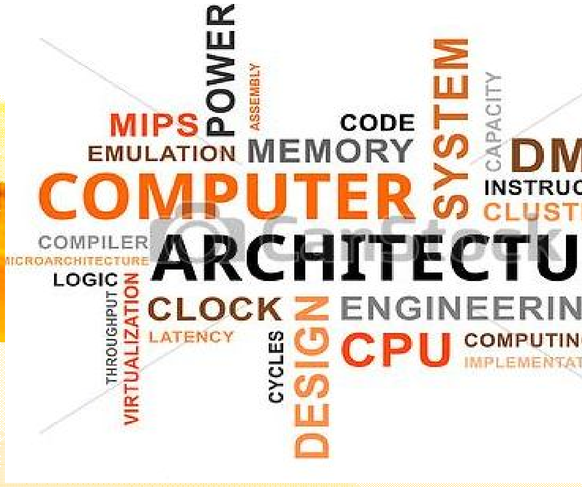
UNIT IV
MEMORY SYSTEM

Basic concepts of Semiconductor RAMs - ROMs – Speed, Size and Cost – Cache memories – Performance consideration – Virtual memory – Memory Management requirements – Secondary storage - **Case Study: Memory Organization in Multiprocessors**



Recap the previous Class

Why Choose a Multiprocessor?

- A single CPU can only go so fast, **use more than one CPU** to improve performance
- Multiple users
- Multiple applications
- Multi-tasking within an application
- Responsiveness and/or throughput
- Share hardware between CPUs

Multiprocessor Symmetry

- In a multiprocessing system, **all CPUs may be equal**, or some may be reserved for special purposes.
- Systems that treat **all CPUs equally** are called symmetric multiprocessing (SMP) systems.
- If all CPUs **are not equal**, system resources may be divided in a number of ways, including asymmetric multiprocessing (ASMP), non-uniform memory access (NUMA) multiprocessing, and clustered multiprocessing.

Instruction and Data Streams

<u>ultiprocessors can be used in different ways:</u>

Uniprossesors (single-instruction, single-data or SISD)

Within a single system to execute **multiple, independent sequences** of instructions in multiple contexts (multiple-instruction, multiple-data or MIMD);

A single sequence of instructions in multiple contexts (single-instruction, multiple-data or SIMD, often used in vector processing);

Multiple sequences of instructions in a single context (multiple-instruction, single-dat or MISD, used for redundancy in fail-safe systems and pipelined processors or hyper threading).

Processor Coupling

ightly-coupled multiprocessor systems:

Contain multiple CPUs that are connected at the bus level.

These CPUs may have access to a central shared memory (Symmetric Multiprocessing, or SMP), or may participate in a memory hierarchy with both local and shared memory (Non-Uniform Memory Access, or NUMA).

Example: IBM p690 Regatta, Chip multiprocessors, also known as multi-core computing.

Processor Coupling

Loosely-coupled multiprocessor systems:

- Often referred as clusters
- Based on multiple standalone single or dual processor commodity computers interconnected via a high speed communication system, such as Gigabit ethernet.
- **Example:** Linux Beowulf cluster

Multiprocessor Communication Architectures

Message Passing

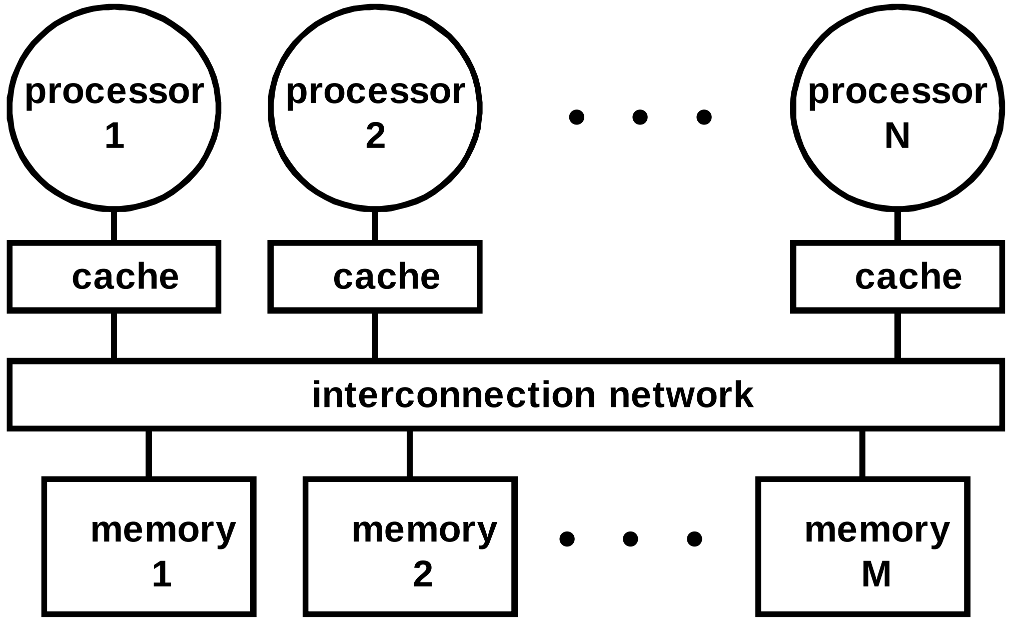
- Separate address space for each processor
- Processors communicate via message passing
- Processors have private memories
- Focuses attention on costly non-local operations

Shared Memory

- Processors communicate with shared address space
- Processors communicate by memory read/write
- Easy on small-scale machines
- Lower latency
- SMP or NUMA

Shared-Memory Processors

- Single copy of the OS (although some parts might be parallel)
- Relatively easy to program and port sequential code to
- Difficult to scale to large numbers of processors



UMA machine block diagram

Types of Shared-Memory Architectures

UMA

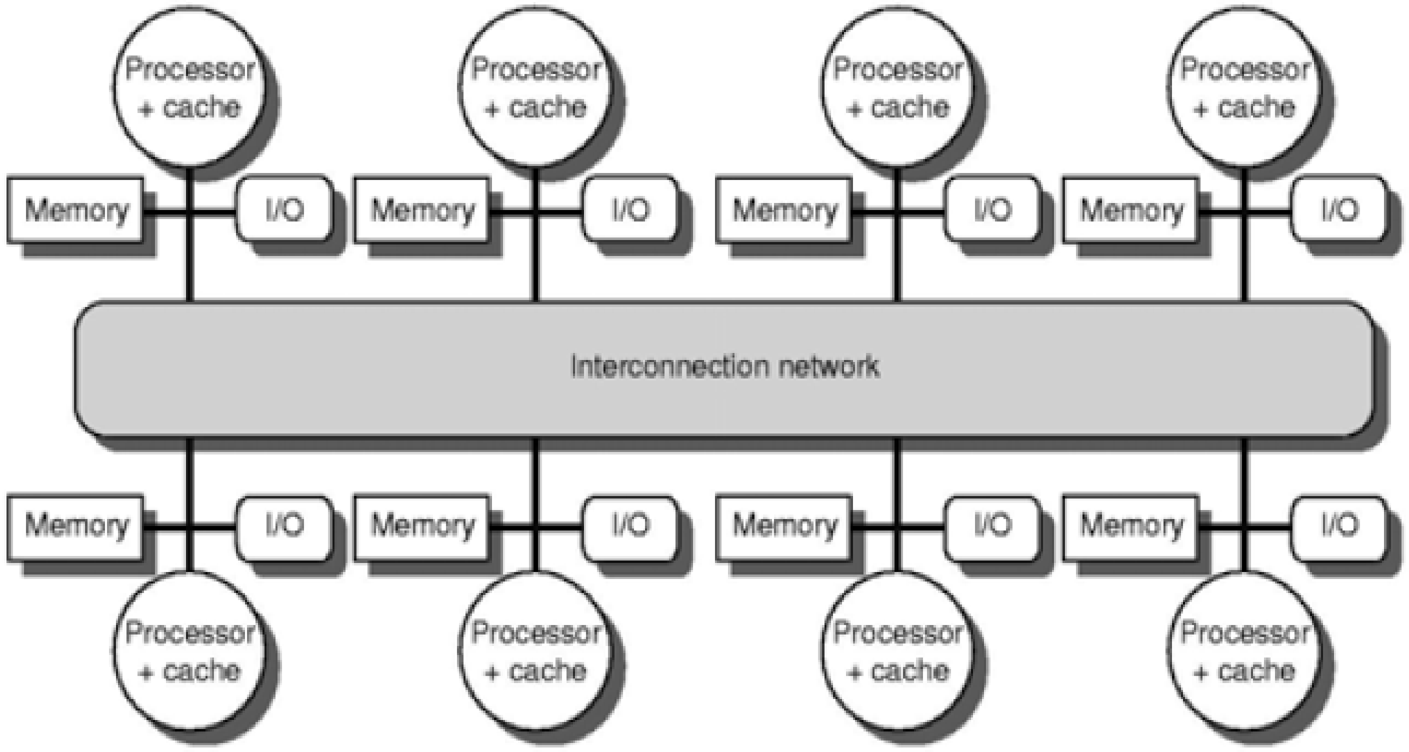
- Uniform Memory Access
- Access to all memory occurred at the same speed for all processors.

