Introduction to High Performance Computers

Richard Gerber

NERSC User Services

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Why Do You Care About Architecture?

- **To use HPC systems well, you need to understand the basics and conceptual design**
  - Otherwise, too many things are mysterious
- **Programming for HPC systems is hard**
  - To get your code to work properly
  - To make it run efficiently (performance)
- **You want to efficiently configure the way your job runs**
- **The technology is just cool!**

Topics

- Terminology
- 5 main parts of an HPC system
- CPUs
- Memory
- Nodes
- Interconnect
- Data Storage

5 Major Parts of an HPC System

We'll take the perspective of an application programmer or a scientist.

What features of an HPC system are important for you to know about?

1. CPUs
2. Memory (volatile)
3. Nodes
4. Inter-node network
5. Non-volatile storage (disks, tape)

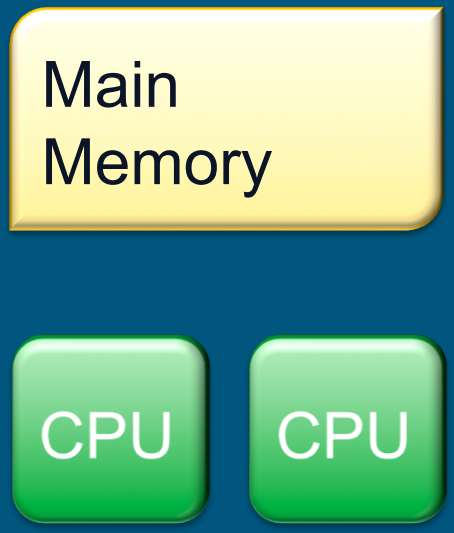
Let's Build an HPC System

We'll talk about the components in just a bit.

