ODTÜ
METU

TÜBİTAK

EURO 4SEE

# Optimizing Deep Learning Systems for Hardware

Assoc. Prof. Erdem AKAGÜNDÜZ, METU

ncc@ulakbim.gov.tr

# Part I : Fundamentals

- PI.a : Why hardware matters in deep learning?
- PI.b : Performance metrics
- PI.c : Case Study: Edge Devices vs Datacenter vs Supercomputers

# Hardware matters!

- Hardware is not universal; it is designed for specific **types** of computation.
- Different hardware handles different operations at different speeds.

# Hardware matters!

- Hardware is not universal; it is designed for specific **types** of computation.
- Different hardware handles different operations at different speeds.

- Some hardware excels at logic/branching, others at matrix math.
- For example:
  - Fixed-point vs floating-point computation:
    each hardware type is optimized differently.

# Computation types?

- 1. Logic Operations
  - AND, OR, XOR, NOT
  - Bitwise operations
  - Comparisons (>, <, ==)

# Computation types?

- 2. Branching / Control Flow
  - Conditional statements (if/else)
  - Loops with unpredictable iteration counts
  - Function calls and returns

# Computation types?

- 2. Branching / Control Flow
  - Conditional statements (if/else)
  - Loops with unpredictable iteration counts
  - Function calls and returns

Logic Design → Computer Architecture → Microprocessors

NCC TÜRKİYE · EURO CC 4SEE

# Computation types?

- 3. Memory Operations
    - Read from memory / cache
    - Write to memory / cache
    - Load/store operations

Logic Design → Computer Architecture → Microprocessors

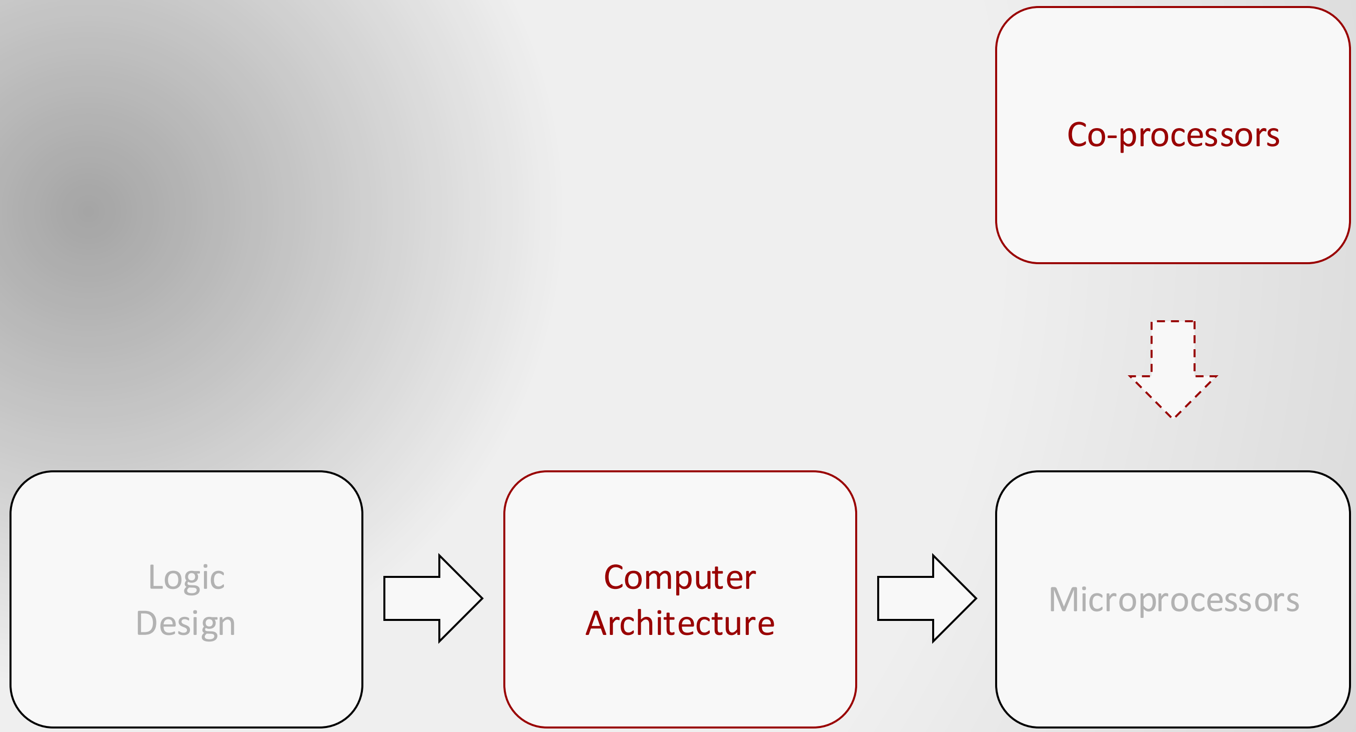
# Computation types?

- 4. Arithmetic Operations
  - Additions / Subtractions
  - Multiplications / Divisions
  - Floating-point vs fixed-point computation
  - Accumulations / reductions (sums, averages)

Logic Design → Computer Architecture → Microprocessors

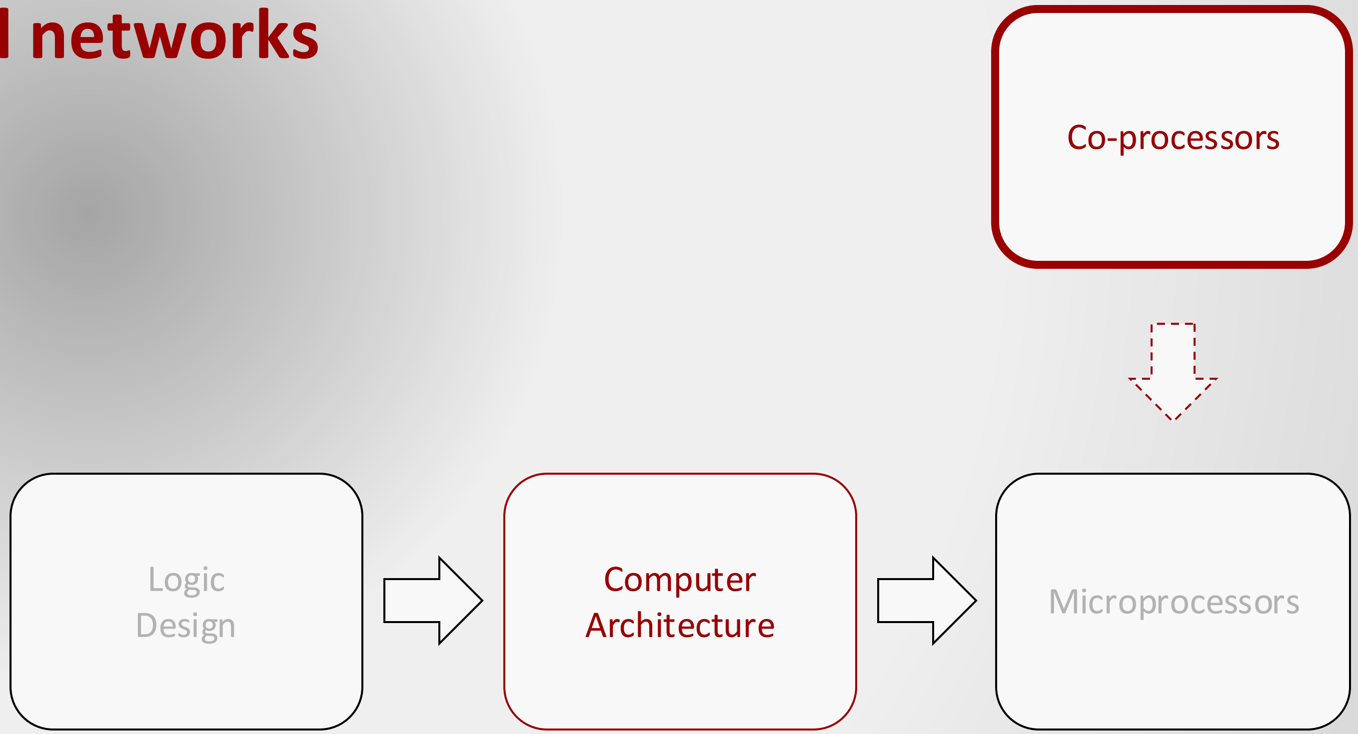
# Computation types?

