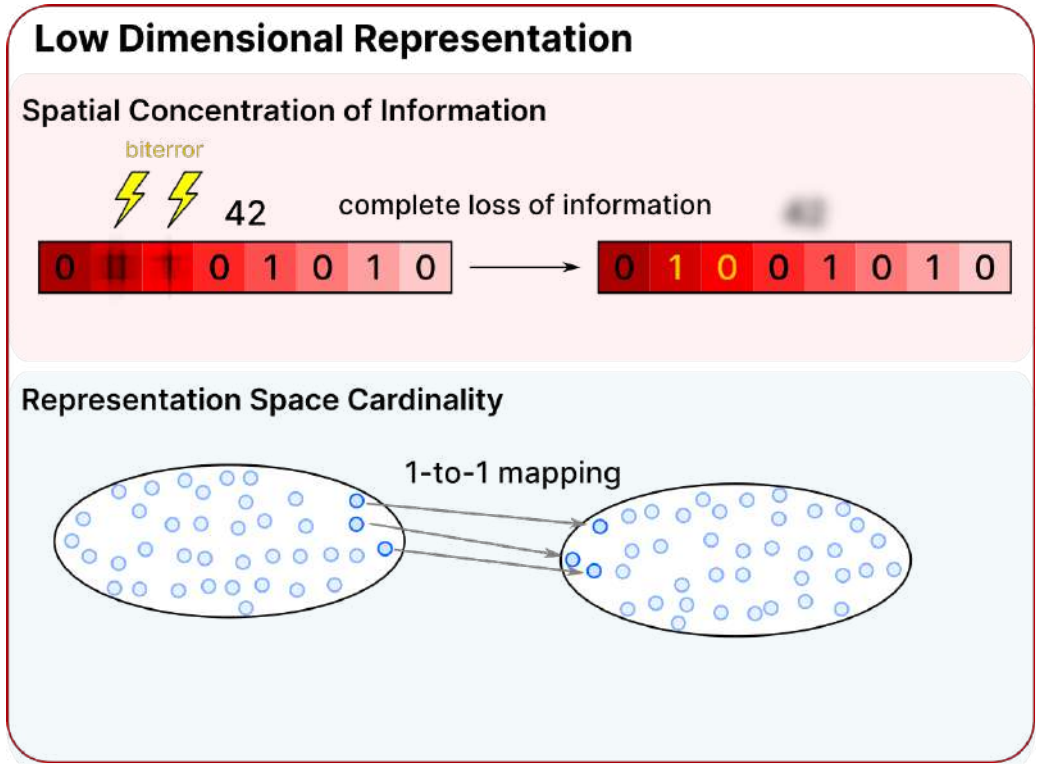
lookup

Symbol	HD- Vector
1	011010110100.....01001
2	110100100110.....11001
…	…….
255	101001101110.....00100