NUMA

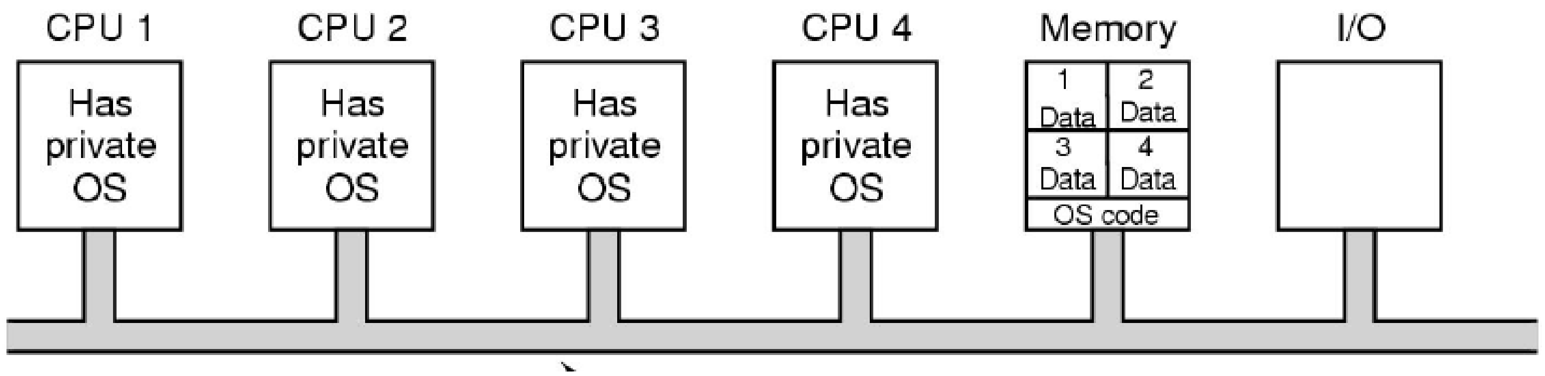
- Non-Uniform Memory Access a.k.a. “Distributed Shared Memory”.
- Typically interconnection is grid or hypercube.
- Access to some parts of memory is faster for some processors than other parts of memory.
- Harder to program, but scales to more processors

NUMA

- All memories can be addressed by all processors, but access to a processor’s own local memory is faster than access to another processor’s remote memory.



OS Option 1



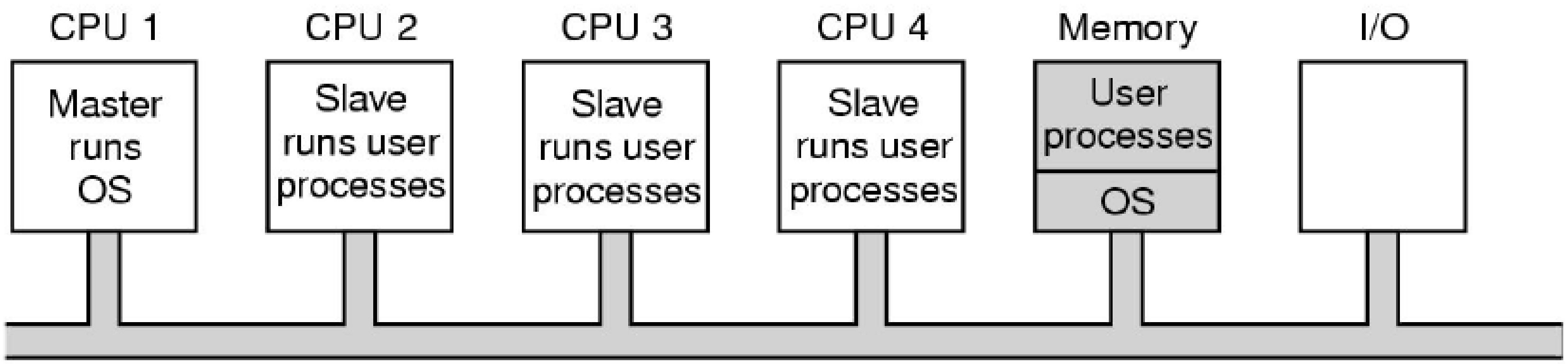
Each CPU has its own OS

- Statically allocate physical memory to each CPU
- Each CPU runs its own independents OS , Share peripherals
- Simple to implement
- Avoids concurrency issues by not sharing
- Issues: 1. Each processor has its own scheduling queue.

  2. Each processor has its own memory partition.

  3. Consistency is an issue with independent disk buffer caches and potentially shared files.

OS Option 2


CPU 1
Master runs OS
CPU 2
Slave runs user processes
CPU 3
Slave runs user processes
CPU 4
Slave runs user processes
Memory
User processes
OS
I/O

aster-Slave Multiprocessors

OS mostly runs on a single fixed CPU.

User-level applications run on the other CPUs.

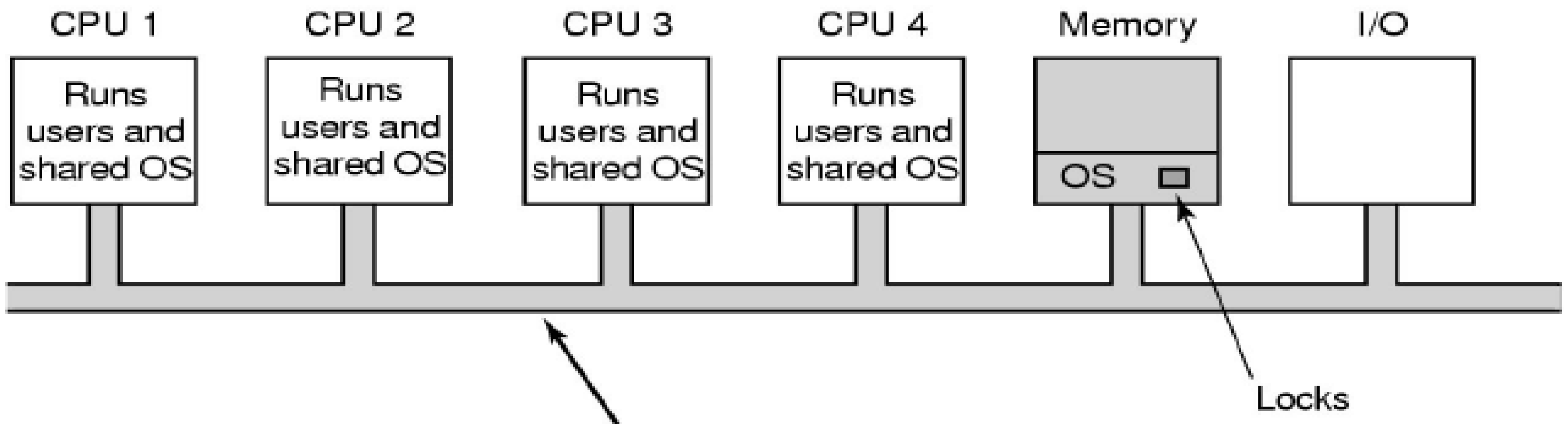
Single to implement, Memory can be allocated as needed to all CPUs.

Issues: Master CPU becomes the bottleneck.

OS Option 3

Symmetric Multiprocessors (SMP)

- OS kernel runs on all processors
- One alternative: A single mutex (mutual exclusion object) that make the entire kernel a large critical section
- Issues: A difficult task; Code is mostly similar to uniprocessor code; hard part is identifying independent parts that don't interfere with each other



Earlier Example
Quad-Processor Pentium Pro

- SMP, bus interconnection.
- 4 x 200 MHz Intel Pentium Pro processors.
- 8 + 8 Kb L1 cache per processor.
- 512 Kb L2 cache per processor.
- Snoopy cache coherence.
- Employed in Compaq, HP, IBM, NetPower.
- OS: Windows NT, Solaris, Linux, etc.

TEXT BOOK

Carl Hamacher, Zvonko Vranesic and Safwat Zaky, “Computer Organization”, McGraw-Hill, 6th Edition 2012.

REFERENCES

1. David A. Patterson and John L. Hennessey, “Computer organization and design”, MorganKauffman ,Elsevier, 5th edition, 2014.
2. William Stallings, “Computer Organization and Architecture designing for Performance”, Pearson Education 8th Edition, 2010
3. John P.Hayes, “Computer Architecture and Organization”, McGraw Hill, 3rd Edition, 2002
4. M. Morris R. Mano “Computer System Architecture” 3rd Edition 2007
5. David A. Patterson “Computer Architecture: A Quantitative Approach”, Morgan Kaufmann; 5th edition 2011

THANK YOU