- 5. Specialized Vector / Matrix Operations
  - Dot products, matrix multiplications
  - Convolutions
  - Tensor contractions
  - Batch operations



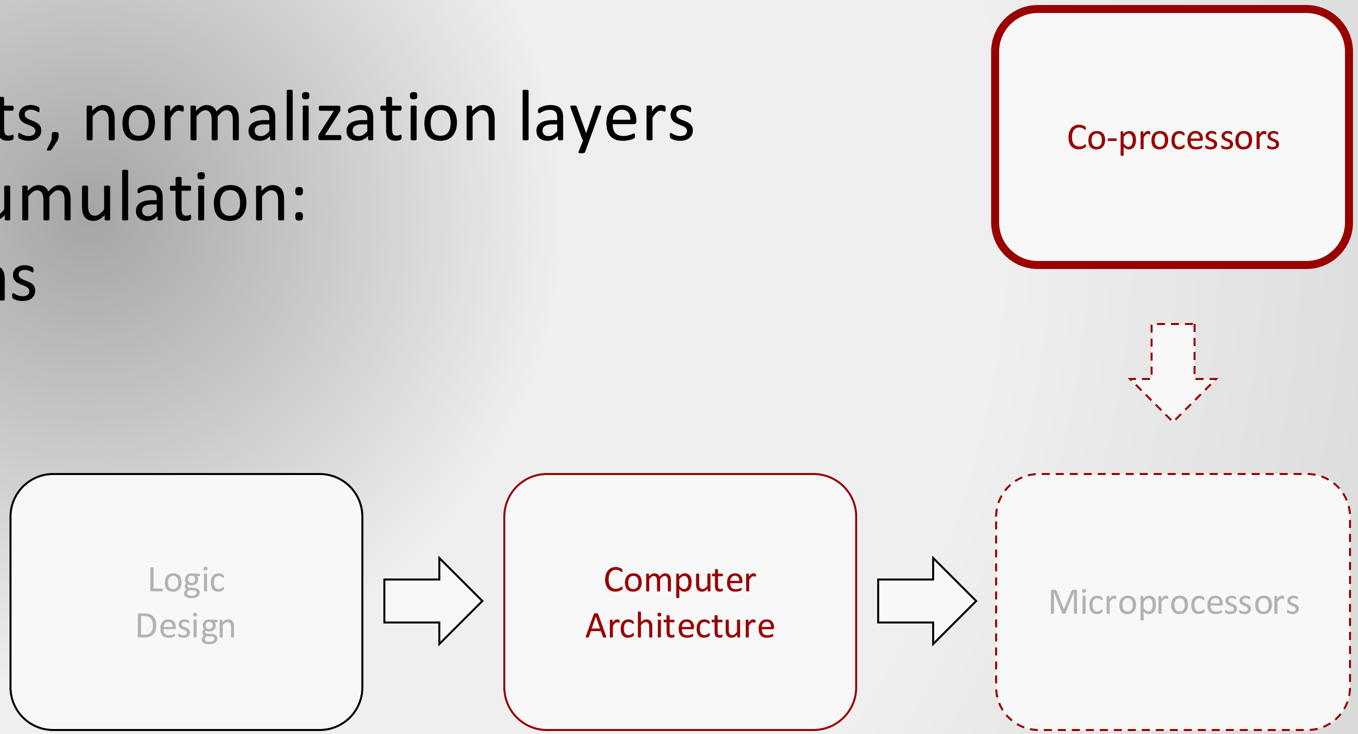
# Computation types?

- 6. Miscellaneous / Specialized
  - Random number generation
  - Transcendental functions (exp, log, sin, cos)
  - **Activation functions in neural networks**
  - **Parallelization**
  - **Quantization**
  - **others…**



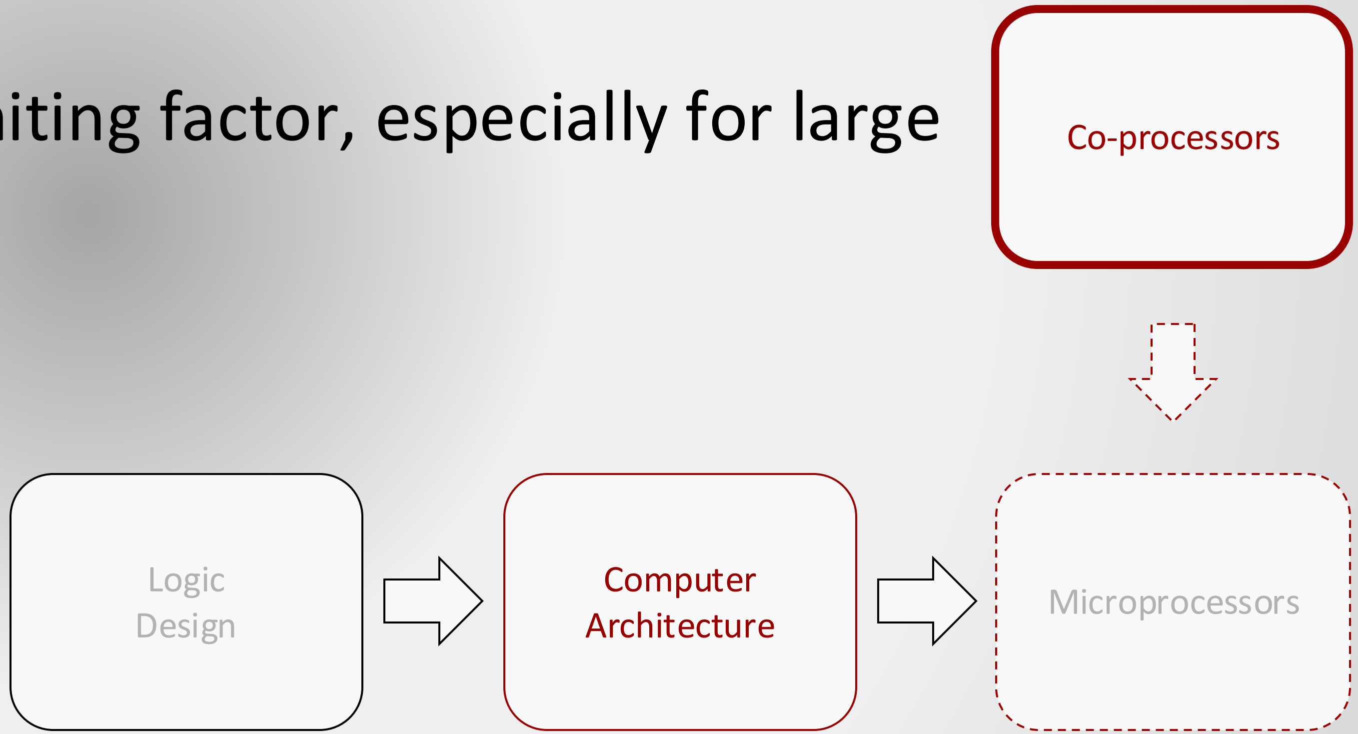
# for Deep Learning?

- Arithmetic-heavy:
  - Matrix / vector operations:
    - Core of forward/backward passes (dense linear algebra)
  - Reductions (sum/average):
    - Computing losses, gradients, normalization layers
  - Multiplication / division / accumulation:
    - Weight updates, activations



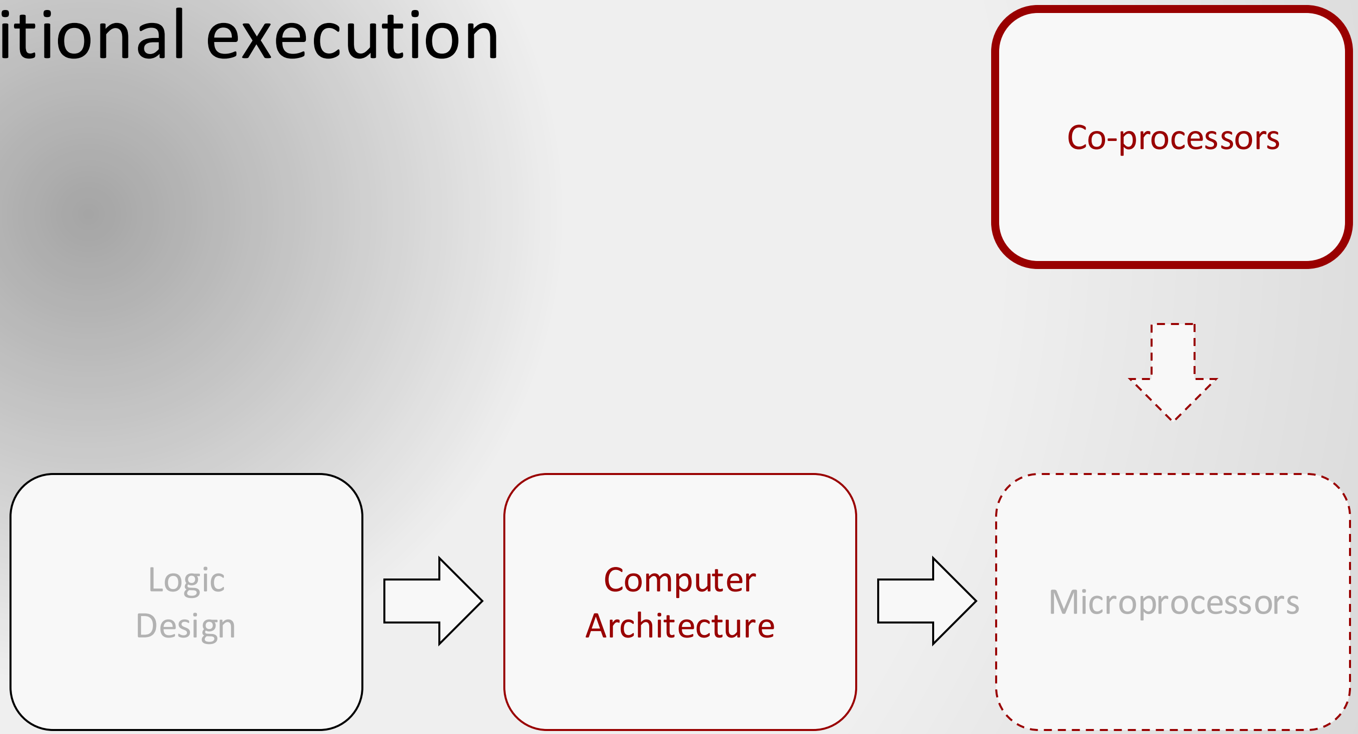
# for Deep Learning?

- Memory-heavy:
  - Read/write/transfer:
    - Loading inputs, activations, weights; storing gradients
  - Memory bandwidth:
    - which often becomes a limiting factor, especially for large models!



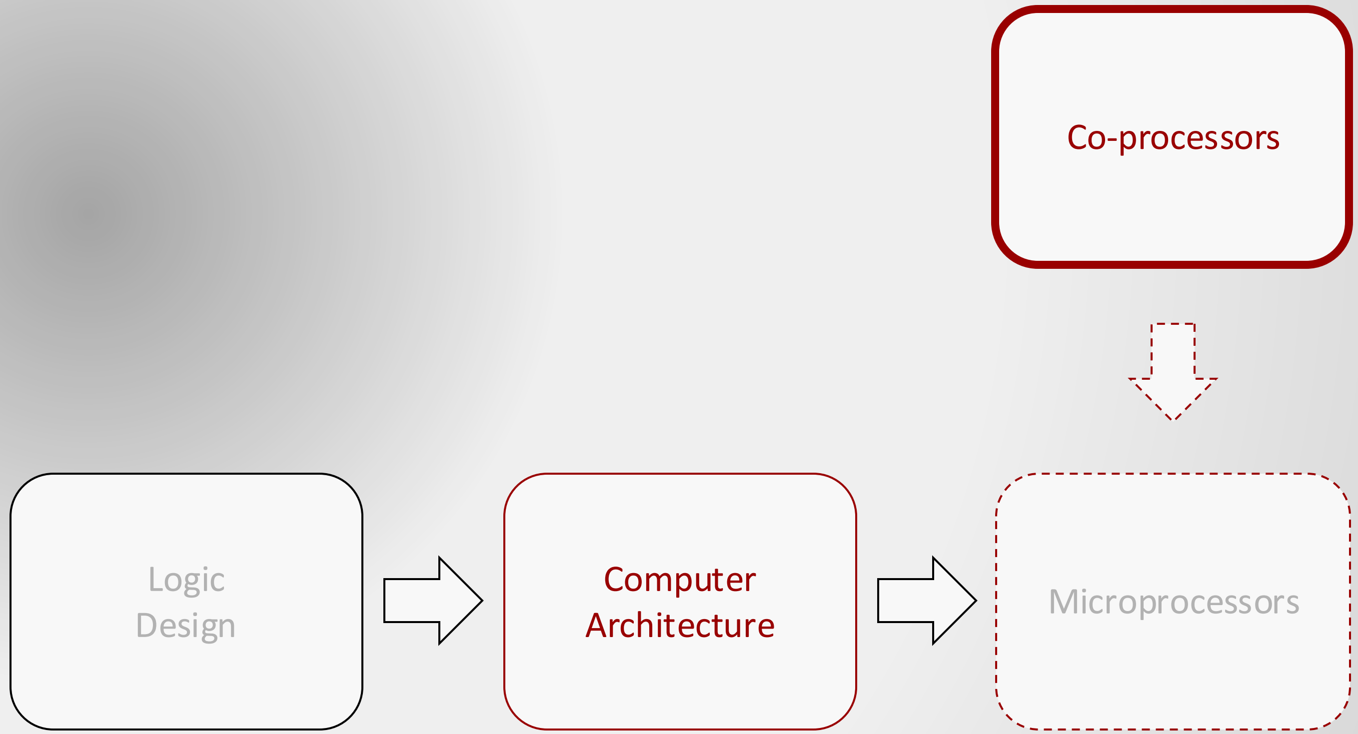
# for Deep Learning?

- Logic / branching:
  - Minimal in standard feed-forward neural nets
  - More significant in models with dynamic architectures, RNNs with variable lengths, or conditional execution
    - BP-thru-time in RNNs



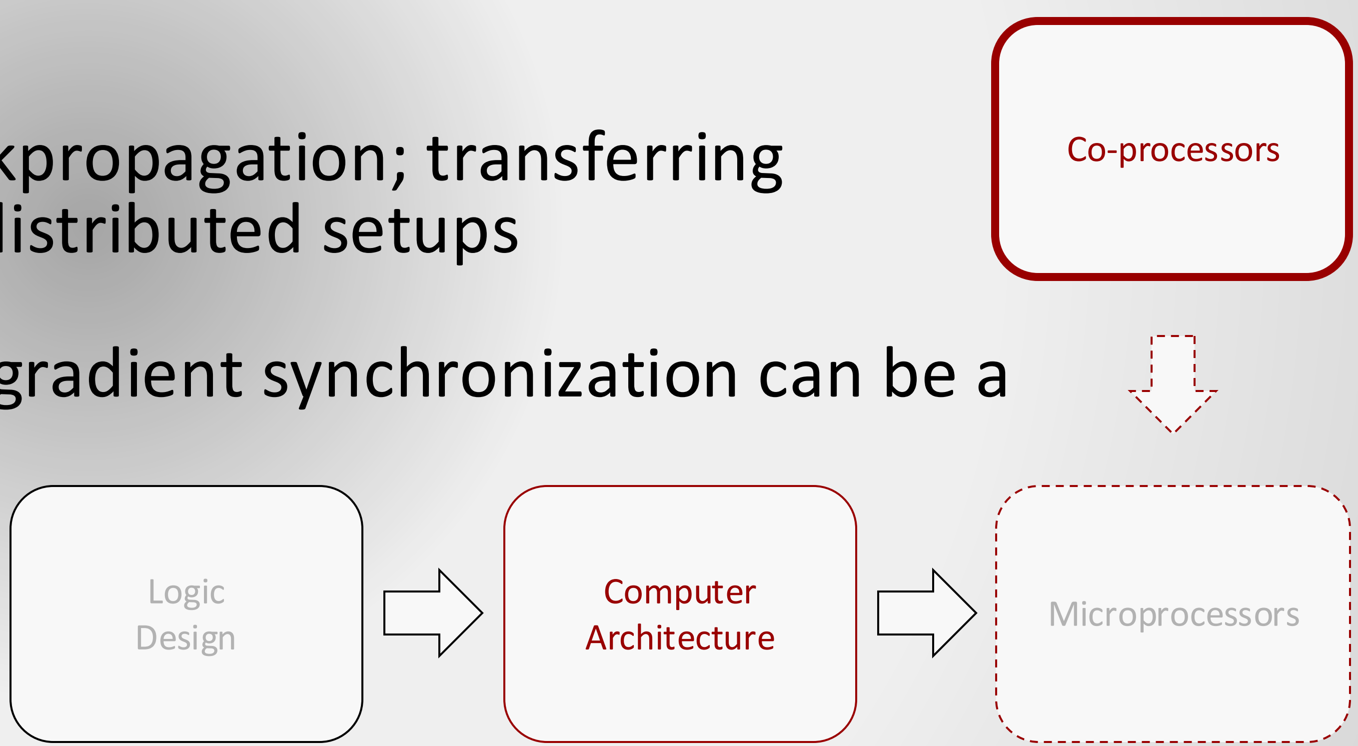
# for Deep Learning?

- Specialized functions:
  - Activation functions (ReLU, Sigmoid, GELU)
  - Softmax, normalization layers, random sampling (dropout)



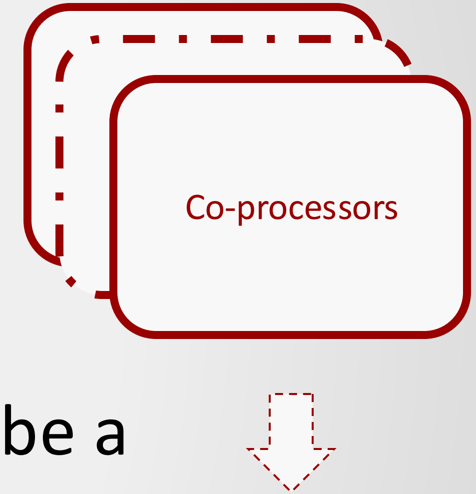
# training vs inference?

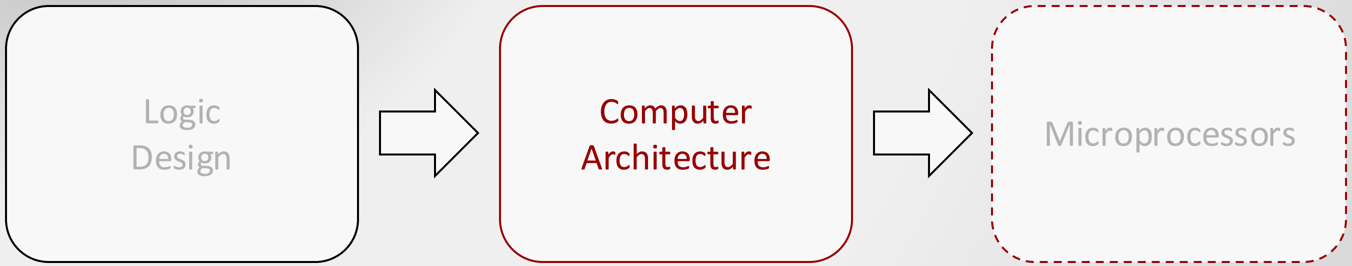
- Training:
  - Arithmetic-bound:
    - Most time spent in matrix multiplications for forward/backward passes
  - Memory-bound:
    - Storing activations for backpropagation; transferring weights across devices in distributed setups
  - Communication-bound:
    - In multi-GPU/HPC setups, gradient synchronization can be a bottleneck



# training vs inference?

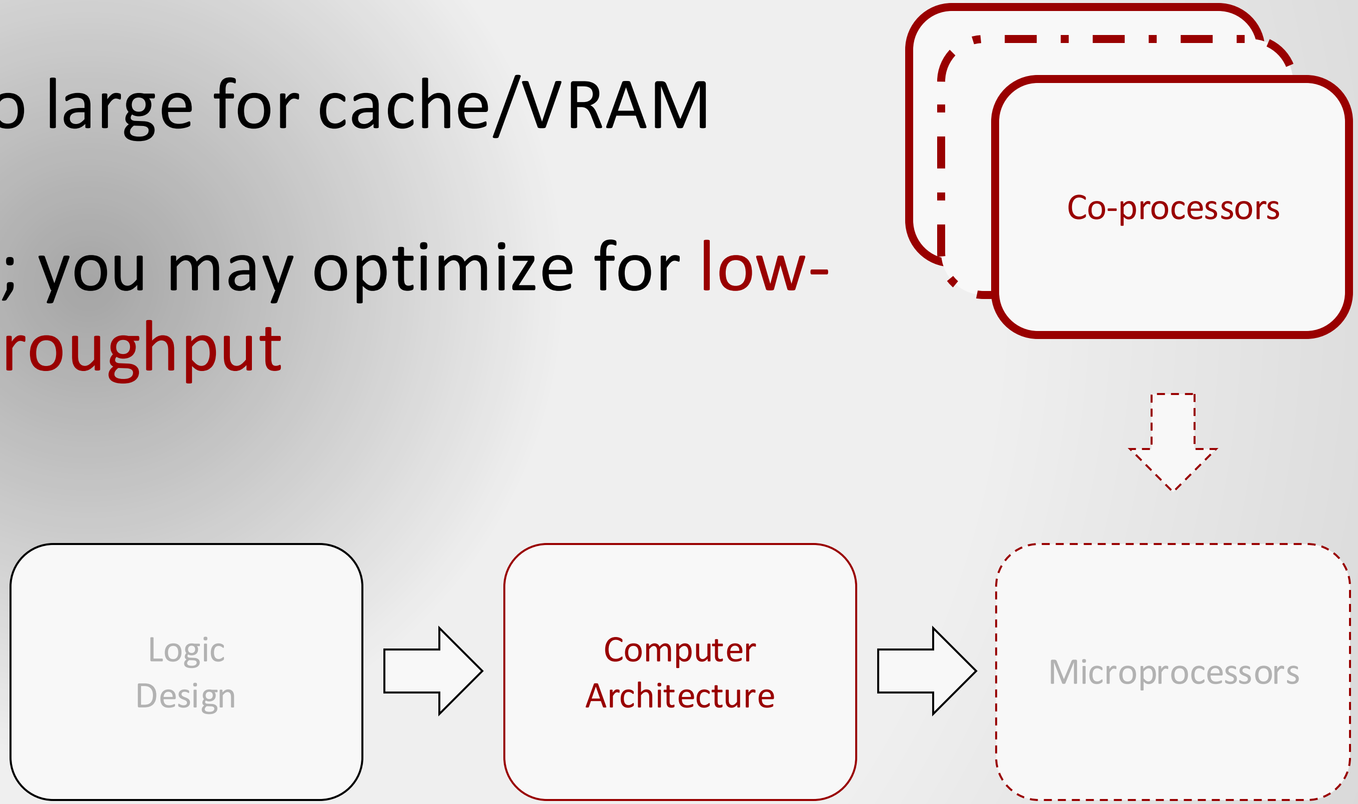
- Inference:
  - Arithmetic-bound:
    - Most time spent in matrix multiplications for forward/backward passes
  - Memory-bound:
    - Storing activations for backpropagation; transferring weights across devices in distributed setups
  - Communication-bound:
    - In multi-GPU/HPC setups, gradient synchronization can be a bottleneck



# training vs inference?

- Inference:
  - Arithmetic-bound for large models, but generally less intensive than training
  - Memory-bound if model is too large for cache/VRAM
  - Latency-sensitive:
    - Especially for edge devices; you may optimize for low-latency rather than max throughput



# training vs inference?

- Inference:
    - Arithmetic-bound for large models, but generally less intensive than training
    - Memory-bound if model is too large for cache/VRAM
    - Latency-sensitive:
        - Especially for edge devices; you may optimize for low-latency rather than max throughput

**latency? throughput?**

Co-processors

Logic Design → Computer Architecture → Microprocessors

# Next: Part I.b Performance Metrics

- PI.a : Why hardware matters in deep learning?
- PI.b : Performance metrics
- PI.c : Case Study: Edge Devices vs Datacenter vs Supercomputers

EURO 4SEE

# Thanks!

Co-funded by the European Union

EuroHPC Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101191697. The JU receives support from the Digital Europe Programme and Germany, Türkiye, Republic of North Macedonia, Montenegro, Serbia, Bosnia and Herzegovina.