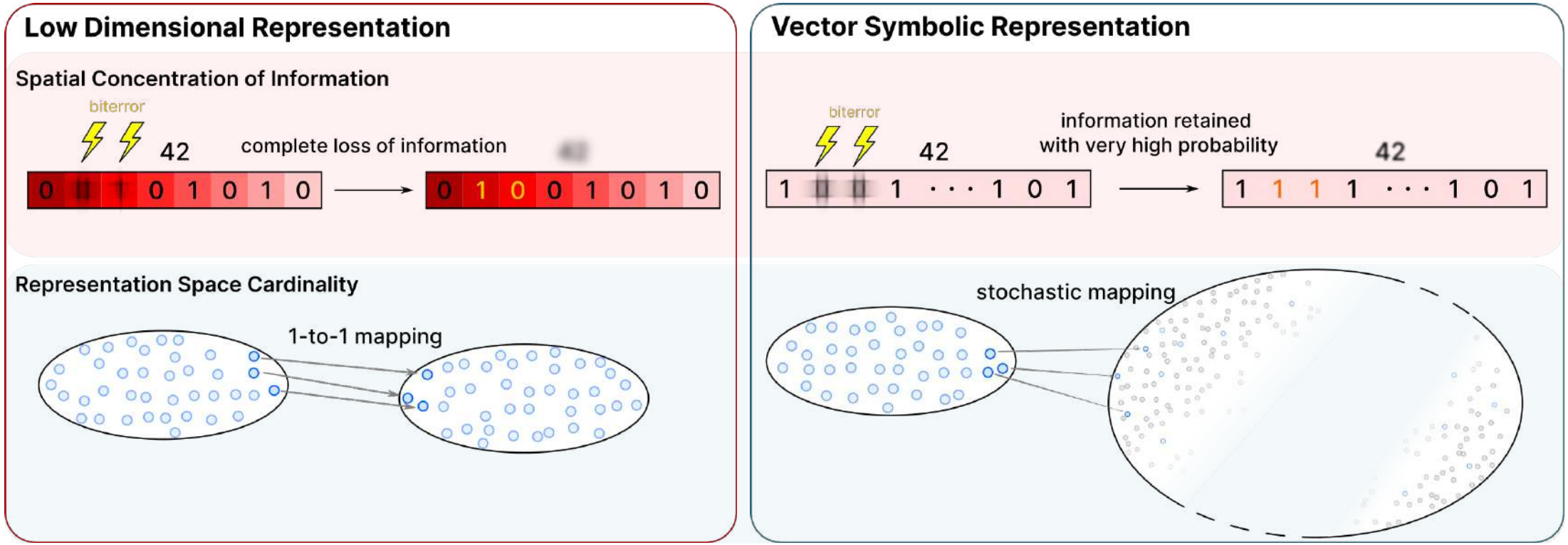
01101101010...101110011

Most similar Element

A Primer on Hyper-Dimensional Computing



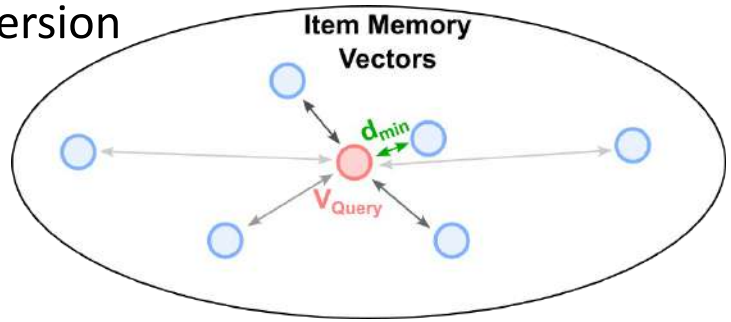
A Primer on Hyper-Dimensional Computing



HDC Operators

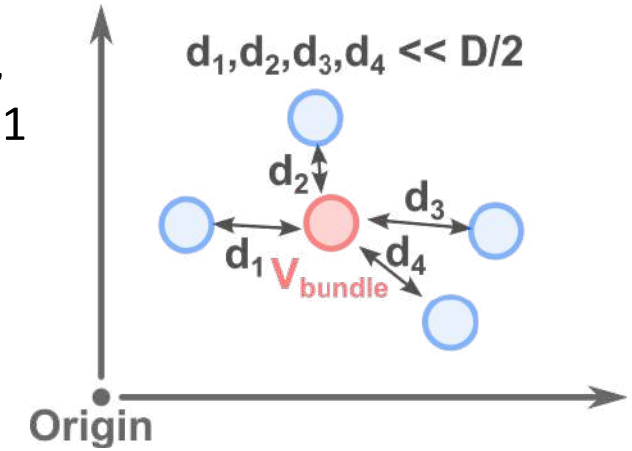
Associative Lookup

- Search for most similar vector in IM
- Recover Item Vector from noisy Version



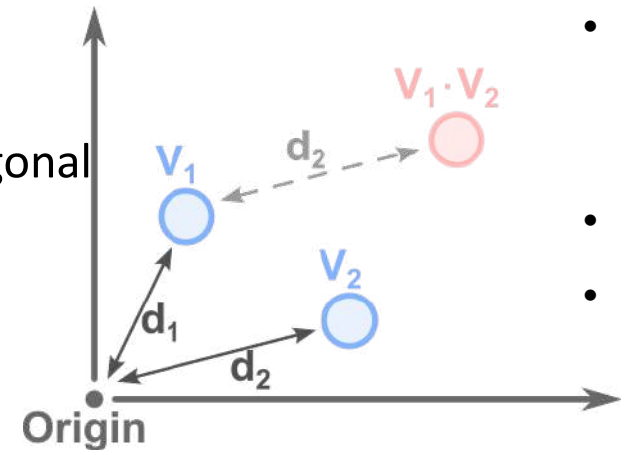
Bundling (+)

