

Programming Techniques for Supercomputers:

Parallel Computers: Shared Memory

Modern multi- and manycore chips

Parallel Computers: Basic Classifications

Parallel Computers: Shared-memory computers

Prof. Dr. G. Wellein^(a,b)

^(a) Erlangen National High Performance Computing Center (NHR@FAU)

^(b) Department für Informatik

Friedrich-Alexander-Universität Erlangen-Nürnberg, Sommersemester 2025



Intel Ice Lake

NVIDIA A100

AMD MI100

Fujitsu A64FX (ARM)

Parallel Computers: Shared Memory

Modern multi- and manycore chips

Parallel Computers: Basic Classifications

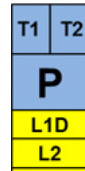
Parallel Computers: Shared-memory computers

Be prepared for more cores with less complexity and slower clock!

Modern Multicore Processors

From single core to multicore

Put n_{core} copies of core with their local caches on a chip and connect to shared cache / memory interface



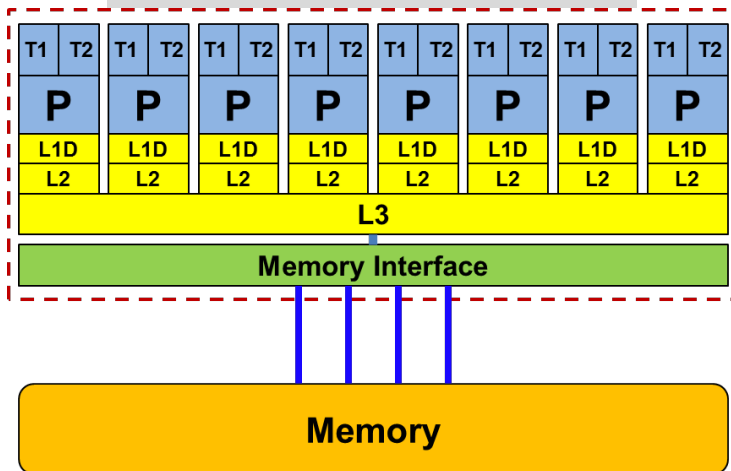
Floating Point (FP) Peak Performance of a single chip:

$$P_{chip} = n_{core} \cdot P_{core}$$

$$P_{core} = n_{super}^{FP} \cdot n_{FMA} \cdot n_{SIMD} \cdot f$$

Intel Xeon (“Ice Lake”)

~ 8,...,40 cores



Intel Xeon Platinum 8360Y (“Ice Lake”):

$$f = 1.8, \dots, 2.6 \text{ GHz (36 cores; AVX512)}$$

$$n_{core} = 36 ; n_{super}^{FP} = 2 ; n_{FMA} = 2 ; n_{SIMD} = 8$$

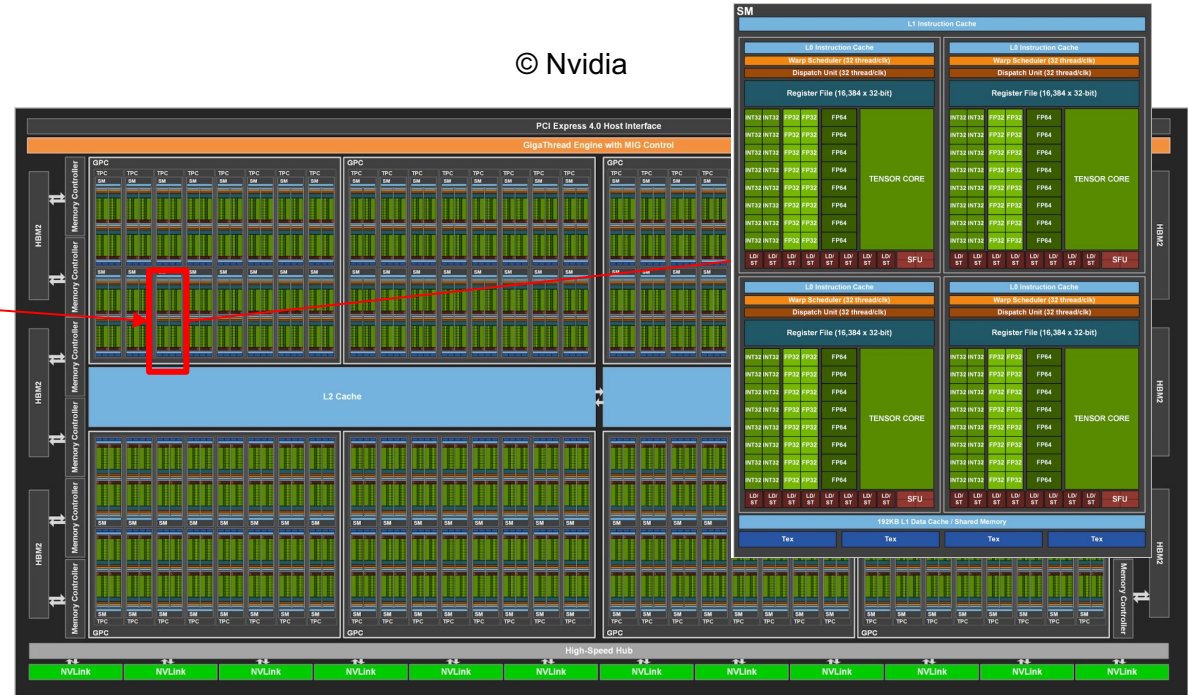
$$\text{TOP1} - 1998 \rightarrow P_{chip} = 2.1, \dots, 3.0 \frac{TF}{s} (\text{double})$$

Nvidia A100 “Ampere” SXM4 specs

Architecture

- 54.2 B Transistors
- ~ 1.4 GHz clock speed
- ~ 108 “SM” units (cores)
 - 64 SP / 32 DP FMA units each (SIMT)
 - 2:1 SP:DP performance
 - 4 “Tensor Cores” each
- 9.7 TFlop/s DP peak (FP64)
- 19.5 TFlop/s DP peak (Tensor)
- 40 MiB L2 Cache
- 40/80 GB (5120-bit) HBM2
- MemBW ~ 1555 GB/s (theoretical)
- MemBW ~ 1400 GB/s (measured)

© Nvidia



$$P_{chip} = n_{core} \cdot n_{super}^{FP} \cdot n_{FMA} \cdot n_{SIMD} \cdot f$$

SMs

1 inst./cy

FMA
factor

#Flop/inst.

$$n_{core} = 108$$

$$n_{SIMD} = 32 \frac{Flop}{inst}$$

$$n_{tensor} = 64 \frac{Flop}{inst}$$

$$n_{super} = 1 \frac{inst}{cy}$$

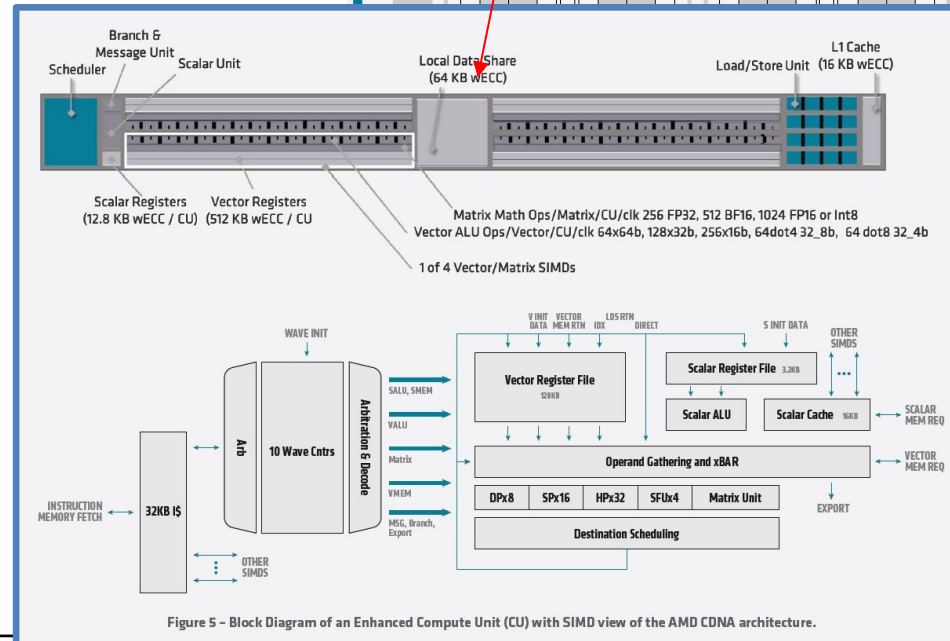
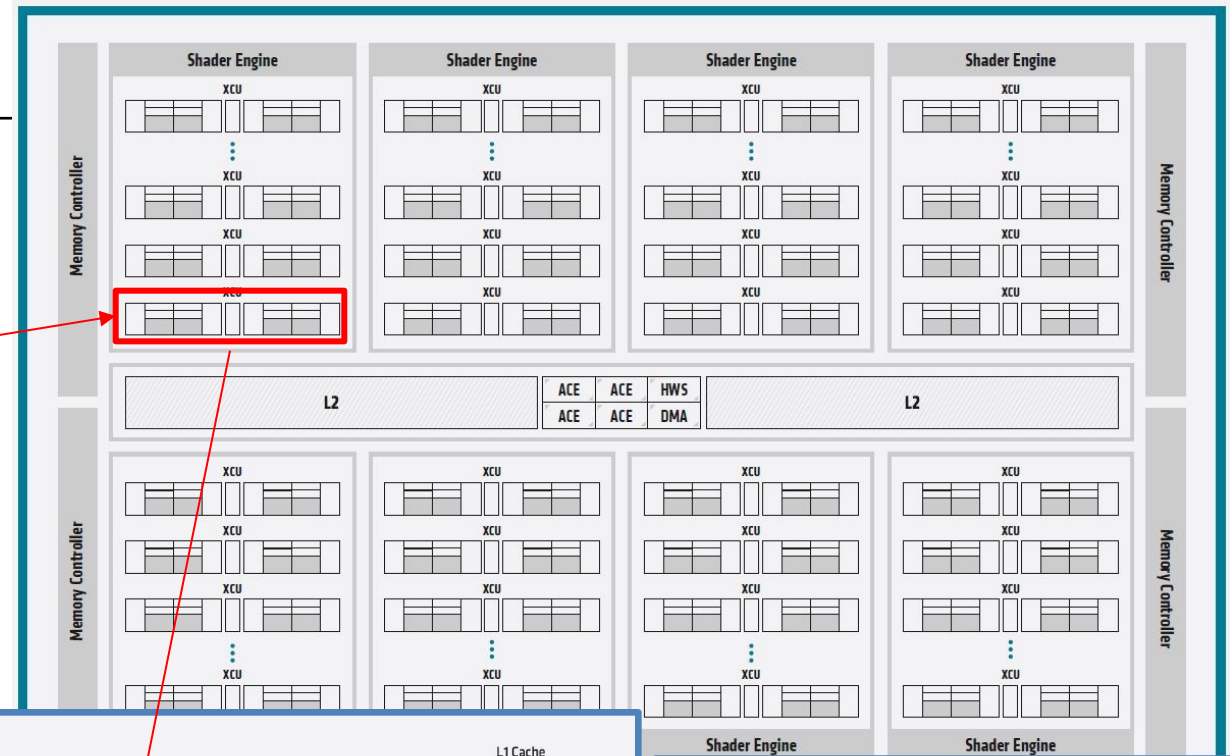
$$n_{FMA} = 2$$

$$f = 1.4 \frac{Gcy}{s}$$

AMD MI100 specs

Architecture

- 50 B Transistors
- ~ 1.5 GHz clock speed
- ~ 120 “CU” units (cores)
 - 64 SP / 32 DP FMA units each (SIMT)
 - 2:1 SP:DP performance
 - No DP matrix units
- 11.5 TFlop/s DP peak (FP64)
- 8 MiB L2 Cache
- 32 GB (4096-bit) HBM2
- 1229 GB/s (theoretical)



$$n_{core} = 120$$

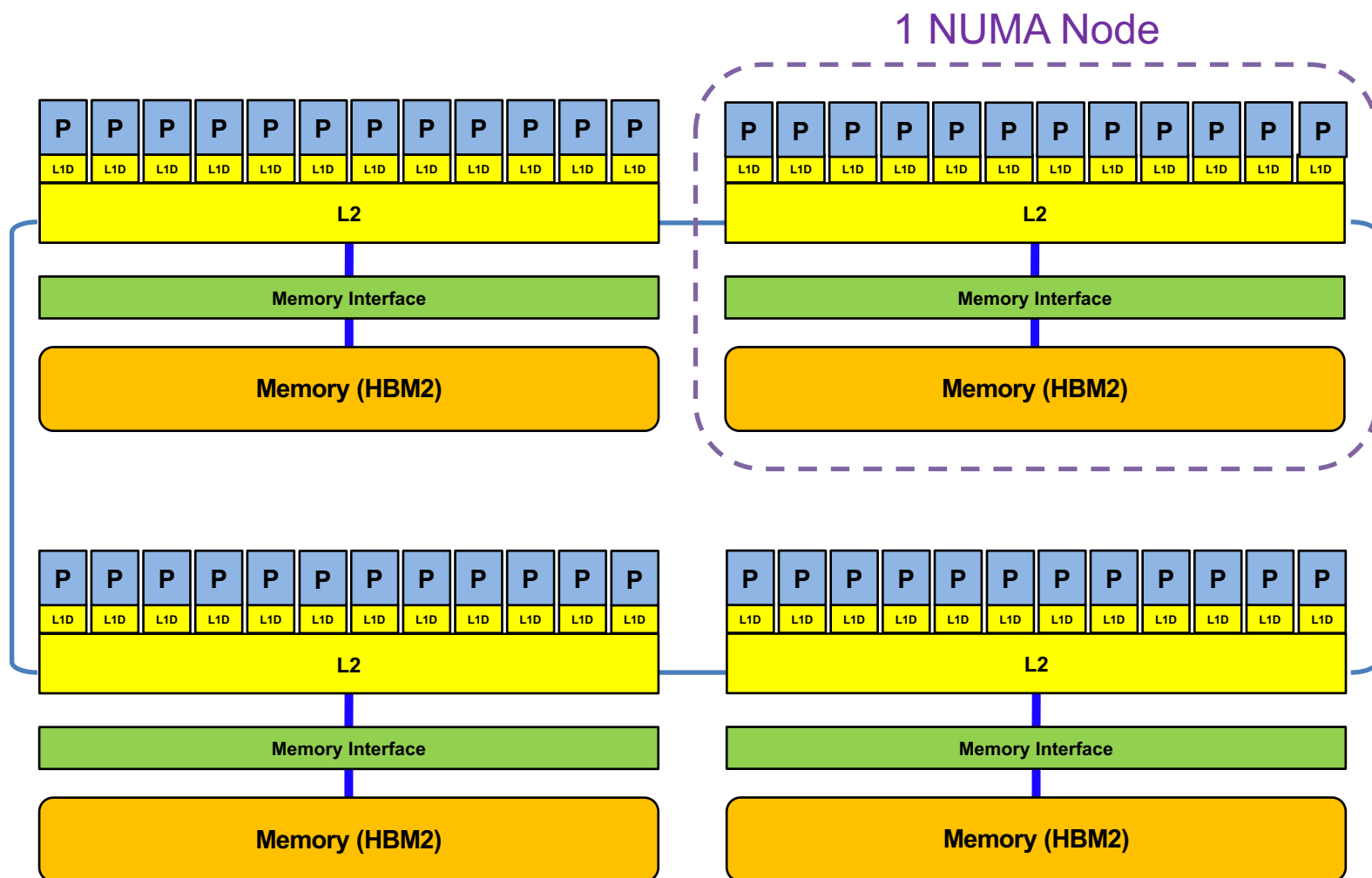
$$n_{SIMD} = 32 \frac{Flop}{inst}$$

$$n_{super} = 1 \frac{inst}{cy}$$

$$n_{FMA} = 2$$

$$f = 1.5 \frac{Gcy}{s}$$

Fujitsu A64FX (FX1000 processor)



Architecture

- 8.8 B Transistors
- Up to 2.2 GHz clock speed
- Up to 12x4 cores + 2-4 assistant cores
 - 2x 512-bit SIMD units each core (ARM SVE)
 - no SMT
- 3.4 TFlop/s DP peak (SP 2x)
- 32 MiB L2 Cache
- 32 GiB HBM2 Memory
 - MemBW ~ 860 GB/s (measured)

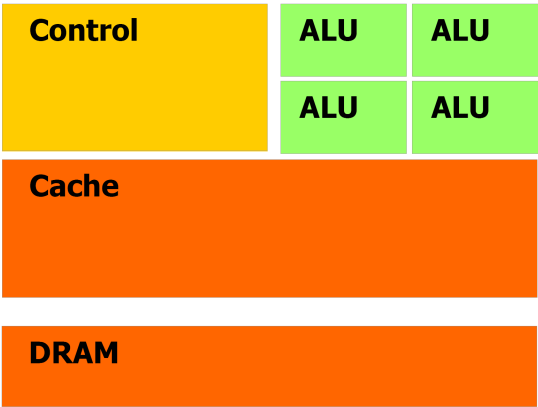
TOP500: #5 (06/2024) “Fugaku”

$$f = 2.2 \text{ GHz}; n_{\text{core}} = 48; n_{\text{super}}^{\text{FP}} = 2; n_{\text{FMA}} = 2; n_{\text{SIMD}} = 8 \rightarrow P_{\text{chip}} = 3.4 \text{ TF/s (double)}$$

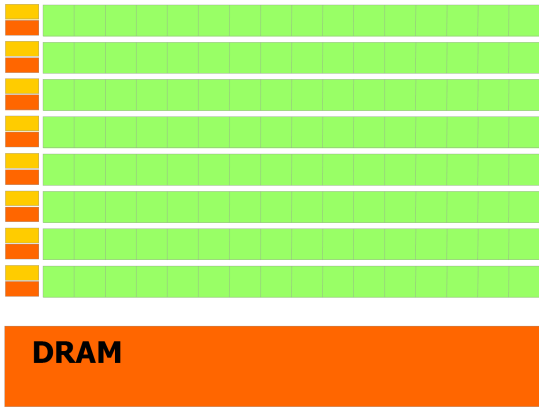
Trading single thread performance for parallelism: *GPGPUs* vs. *CPUs*

GPU/A64FX vs. CPU
light speed estimate
(per device)

MemBW ~ 2-5 x
PeakPerf (DP) ~ 3-4 x



CPU



GPU

	2x Intel Xeon Platinum 8360Y	Fujitsu A64FX	NVidia A100 “Ampere”
Cores@Clock	2 x 36 @ ≥1.8 GHz	48 @ 2.2 GHz	108 SMs @ ~1.4 GHz
DP peak	4.2,...,6 TFlop/s	3.4 TFlop/s	19.5 TFlop/s
Stream BW (meas.)	2 x 170 GB/s	860 GB/s (HBM)	1400 GB/s
Transistors / TDP	~? Billion / 2x250 W	8 Billion / ~200W	54 Billion/400W
Threads to saturate bandwidth	~30	~20	~20.000

There is no single driving force for single core performance!

$$P_{chip} = n_{core} \cdot n_{super}^{FP} \cdot n_{FMA} \cdot n_{SIMD} \cdot f$$

	n_{core}	n_{super}^{FP} inst./cy	n_{FMA}	n_{SIMD} ops/inst		Chip	f [GHz]	P_{chip} [GF/s]
Nehalem	4	2	1	2	Q1/2009	X5570	2.93	46.8
Westmere	6	2	1	2	Q1/2010	X5650	2.66	63.6
Sandy Bridge	8	2	1	4	Q1/2012	E5-2680	2.7	173
Ivy Bridge	10	2	1	4	Q3/2013	E5-2660 v2	2.2	176
Haswell	14	2	2	4	Q3/2014	E5-2695 v3	2.3	515
Broadwell	22	2	2	4	Q1/2016	E5-2699 v4	2.2	774
Skylake	28	2	2	8	Q3/2017	Platinum 8180	2.5	2,240
Ice Lake	36	2	2	8	Q2/2021	Platinum 8360Y	1.8 / 2.6	2,073/2,995
Sapphire Rapids	52	2	2	8	Q1/2023	Platinum 8470	2.0	3,328
AMD Rome	64	2	2	4	Q4/2019	EPYC 7742	2.25	2,304
IBM POWER8	10	2	2	2	Q2/2014	S822LC	2.93	234
Fujitsu A64 FX	48	2	2	8			2.2	3,400

AVX-512 (36c):
Base / Max. Turbo

There is no single driving force for single core performance!

$$P_{chip} = n_{core} \cdot n_{super}^{FP} \cdot n_{FMA} \cdot n_{SIMD} \cdot f$$


Cores Super-scalarity FMA factor SIMD factor Clock Speed

	n_{core}	n_{super}^{FP} inst./cy	n_{FMA}	n_{SIMD} ops/inst		Chip	f [GHz]	P_{chip} [GF/s]
Nvidia P100	56	1	2	32	Q2/2016		1.3	4,660
Nvidia A100	108	1	2	32	2020		1.4	9,700
Nvidia H100	132	2	2	32	2024	GH100	2.0	33,400
AMD MI100	120	1	2	32			1.5	11,500
AMD MI250	2*110	1	2	64	2022		1.7	24,000
<i>Xeon Phi 7250</i>	68	2	2	8	Q2/2016		1.4	3,046

Parallel Computers: Shared Memory

Modern multi- and manycore chips

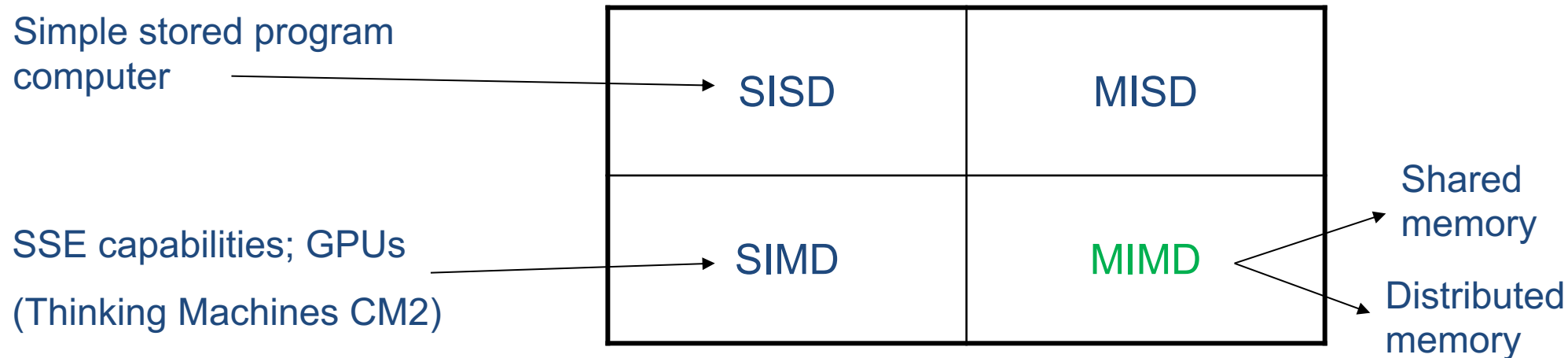
Parallel Computers: Basic Classifications

Parallel Computers: Shared-memory computers



Parallel computers – Classifications

- **Parallel Computing**: A number of compute elements solve a problem in a cooperative way
- **Parallel Computer**: A number of compute elements connected such way to do parallel computing for a large set of applications
- Classification according to Flynn: **Multiple Instruction Multiple Data (MIMD)**

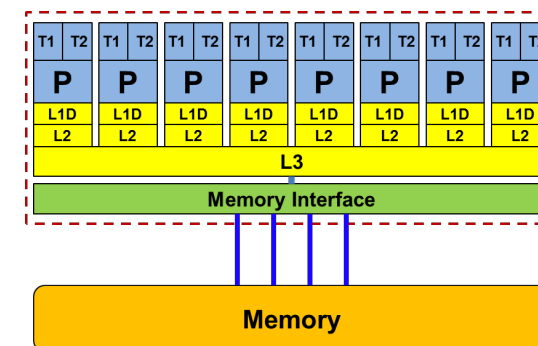


Parallel Computers - Classifications

Classification according to address space organization

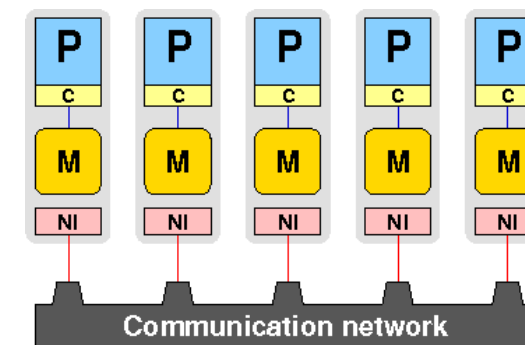
- **Shared-Memory** Architectures:

Cache-Coherent Single Address Space

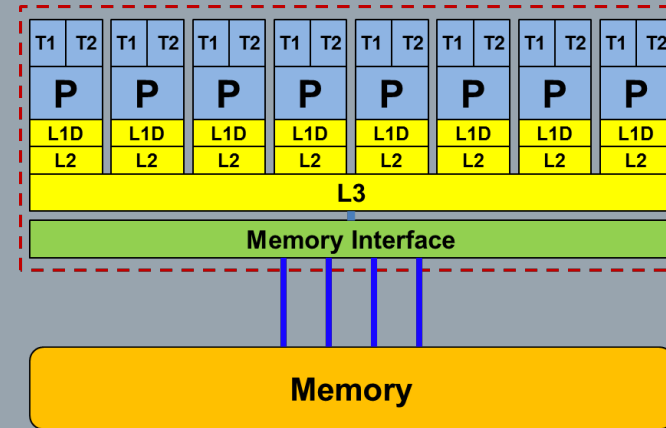
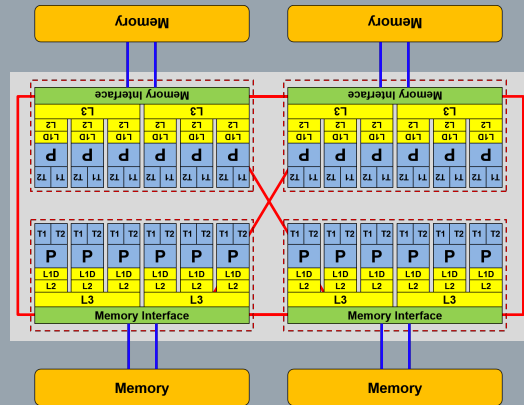


- **Distributed-Memory** Architectures

No (Cache-Coherent) Single Address Space



Hybrid architectures containing both concepts are state-of-the art (e.g., “Emmy” / ”Meggie” / “Fritz” cluster @ RRZE)



Parallel Computers: Shared Memory

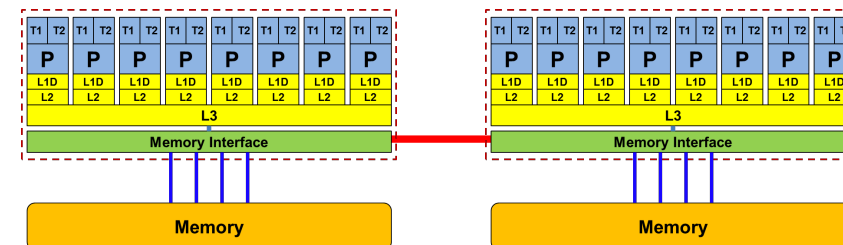
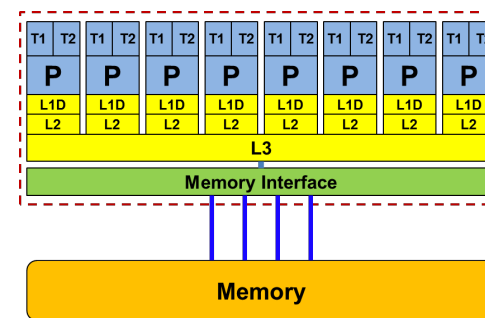
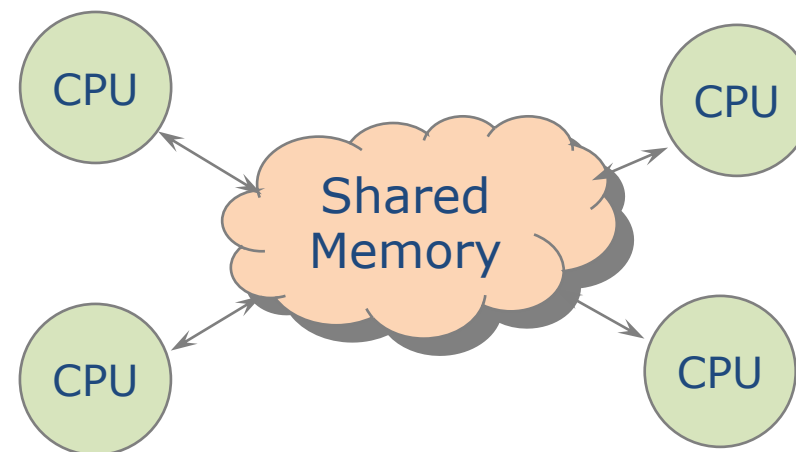
Modern multi- and manycore chips

Parallel Computers: Basic Classifications

Parallel Computers: Shared-memory computers

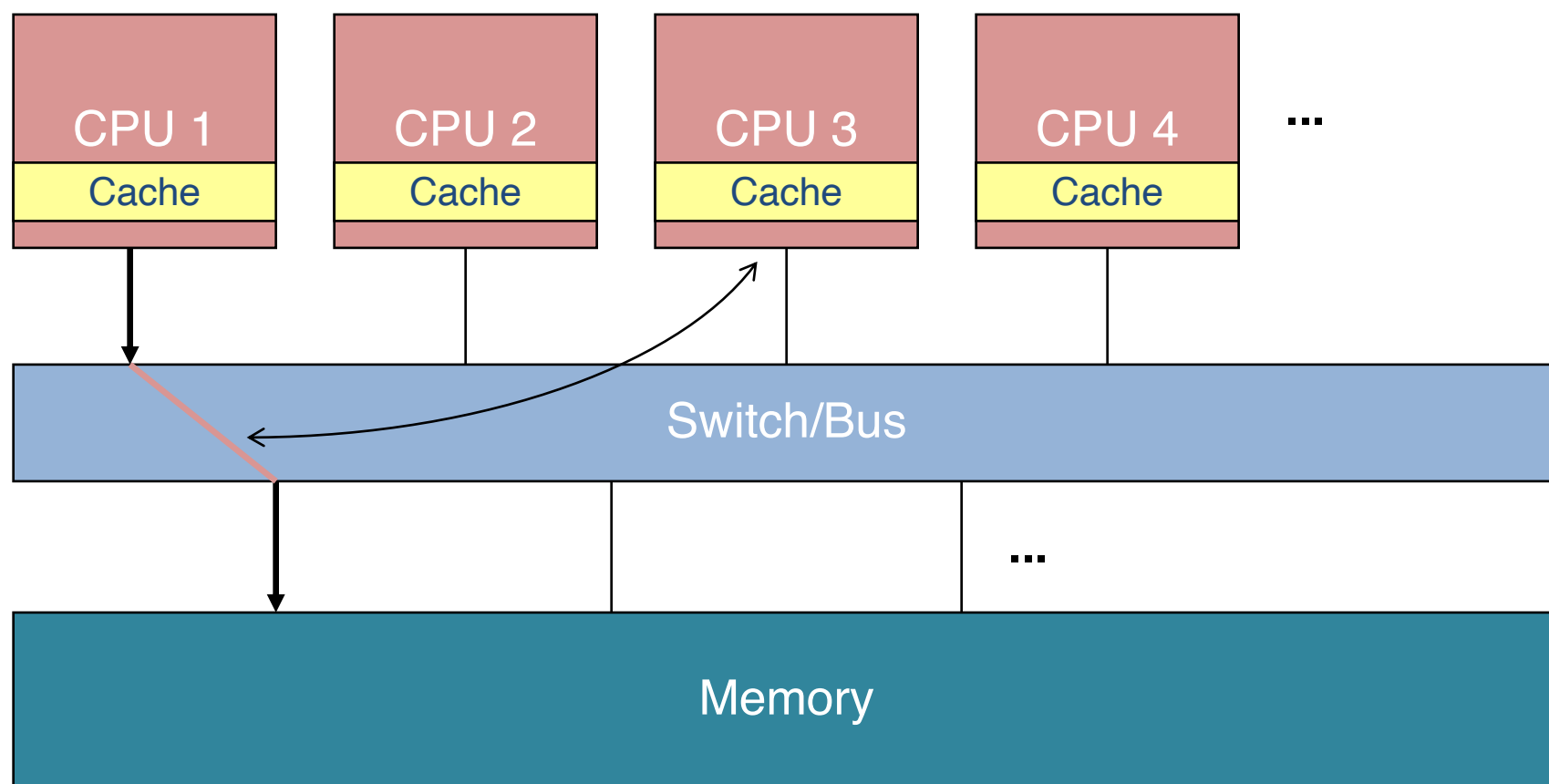
Parallel computers – Shared-Memory Architectures

- Shared memory computers provide
 - Single shared address space for all processors
 - All processors share the same view of the address space!
- Two basic categories of shared memory systems
 - **Uniform Memory Access (UMA):**
Memory is equally accessible to all processors with the same performance (Bandwidth & Latency)
 - **cache-coherent Non Uniform Memory Access (ccNUMA):**
Memory is physically distributed:
Performance (Bandwidth & Latency) is different for local and remote memory access



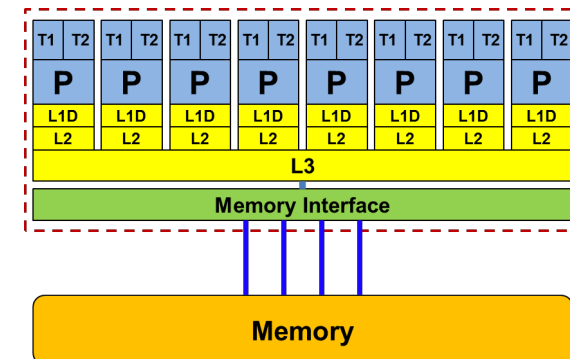
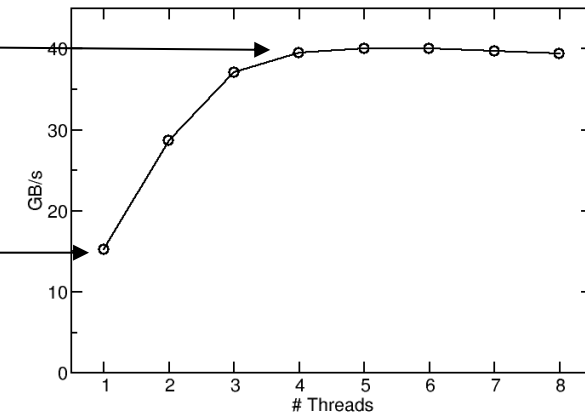
Parallel computers: Shared-memory: UMA

- **UMA** Architecture: switch/bus arbitrates memory access
 - Special protocol ensures cross-CPU cache data consistency
 - Flat memory: Access time to a given memory location is same for all CPUs



Parallel computers: Shared-memory: UMA / Bus based

- Worst case: **bus system** single memory path to multiple processors
 - Only “**one consumer**” at a time can use the bus and access memory at any one time
 - Collisions occur frequently, causing one or more CPUs to wait for “bus ready” (contention) → **Saturation**
 - One consumer / core may not be able to saturate the bus
 - Multi-core architectures:
Bus-type Memory Interface
→UMA
→Bandwidth saturates when increasing core count (i.e. utilization)

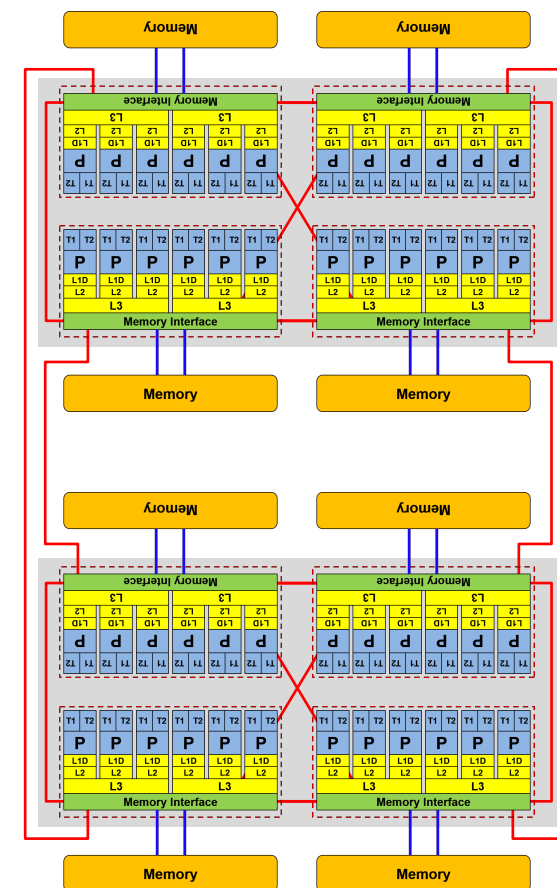


Parallel computers: Shared Memory: UMA Nodes

- Examples:
 - Dual-/quad-/hexa-/octo-/.../32-core laptop/desktop/server processor
 - IBM Power8/BlueGene processor series
 - NEC vector systems
 - nVIDIA GPUs
 - Intel Xeon Phi (KNC, KNL,...)
 - Advantages
 - Cache Coherence (see below) is "easy" to implement → single controller
 - Easy to optimize memory access
 - Incremental parallelization
 - Disadvantages
 - Bus-type memory bandwidth limits scalability in terms of consumers (2 – 30 cores per UMA node)
-

Parallel shared memory computers: ccNUMA/Node Layout

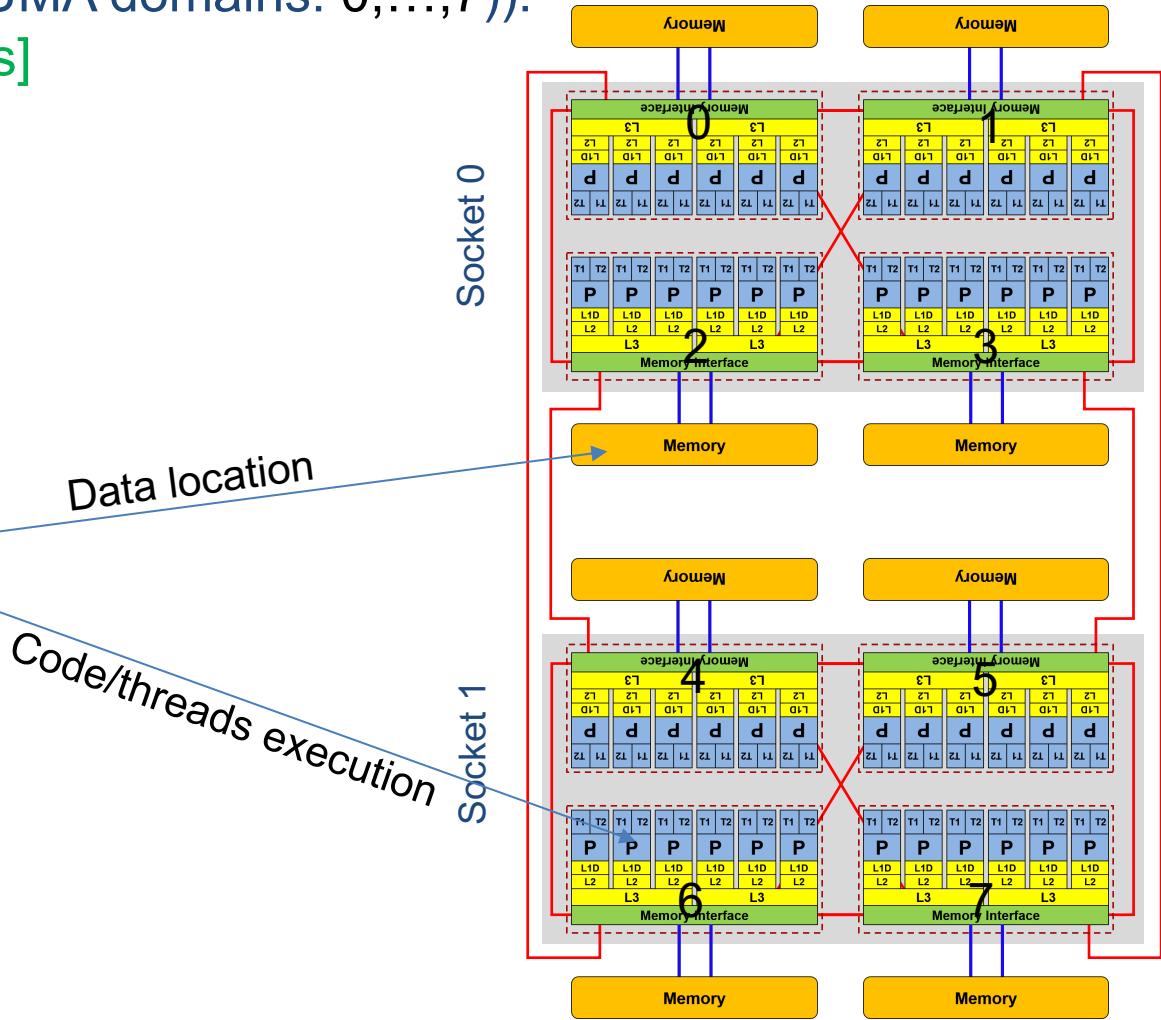
- ccNUMA:
 - Single cache coherent address space although multiple physically distributed memory (interfaces/controllers) are used
 - Hardware and software layers establish a shared address space and the cache coherency
 - Access time to a given memory location may depend on the CPU/core requesting the data (topology)
 - Example: AMD “Naples” dual-socket system (2 sockets, 48 cores) with 8 Memory Interfaces (“NUMA Domains”)



Parallel shared memory computers: ccNUMA/Node Layout

Example: AMD “Naples” dual-socket system (8 NUMA domains: 0,...,7)):
STREAM Triad bandwidth measurements [Gbyte/s]

		Socket 0				Socket 1			
		Threads run on NUMA domain							
		0	1	2	3	4	5	6	7
Data in NUMA domain	0	32.4	21.4	21.8	21.9	10.6	10.6	10.7	10.8
	1	21.5	32.4	21.9	21.9	10.6	10.5	10.7	10.6
	2	21.8	21.9	32.4	21.5	10.6	10.6	10.8	10.7
	3	21.9	21.9	21.5	32.4	10.6	10.6	10.6	10.7
	4	10.6	10.7	10.6	10.6	32.4	21.4	21.9	21.9
	5	10.6	10.6	10.6	10.6	21.4	32.4	21.9	21.9
	6	10.6	10.7	10.6	10.6	21.9	21.9	32.3	21.4
	7	10.7	10.6	10.6	10.6	21.9	21.9	21.4	32.5



Parallel shared memory computers: ccNUMA/Node Layout

■ ccNUMA:

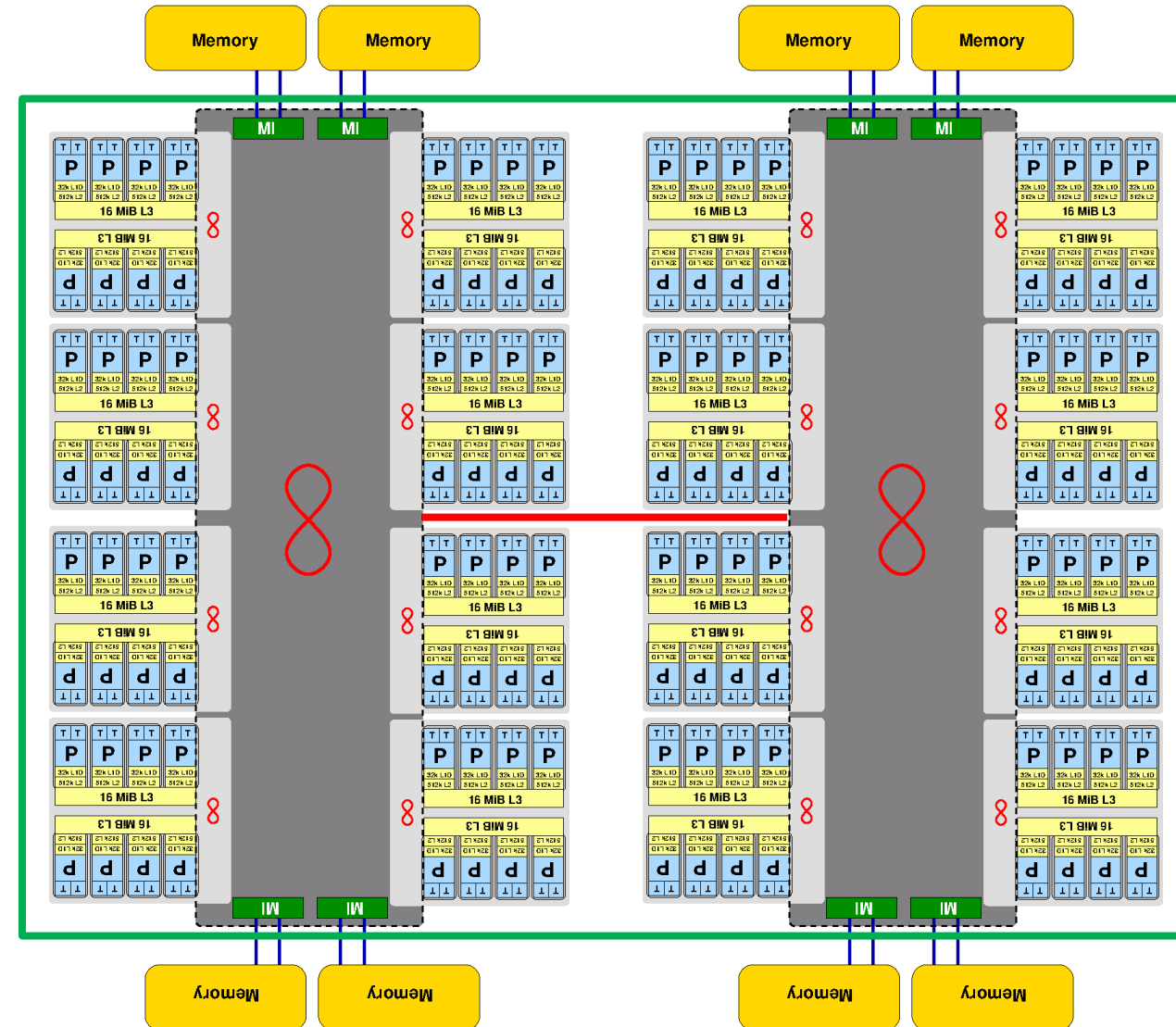
■ Advantages:

- Aggregate memory bandwidth is scalable within node (multiple memory controller)
- Systems with >1024 cores are available
- Example: 2 socket AMD Rome/Milan (64c) compute node ("Alex"):
8 Memory Interfaces connected by Infinity fabric

■ Disadvantages:

- Cache Coherence hard to implement / expensive
- Performance depends on topology, i.e. access to local or remote memory

- All modern multi-socket compute nodes



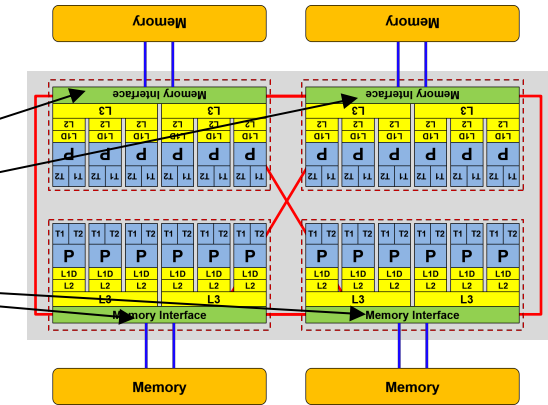
ccNUMA in a single multicore processor!

AMD Magny-Cours+ & Intel Cluster on Die mode

- ccNUMA can be found within a single multicore processor chip

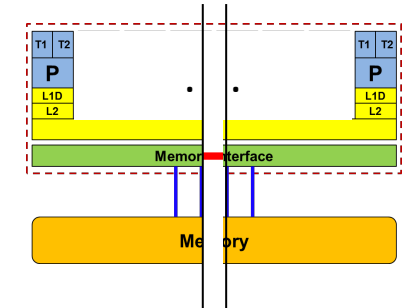
- AMD: **single chip ccNUMA** since Magny Cours:

“Naples/EPYC” has 4 memory controllers per chip!

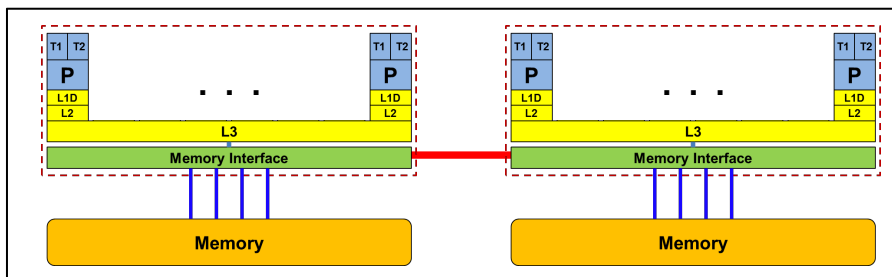


- Intel: Cluster on Die (CoD) mode since Haswell
(sub-NUMA Cluster / **SNC** for Skylake+)

BIOS boot-time option: single chip UMA or ccNUMA



- Standard 2 socket HPC server → always NUMA



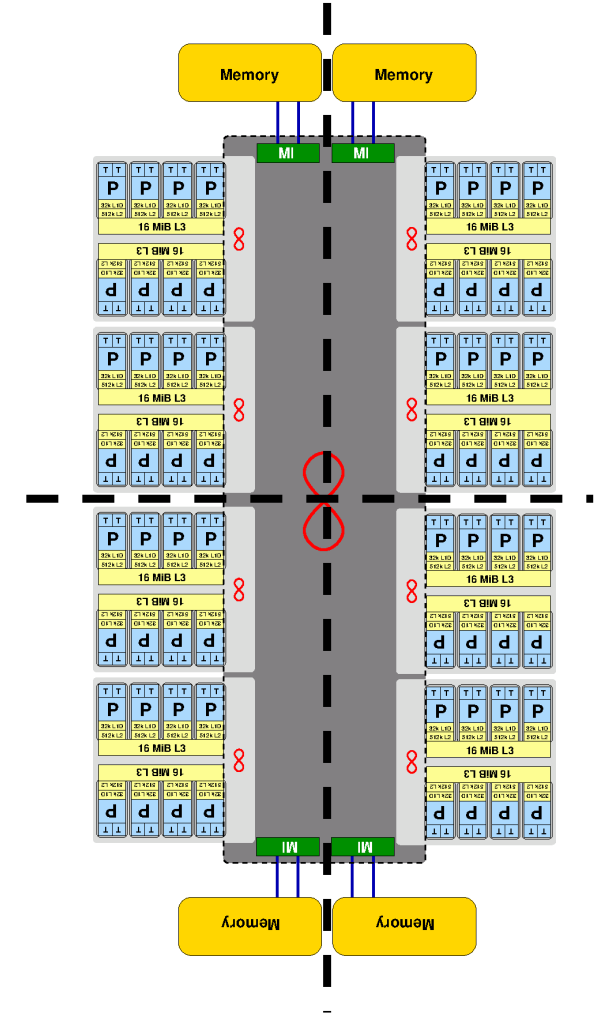
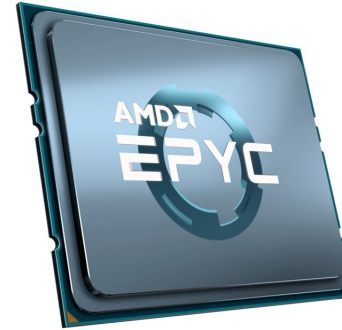
Where does my data end up?
→ OpenMP programming lectures

Intel with CoD / SNC NOT enabled

ccNUMA in a single multicore processor

- AMD / Intel processors may support NUMA-modes in a single multicore processor chip

- Boot time option → BIOS
- AMD: **Single chip ccNUMA** since Magny Cours
- AMD Rome/Milan 64 cores (c) supports 4 options ("NPS mode")
 - NPS = 1 – UMA mode (no NUMA characteristics): 64 c + 4 memory interfaces (MI)
 - NPS = 2 – 2 NUMA domains with 32 c + 2 MI each
 - NPS = 4 – 4 NUMA domains with 16 c + 1 MI each



How to determine NUMA configuration of node / chip → LIKWID topology
Where does my data end up? → OpenMP programming lectures

Multicore nomenclature: 2 x AMD Rome 64c (NPS=4)

