CS343: Operating System

System Arch & OS Structure, Mode and Services

Lect03 : 4th Aug 2023

Dr. A. Sahu

Dept of Comp. Sc. & Engg.

Indian Institute of Technology Guwahati

Outline

- Storage Structure
- Caching DMA
- System Architecture: MP
- OS Structure
- OS Mode
- OS Services

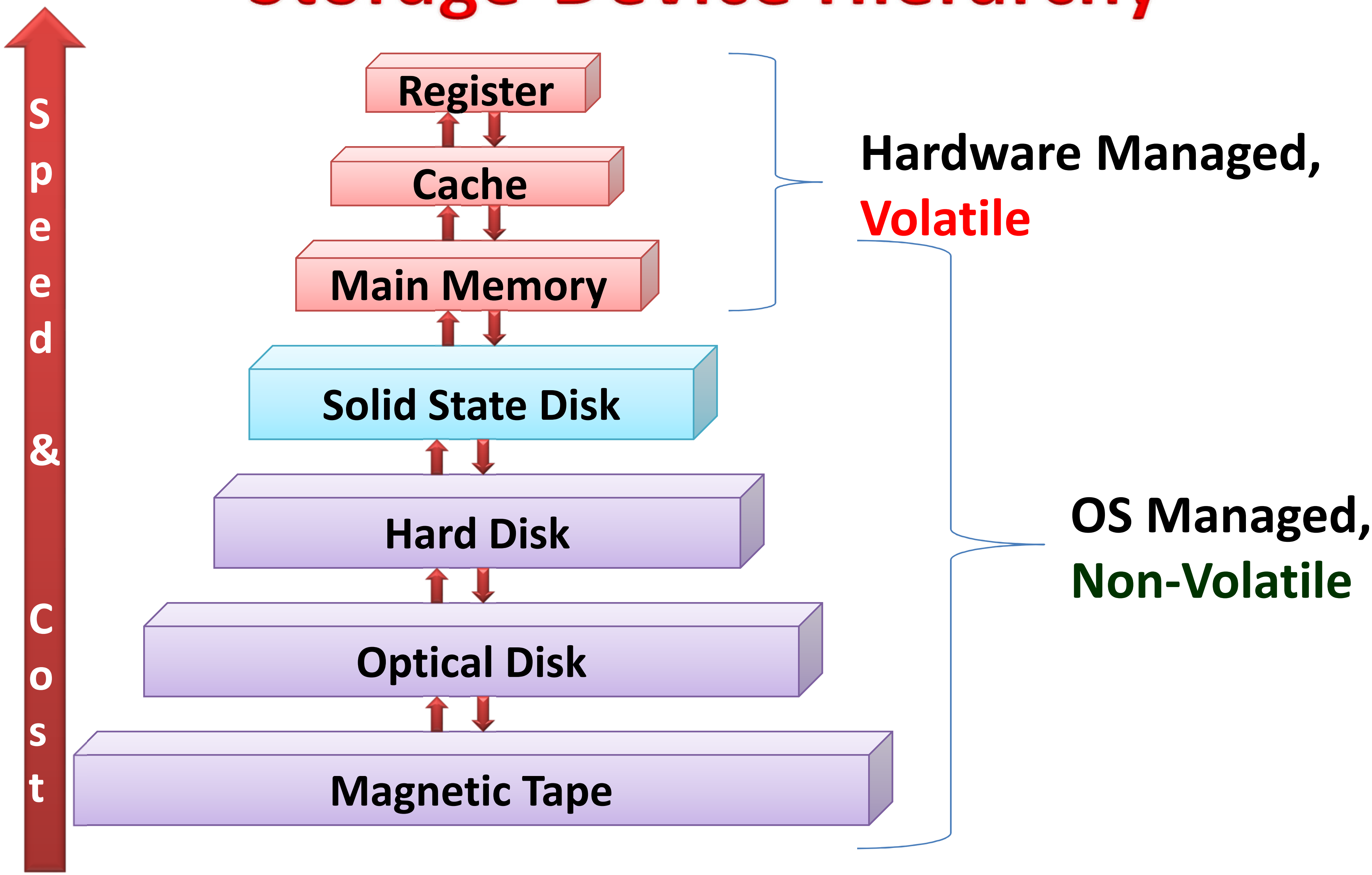
Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - **Random access,** Typically **volatile**
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies, Becoming more popular

Storage Hierarchy

- Storage systems organized in hierarchy
  - Speed, Cost, Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
  - Locality of Reference
  - Locality Principle (Spatial and Temporal Locality)
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel

Storage-Device Hierarchy



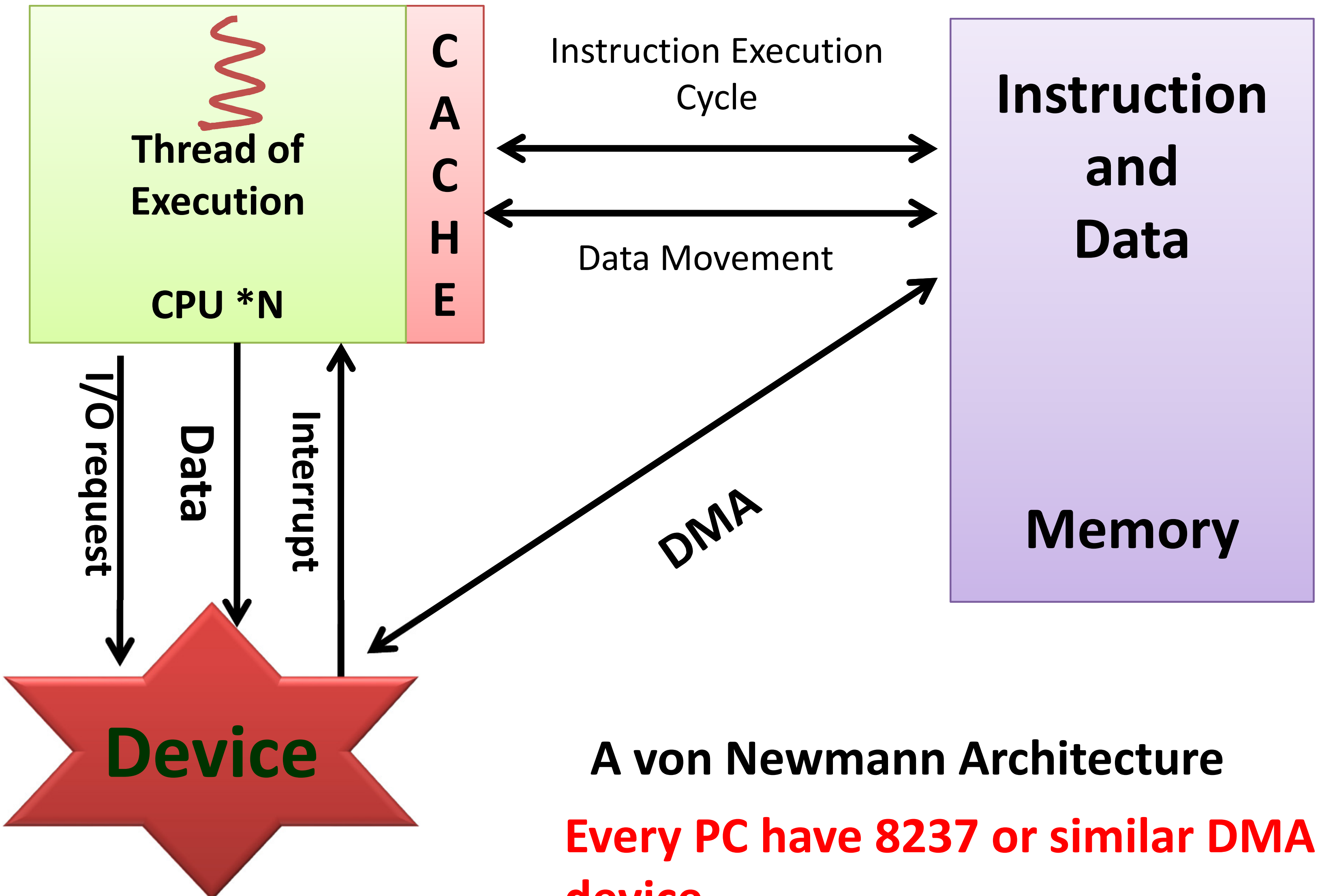
Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied
  - From slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy

Direct Memory Access Structure

- **DMA**
- Used for high-speed I/O devices able to transmit information at close to memory speeds
- ***Device controller*** transfers blocks of data from buffer storage directly to main memory without CPU intervention
  - ***Taxi Wala to handle all Pickup and Drop of Conference***
- Only one interrupt is generated per block, rather than the one interrupt per byte

I/O transfer in a Modern Computer



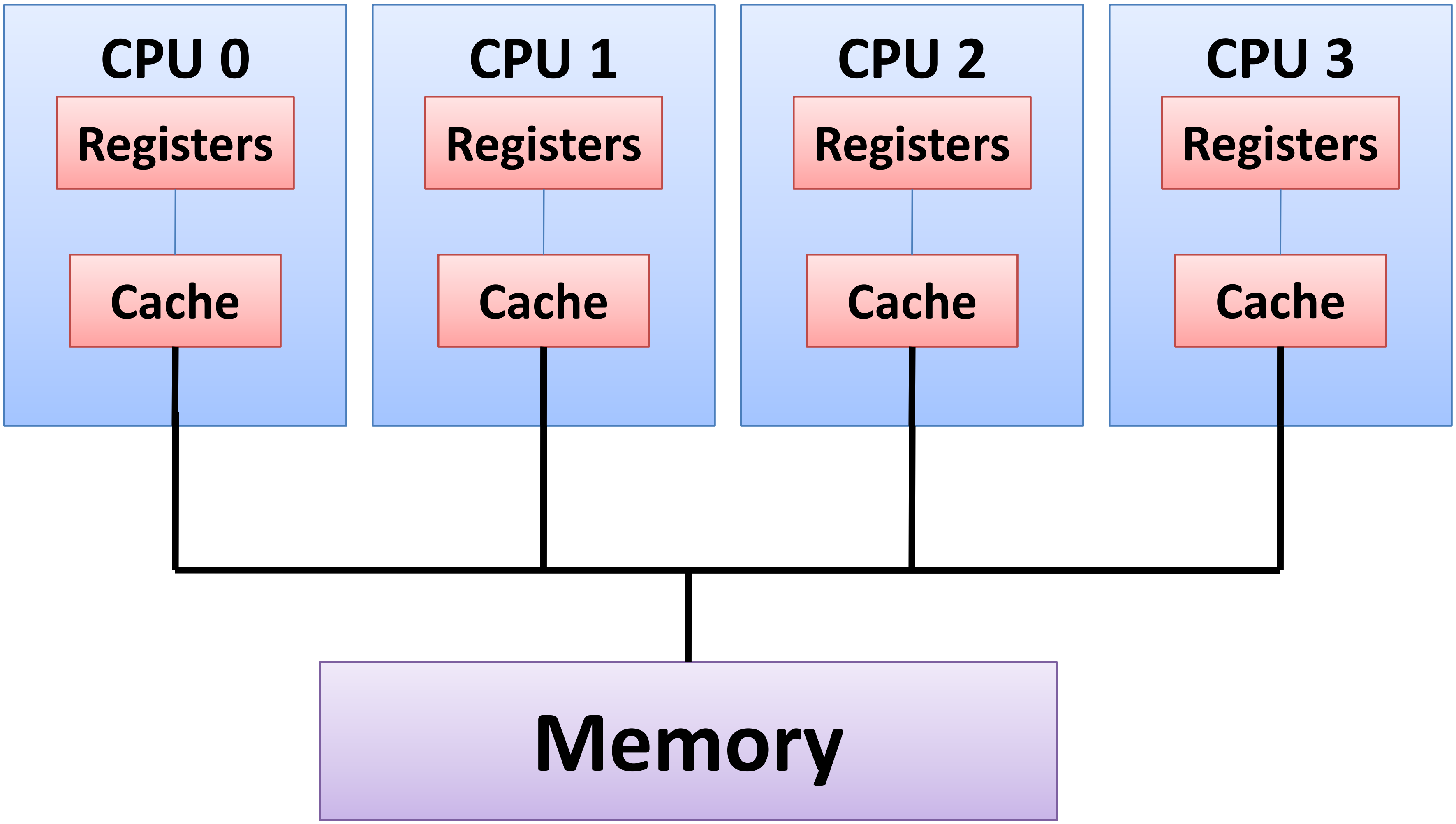
A von Newmann Architecture

Every PC have 8237 or similar DMA device

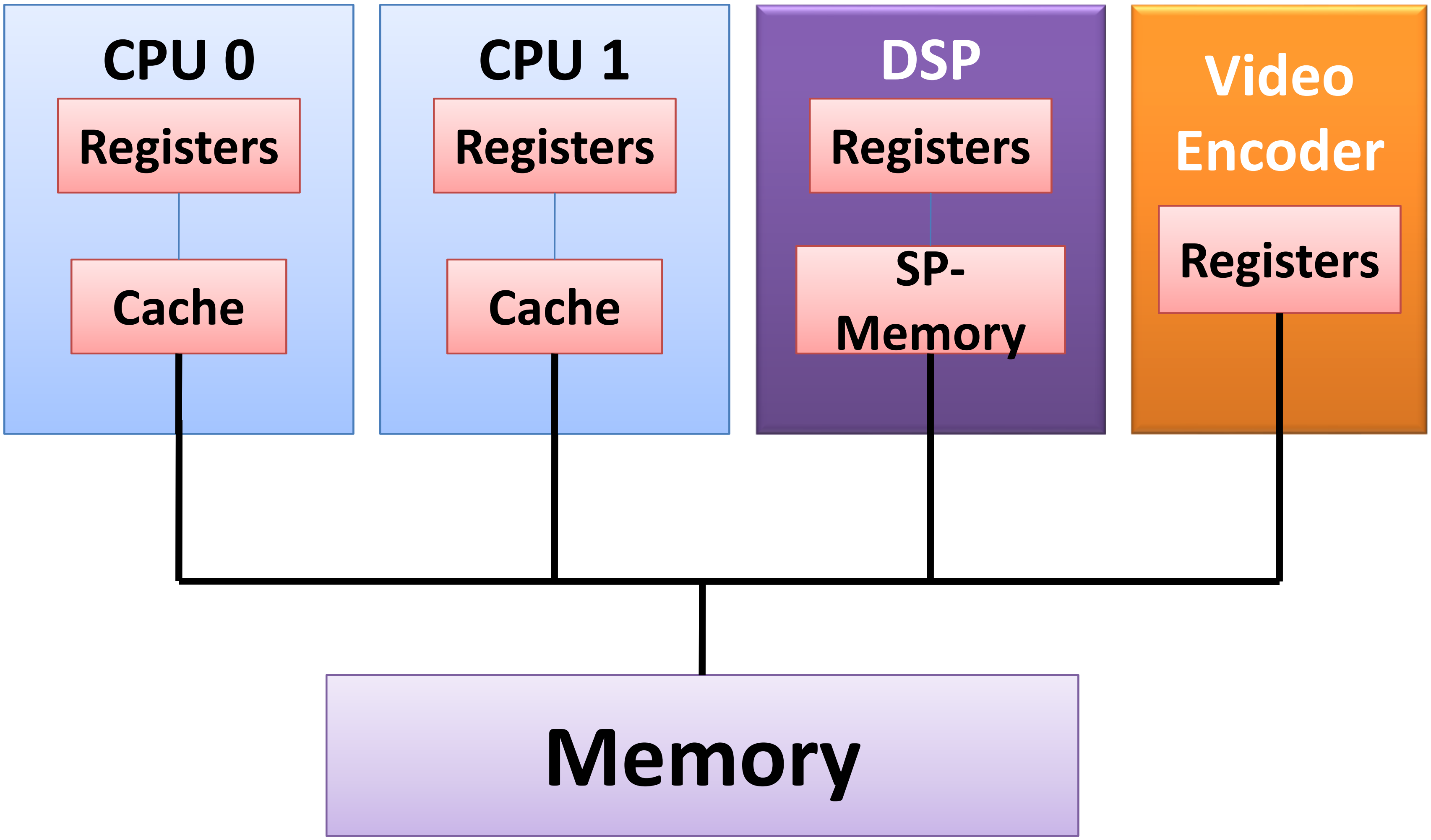
Computer-System Architecture

- Most systems use a general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** growing in use and importance
- Also known as **parallel systems**, **tightly-coupled systems** Advantages include:
  - **Increased throughput, Economy of scale**
  - **Increased reliability** – graceful degradation or fault tolerance
- **Asymmetric Multiprocessing** – each processor is assigned a specific task.
- **Symmetric Multiprocessing** – each processor performs all tasks

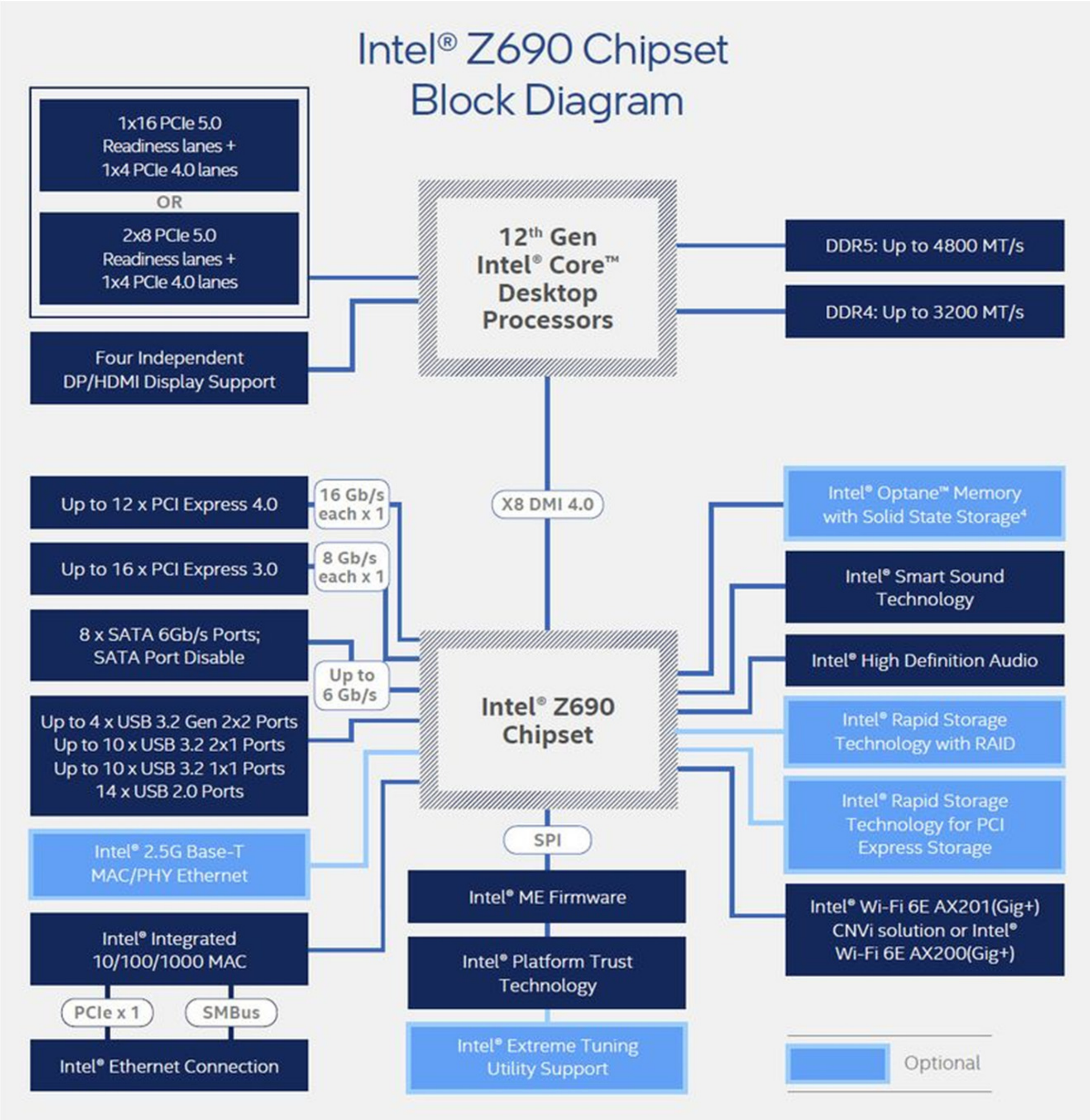
Symmetric Multiprocessing Architecture

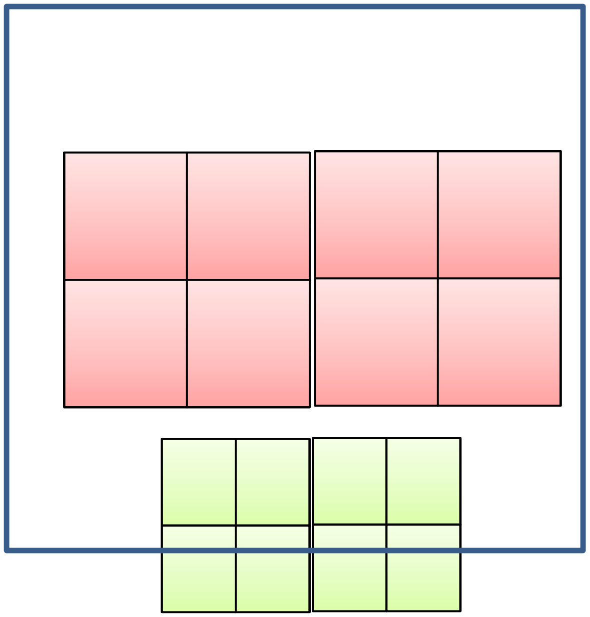


Asymmetric Architecture



Core i7 12th and Z690 ChipSet





Cores: 8 P-cores + 8 E-cores, 24 threads
Max P-core freq: 5.1 GHz
Max E-core freq: 3.9 GHz
L3 Cache: 30MB
L2 Cache: 14MB

Number of Cores & Thread

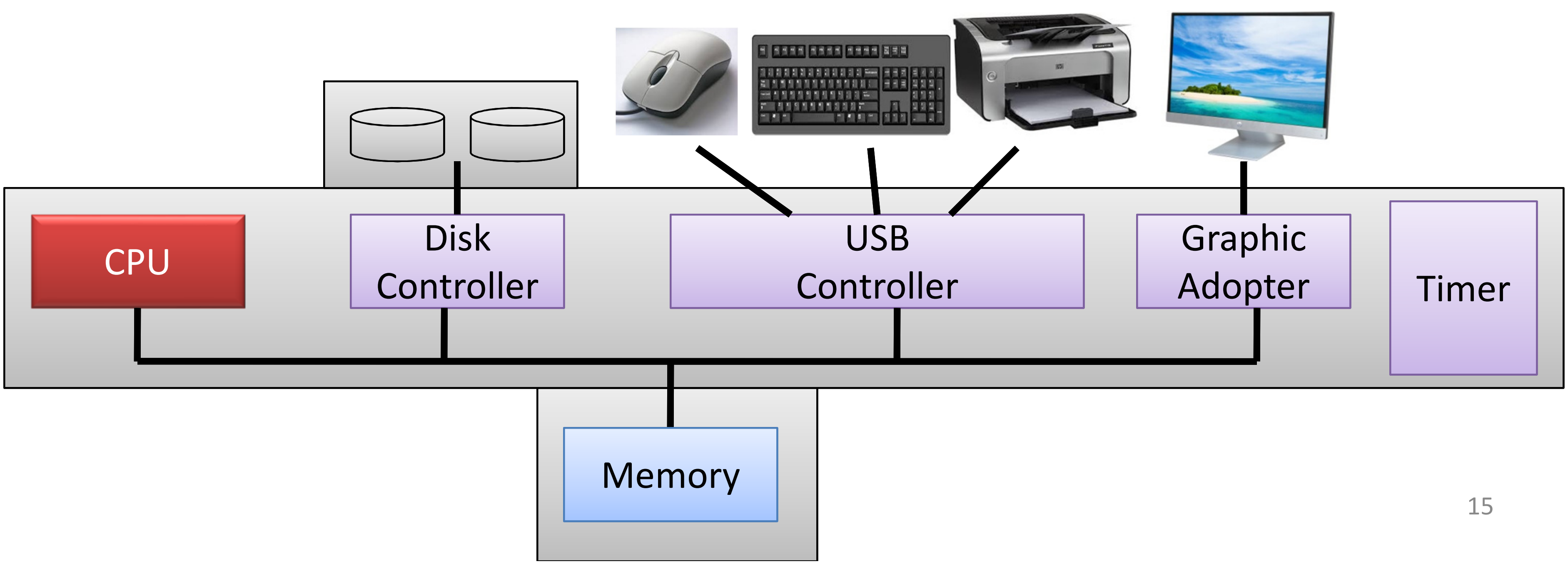
- Intel Xeon Phi: 72 cores, 4T/C → 288 HTs
- AMD ThreadRipper: 128C → 256 HTs
- NetLogic MicroSystems XLP: 32 core, 128 threads
- Tilera   TILE64: 64-core
- ClearSpeed   CSX700: 192 cores
- **GPUs Nvidia & AMD : Thousands of Tiny Cores**

Multithreaded Architecture

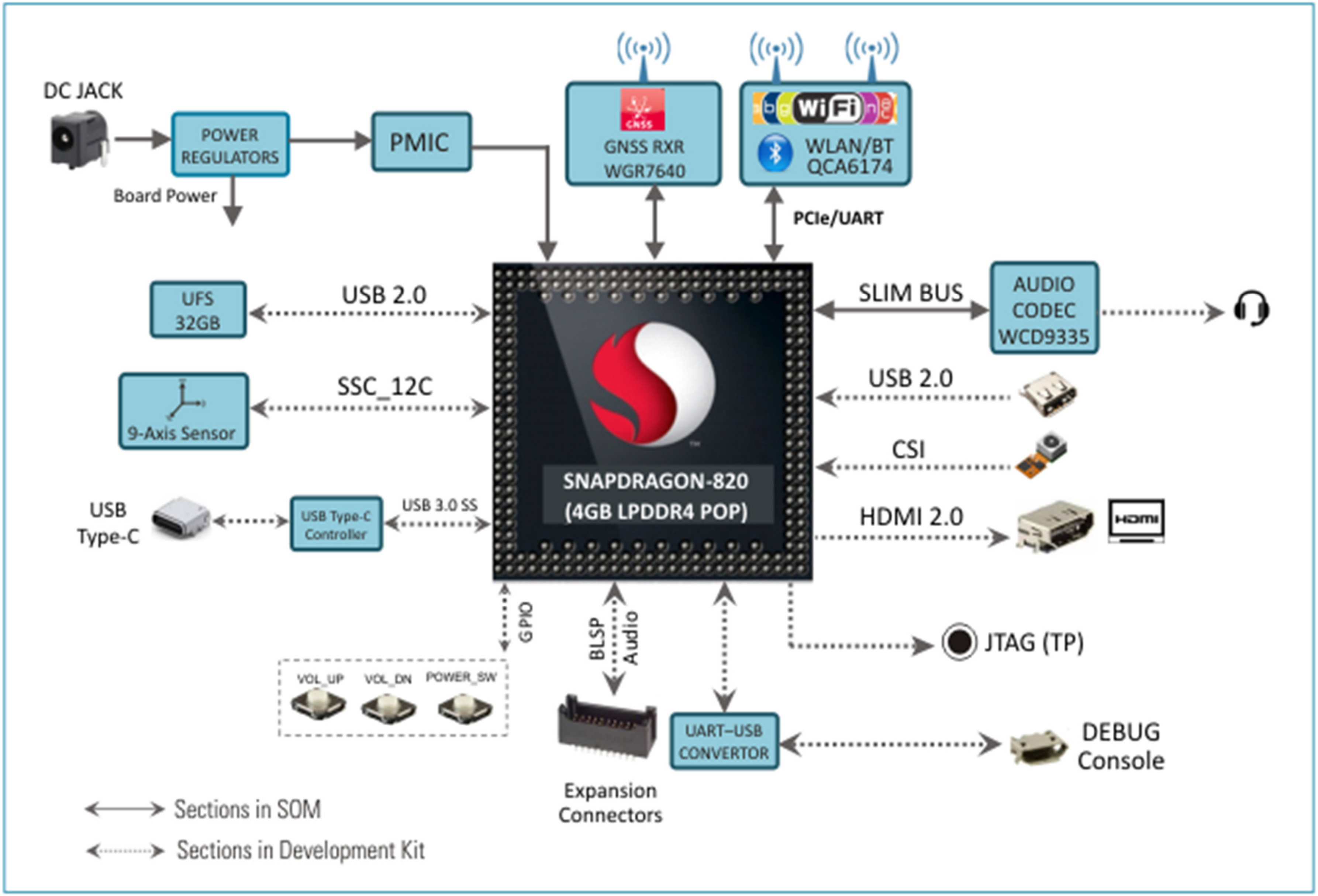
- Core/Processor
- Each processor can be threaded
  - How many thread can be handled: 2,4,8
  - Hardware threading (Ex-Intel Hyper Threading)
- Example
  - Intel Core i7 : 4 cores, 8 Threads
  - Intel PIV HT : 1 core, 2 Thread
  - Intel Xeon E5-2687: 8 cores, 16 threads
  - Intel Atom C2780 : 8 cores, 8 threads

Processor Vs Micro-controller Vs SOC

- Processor : In a Single Chip
- Micro-controller: In a single chip
  - Processors + (USB, Disk,..) Controllers +Timers+GA
- SOC : System on Chip, **All mobile Platforms**
  - Microcontroller + Memory + *Wireless Controller (Opt)*



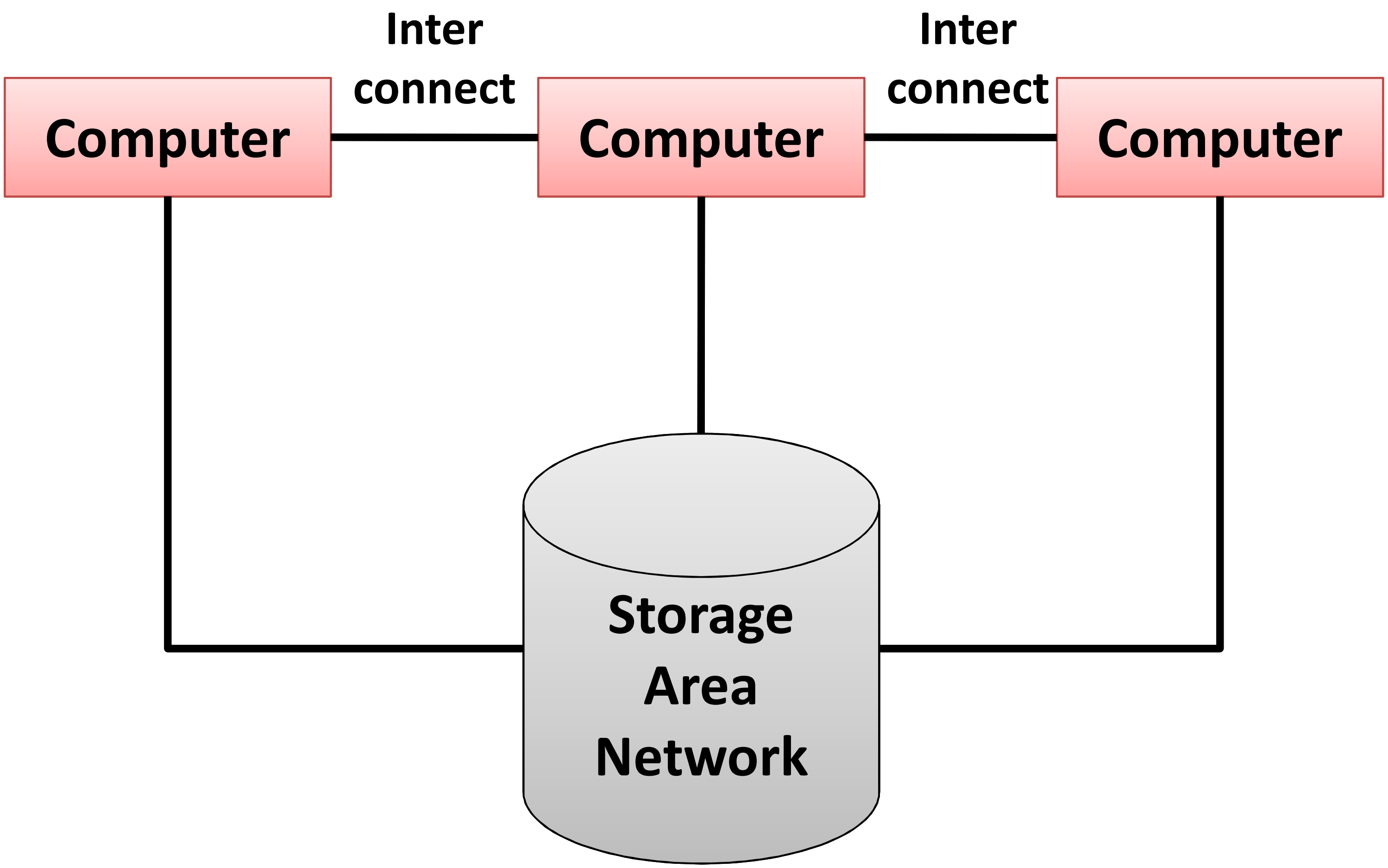
Qualcomm SD-Block Diagram



Clustered Systems

- Multiple systems working together
- Usually sharing storage via a **storage-area network(SAN)**
- Provides a **high-availability** service which survives failures
  - **Asymmetric clustering** has one machine in hot-standby mod
  - **Symmetric clustering** has multiple nodes running applications, monitoring each other
- Some clusters are for **HPC**
  - Applications must be written to use **parallelization**
- Some have **distributed lock manager** (**DLM**) to avoid conflicting operations

Clustered Systems



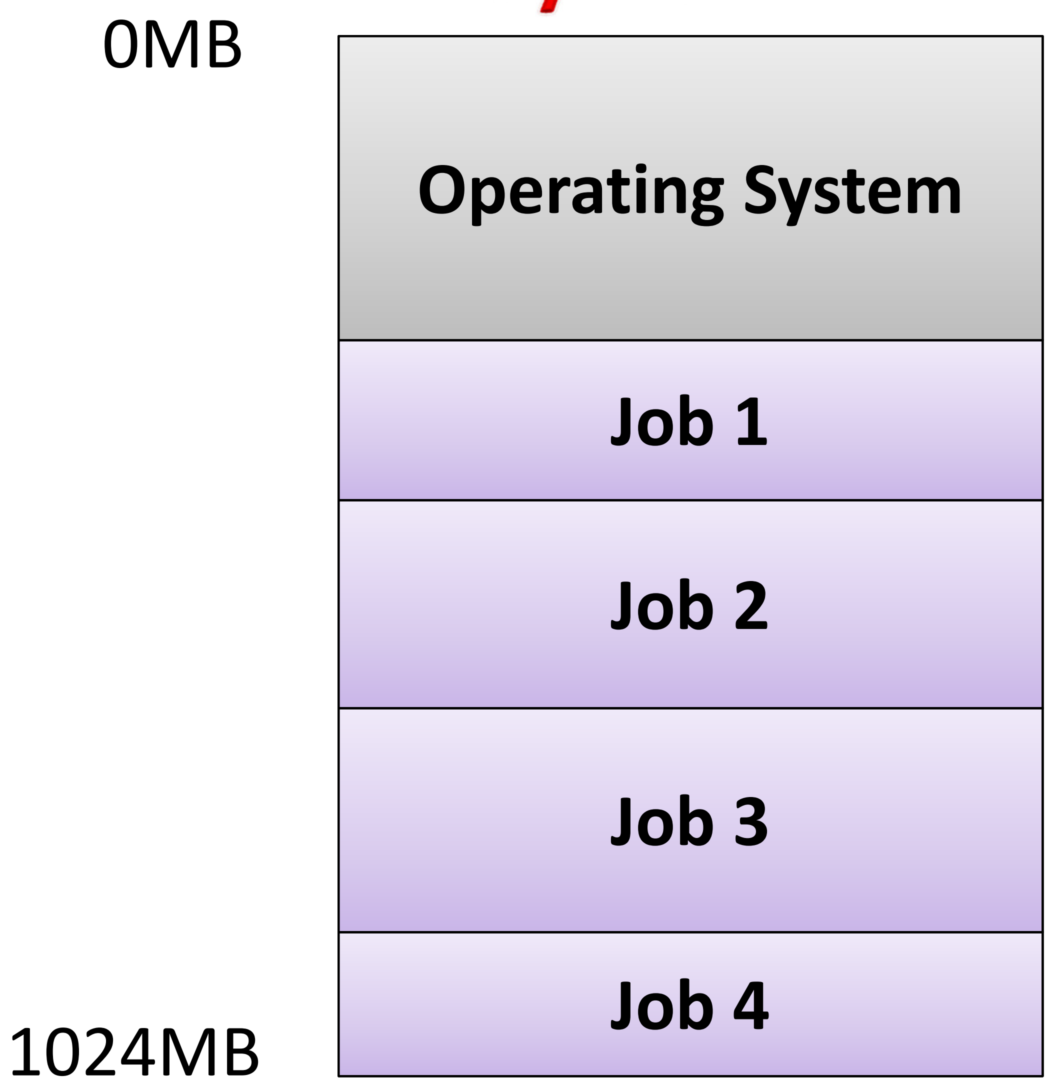
OS Structures Basics

Multiprogramming & Multi Tasking

OS Structure: Multiprogramming

- **Multiprogramming**/**Batch system**
  - needed for efficiency
- Single user cannot keep CPU and I/O devices busy at all times
- Multiprogramming organizes jobs (code and data), so CPU always has one to execute
- A subset of total jobs in system is kept in memory
- One job selected and run via **job scheduling**
- When it has to wait (for I/O for example), OS switches to another job

Memory Layout for Multi-programmed System



OS Structure: Time Sharing

- Timesharing (multitasking)
  - Is logical extension
- CPU switches jobs so frequently
  - That users can interact with each job while it is running
  - Creates an **interactive** computing

Time Sharing (Multitasking)

- **Response time** should be < 1 second
- Each user has at least one program executing in memory ⇨ **process**
- If several jobs ready to run at the same time ⇨ **CPU scheduling**
- If processes don't fit in memory, **swapping** moves them in and out to run
- **Virtual memory** allows execution of processes not completely in memory

OS Operation Mode

Interrupt, Kernel Mode & User Mode

Operating-System Operations

- **Interrupt driven** (H/W and S/W)
  - H/W interrupt by one of the devices
  - S/W interrupt (**exception** or **trap):**
    - Software error (e.g., division by zero)
    - Request for OS service
    - Other process problems include
      - infinite loop
      - processes modifying each other or the operating system

Operating-System Operations (cont.)

- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - Provides ability to distinguish when system is running user code or kernel code
    - Some instructions designated as **privileged**, only executable in kernel mode
    - System call changes mode to kernel, return from call resets it to user
- Increasingly CPUs support multi-mode operations
  - i.e. **virtual machine manager** (**VMM**) mode for guest **VMs**

Thanks