- For each bit position, take majority of 0 or 1
- Output similar to all operands



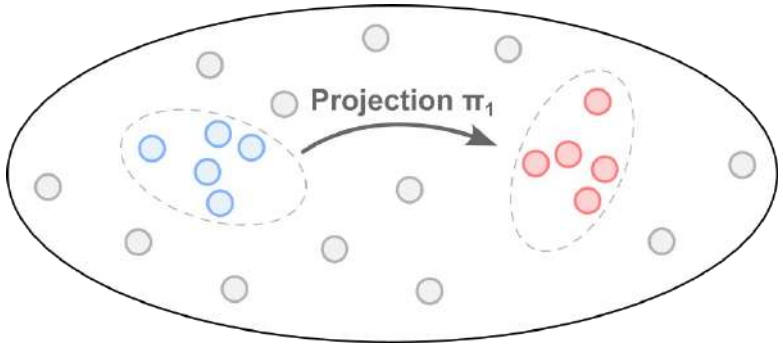
Binding (⊙)

- Take XOR of both vectors
- Output quasi-orthogonal to both operands
- Binding with same vector preserves similarity



Permutation (π)

- Permutate bit positions with fixed Permutation
- Unary Operation
- Maps vectors to orthogonal subspace



Building an ULP Accelerator for HDC

Associative Memory

Item Memory

Flexible Control Path

Bundling & Binding

Building an ULP Accelerator for HDC

Associative Memory

Item Memory

Flexible Control Path

Bundling & Binding

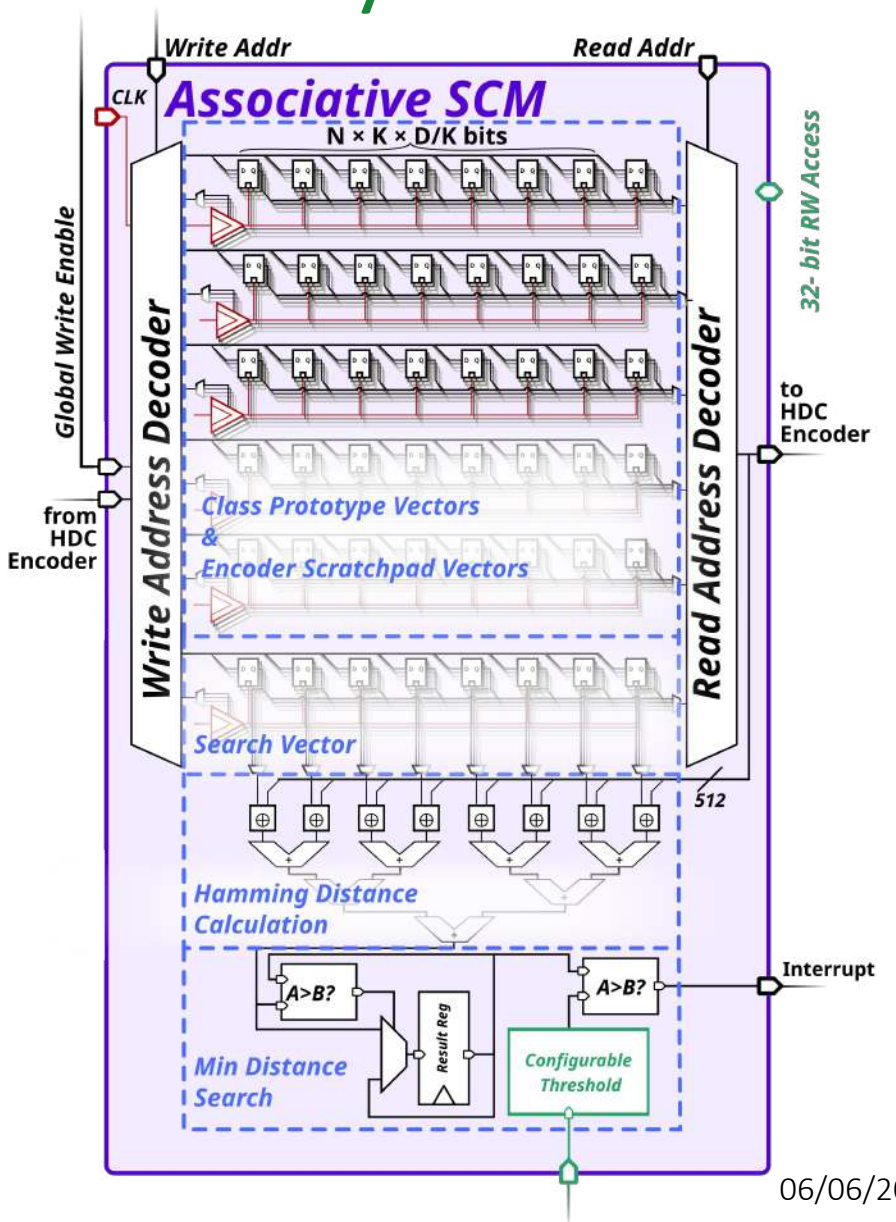
All-digital SCM-Based Associative Memory