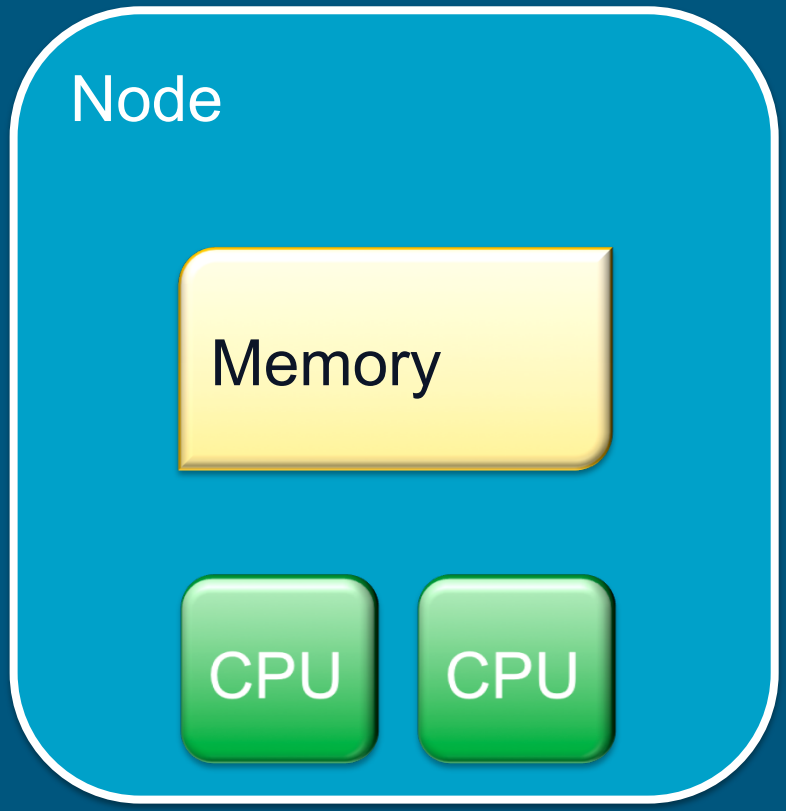
A distributed-memory HPC system

A distributed-memory HPC system

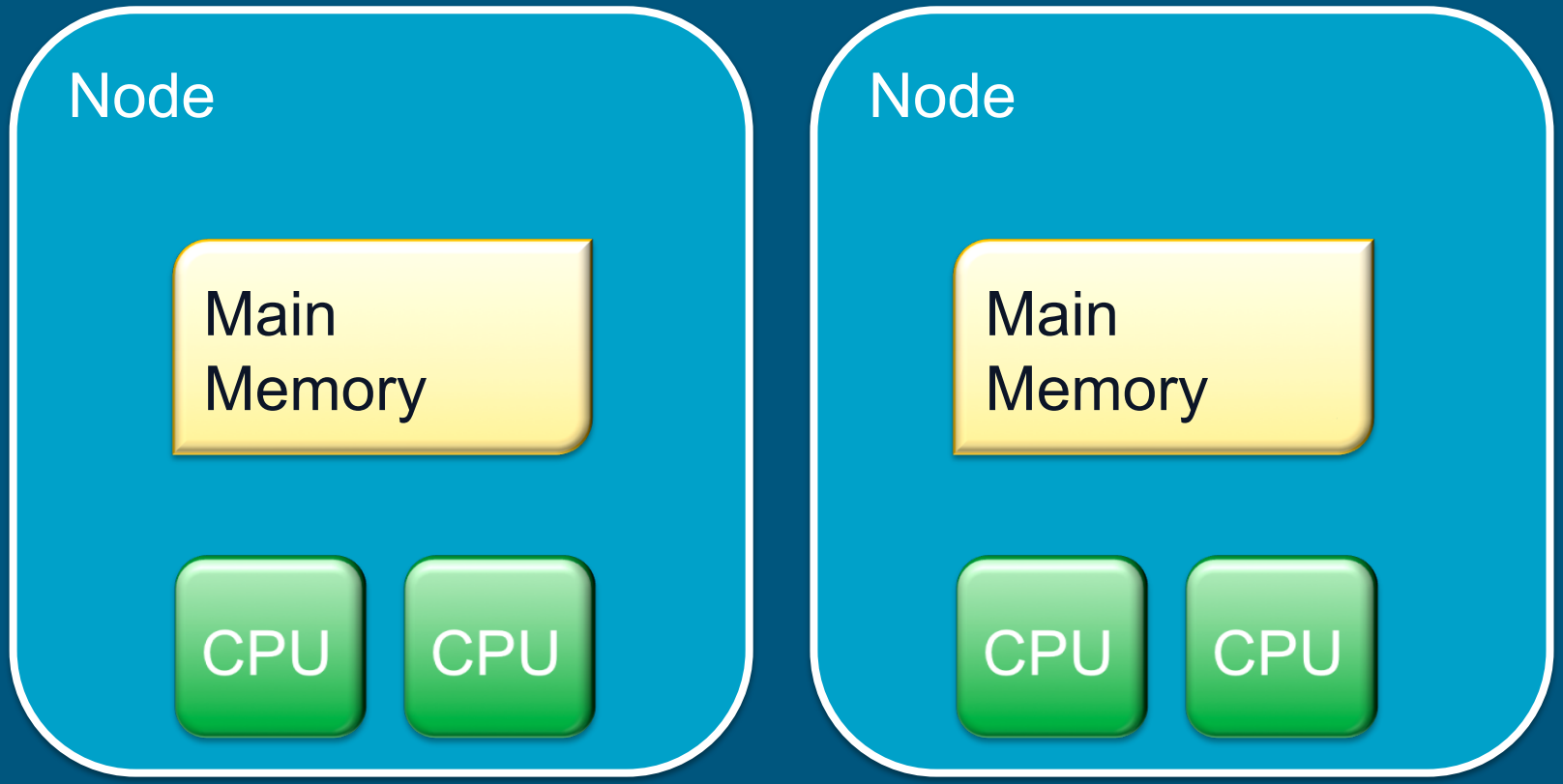
A distributed-memory HPC system



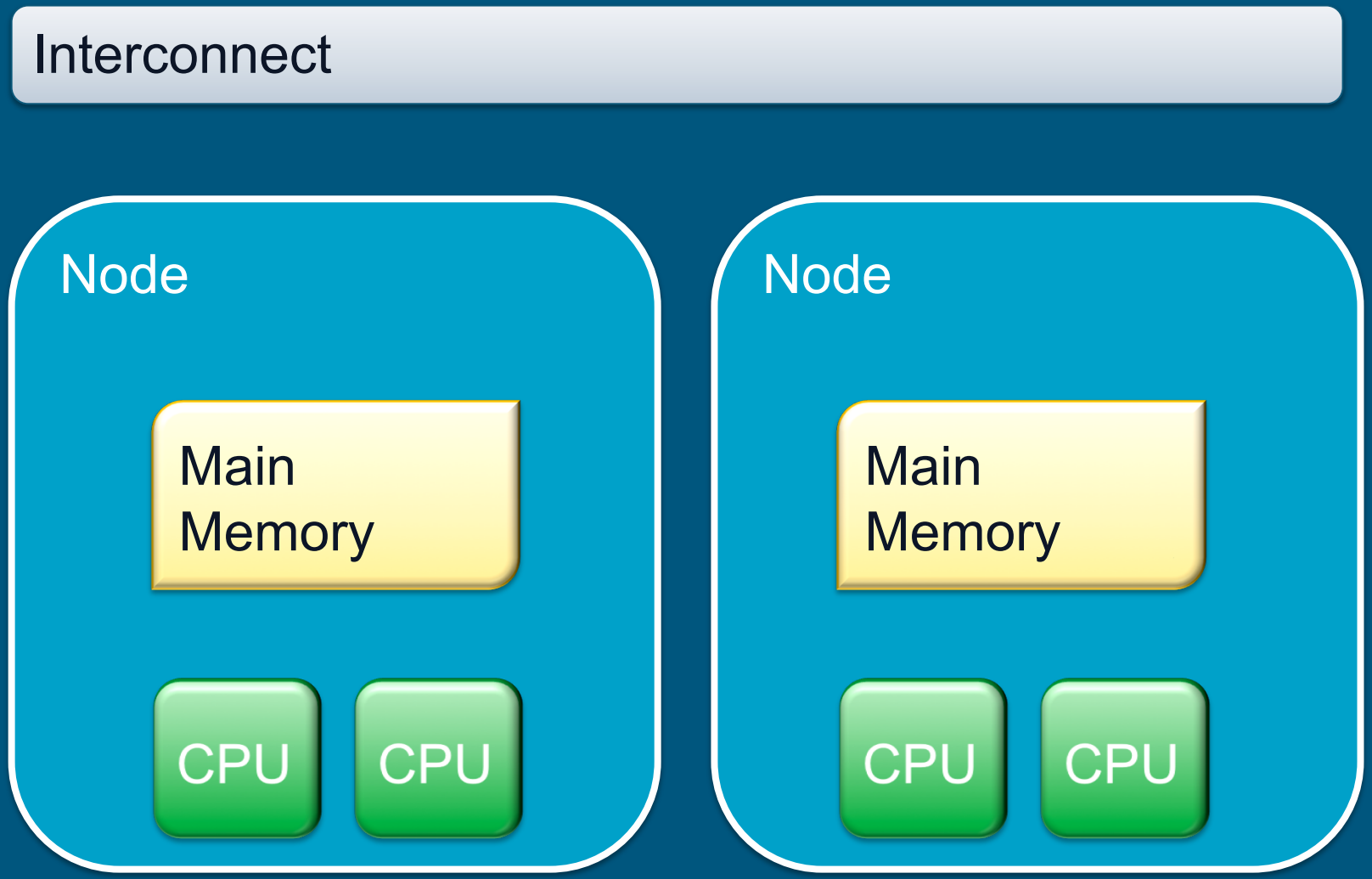
A distributed-memory HPC system



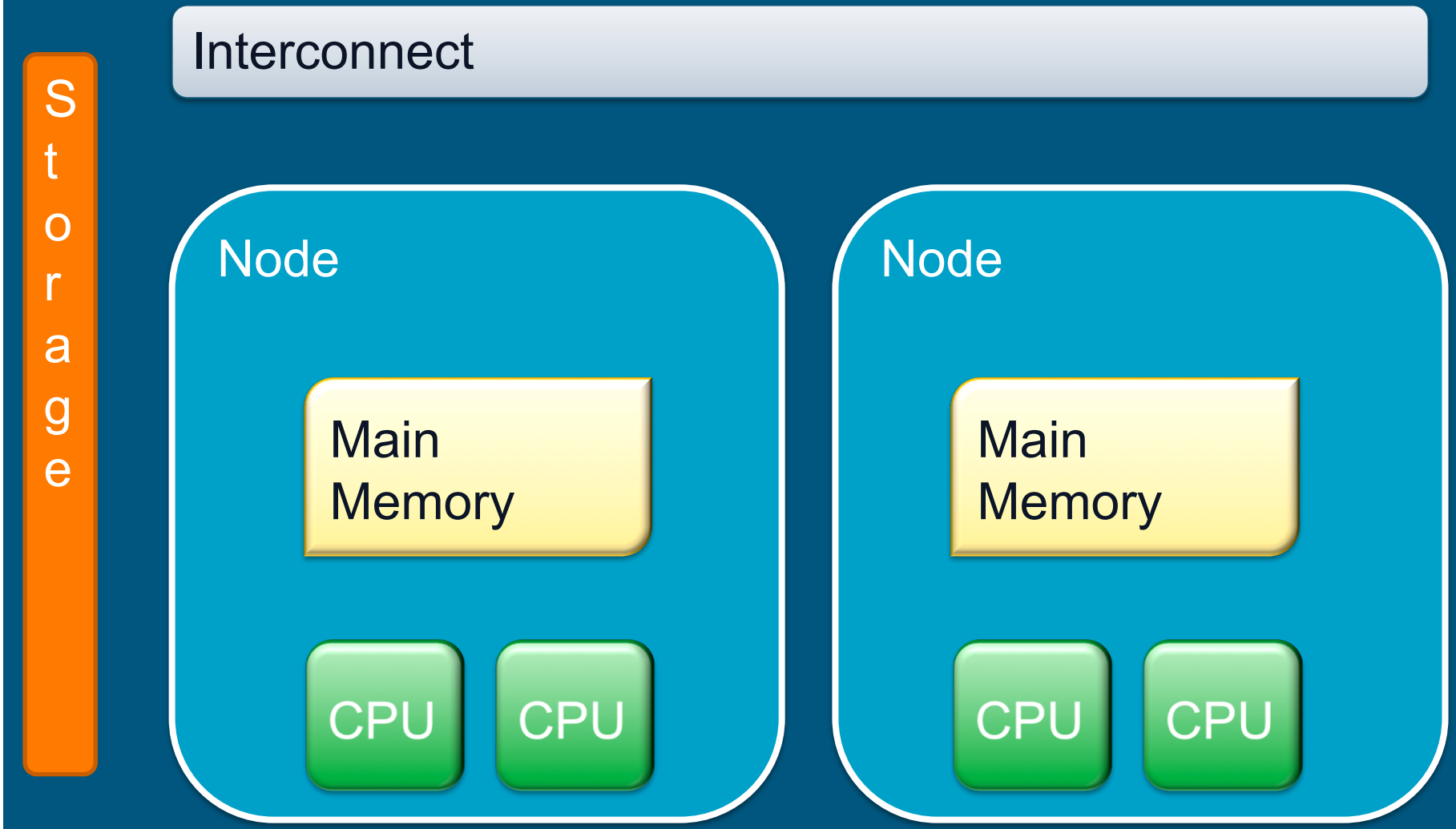
A distributed-memory HPC system



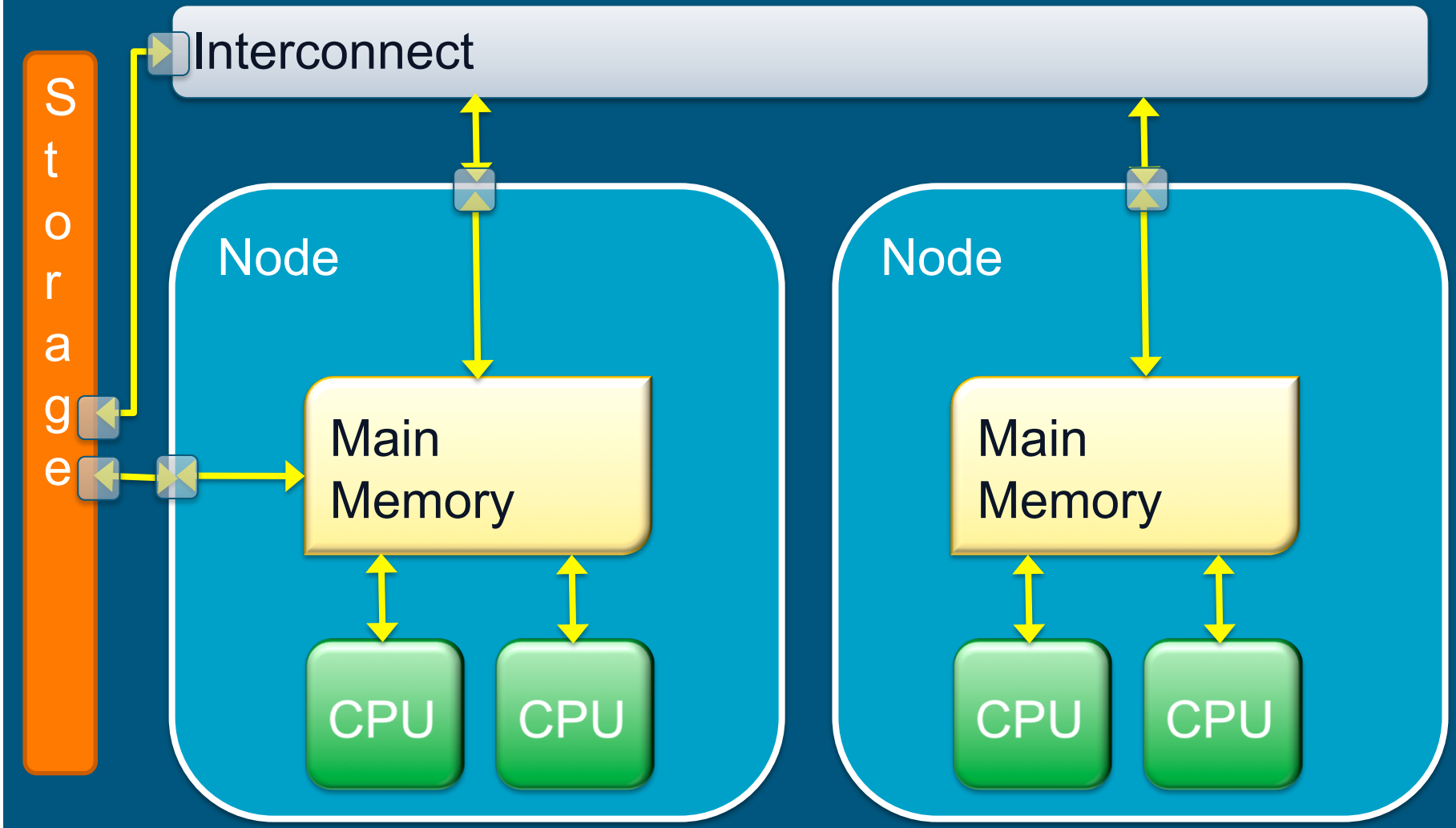
A distributed-memory HPC system



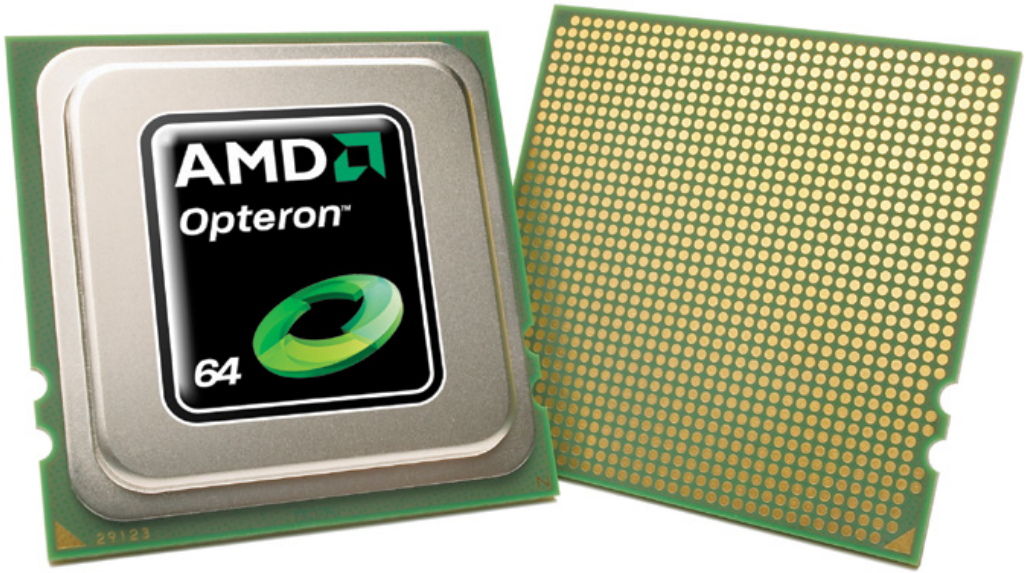
A distributed-memory HPC system



A distributed-memory HPC system



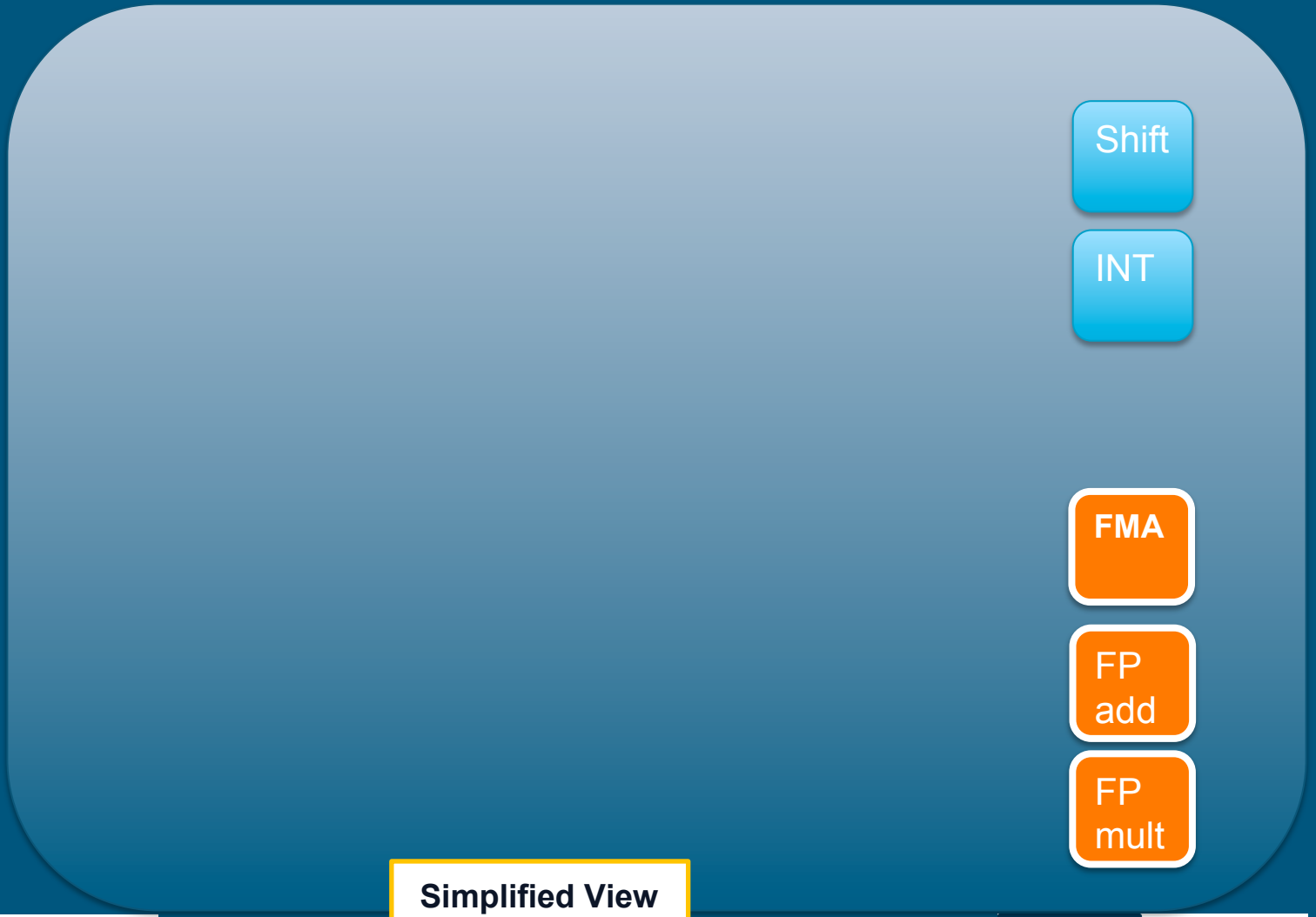
CPUs



Stored Program Computers

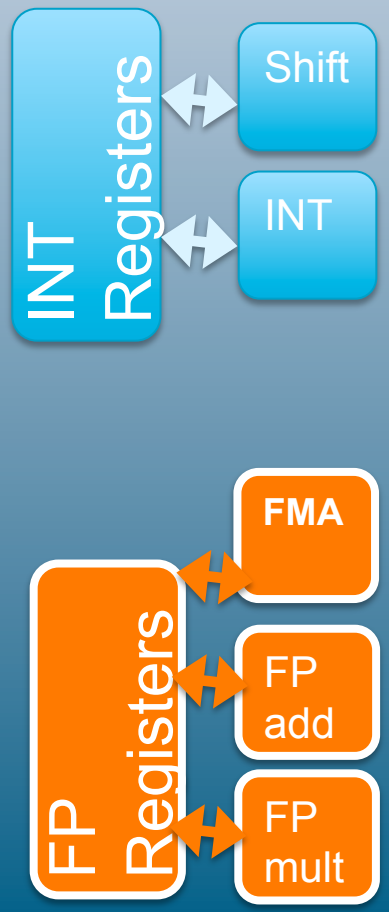
- **Modern computers are "stored program computers"**
  - Conceived by Turing in 1936
  - Implemented in 1949 (EDVAC)
- **Instructions are stored as data in memory**
  - Read and executed by control units
- **Arithmetic and logic**
  - Performed by functional units separate from instruction control units

CPU (1 core)



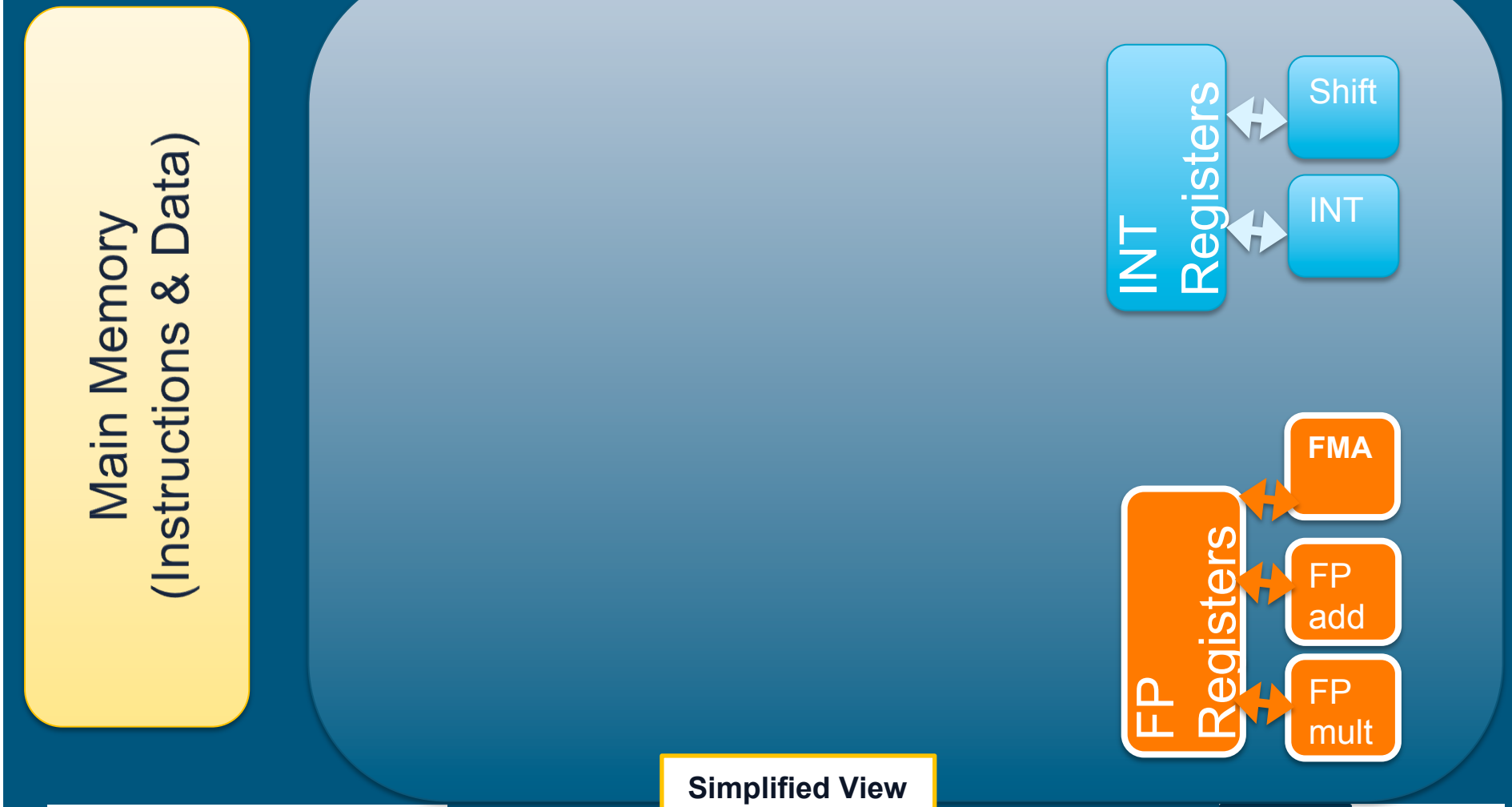
Simplified View

CPU (1 core)



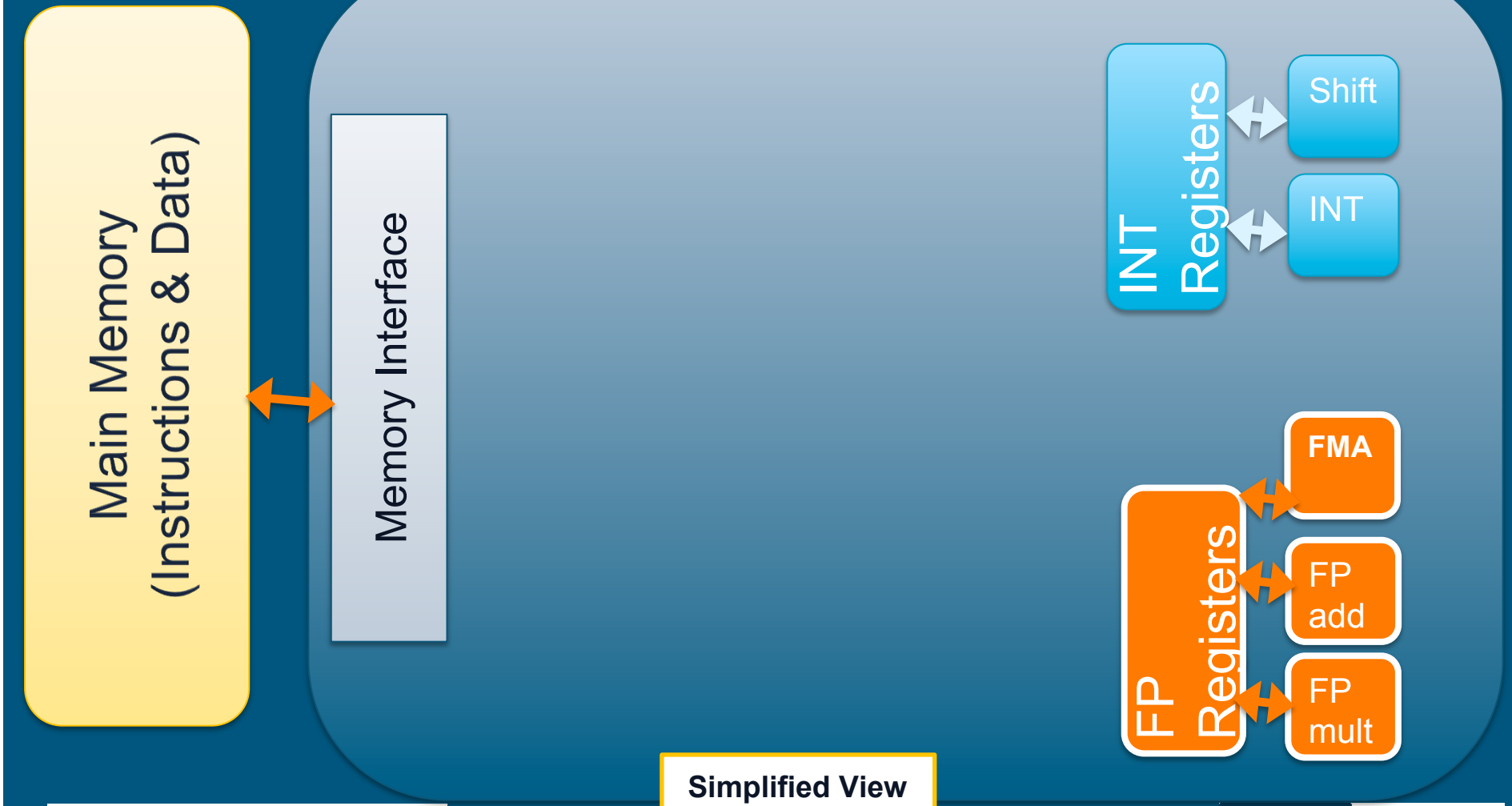
Simplified View

CPU (1 core)



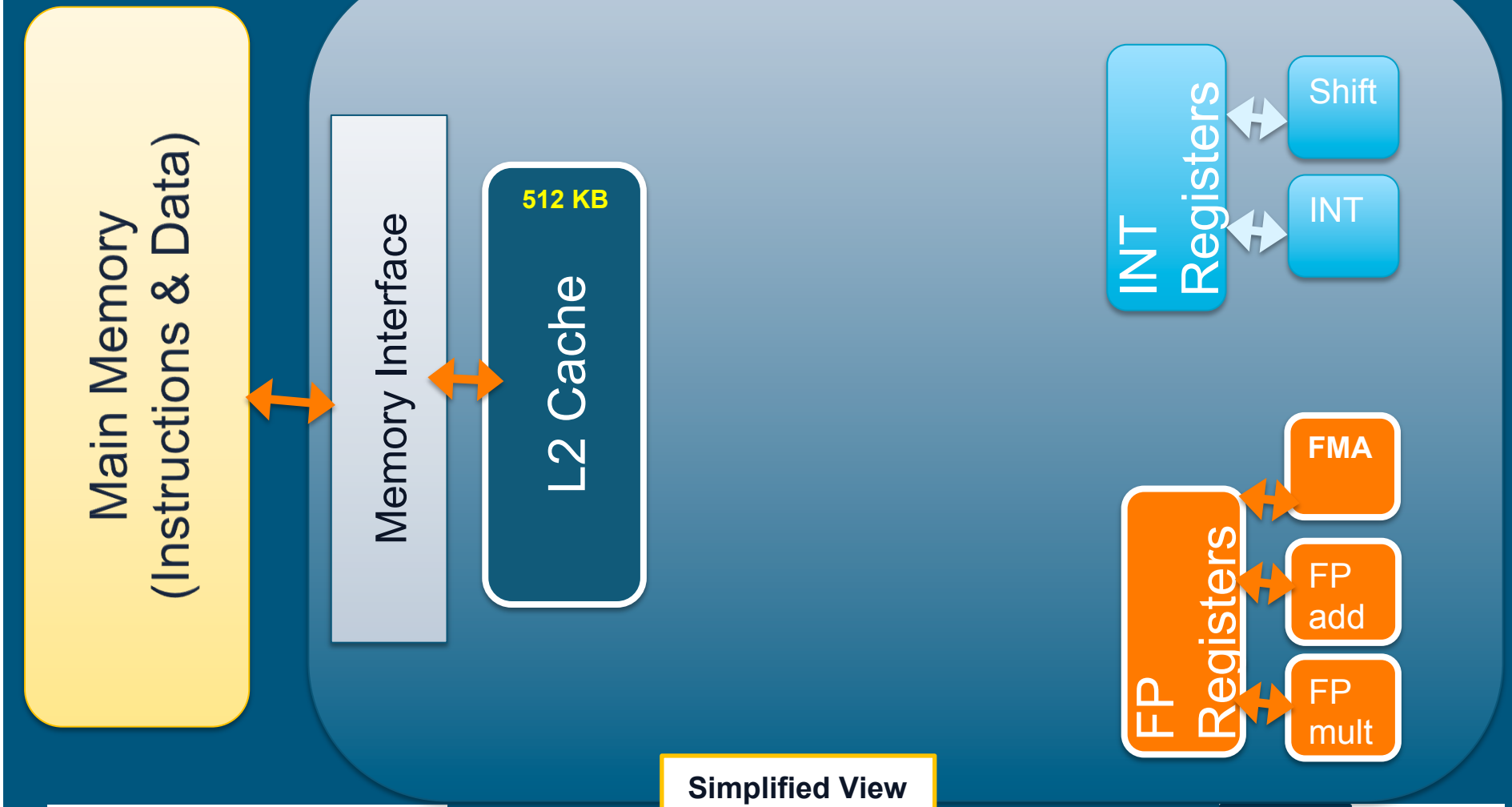
Simplified View

CPU (1 core)



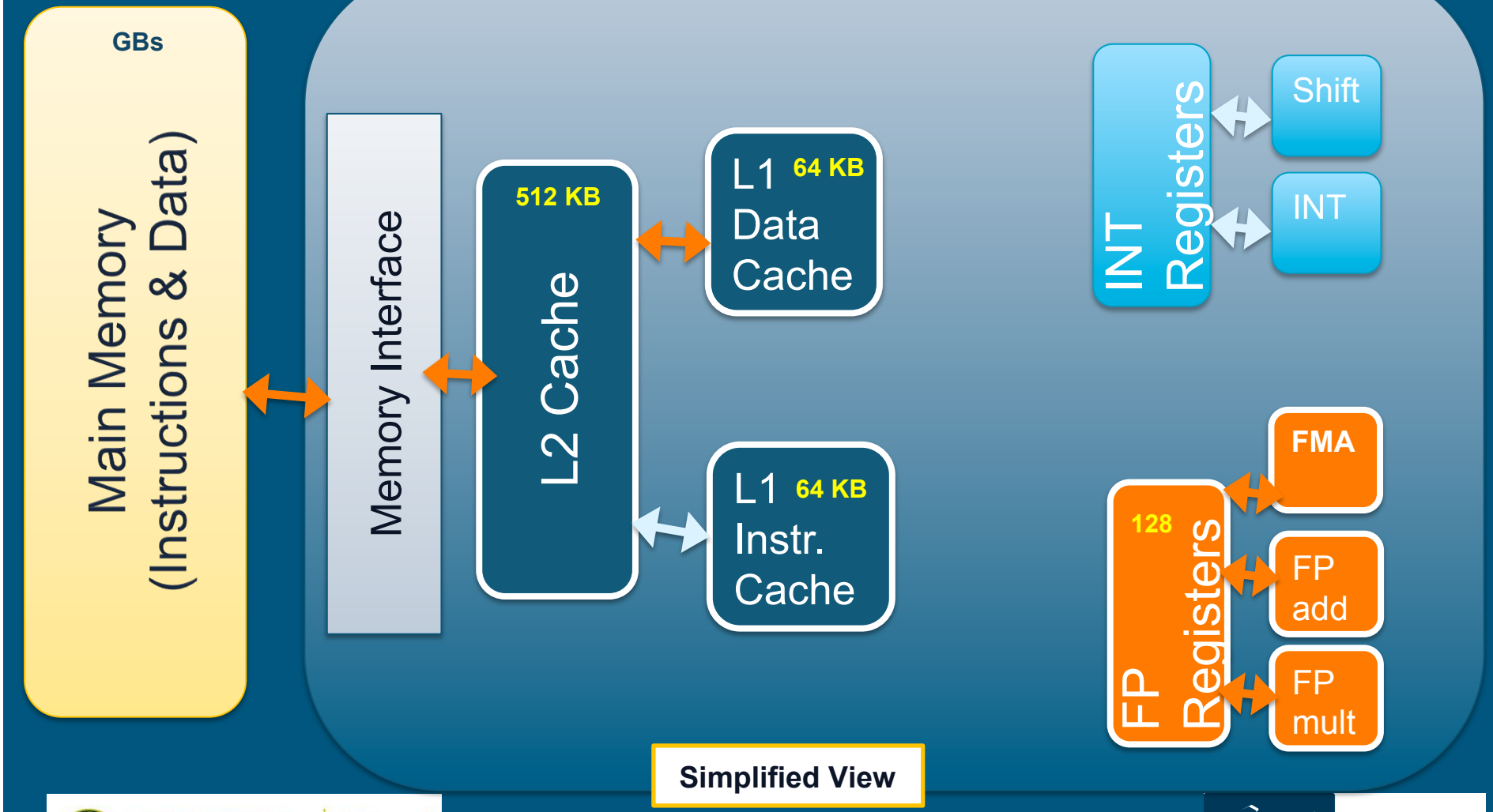
Simplified View

CPU (1 core)

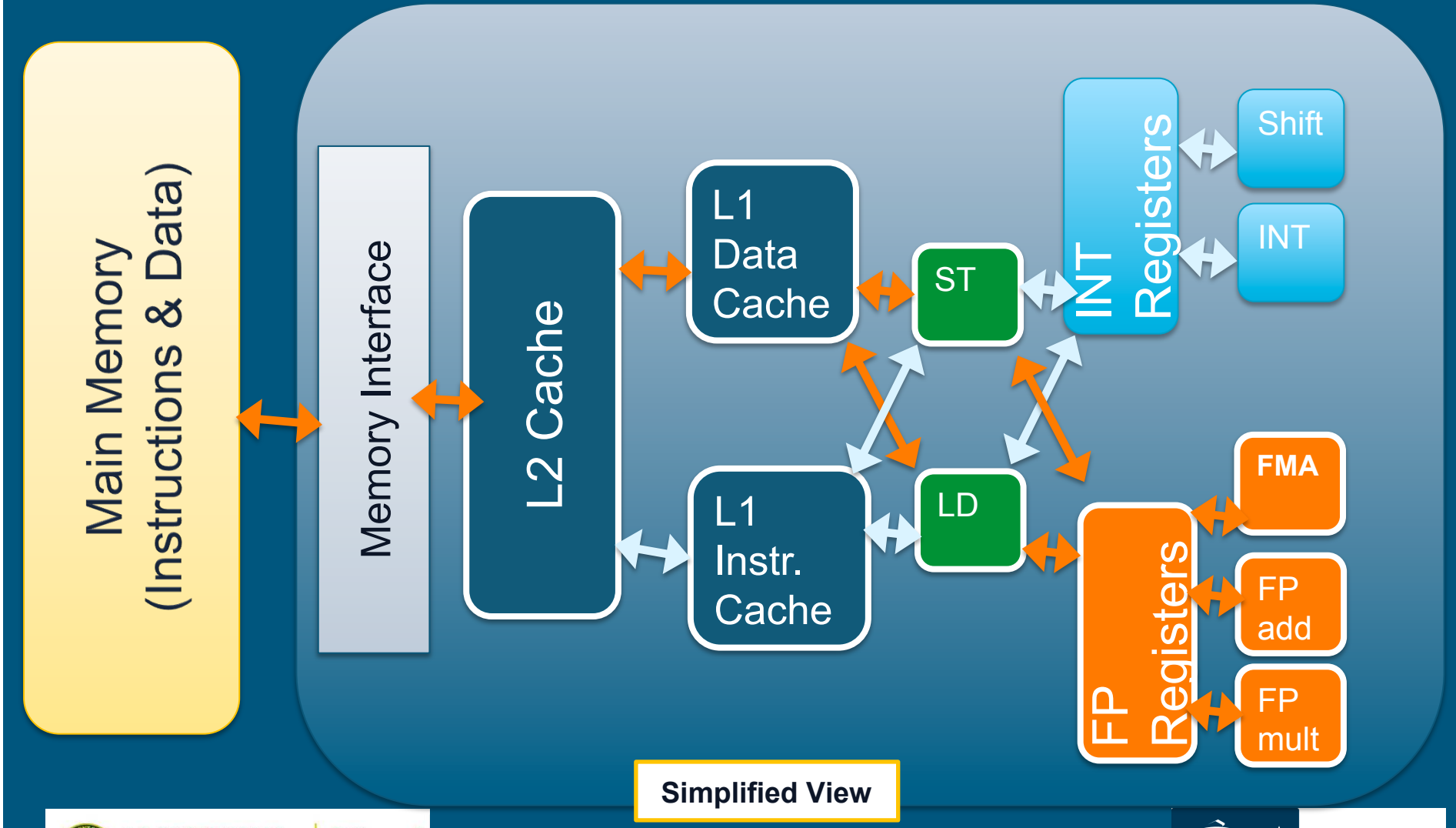


Simplified View

CPU (1 core)


GBs
Main Memory
(Instructions & Data)
Memory Interface
512 KB
L2 Cache
L1 64 KB
Data
Cache
L1 64 KB
Instr.
Cache
INT
Registers
Shift
INT
FP
Registers
128
FMA
FP
add
FP
mult
Simplified View

CPU (1 core)


Main Memory
(Instructions & Data)
Memory Interface
L2 Cache
L1
Data
Cache
L1
Instr.
Cache
ST
LD
INT
Registers
Shift
INT
FP
Registers
FMA
FP
add
FP
mult
Simplified View

Additional Functional Units

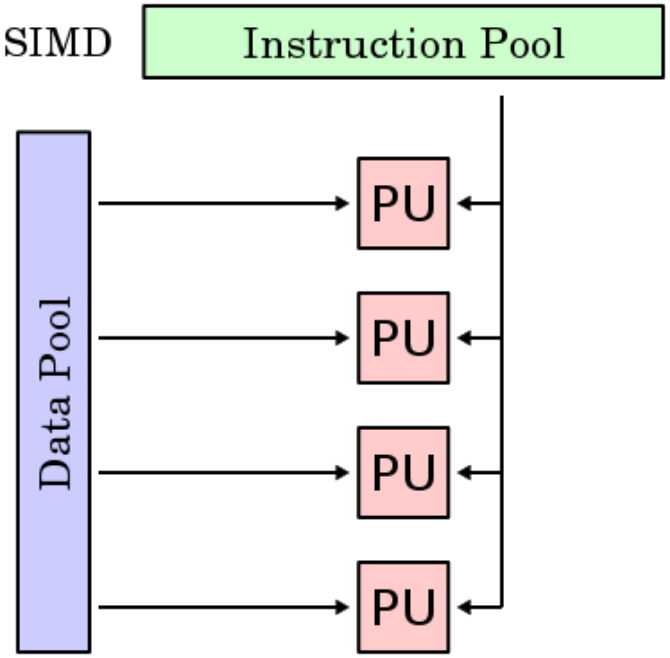
- **There's a lot more on the CPU than shown previously, e.g.**
  - L3 cache (~10 MB)
  - SQRT/Divide/Trig FP unit
  - "TLB" to cache memory addresses
  - Instruction decode
  - …

On-Chip Performance Enhancements

- **Chip designers have added lots of complexity to increase performance**
- **Instruction Parallelism**
  - Pipelined functional units (e.g. FPU)
  - Superscalar processors
- **Data Parallelism**
  - SIMD
- **Cache lines**
  - Data brought into cache in contiguous chunks
- **Out of order & speculative execution**

SIMD

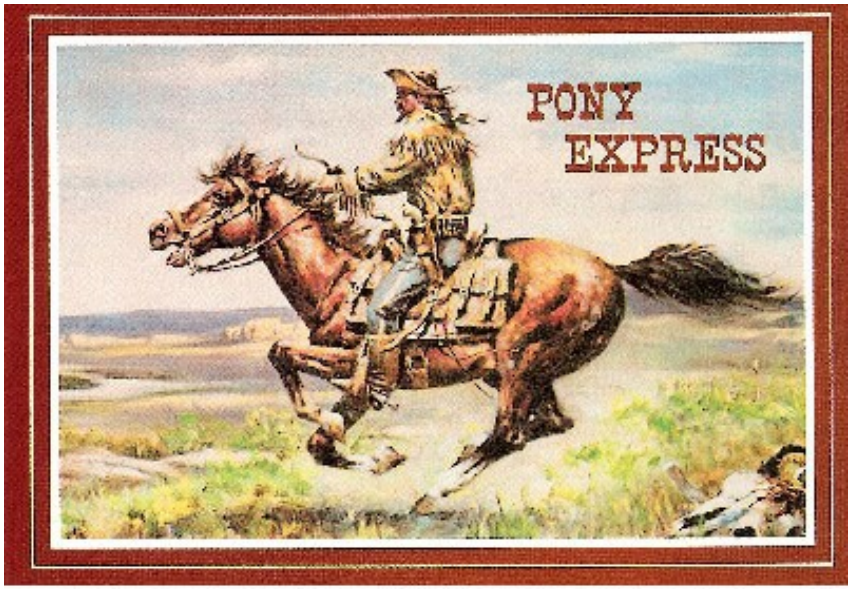
- Special registers hold multiple words of data
- A single instruction (e.g. floating point multiply) is applied to all the data at once
- "SSE[2-4]" : Streaming SIMD Extension instruction set for x86
- aka "Vectorization"



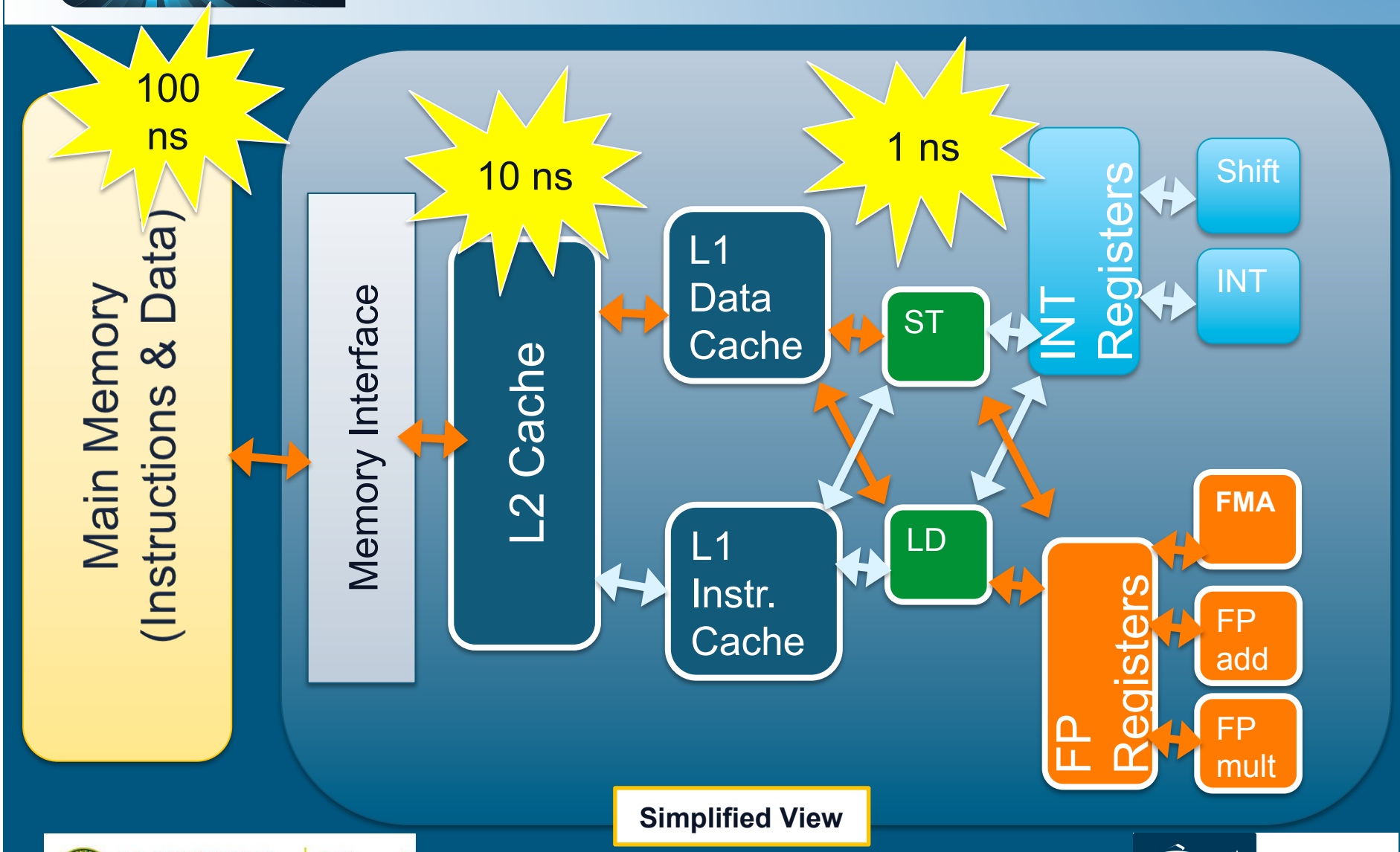
Latency

Time to send a zero-length message.

“Data Access Time”

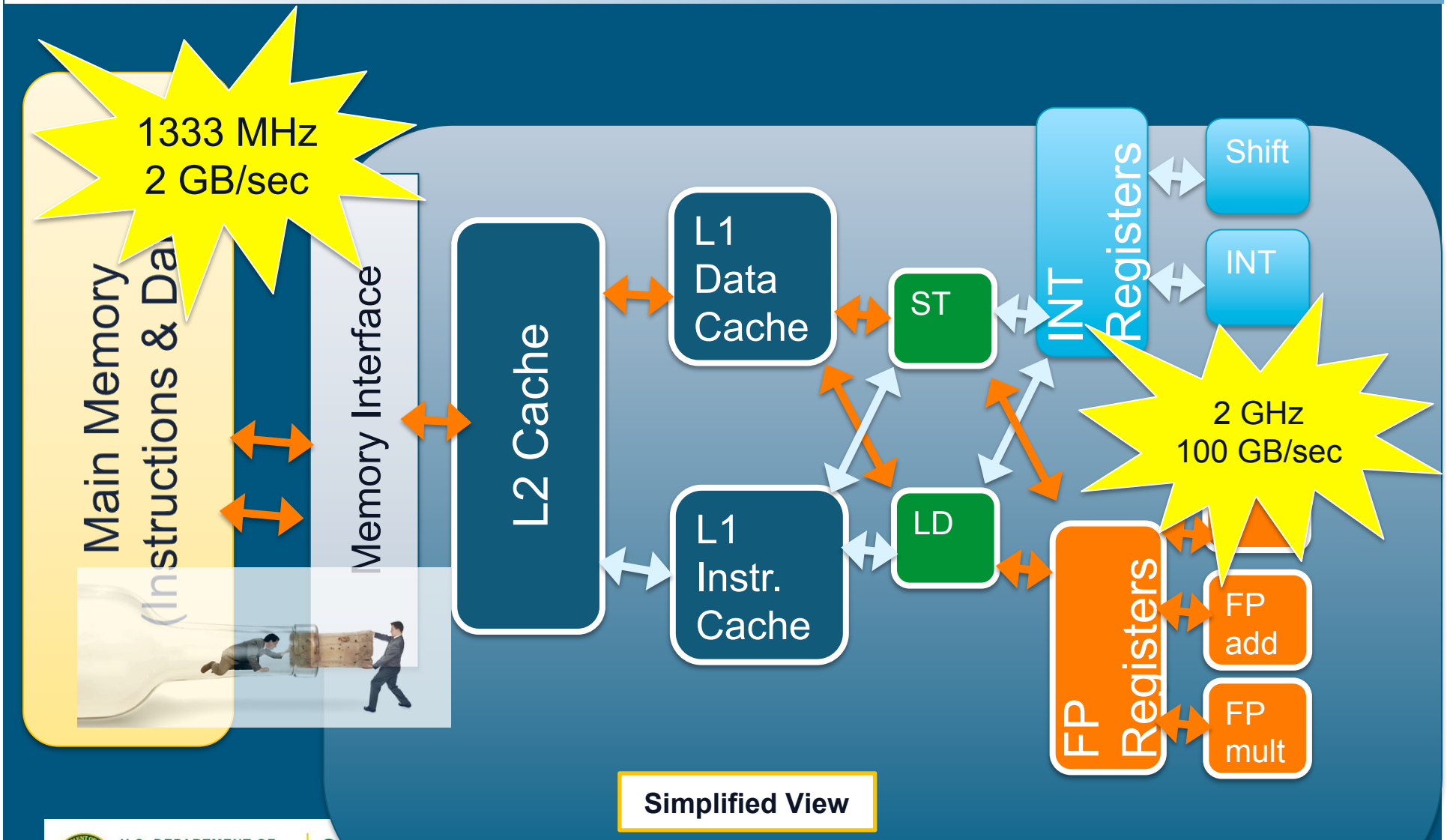


Latencies



Simplified View

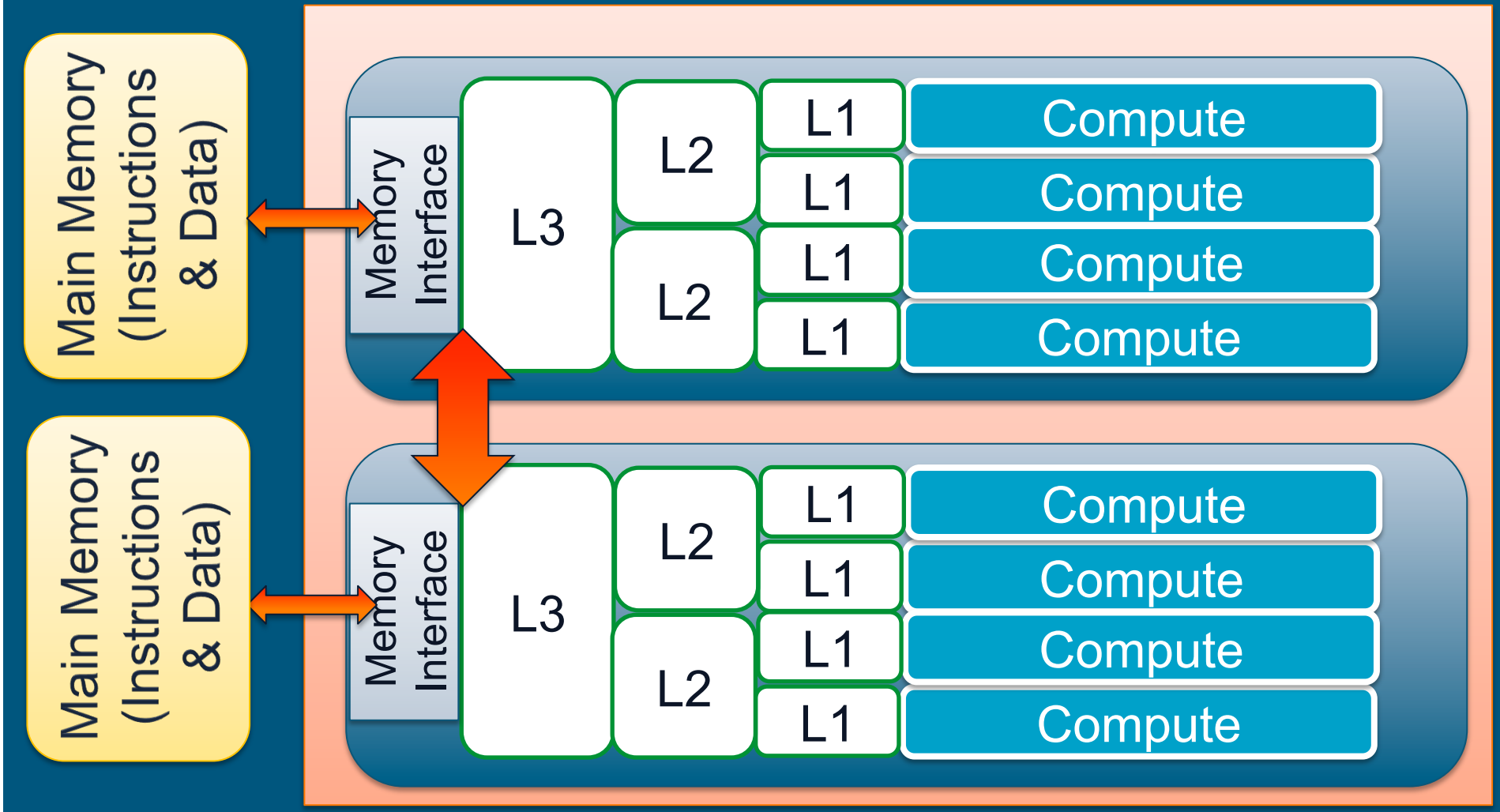
Memory Bandwidth Bottleneck



Multicore Processors

- All CPUs (processors) now have multiple compute "cores" on a single "chip" or "die" with possibly multiple chips per "socket" (the unit that plugs into the motherboard)
- Increased complexity
- The trend is for ever-more cores per die, mainly to hold down power needs

Hypothetical Socket (8 core)



HPC Nodes

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

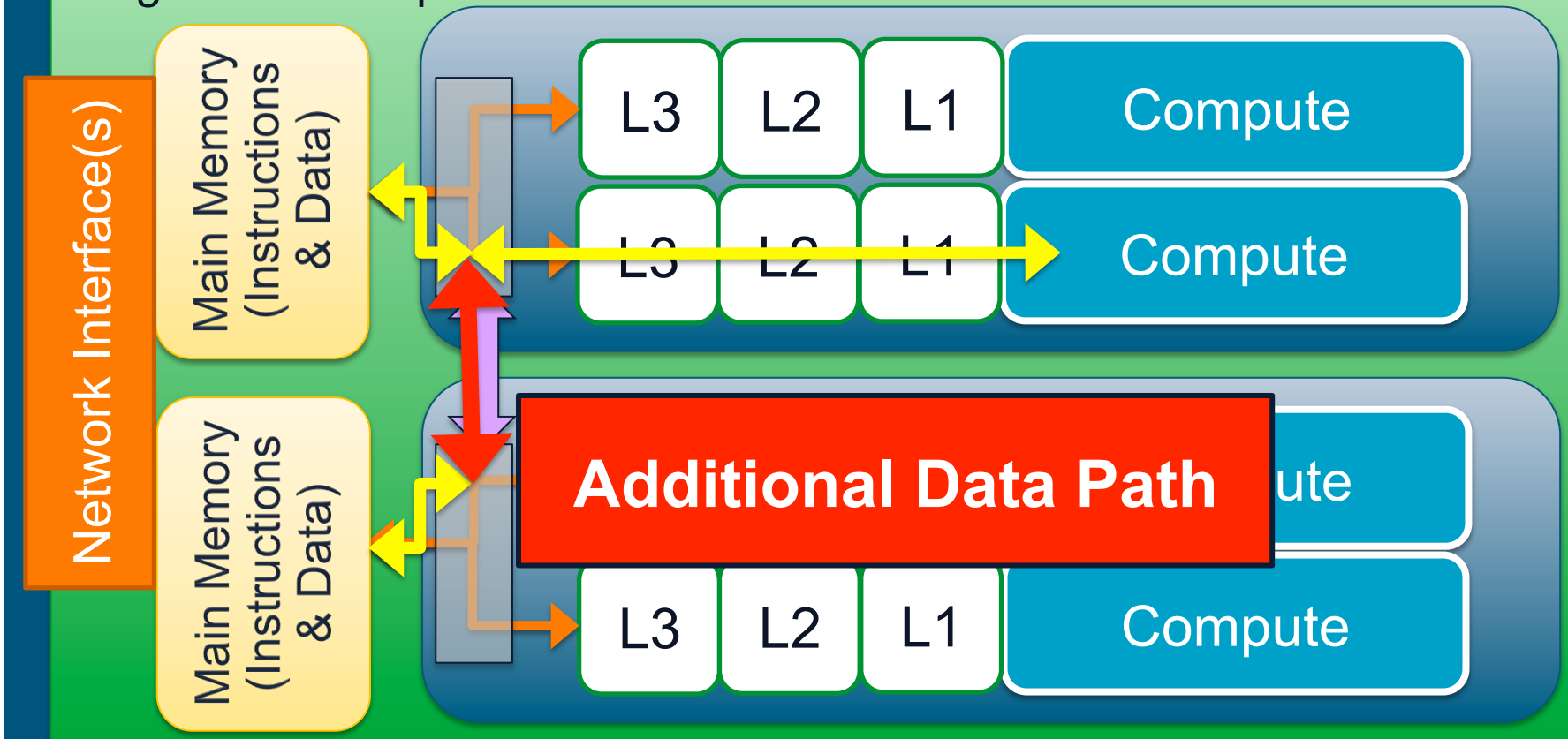
HPC Node

- **A "node" is a (physical) collection of CPUs, memory, and interfaces to other nodes and devices.**
  - Single memory address space
  - Memory access "on-node" is significantly faster than "off-node" memory access

Example NUMA Node

NUMA Node – Non-Uniform Memory Access

Single address space



GPUs

- **Many of top-ranked HPC systems are GPU-accelerated**
- **"Graphics Processing Units" are composed of 100s of simple "cores" that provide data-level on-chip parallelism**
- **Private, limited memory**
- **Yet more low-level complexity to consider**
  - Another memory hierarchy
  - Have to move data to GPU memory "by hand"
  - 400-800 cycle latency to GPU memory
- **Computations are extremely fast once data is in GPU memory**
- **Programmability is currently poor**
- **Legacy codes may have to be rewritten to minimize data movement**
- **Not all algorithms map well to GPUs**
- **What is their future in HPC?????**

Internode Networks

U.S. DEPARTMENT OF ENERGY | Office of Science

NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Distributed Memory Systems

- **Most HPC systems are "distributed memory"**
  - Many nodes, each with its own local memory and distinct memory space
  - Nodes communicate over a specialized high-speed, low-latency network

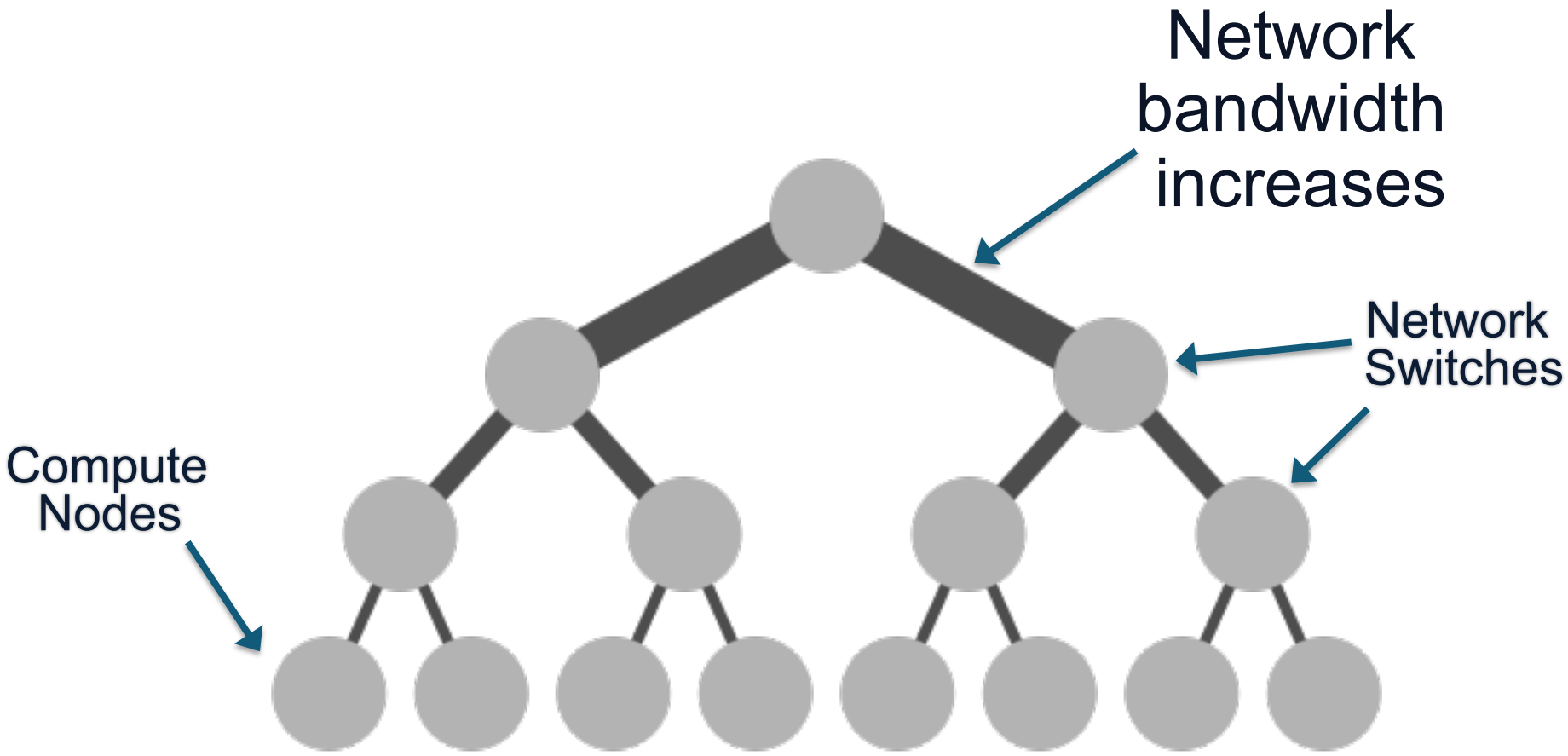
Interconnect Characteristics

- **Latency**
  - Latencies between different nodes may be different
  - Typically ~ a few µsec
- **Bandwidth**
  - Typically ~ a few GB/sec in/out of a node

Main Networks Types

- **Switched**
  - Network switches connect and route network traffic over the interconnect
- **Mesh**
  - Each node sends and receives its own data, and also relays data from other nodes
  - Messages hop from one node to another until they reach their destination (must deal with routing around down nodes)

Fat Tree Switched Network

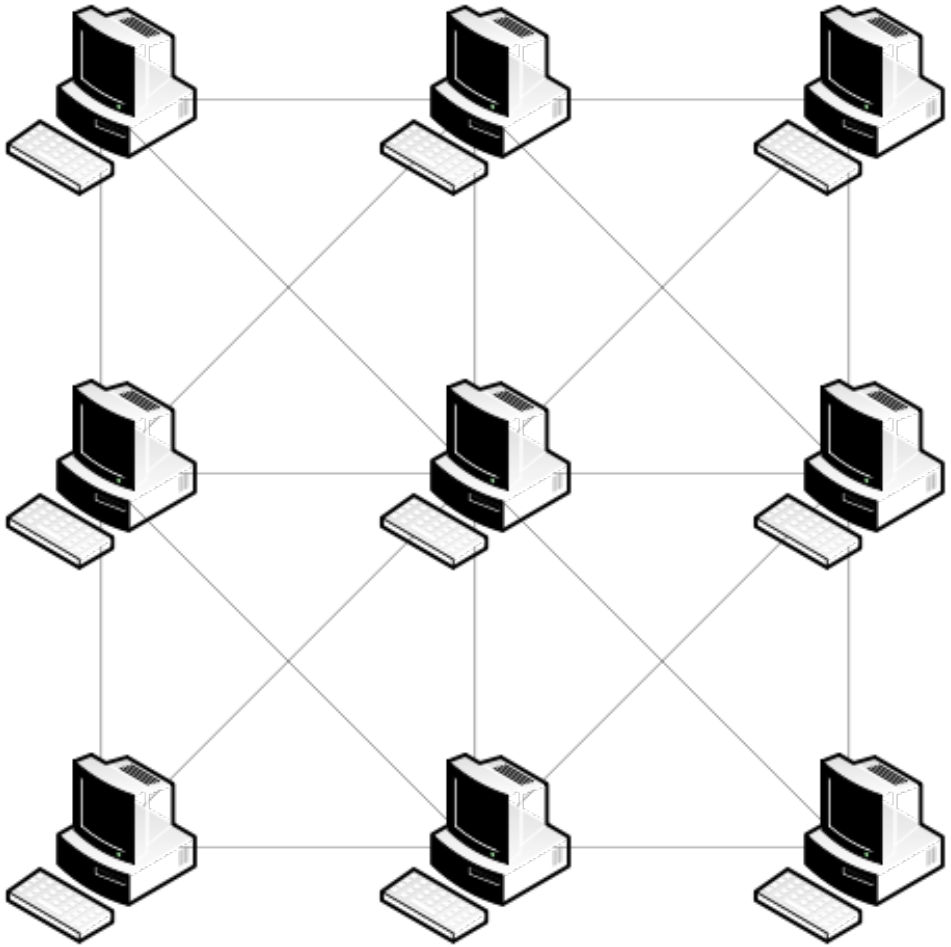


e.g., Infiniband (IB)

Mesh Networks

Mesh network topologies can be complex

Grids
Cubes
Hypercubes
Tori



Disk and Tape Storage

U.S. DEPARTMENT OF ENERGY | Office of Science

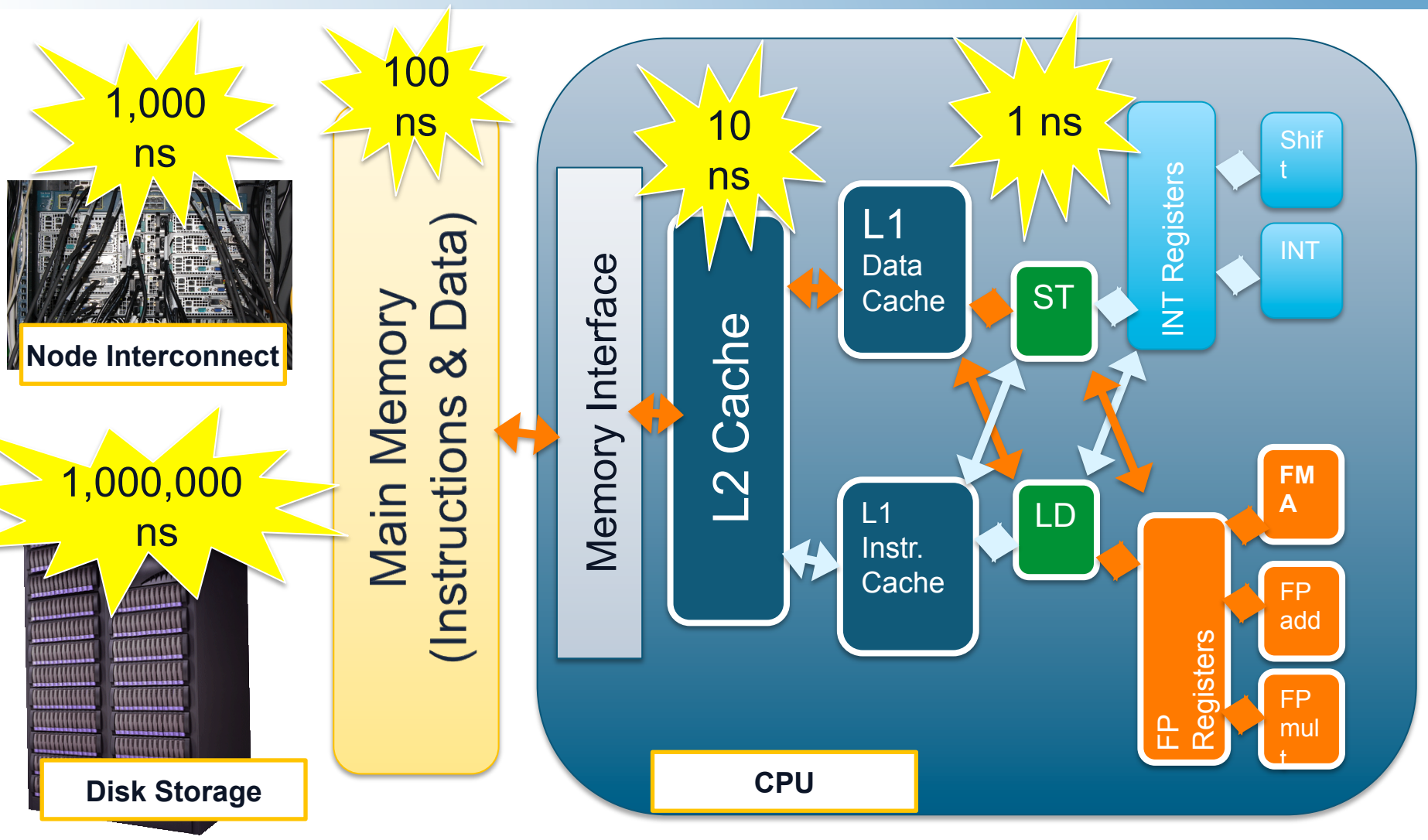
NERSC National Energy Research Scientific Computing Center

BERKELEY LAB Lawrence Berkeley National Laboratory

Largest and Slowest Memory

- **File storage is the slowest level in the data memory hierarchy**
  - Not uncommon for checkpoints / memory dumps to be taking a large fraction of total run time (>50%?)
  - NERSC users say they want no more than 10% of time to be IO

Latencies


1,000 ns
Node Interconnect
100 ns
Main Memory (Instructions & Data)
1,000,000 ns
Disk Storage
10 ns
Memory Interface
L2 Cache
1 ns
L1 Data Cache
L1 Instr. Cache
ST
LD
INT Registers
Shift
INT
FP Registers
FMA
FP add
FP mult
CPU

Archival Storage

- **Large, Permanent Storage**
  - Many PBs
  - Often tape storage fronted by a disk cache
  - Often accessed via ftp, grid tools, and/or custom clients (e.g. hsi for HPSS)

HPC Operating Systems

- **Most HPC OSs are Linux-Based**
  - IBM AIX on POWER (also offers Linux)
- **"Generic" Cluster Systems**
  - Full Linux OS on each node
- **Specialized HPC Systems (e.g., Cray XT series, IBM Blue Gene)**
  - Full Linux OS on login, "services" nodes
  - Lightweight kernel on compute nodes
    - Helps performance
    - May hinder functionality (DLLs, dynamic process creation, some system calls may not be supported.)

Summary: Why Do You Care About Architecture?

- **Details of machine are important for performance**
  - Processor and memory system (not just parallelism)
  - What to expect? Use understanding of hardware limits
- **There is parallelism hidden within processors**
  - Pipelining, SIMD, etc
- **Locality is at least as important as computation**
  - Temporal: re-use of data recently used
  - Spatial: using data nearby that recently used
- **Machines have memory hierarchies**
  - 100s of cycles to read from DRAM (main memory)
  - Caches are fast (small) memory that optimize average case
- **Can rearrange code/data to improve locality**

National Energy Research Scientific Computing Center

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory