Illinois State University
School of Information Technology
ITK 325
Computer Organization

Fall, 2003

November 20, 2003

Chapter 18

Parallel Processing

Multiple Processor Organization

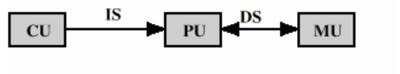
- Single instruction, single data stream - SISD
- Single instruction, multiple data stream - SIMD
- Multiple instruction, single data stream - MISD
- Multiple instruction, multiple data stream- MIMD

Single Instruction

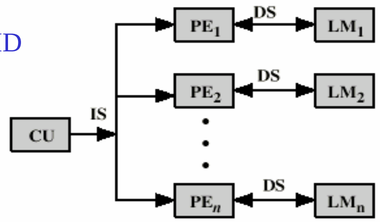
- Single Data Stream – SISD
  - Single processor
  - Single instruction stream
  - Data stored in single memory
  - Uni-processor
- Multiple Data Stream – SIMD
  - Single machine instruction
  - Controls simultaneous execution
  - Number of processing elements
  - Lockstep basis
  - Each processing element has associated data memory
  - Each instruction executed on different set of data by different processors
  - Vector and array processors

Parallel Organizations (1)

- SISD



- SIMD



Multiple Instruction

- Single Data Stream - MISD
  - Sequence of data
  - Transmitted to set of processors
  - Each processor executes different instruction sequence
  - Never been implemented
- Multiple Data Stream- MIMD
  - Set of processors
  - Simultaneously execute different instruction sequences
  - Different sets of data
  - SMPs, clusters and NUMA systems

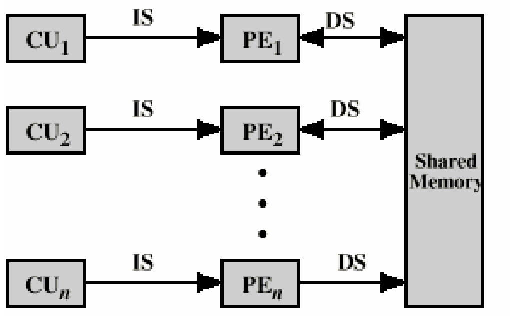
MIMD - Overview

- General purpose processors
- Each can process all instructions necessary
- Further classified by method of processor communication
  - Tightly Coupled
    - Symmetric Multiprocessor (SMP)
    - Nonuniform Memory Access (NUMA)
  - Loosely Coupled
    - Cluster, a collection of independent uniprocessors or SMPs.

7

Parallel Organizations (2)

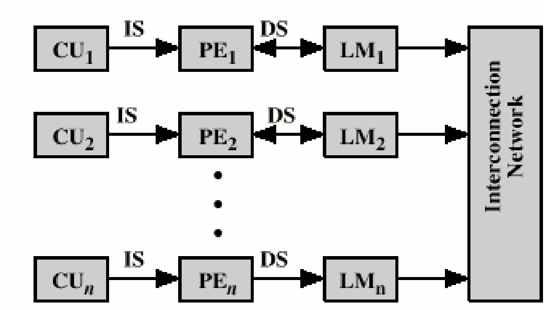
MIMD Shared Memory



8

Parallel Organizations (3)

MIMD Distributed Memory



9

Symmetric Multiprocessors (SMP)

A stand alone computer with the following characteristics

- Two or more similar processors of comparable capacity
- Processors share same memory and I/O
- Processors are connected by a bus or other internal connection
- Memory access time is approximately the same for each processor
- All processors share access to I/O
  Either through same channels or different channels giving paths to same devices
- All processors can perform the same functions (hence symmetric)
- System controlled by integrated operating system
  - Providing interaction between processors
  - Interaction at job, task, file and data element levels

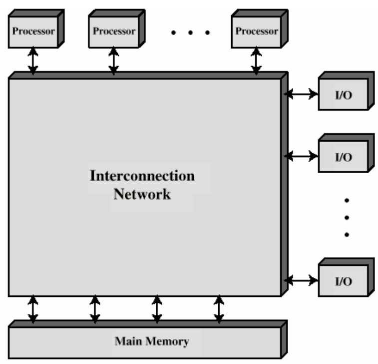
10

SMP Advantages over Uniprocessor

- Performance
  If some work can be done in parallel
- Availability
  Since all processors can perform the same functions, failure of a single processor does not halt the system
- Incremental growth
  User can enhance performance by adding additional processors
- Scaling
  Vendors can offer range of products based on number of processors

11

SMP Block Diagram



12

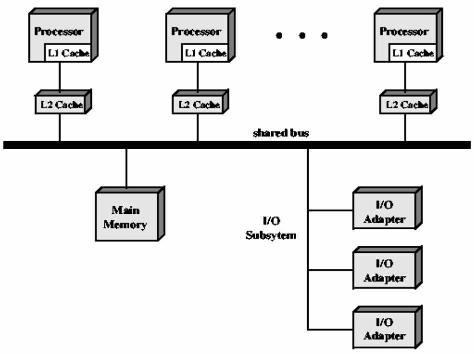
SMP Organization Classification

- Time shared or common bus
- Multiport memory
- Central control unit

SMP Organization
Time Shared Bus

- The simplest form
- Structure and interface similar to single processor system
- Following features provided (as uniprocessor)
  - Addressing - distinguish modules on bus
  - Arbitration - any module can be temporary master
  - Time sharing - if one module has the bus, others must wait and may have to suspend
- Now have multiple processors as well as multiple I/O modules – need O/S support

SMP Organization
Time Shared Bus Diagram

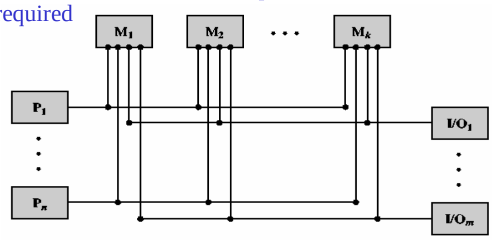


SMP Organization
Time Share Bus Pros and Cons

- Advantages
  - Simplicity: Time-sharing logical remain the same as single-processor system.
  - Flexibility: Easy to add processors to the system
  - Reliability: Bus system is very realiable, the failure of any attached device should not cause failure of the whole system.
- Disadvantage
  - Performance limited by bus cycle time
  - Each processor should have local cache
    - Reduce number of bus accesses
  - Leads to problems with cache coherence
    - Solution – 18.3

SMP Organization
Multiport Memory

- Direct independent access of memory modules by each processor
- Logic required to resolve conflicts
- Little or no modification to processors or modules required



SMP Organization
Multiport Memory Pros and Cons

- More complex
  - Extra logic in memory and I/O modules
- Better performance
  - Each processor has dedicated path to each module
- Can configure portions of memory as private to one or more processors
  - Increased security
- Write through cache policy need to be used since there is no other convenient methods to alert other processors to a memory update.

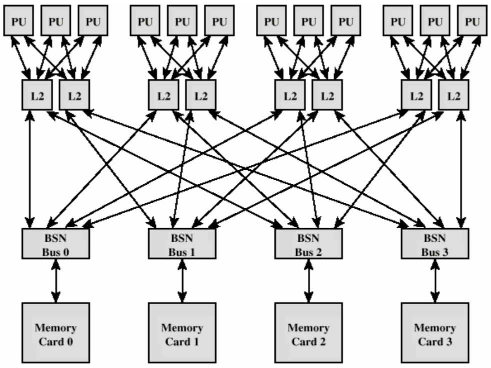
SMP Organization
Central Control Unit

- Funnels separate data streams between independent modules
- Can buffer requests
- Performs arbitration and timing
- Pass status and control
- Perform cache update alerting
- Interfaces to modules remain the same
- Central control unit has potential performance bottleneck.

SMP Operating System Issues

- Simultaneous concurrent processes
  OS routine need to be re-entrant to allow several processors to execute the same OS code simultaneously
- Scheduling
  Scheduling conflicts must be avoided among processors
- Synchronization
  Must synchronize shared resources
- Memory management
  In additional to normal memory management provided by uniprocessor, it needs to exploit the available hardware parallelism.
- Reliability and fault tolerance
  Provide graceful degradation

IBM S/390 Mainframe SMP



Cache Coherence and MESI Protocol

- Problem - multiple copies of same data in different caches
- Can result in an inconsistent view of memory
- Write back policy can lead to inconsistency
- Write through can also give problems unless caches monitor memory traffic

Cache Coherence
Software Solutions

- Compiler and operating system deal with problem
- Overhead transferred to compile time
- Design complexity transferred from hardware to software
- However, software tends to make conservative decisions
  - Inefficient cache utilization
- Analyze code to determine safe periods for caching shared variables

Cache Coherence
Hardware Solution

Generally referred to cache coherence protocols. These approaches are transparent to the programmer and the compiler, reducing the software development burden.

- Hardware schemes are generally divided into two categories:
  - Directory protocols
  - Snoopy protocols

Cache Coherence Hardware Solution
Directory Protocols

- A kind of centralized control.
- Collect and maintain information about copies of data in cache
- Directory stored in main memory
- Requests are checked against directory
- Get exclusive access first then work on appropriate transfers on cache lines
- Creates central bottleneck and communication overhead between cache controller and central controller.
- Effective in large scale systems with multiple buses or complex interconnection schemes

25

Cache Coherence Hardware Solution
Snoopy Protocols

- Distribute cache coherence responsibility among cache controllers
- Cache recognizes that a line is shared
- Updates on shared cache line must be announced to all other caches using broadcast.
- Each cache controller is able to “snoop” on the network
- Suited to bus based multiprocessor
- Increases bus traffic
- Two basic approaches:
  - Write invalidate
  - Write update (or write broadcast)

26

Snoopy Protocols

- Write Invalidate
  - Multiple readers, one writer
  - When a write is required, all other caches of the line are invalidated
  - Writing processor then has exclusive (cheap) access until line required by another processor
  - Used in Pentium II and PowerPC systems
  - State of every line is marked as Modified, Exclusive, Shared or Invalid (MESI)
- Write Update
  - Multiple readers and writers
  - Updated word is distributed to all other processors
- Some systems use an adaptive mixture of both solutions

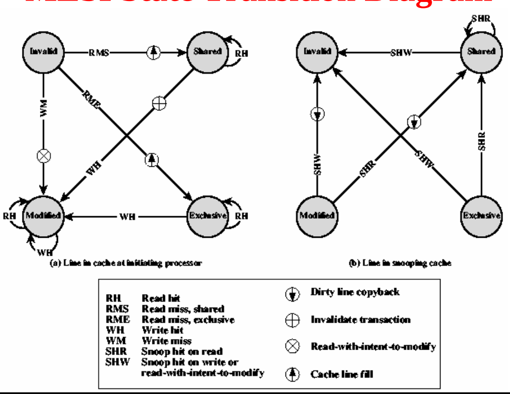
27

The MESI Protocol

- Data cache line includes two status bits per tag, they represent four states.
  - Modified: The line in the cache has been modified and is available only in this cache
  - Exclusive: The line in the cache is the same as that in main memory and is not present in any other cache.
  - Shared: The line in the cache is the same as that in main memory and may be present in another cache.
  - Invalid: The line in the cache does not contain valid data
- Read Miss, Read Hit, Write Miss, Write Hit will change the state of the cache based on the current cache state.

28

MESI State Transition Diagram



29