CS516: Parallelization of Programs

Overview of Parallel Architectures

Vishwesh Jatala

Assistant Professor

Department of CSE

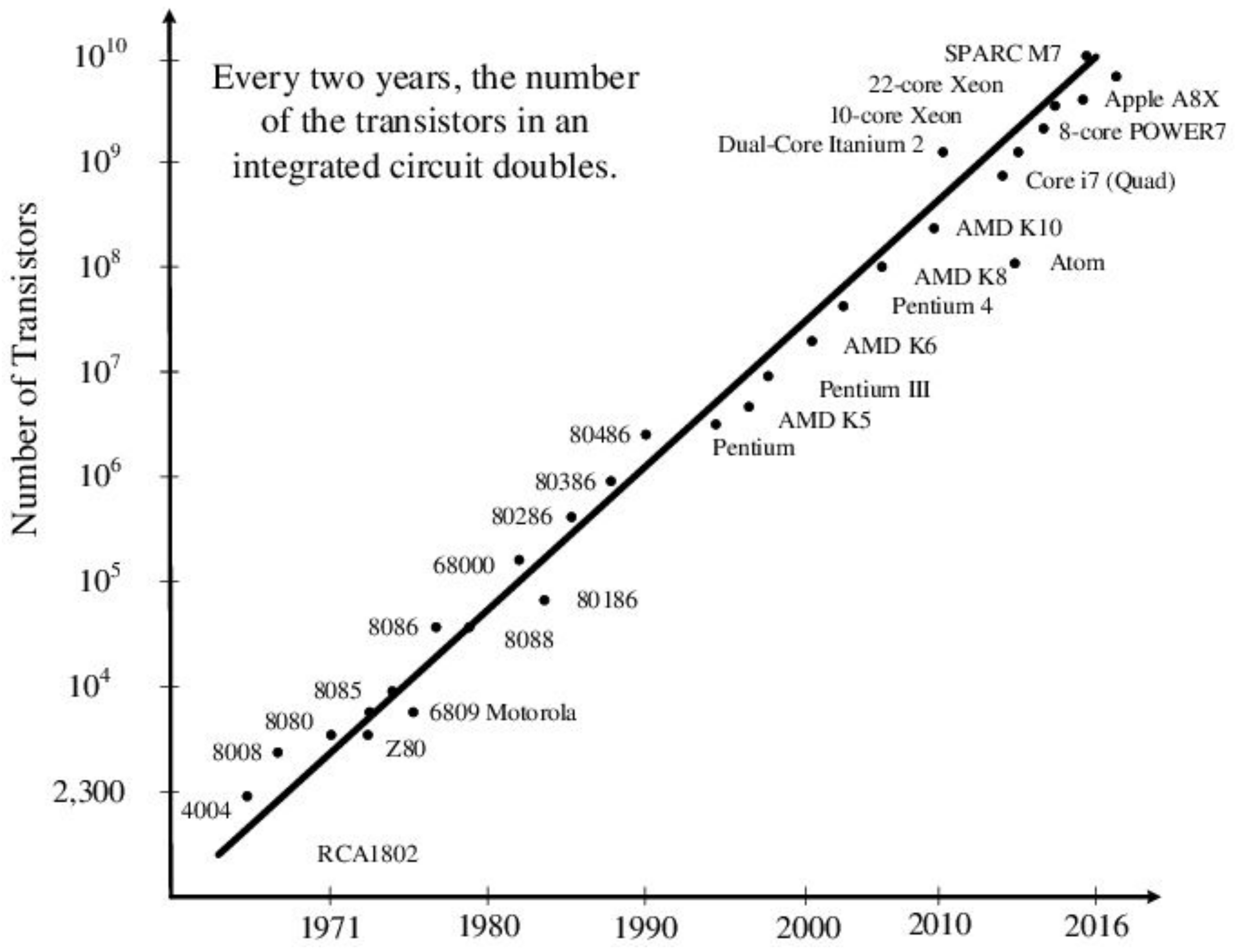
Indian Institute of Technology Bhilai

vishwesh@iitbhilai.ac.in

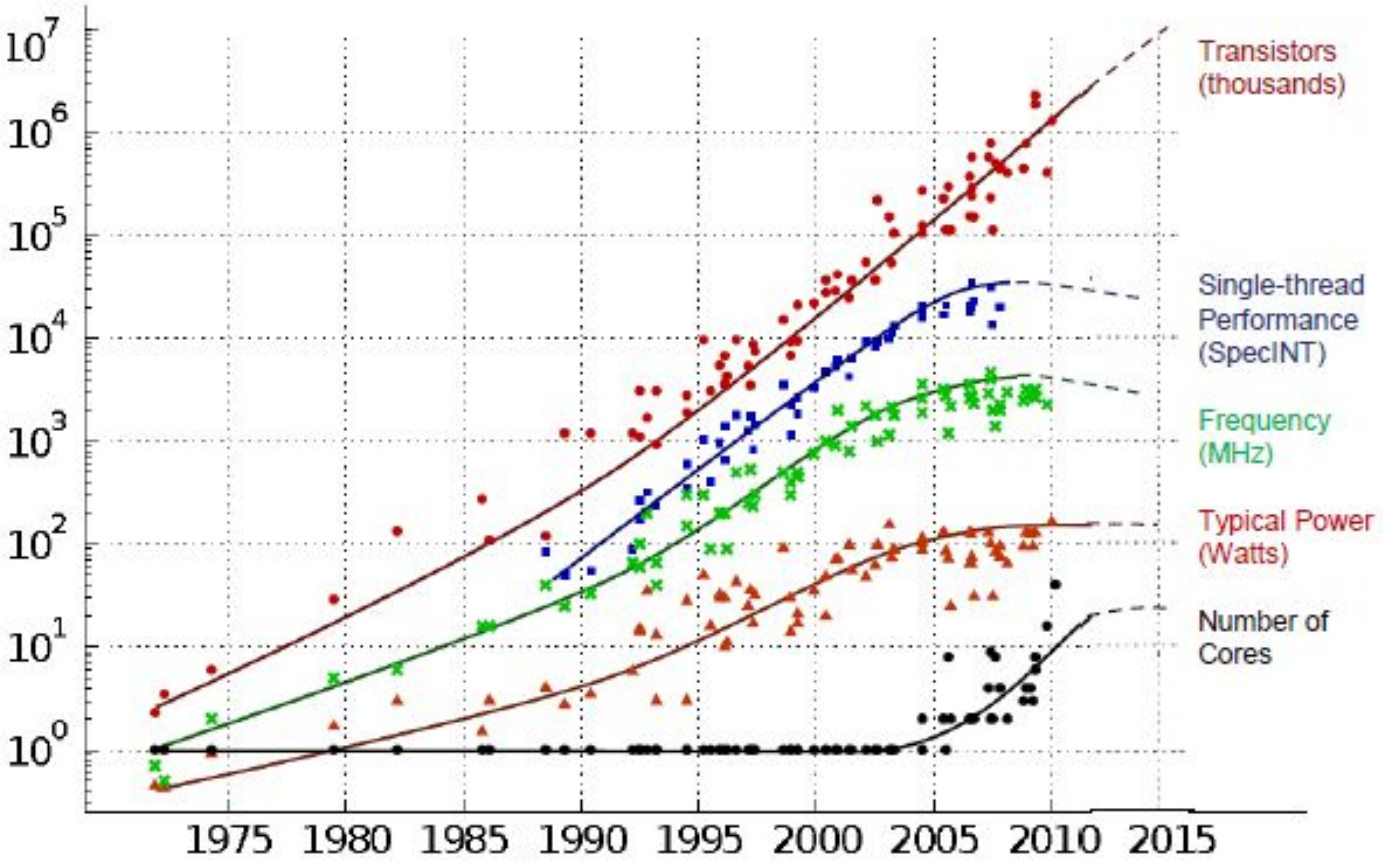
2023-24 W

Recap: Why Parallel Architectures?

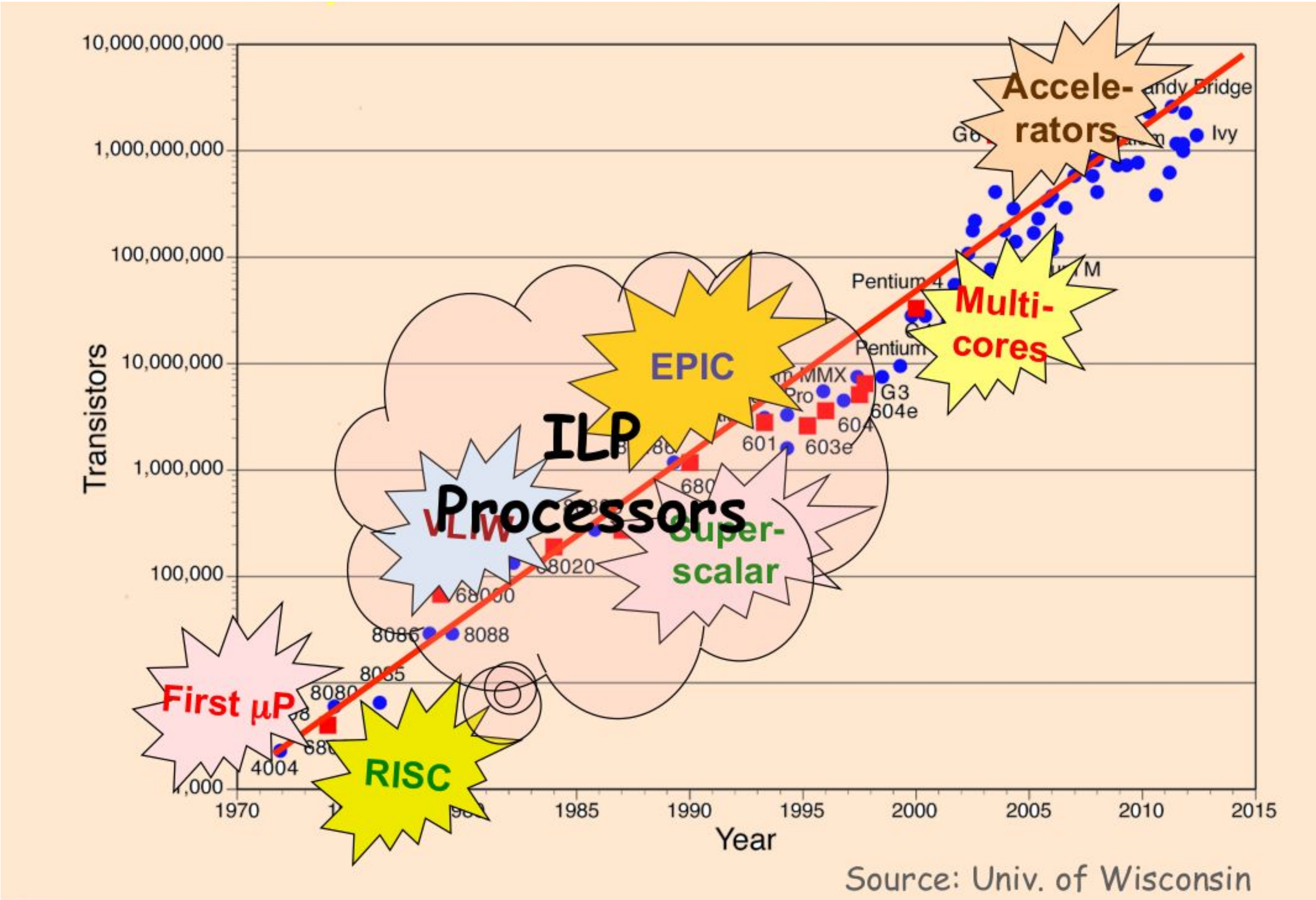
- Moore's Law: The number of transistors on a IC doubles about every two years



Recap: Moore's Law Effect



Processor Architecture RoadMap

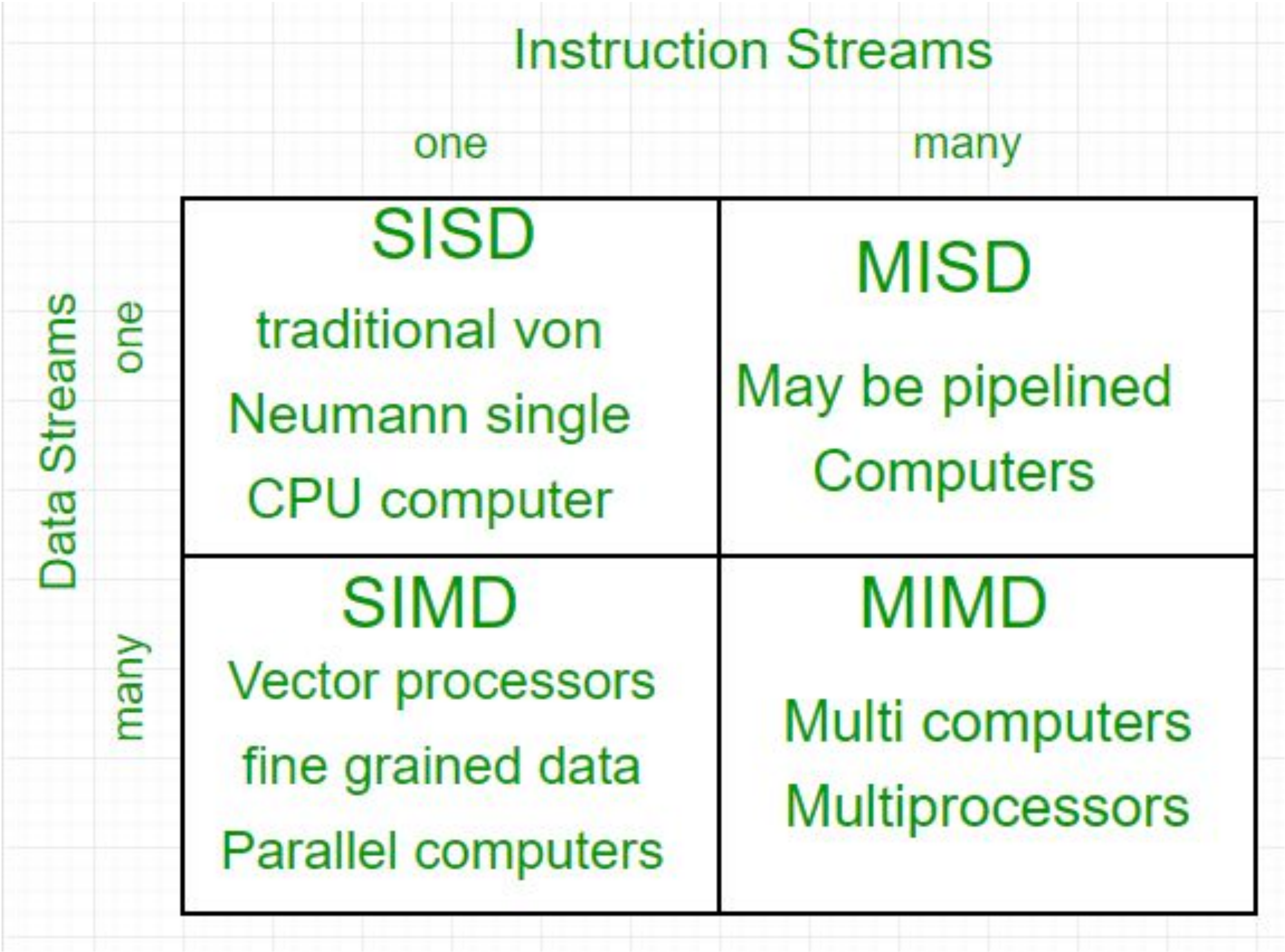


Course Outline

- Introduction
- Overview of Parallel Architectures
- Performance
- Parallel Programming
  - GPUs and CUDA programming
- Case studies
- Extracting Parallelism from Sequential Programs Automatically

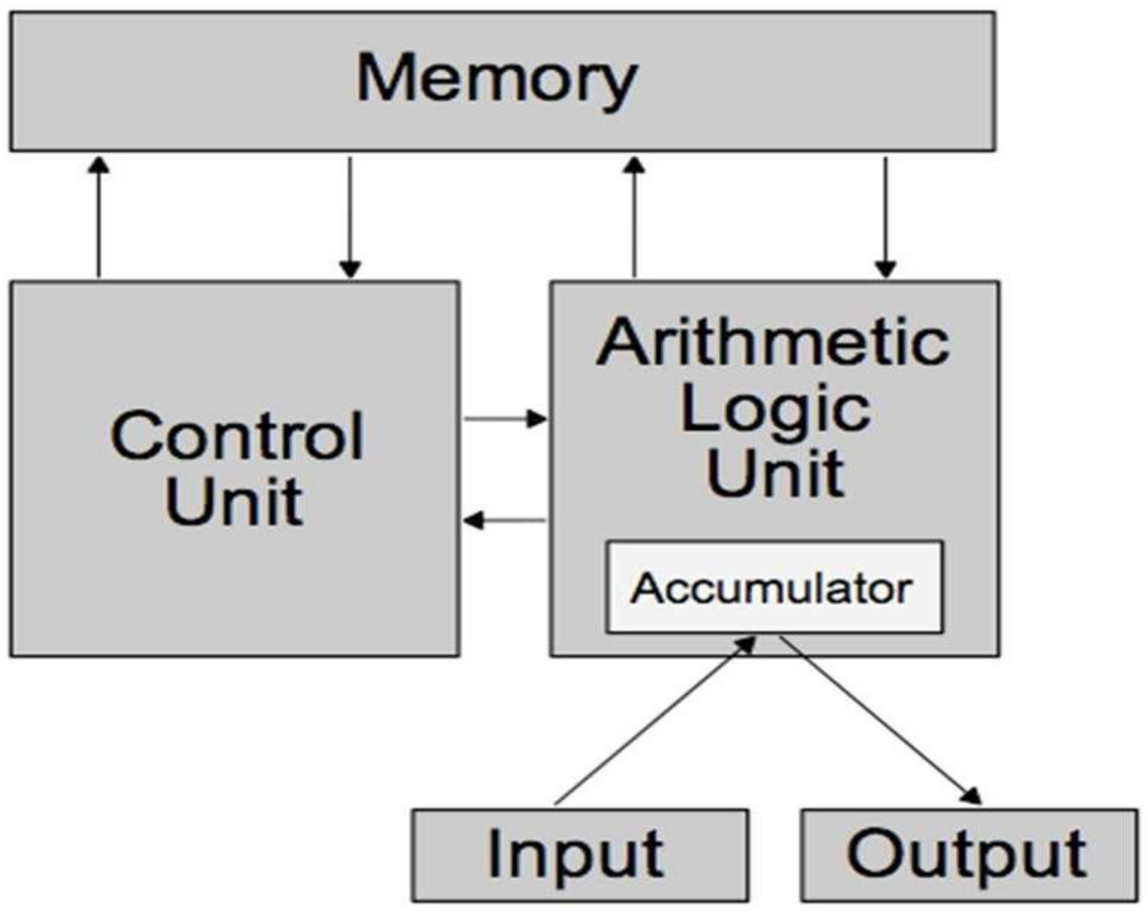
Flynn's Taxonomy

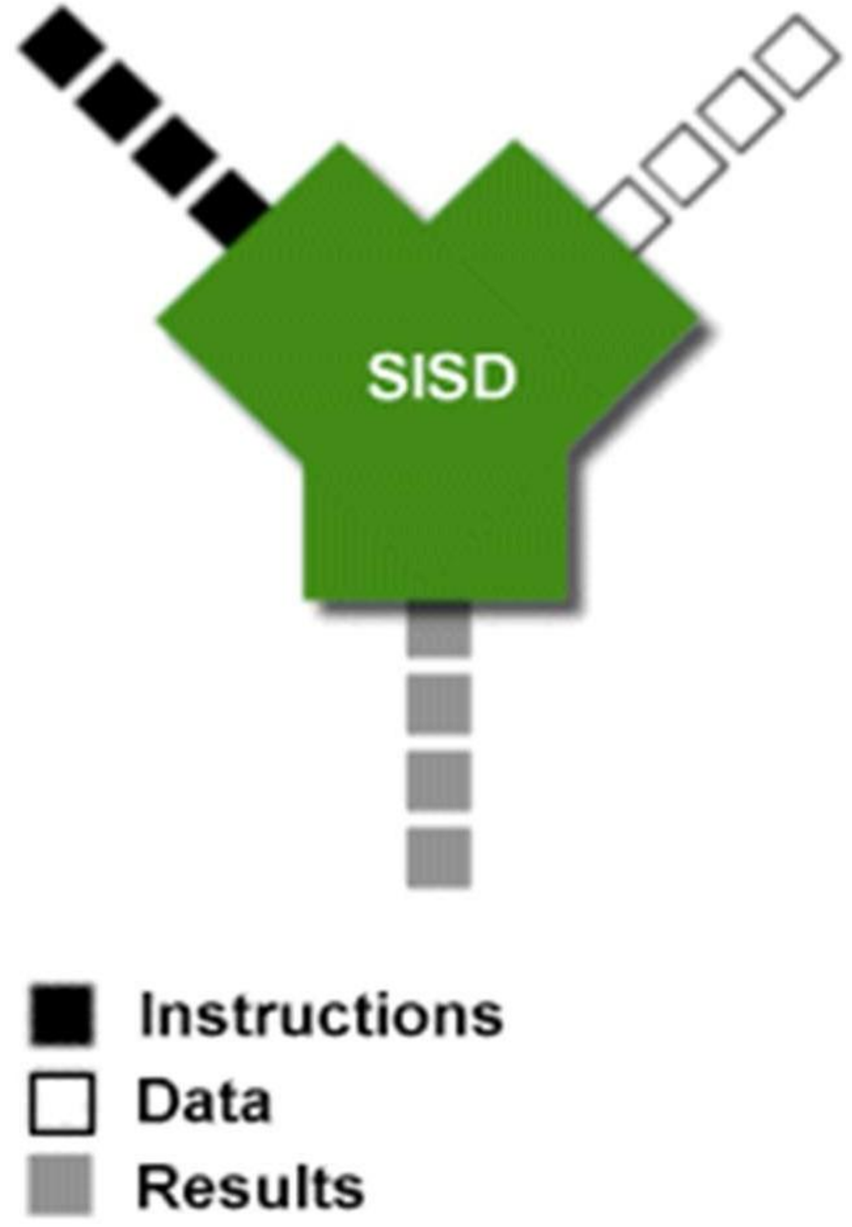
- Flynn's classification of computer architecture



SISD: Single Instruction, Single Data

- The von Neumann architecture
  - Implements a universal Turing machine
  - Conforms to serial algorithmic analysis



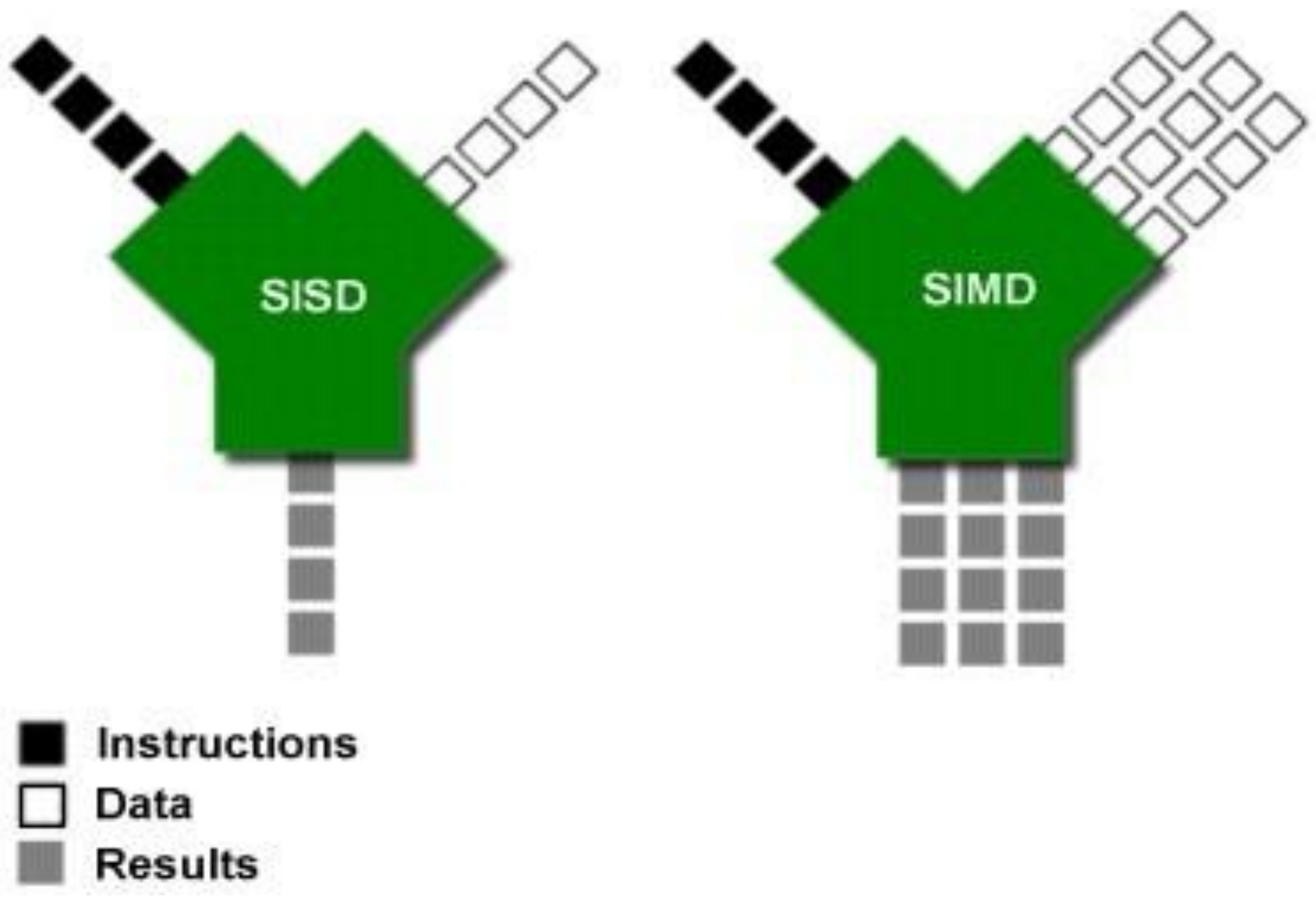


From http://arstechnica.com/paedia/c/ cpu/part-1/cpu1-1.html

SIMD: Single Instruction, Multiple Data

- Single control stream
  - All processors operating in lock step
  - Fine-grained parallelism

SIMD: Single Instruction, Multiple Data

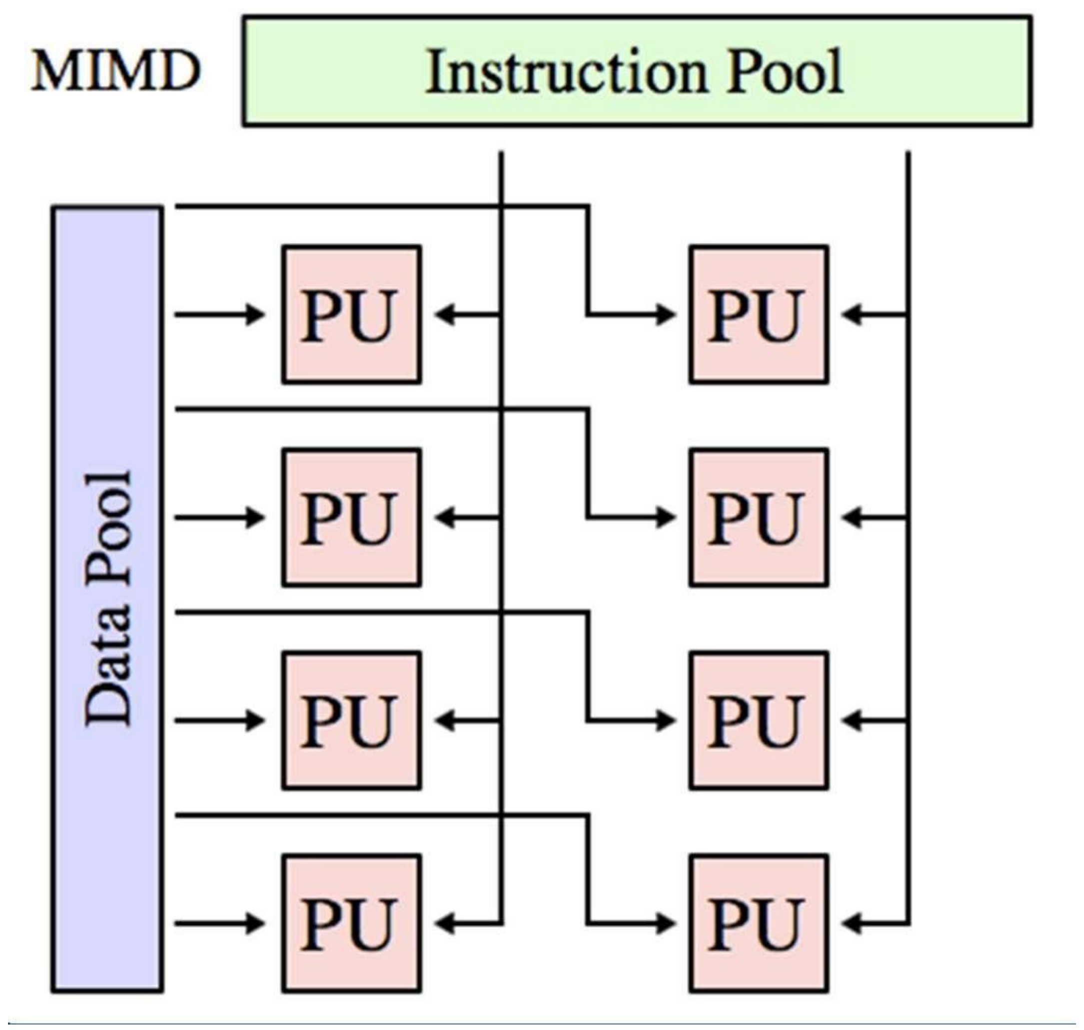


From http://arstechnica.com/paedia/c/c pu/part-1/cpu1-1.html

- Example: GPUs

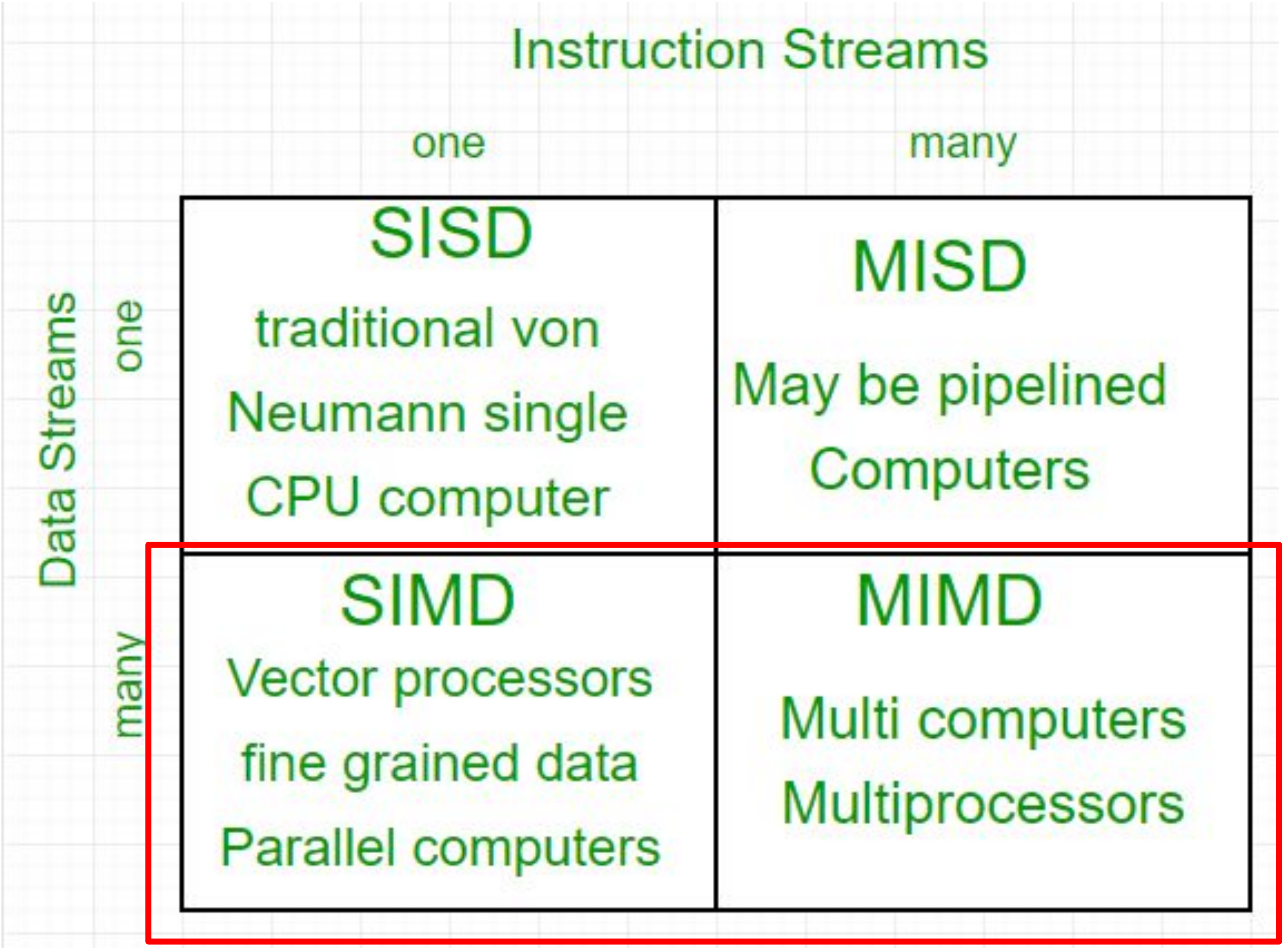
MIMD: Multiple Instructions, Multiple Data

- Most the machines that are prevalent
    - Multi-core, SMP, Clusters, NUMA machines, etc.



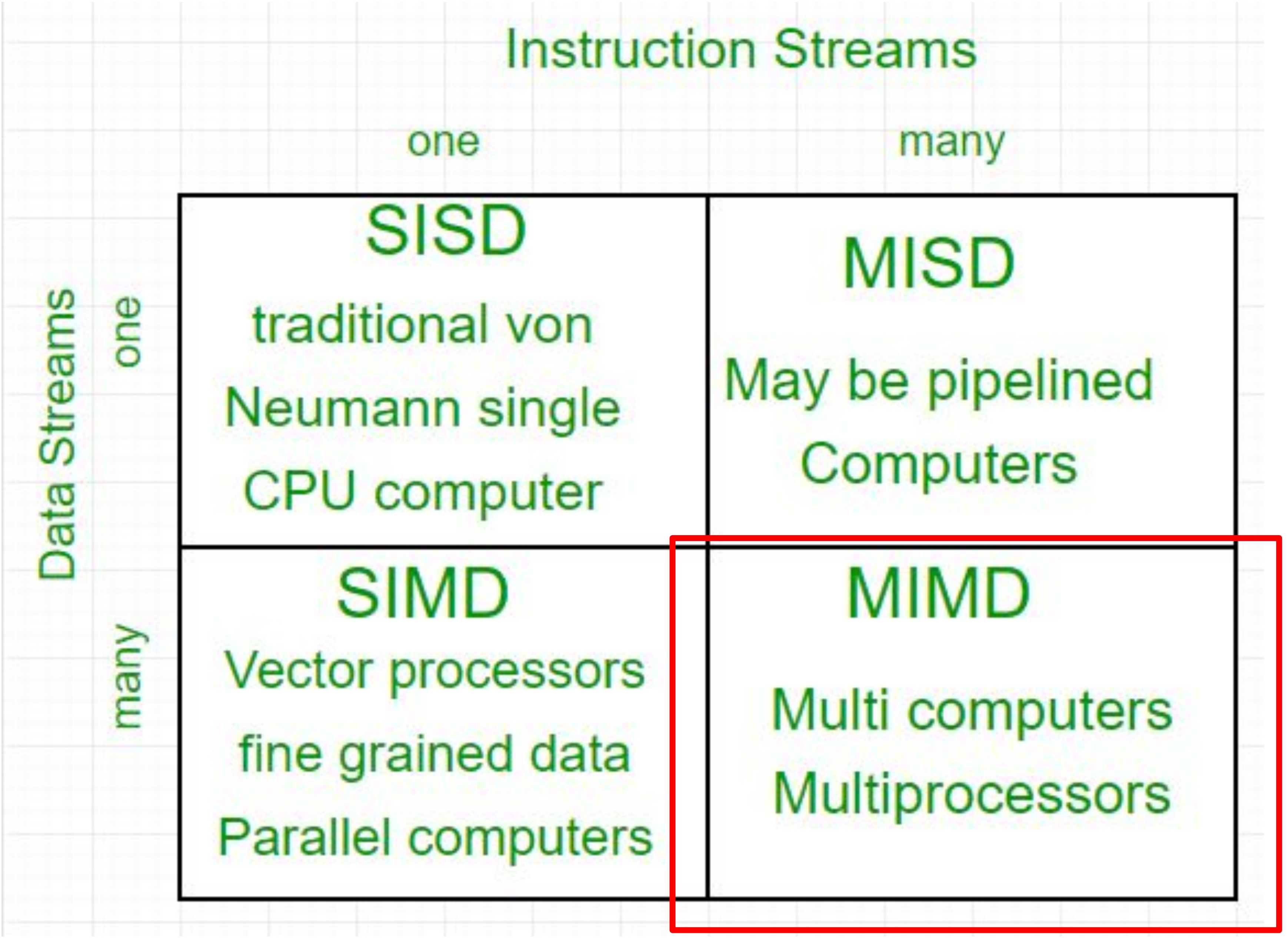
Rest of the today's lecture…

- Flynn's classification of computer architecture



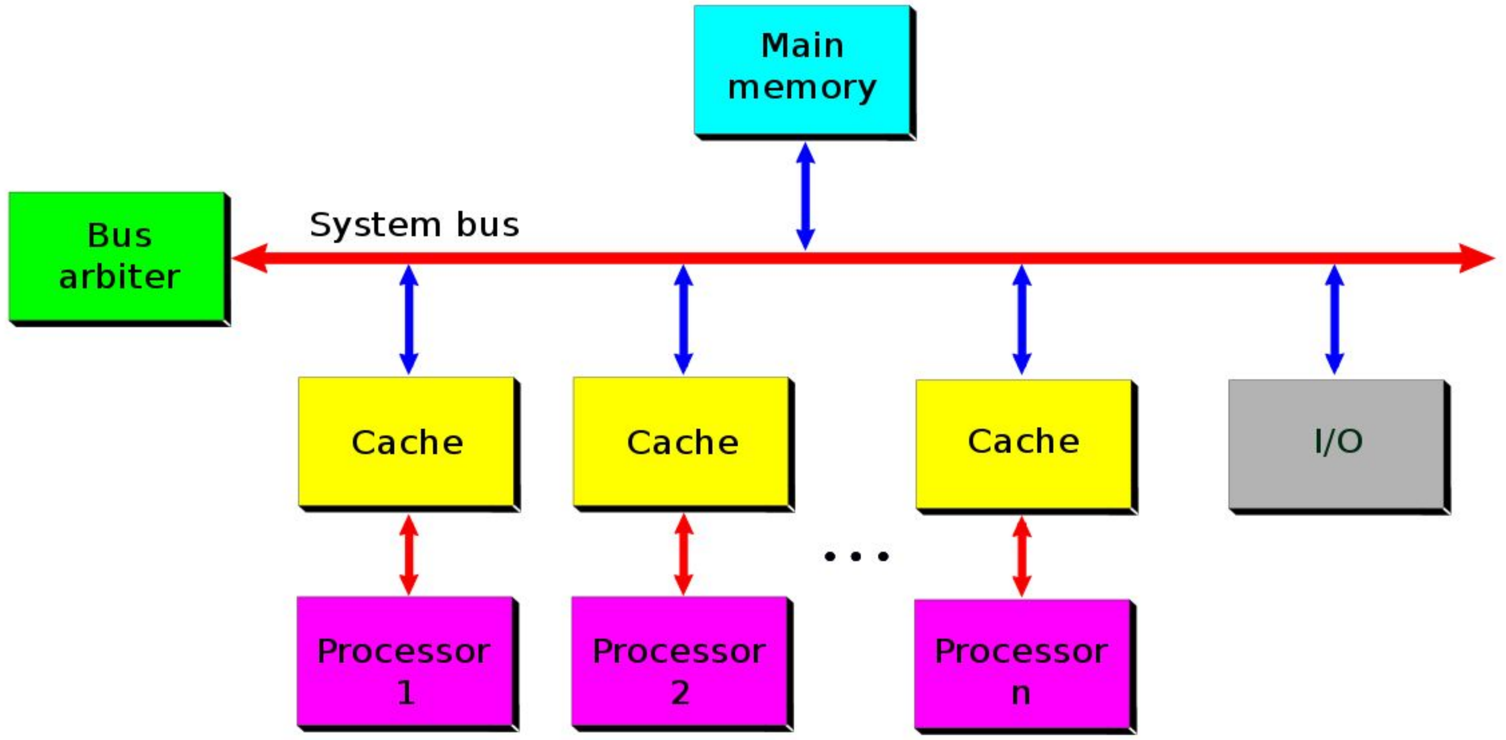
Flynn's Taxonomy

- Flynn's classification of computer architecture

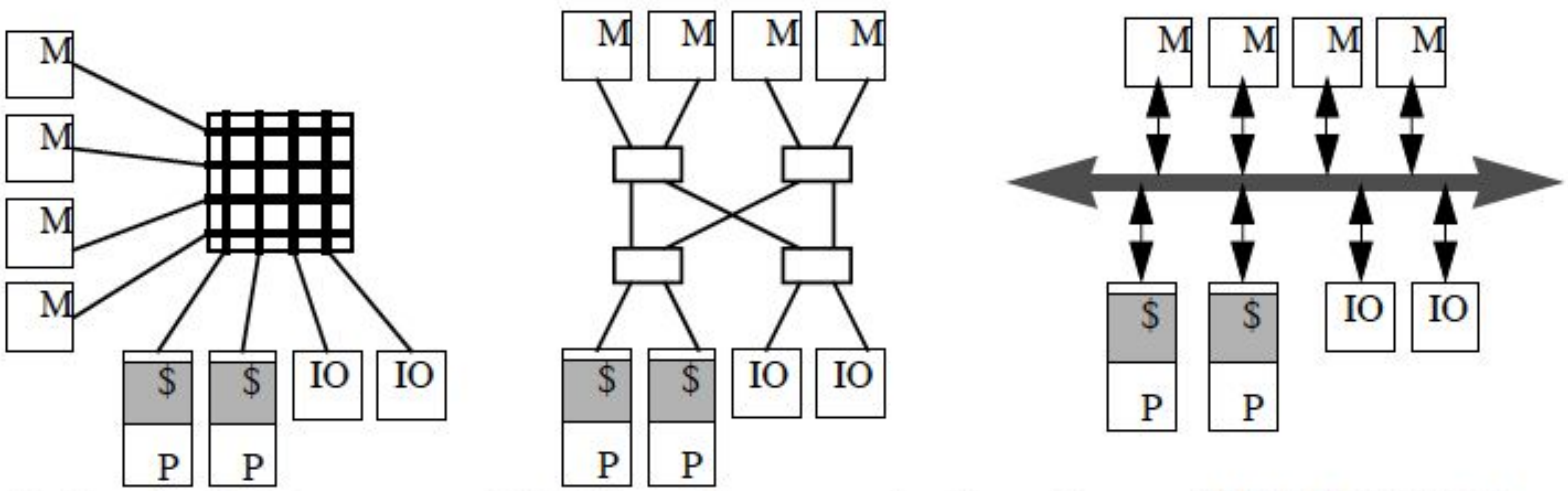


MIMD: Shared Memory Multiprocessors

- Tightly coupled multiprocessors
  - Shared global memory address space
  - Traditional multiprocessing: symmetric multiprocessing (SMP)
    - Existing multi-core processors, multithreaded processors
  - Programming model similar to uniprocessors (i.e., multitasking uniprocessor) except
    - Operations on shared data require synchronization

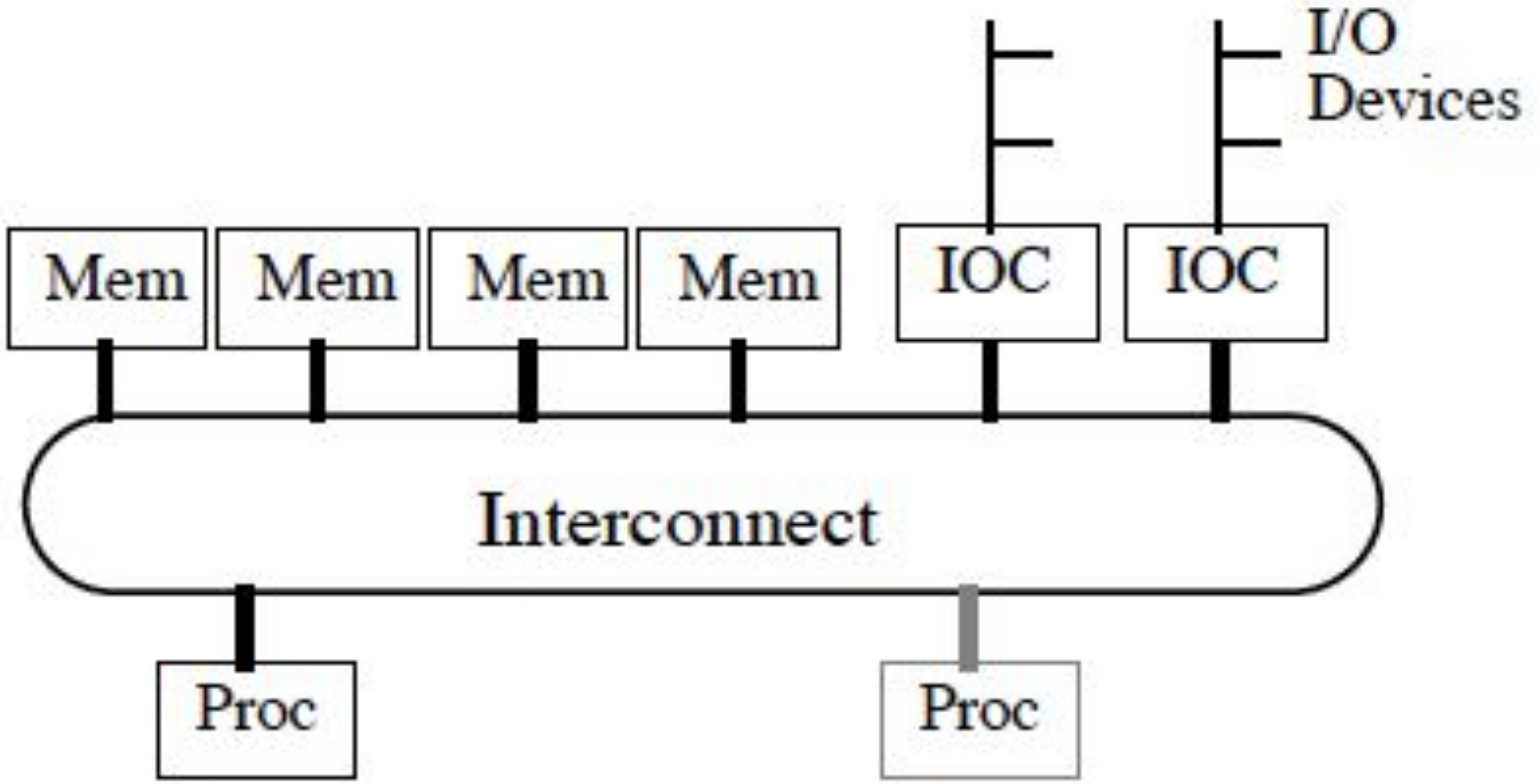


Interconnection Schemes for SMP



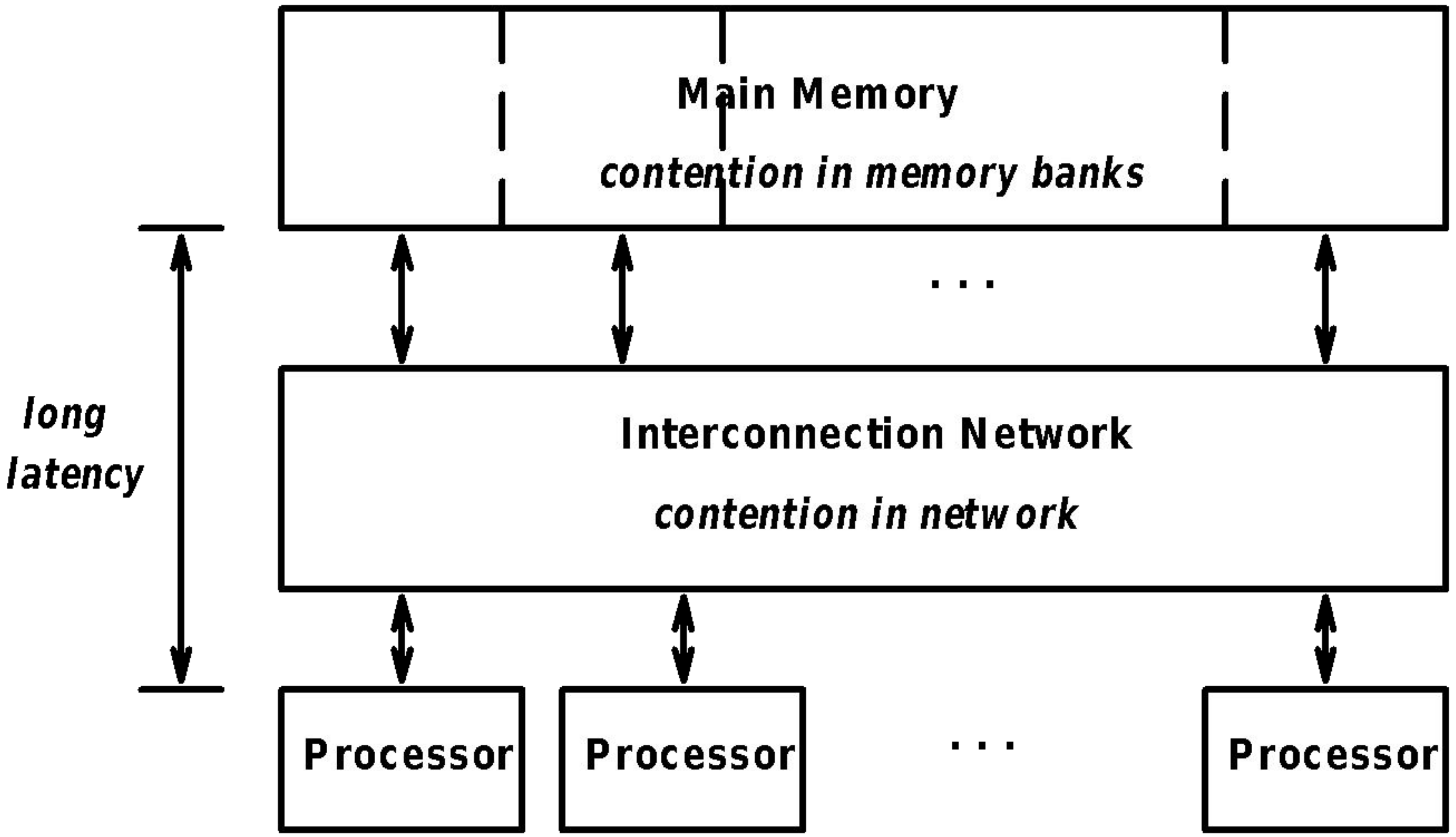
(a) Cross-bar Switch (b) Multistage Interconnection Network (c) Bus Interconnect

SMP Architectures



UMA: Uniform Memory Access

- All processors have the same uncontended latency to memory
- Symmetric multiprocessing (SMP) ~ UMA with bus interconnect

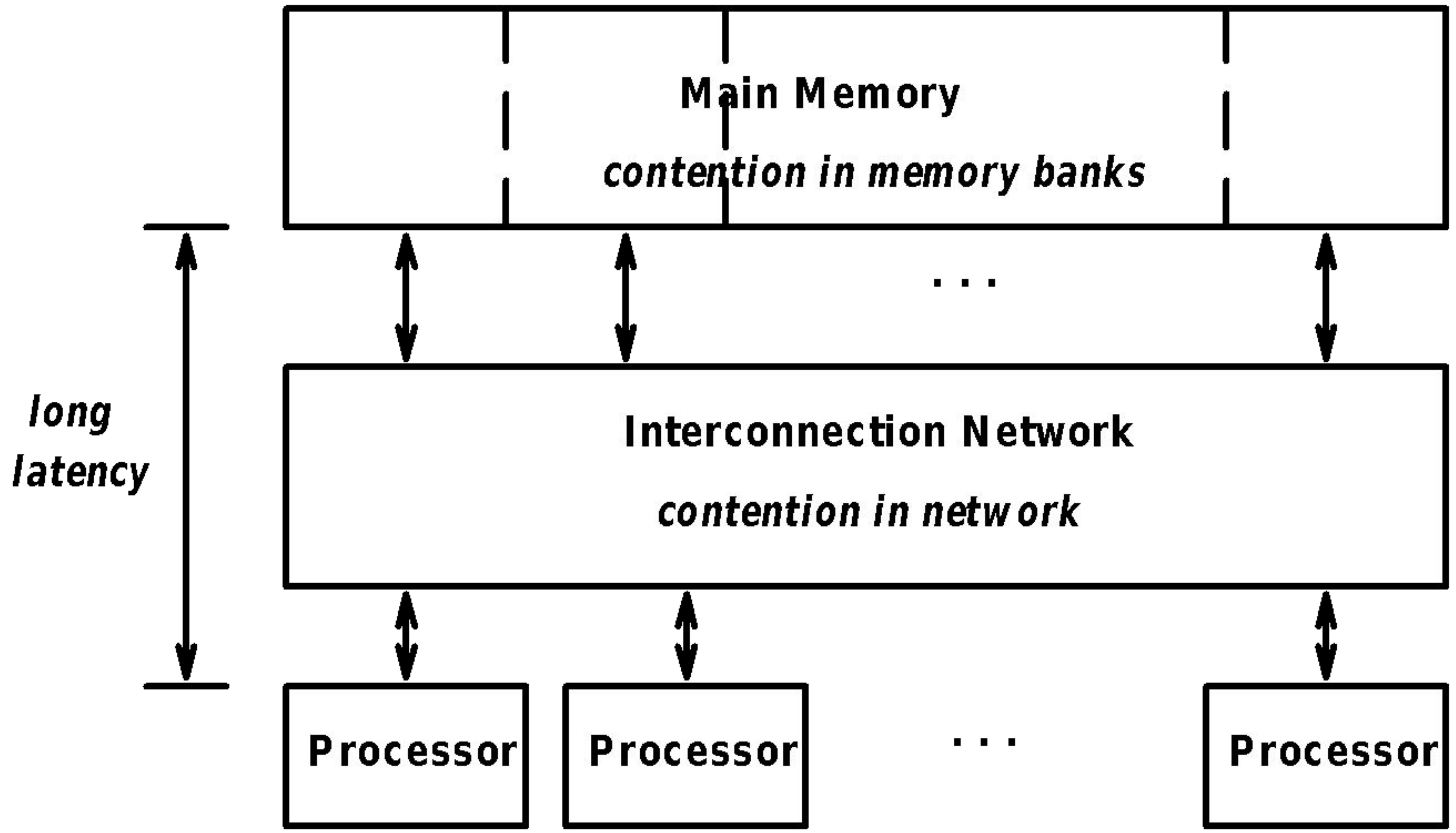


UMA: Uniform Memory Access

+ Data placement unimportant/less important (easier to optimize code and make use of available memory space)

- Scaling the system increases all latencies

- Contention could restrict bandwidth and increase latency



How to Scale Shared Memory Machines?

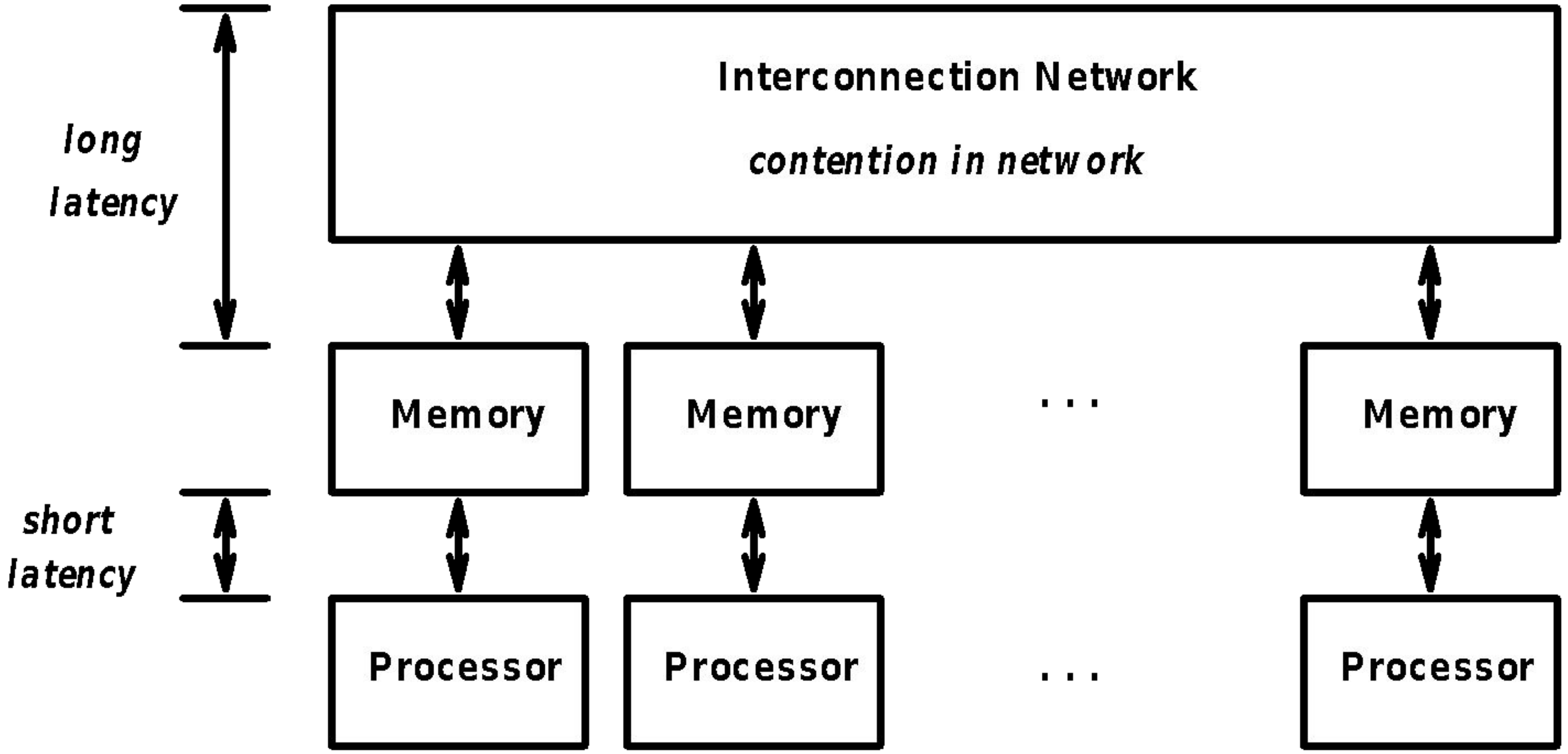
- Two general approaches

- Maintain UMA
  - Provide a scalable interconnect to memory
  - Scaling system increases memory latency

- Interconnect complete processors with local memory
  - NUMA (Non-uniform memory access)
    - Local memory faster than remote memory
  - Still needs a scalable interconnect for accessing remote memory

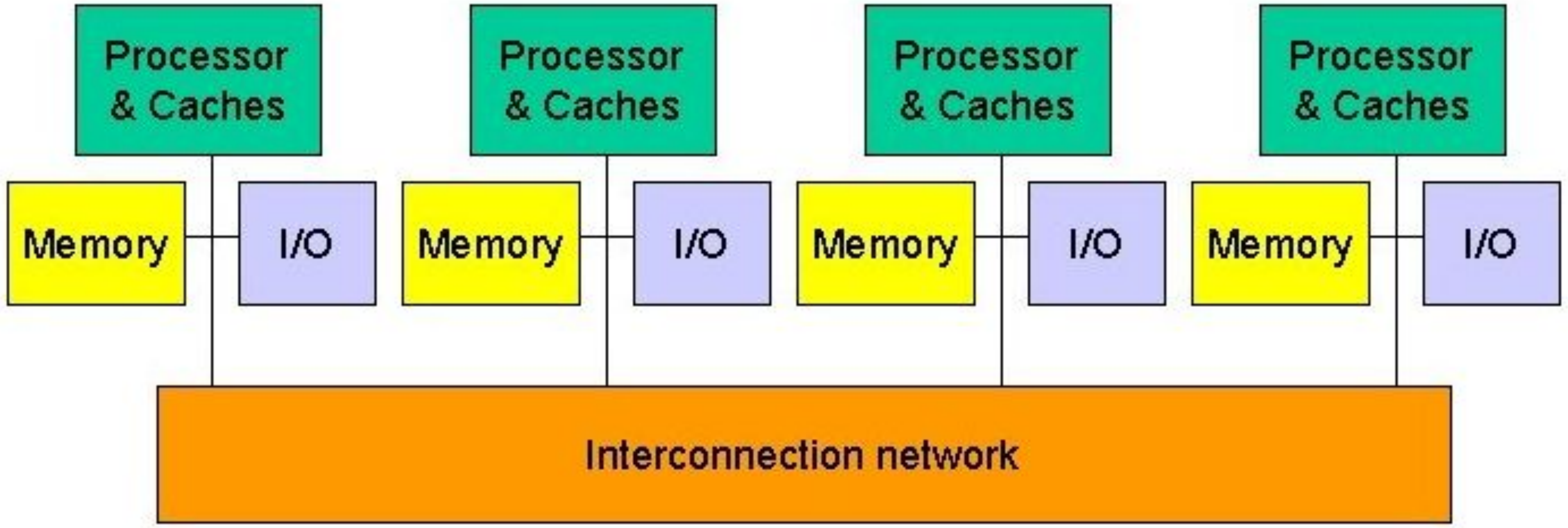
NUMA: Non Uniform Memory Access

- Shared memory as local versus remote memory
  + Low latency to local memory
  - Much higher latency to remote memories
  + Bandwidth to local memory may be higher
  - Performance very sensitive to data placement

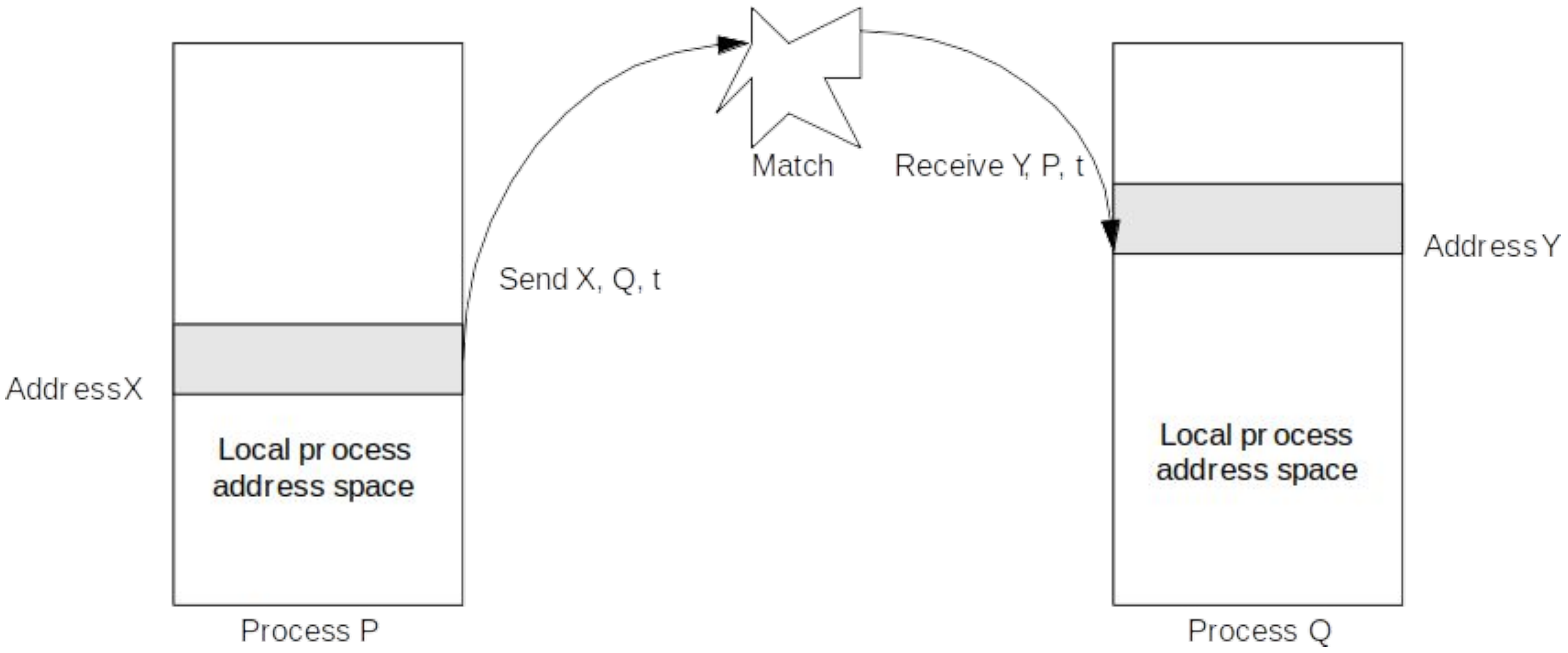


MIMD: Message Passing Architectures

- Loosely coupled multiprocessors
  - No shared global memory address space
  - Multicomputer network
  - Network-based multiprocessors
  - Usually programmed via message passing
  - Explicit calls (send, receive) for communication

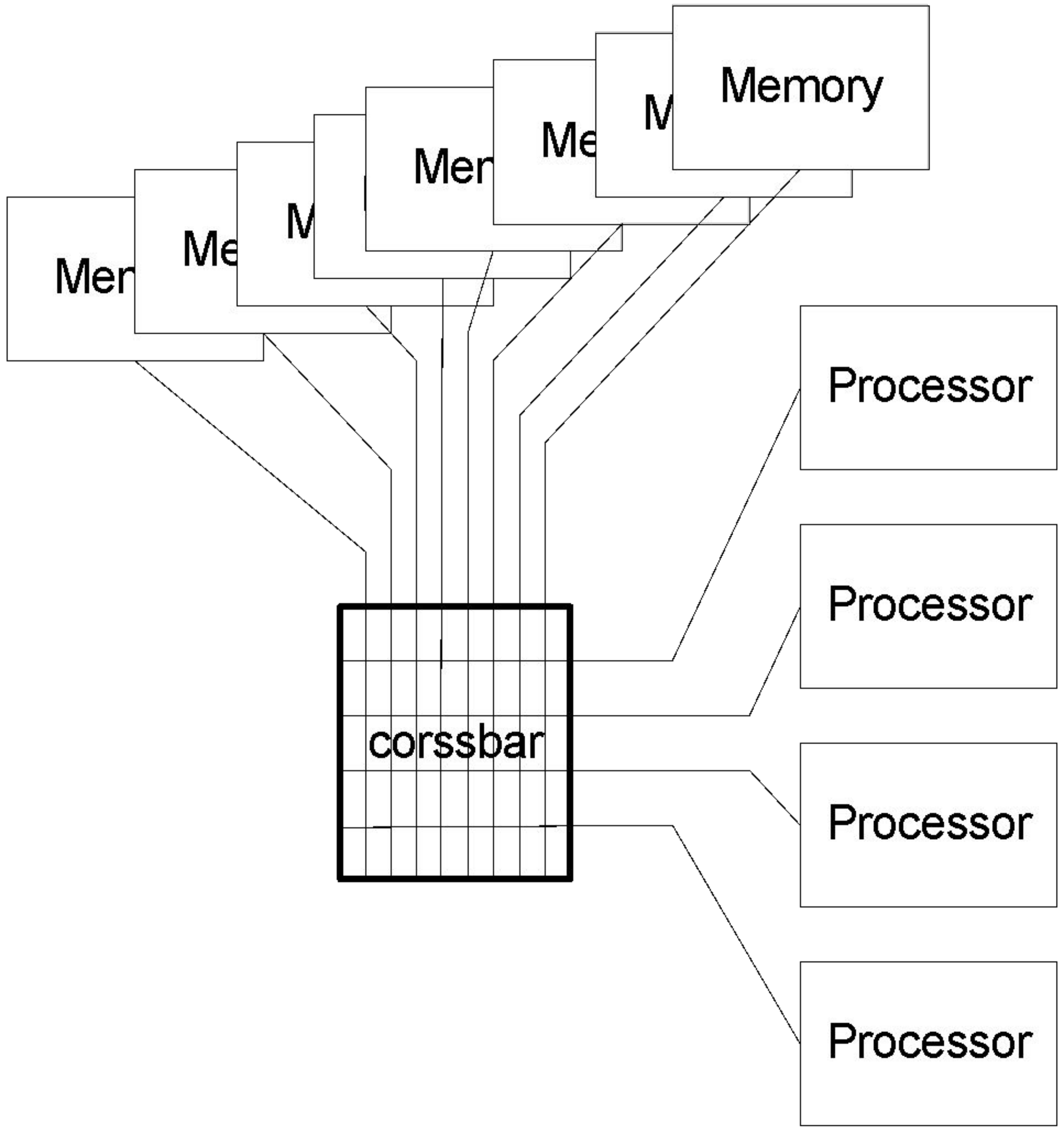


MIMD: Message Passing Architectures



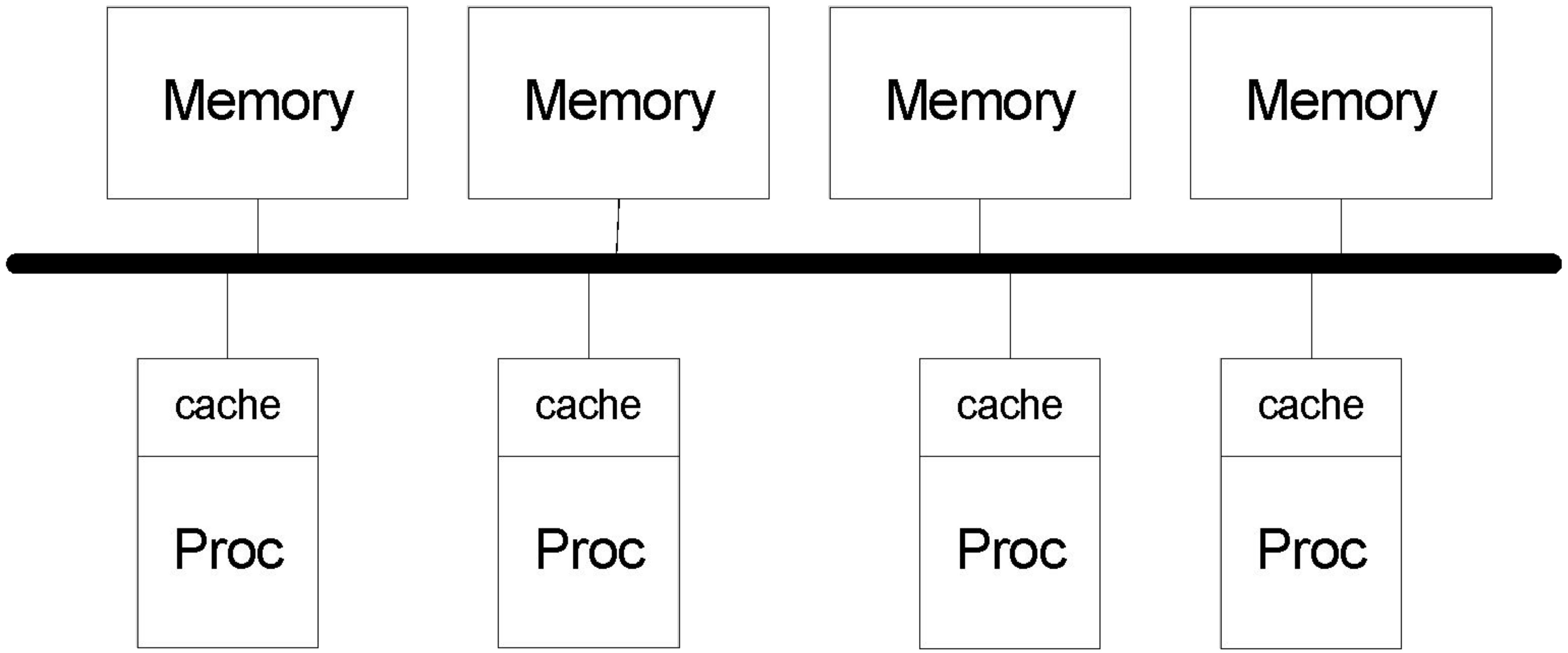
Historical Evolution: 1960s & 70s

- Early MPs
  - Mainframes
  - Small number of processors
  - crossbar interconnect
  - UMA



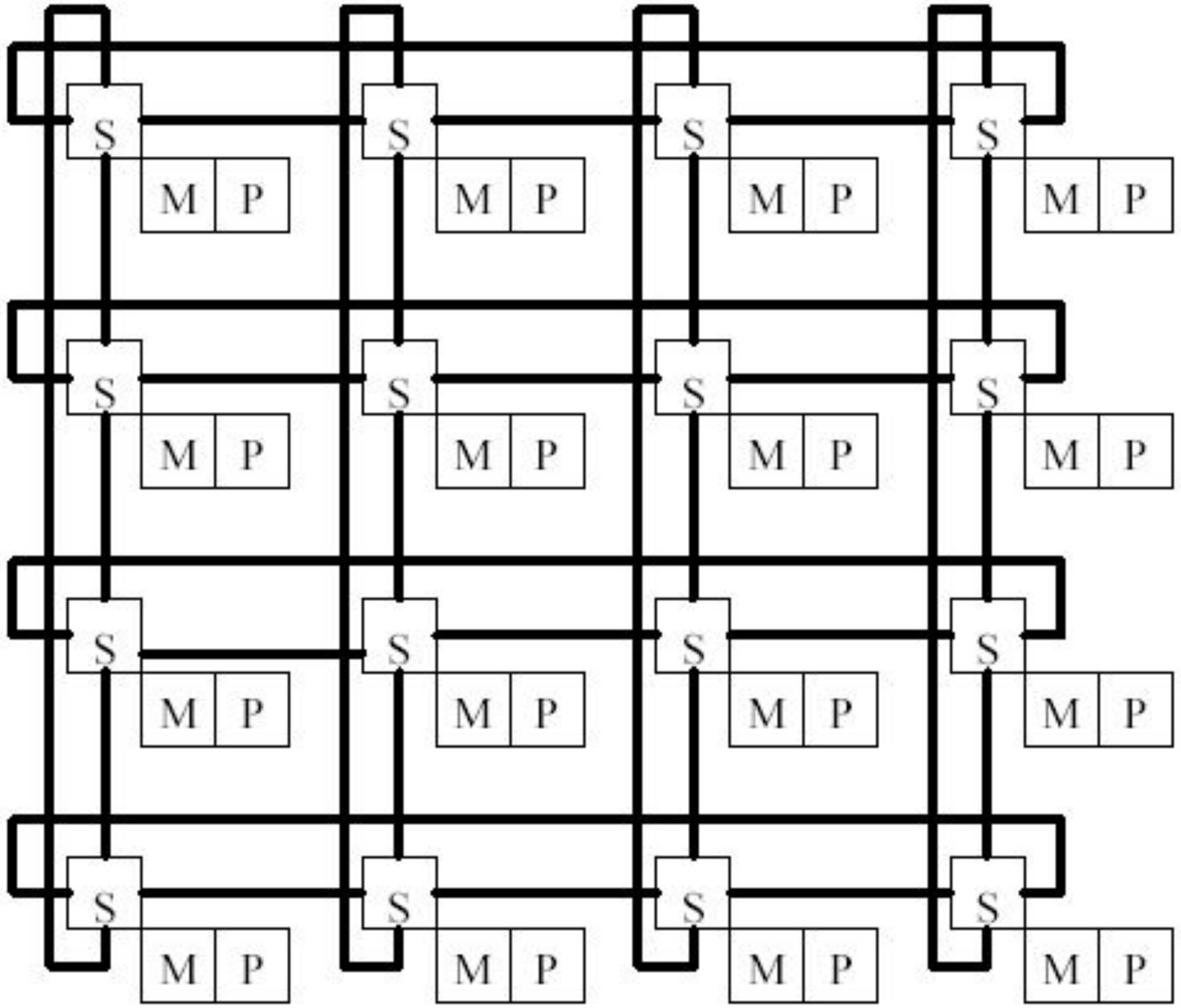
Historical Evolution: 1980s

- Bus-Based MPs
  - enabler: processor-on-a-board
  - economical scaling
  - precursor of today's SMPs
  - UMA



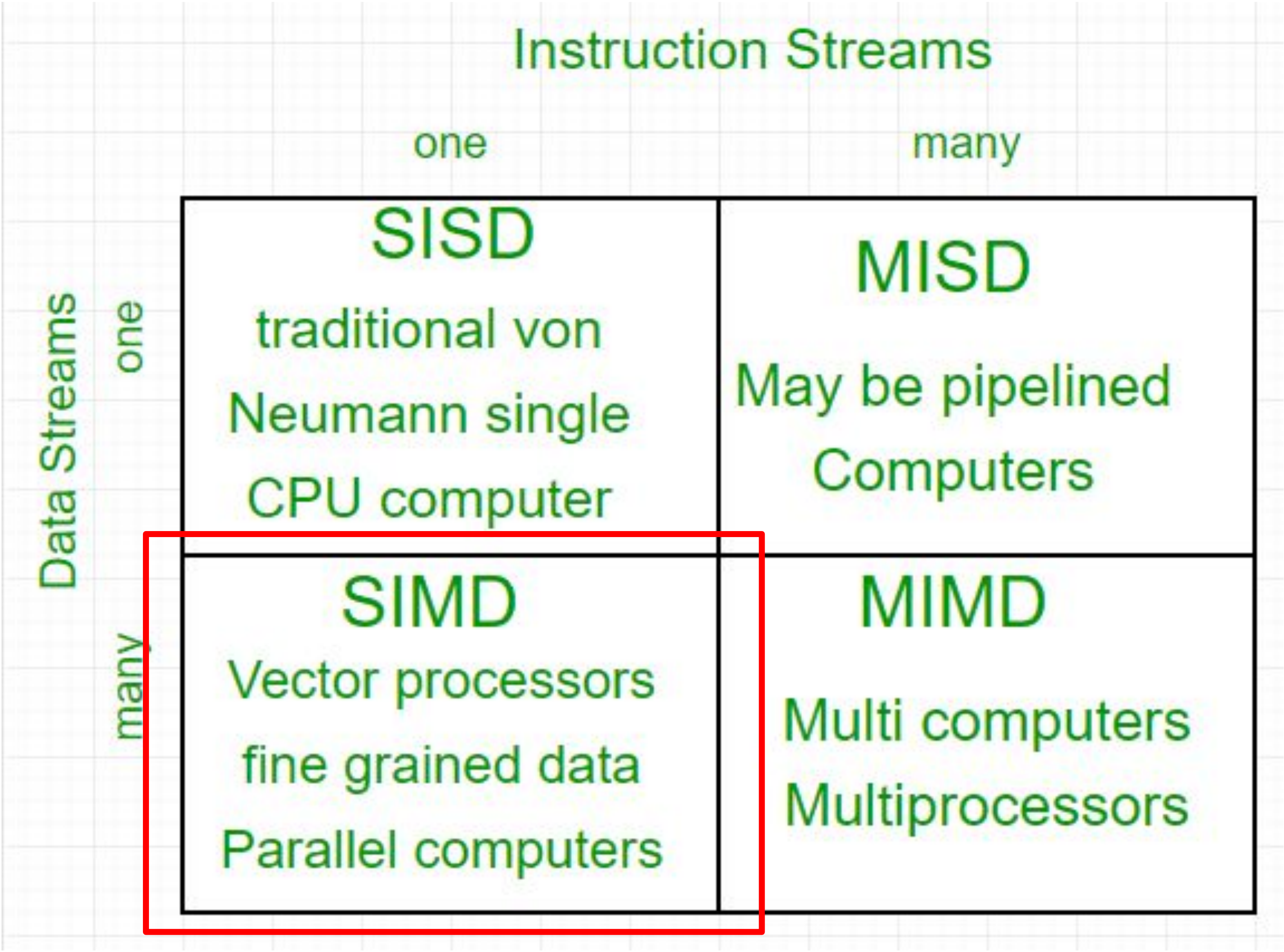
Historical Evolution: Late 80s, mid 90s

- Large Scale MPs (Massively Parallel Processors)
  - multi-dimensional interconnects
  - each node a computer (proc + cache + memory)
  - NUMA
  - still used for “supercomputing”

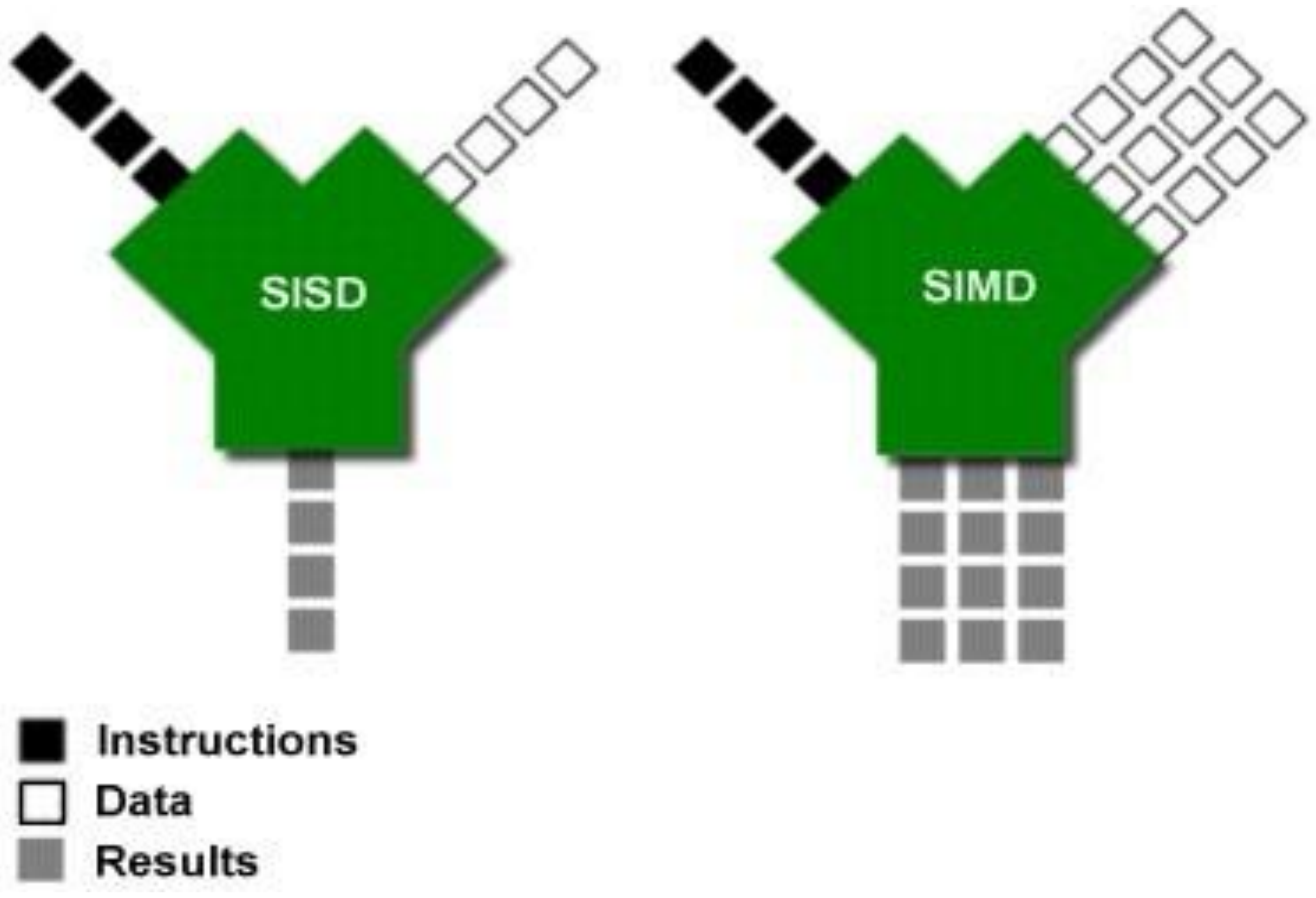


Flynn’s Taxonomy

- Flynn’s classification of computer architecture



SIMD: Single Instruction, Multiple Data



From http://arstechnica.com/paedia/c/c pu/part-1/cpu1-1.html

- Example: GPUs

Data Parallel Programming Model

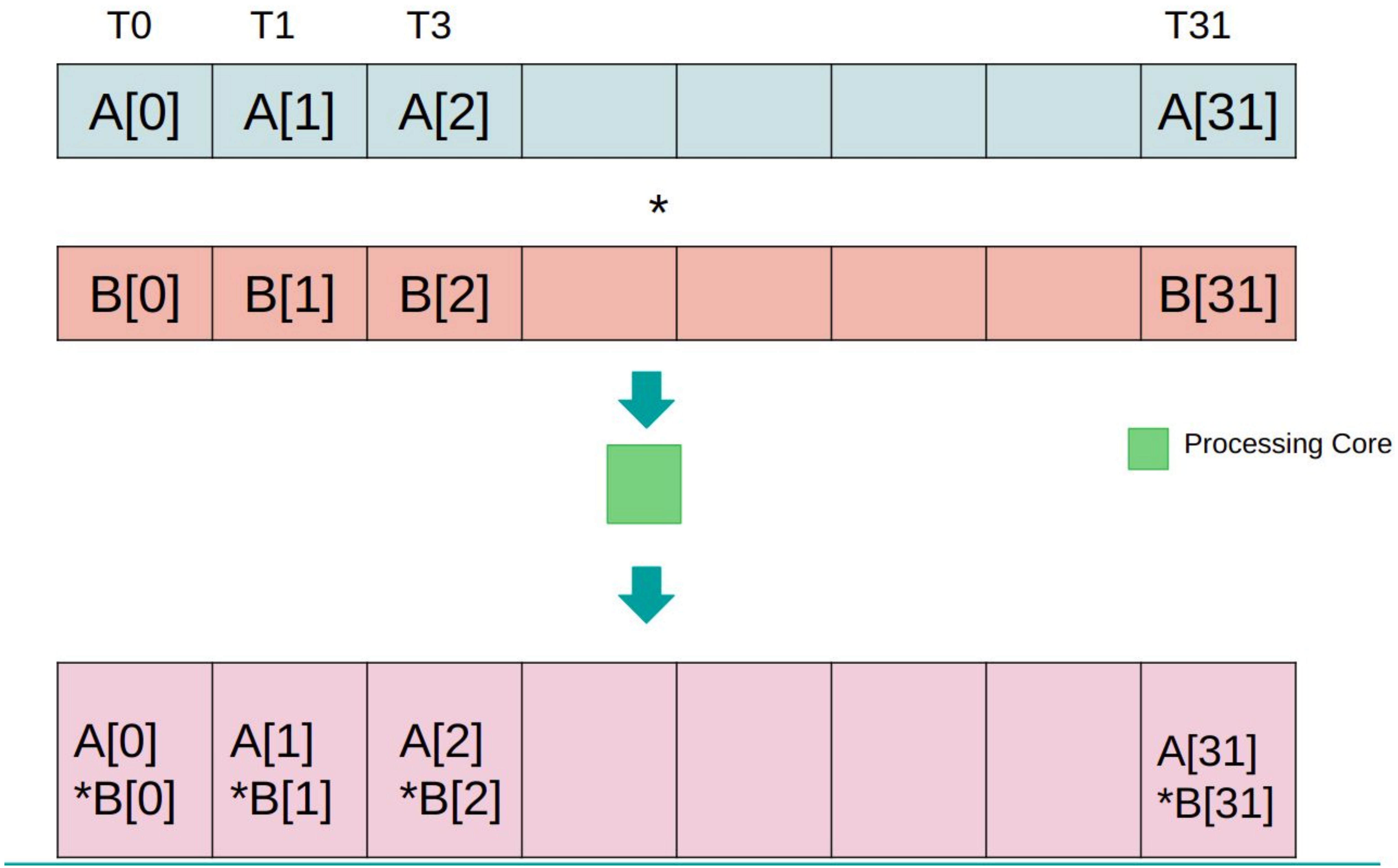
- Programming Model
  - Operations are performed on each element of a large (regular) data structure (array, vector, matrix)

- Simple example (A, B and C are vectors)

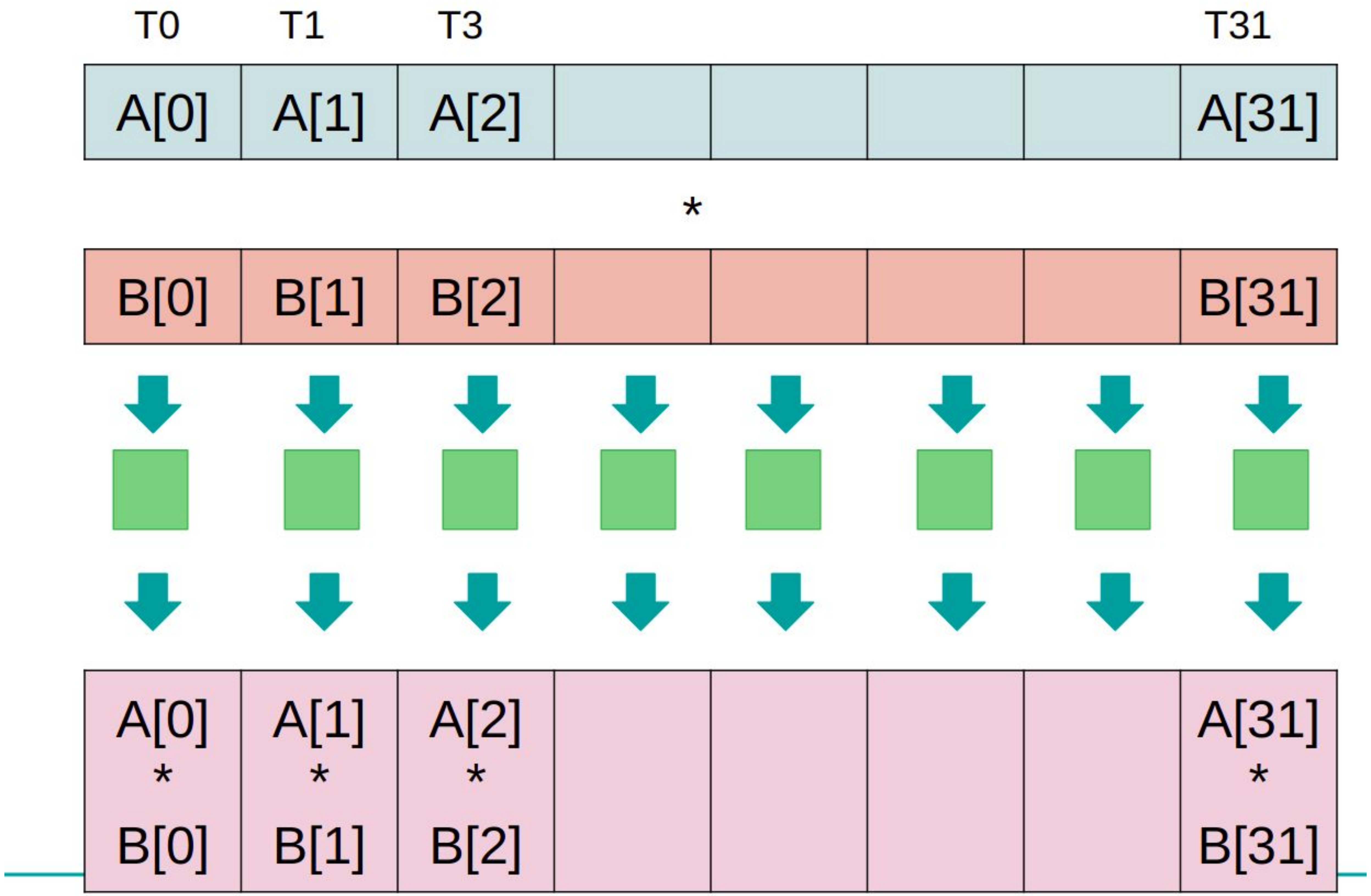
$$C = (A * B)$$

- The operations can be executed in sequential or parallel steps
- Language supports array assignment

On Sequential Hardwares



On Data Parallel Hardwares



Data Parallel Architectures

- Early architectures directly mirrored programming model

- Single control processor (broadcast each instruction to an array/grid of processing elements)

- Examples: Connection Machine, MPP (Massively Parallel Processor)

Data Parallel Architectures

- Later data parallel architectures
  - Higher integration $\rightarrow$ SIMD units on chip along with caches
  - More generic $\rightarrow$ multiple cooperating multiprocessors (GPUs)
  - Specialized hardware support for global synchronization

SIMD: Graphics Processing Units

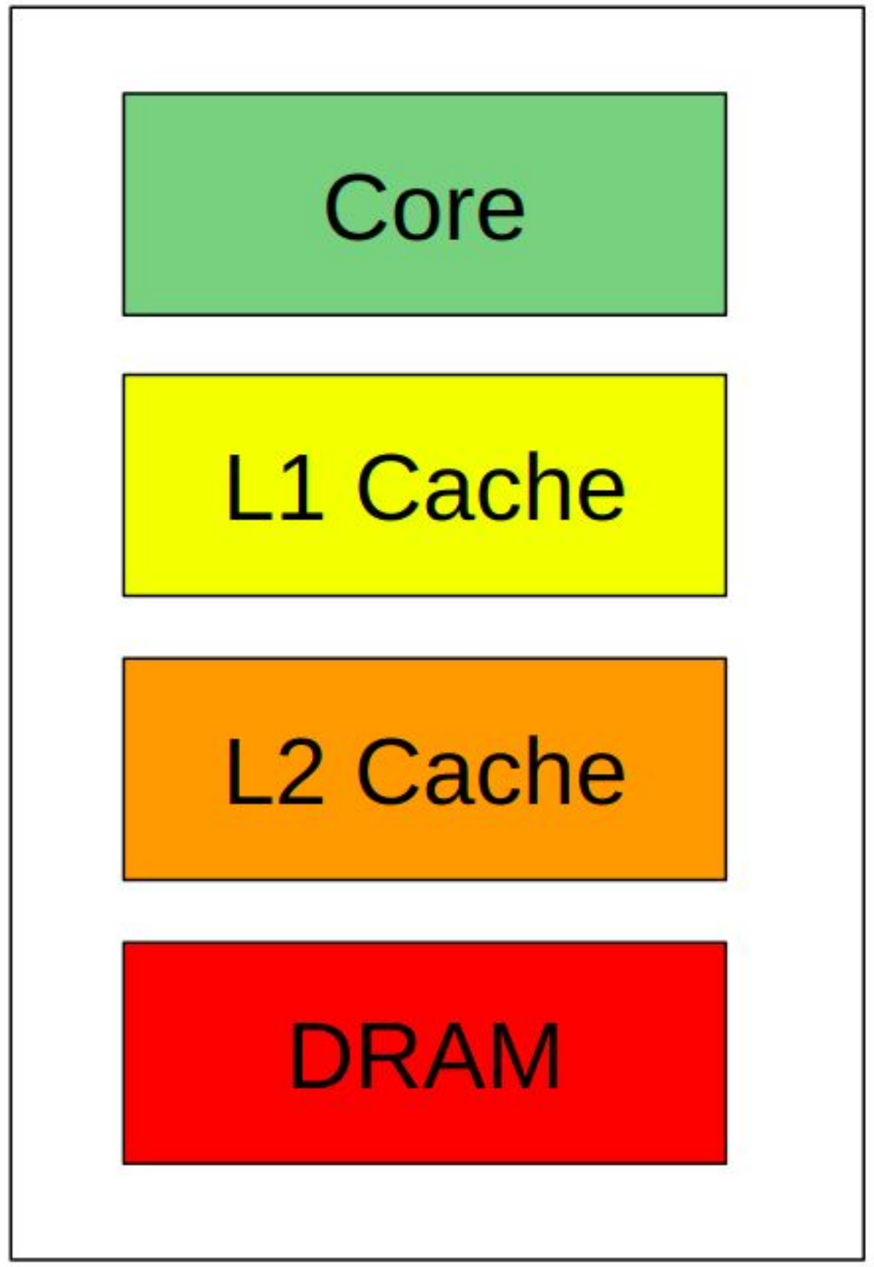
- The early GPU designs
  - Specialized for graphics processing only
  - Exhibit SIMD execution
  - Less programmable

- In 2007, fully programmable GPUs
  - CUDA released

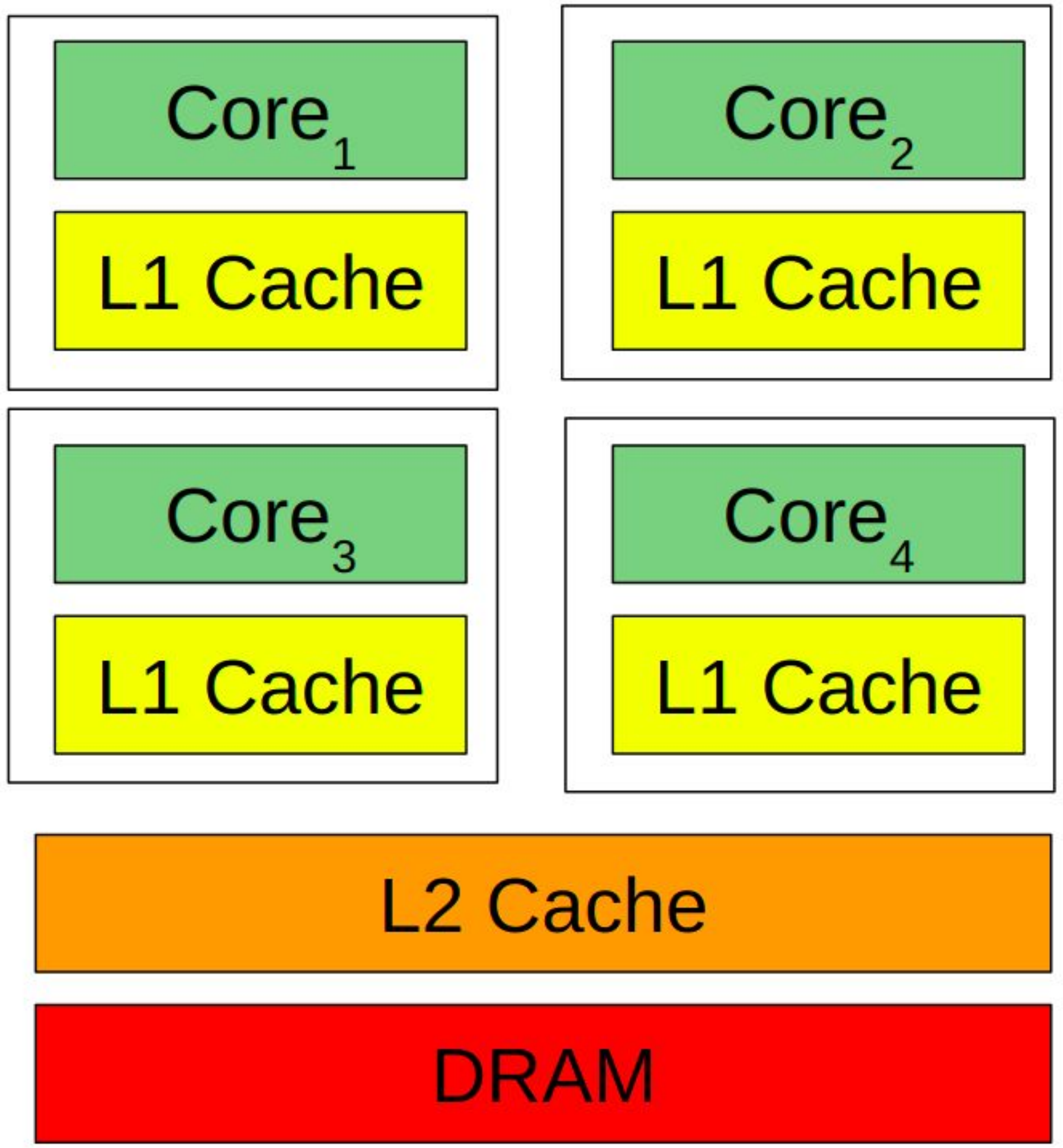
NVIDIA GeForce 256

NVIDIA CUDA

Single-core CPU vs Multi-core vs GPU

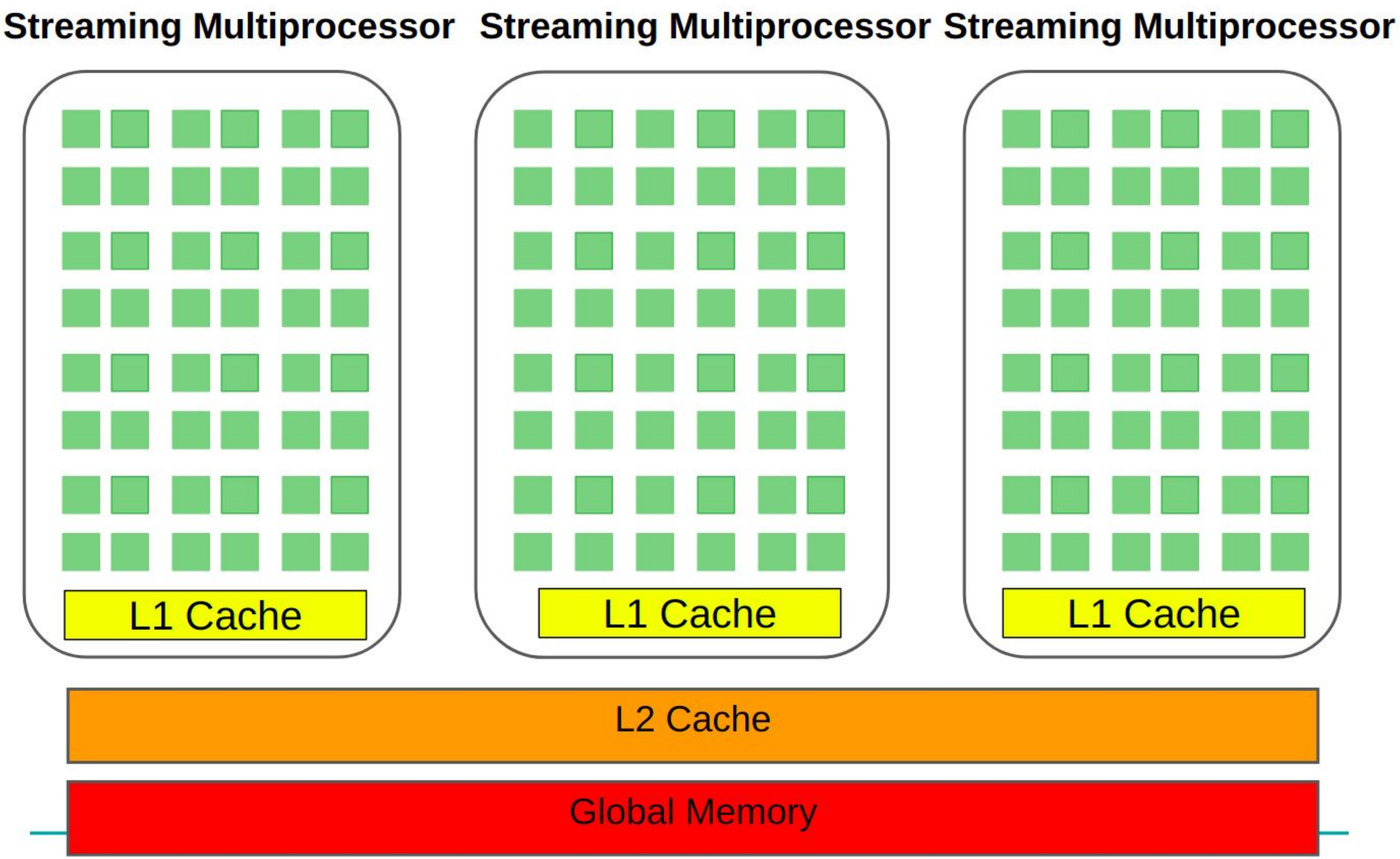


Single-core CPU



Multi-core CPU

Single-core CPU vs Multi-core vs GPU



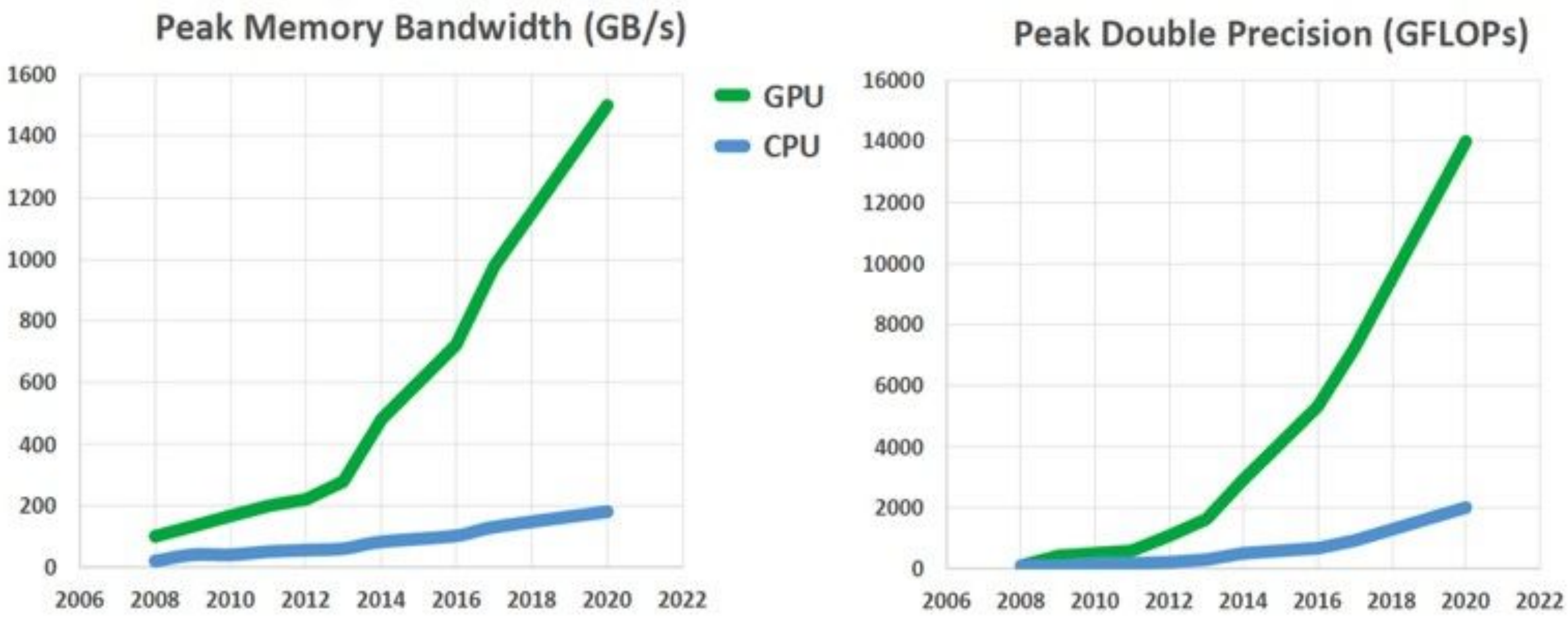
NVIDIA V100 GPU

https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf

Specifications

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm^2	601 mm^2	610 mm^2	815 mm^2
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

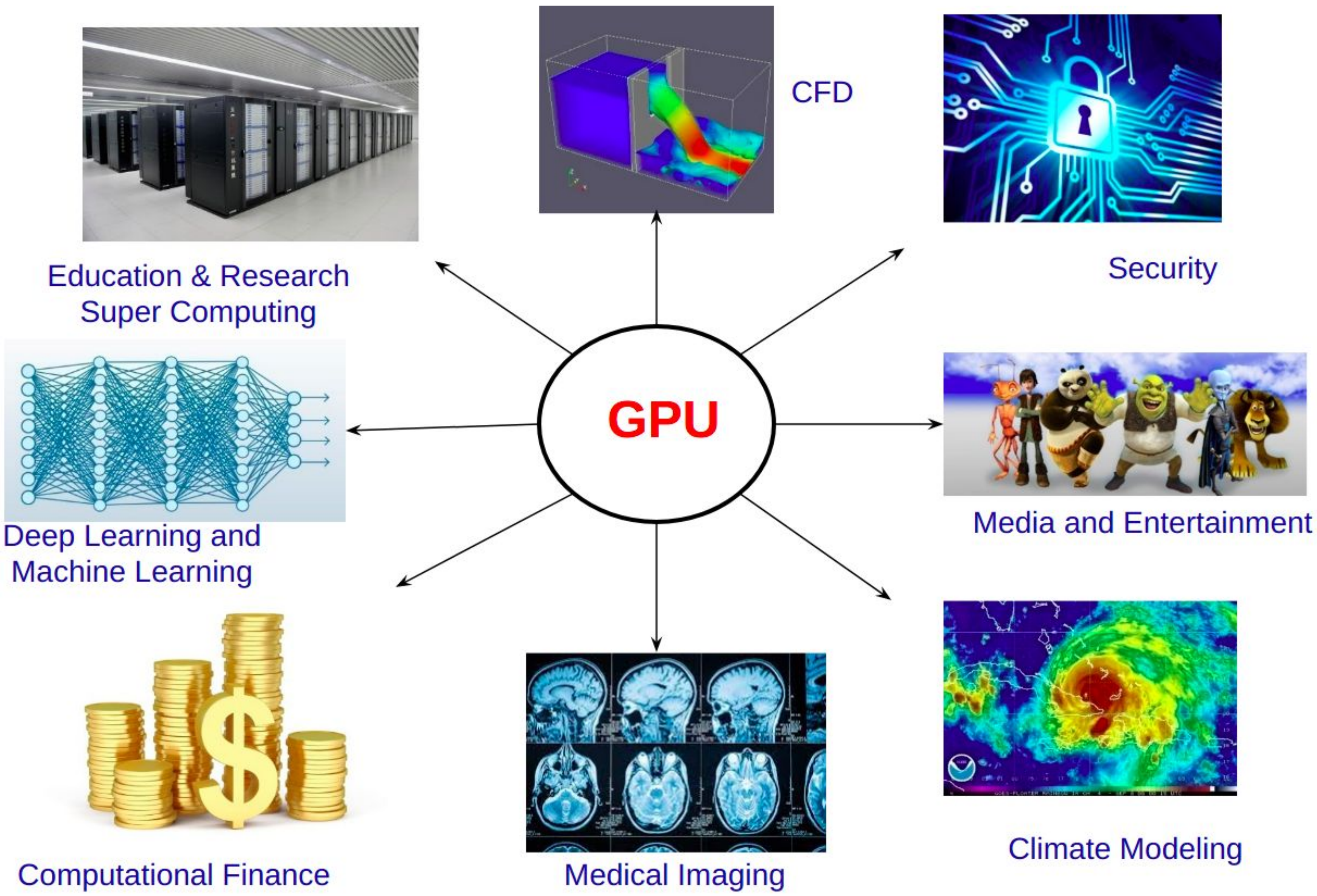
CPUs vs GPUs



Chip to chip comparison of peak memory bandwidth in GB/s and peak double precision gigaflops for GPUs and CPUs since 2008.

https://www.nextplatform.com/2019/07/10/a-decade-of-accelerated-computing-augurs-well-for-gpus

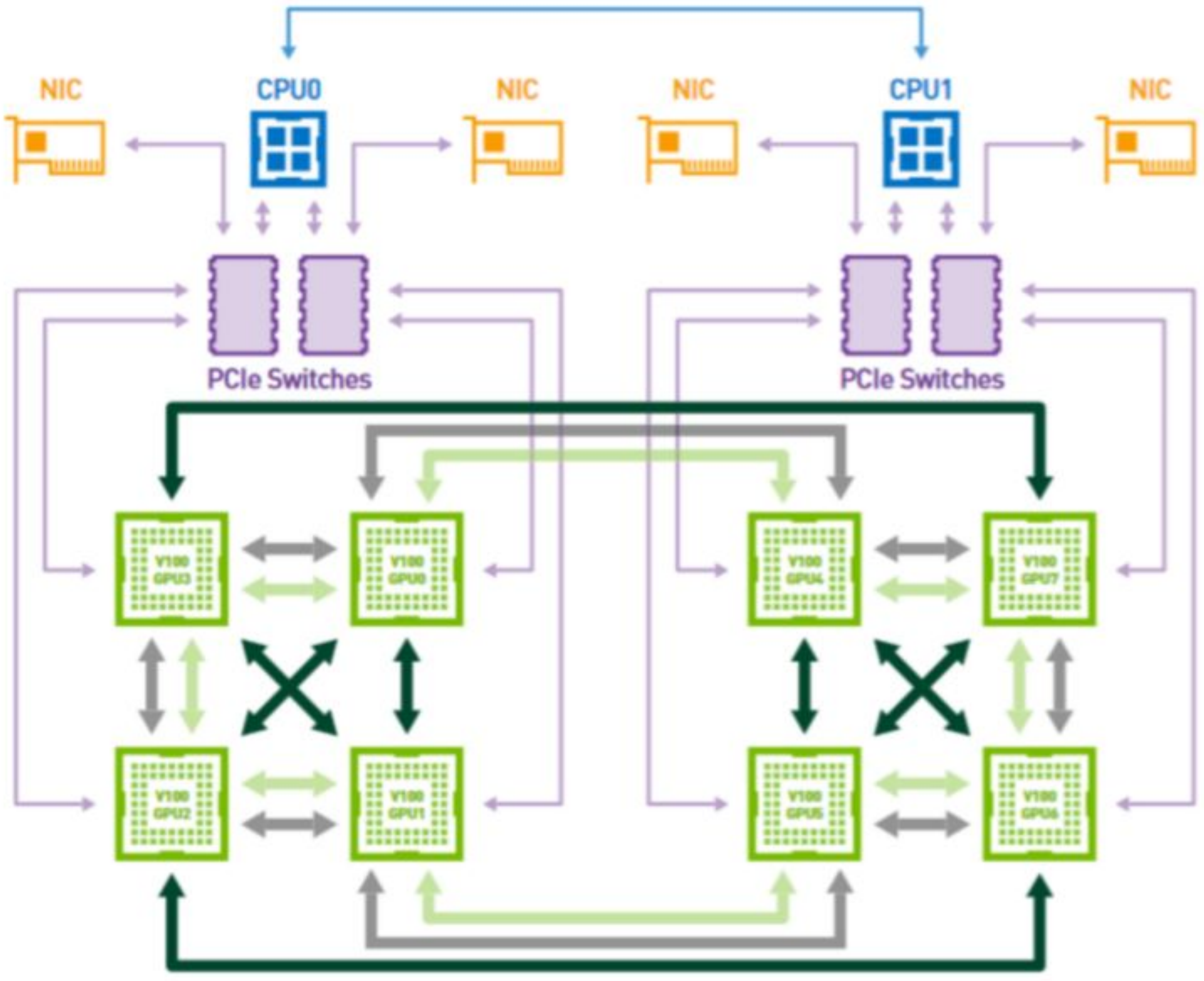
GPU Applications



Specifications

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

Multi-GPU Systems



https://www.azken.com/images/dgx1_images/dgx1-system-architecture-whitepaper1.pdf

Summary

- Parallel architectures are inevitable
- Different architectures are evolved
- Flynn’s taxonomy:
  - SISD
  - MISD
  - MIMD
  - SIMD

References

- David Culler, Jaswinder Pal Singh, and Anoop Gupta. 1998. Parallel Computer Architecture: A Hardware/Software Approach. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA
- https://safari.ethz.ch/architecture/fall2020/doku.php?id=schedule
- https://www.cse.iitd.ac.in/~soham/COL380/page.html
- https://s3.wp.wsu.edu/uploads/sites/1122/2017/05/6-9-2017-slides-vFinal.pptx
- https://ebhor.com/full-form-of-cpu/
- Miscellaneous resources on internet

Thank You