- **Why digital?**

  Technology-agnostic, easy integration in every digital SoC

- **Why a Standard-Cell Memory (SCM)?**
  - Supports weird aspect ratios (e.g. 32 x 1024 bit)
  - Natively support aggressive voltage scaling
  - Allows parallel access to all words if required (e.g. during min distance search)

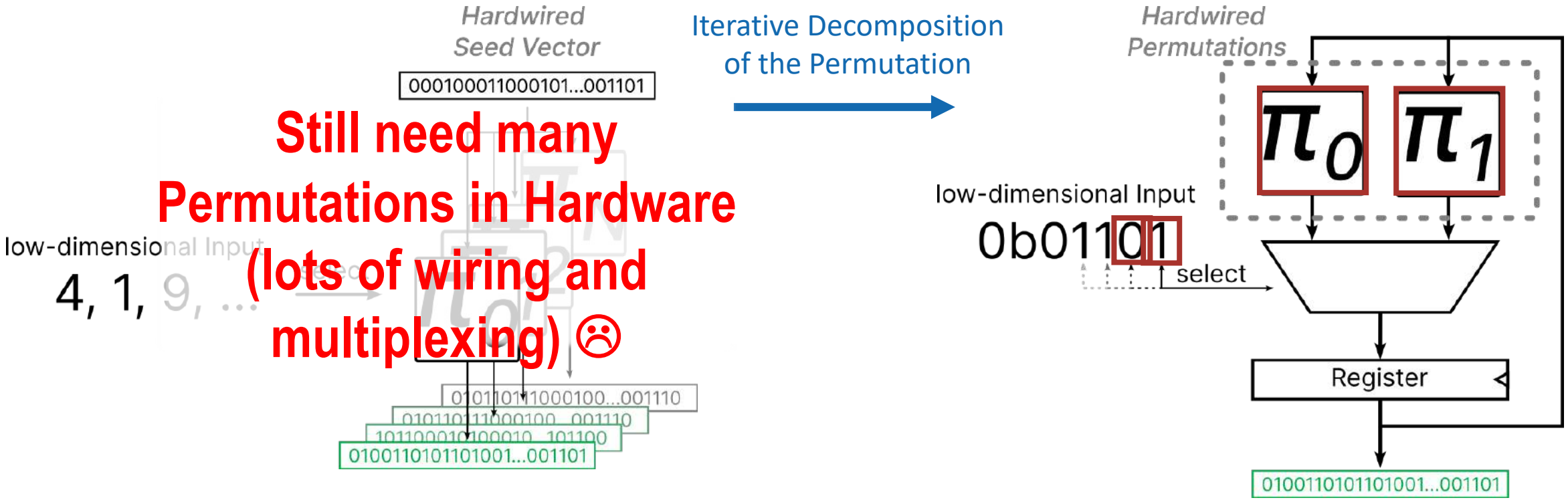
  Near Memory!



An Ephemeral Item Memory through Re-materialization

Observation: Storing many Item Vectors in a ROM is costly!

Idea: Use hardwired seed vector + random permutations



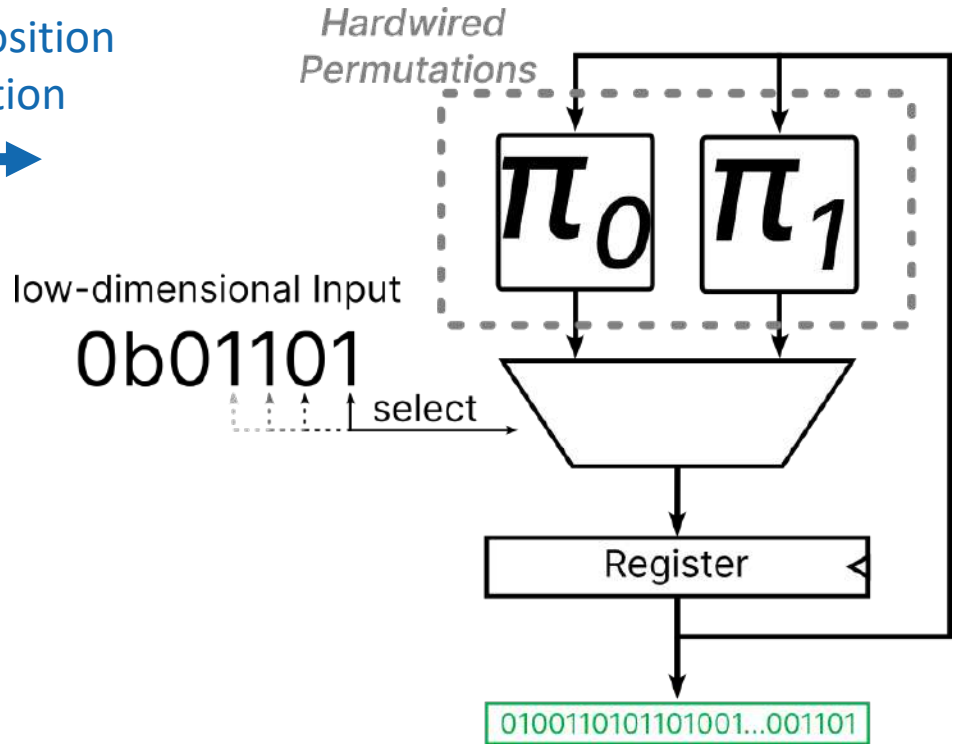
An Ephemeral Item Memory through Re-materialization

Observation: Storing many Item Vectors in a ROM is costly!

Idea: Use hardwired seed vector + random permutations

- \+ **No ROM required**
- \+ **Arbitrary number of Items supported**
- \+ **Random Access to any Item Vector in $log_2(n)$ cycles**

Iterative Decomposition of the Permutation



Building an ULP Accelerator for HDC

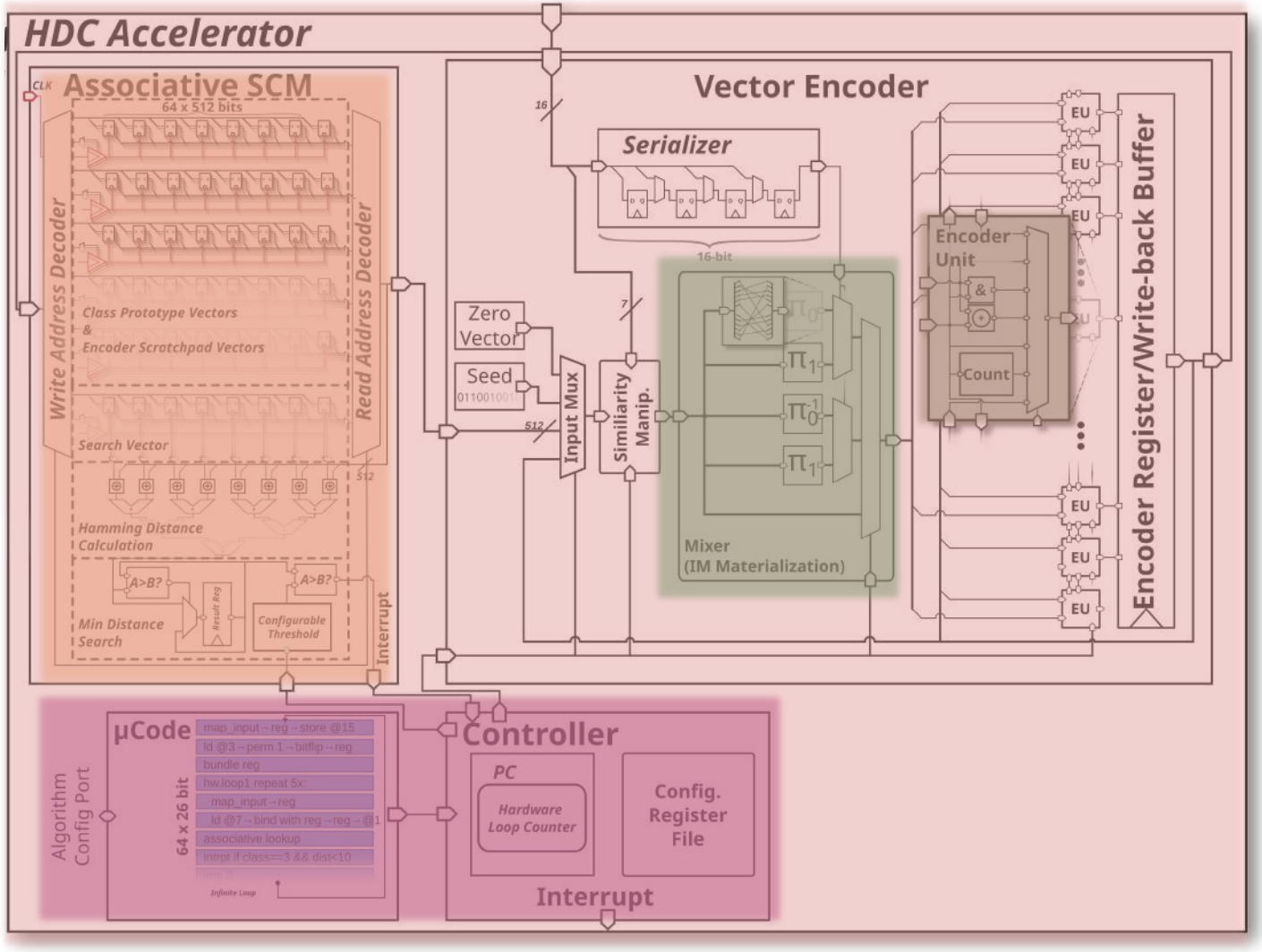
Associative Memory

Item Memory

Flexible Control Path

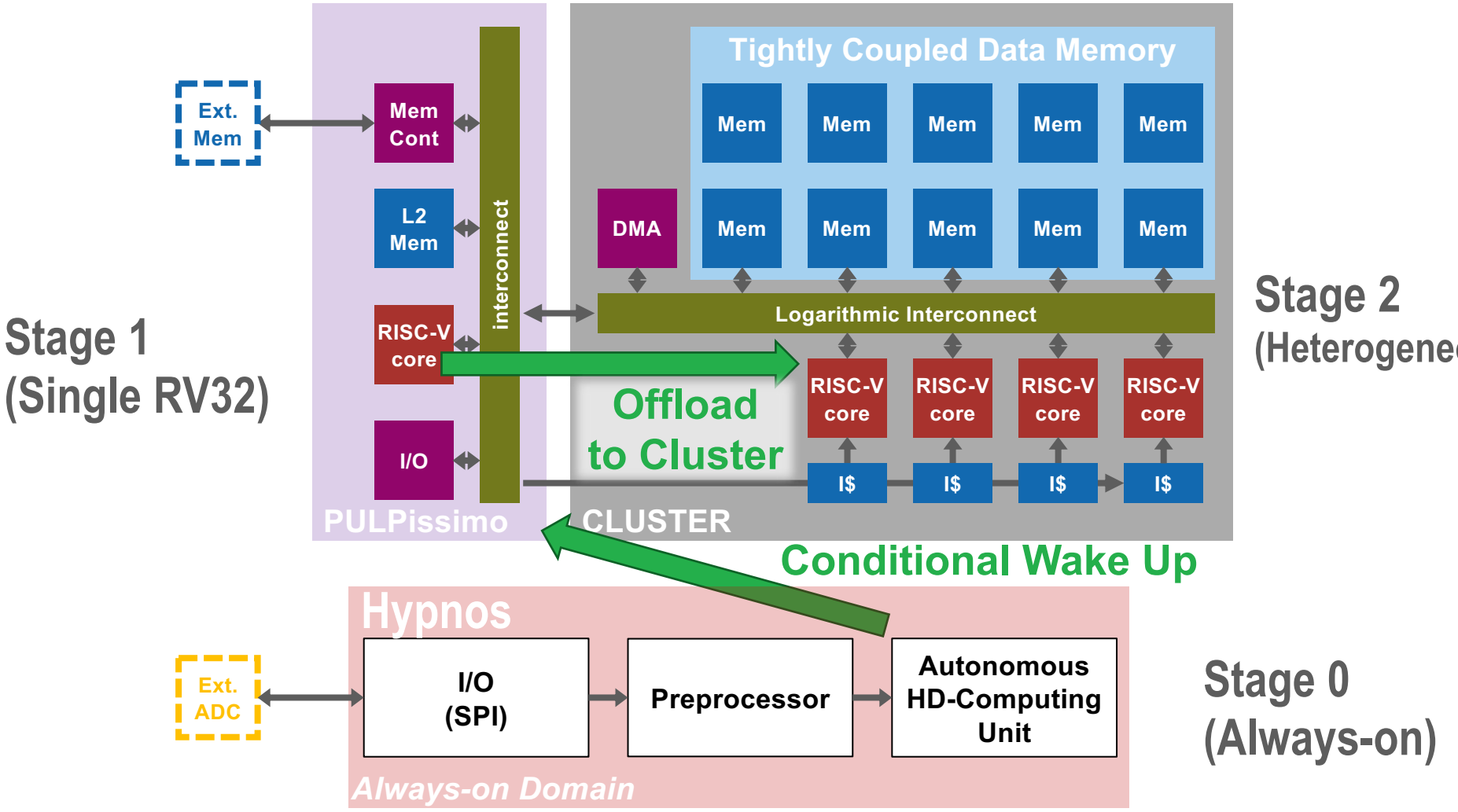
Bundling & Binding

Hypnos – An Ultra-Low Power HDC Accelerator



Vega: A Ten-Core SoC for IoT Endnodes in GF22

Published in ISSCC21/JSSC Volume 57



Vega: A Ten-Core SoC for IoT Endnodes in GF22

Published in ISSCC21/JSSC Volume 57

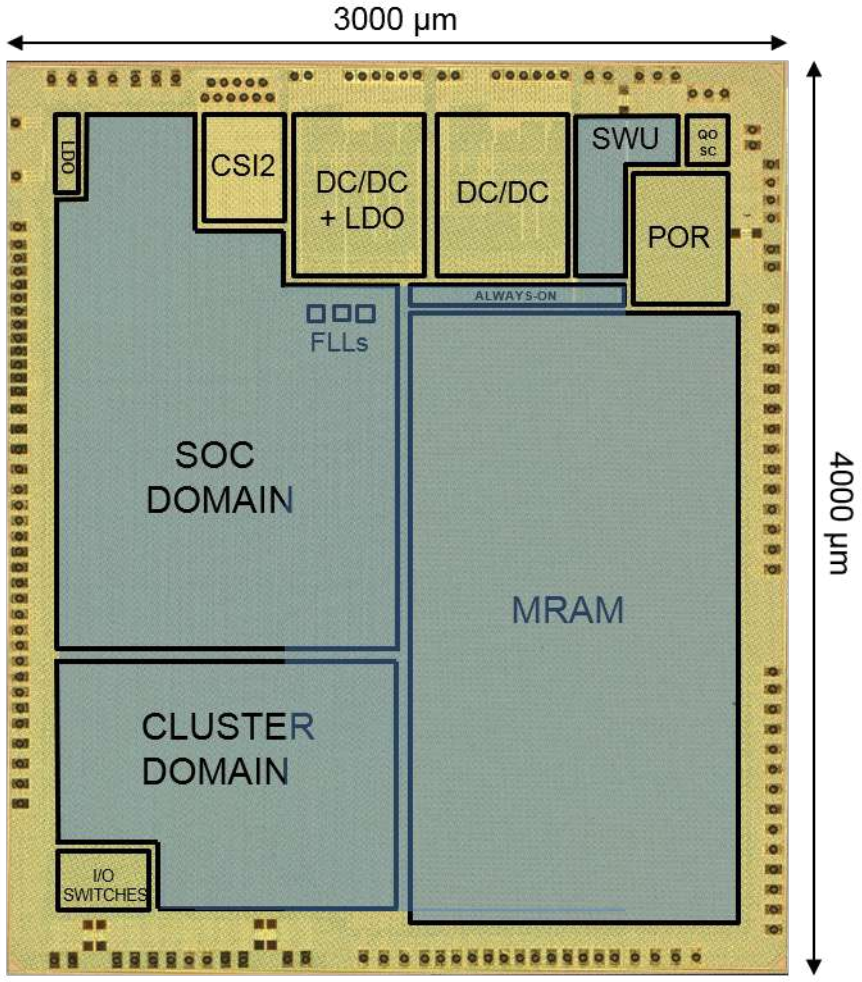
Specifications	
Technology	GF22 UHT
Area	670kGE
Max. Frequency	3 MHz
SCM-Memory	32 kBit
VDD	0.6V

3-channel HDC Inference Algorithm

f_{clk}	32kHz	200kHz
max. Sampling Rate	150 SPS/Channel	1kSPS/Channel
$P_{SWU,\ dynamic}$	0.99uW	6.21uW
$P_{SWU,\ leakage}$	0.7uW	0.7uW
$P_{SPI,\ dynamic}$	1.28uW	8.00uW
$P_{SWU,\ total}$	**2.97uW**	**14.9uW**

Vega: A Ten-Core SoC for IoT Endnodes in GF22

Published in ISSCC21/JSSC Volume 57



Technology	22nm FDSOI
Chip Area	12mm^2
SRAM	1728 kB
MRAM	4 MB
WU Sources	GPIO, RTC, Cognitive
VDD range	0.5V - 0.8V
VBB range	0V - +1.1V
Freq. Range	32 kHz - 450 MHz
Pow. Range	1.7 µW - 49.4 mW
Int. Perf.	15.6 GOPS
Int. Eff.	614 GOPS/W
F.P. Perf.	3.3 GFLOPS
F.P. Eff.	129 GFLOPS/W
Int. ML Perf.	32.2 GOPS
Int. ML Eff.	1.3 TOPS/W

PULP Platform

Open Source Hardware, the way it should be!

Gamze Islamoglu gislamoglu@iis.ee.ethz.ch

Manuel Eggimann meggimann@iis.ee.ethz.ch

Institut für Integrierte Systeme – ETH Zürich
Gloriastrasse 35
Zürich, Switzerland

DEI – Universitá di Bologna
Viale del Risorgimento 2
Bologna, Italy

@pulp_platform

pulp-platform.org

youtube.com/pulp_platform

ETH zürich

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA