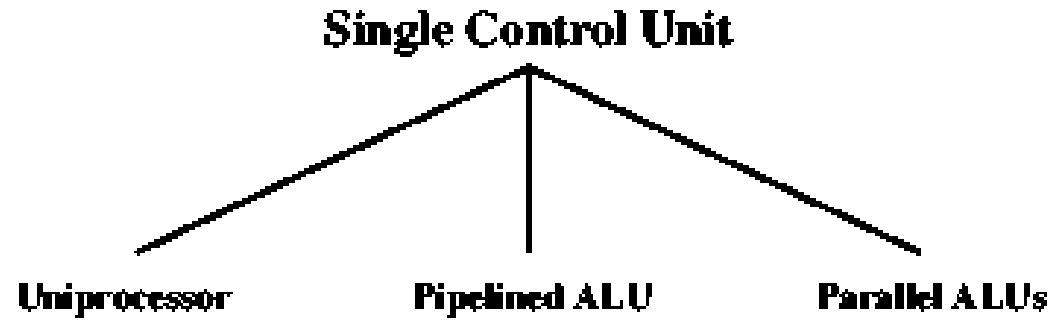
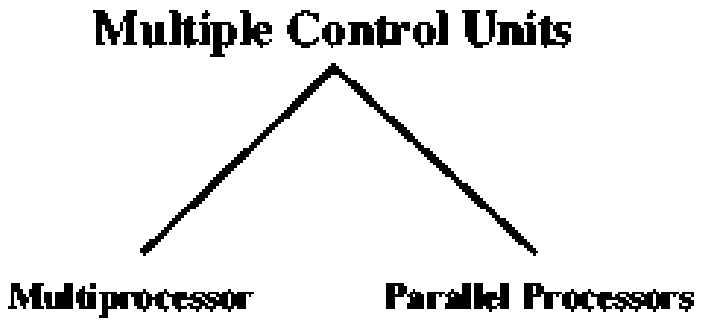
Chapter 17

Parallel Processing

Computer Organizations





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

Single Instruction, Multiple Data Stream - SIMD

- Single machine instruction
  - Each instruction executed on different set of data by different processors
- Number of processing elements
  - Machine controls simultaneous execution
    - Lockstep basis
  - Each processing element has associated data memory
- Application: Vector and array processing

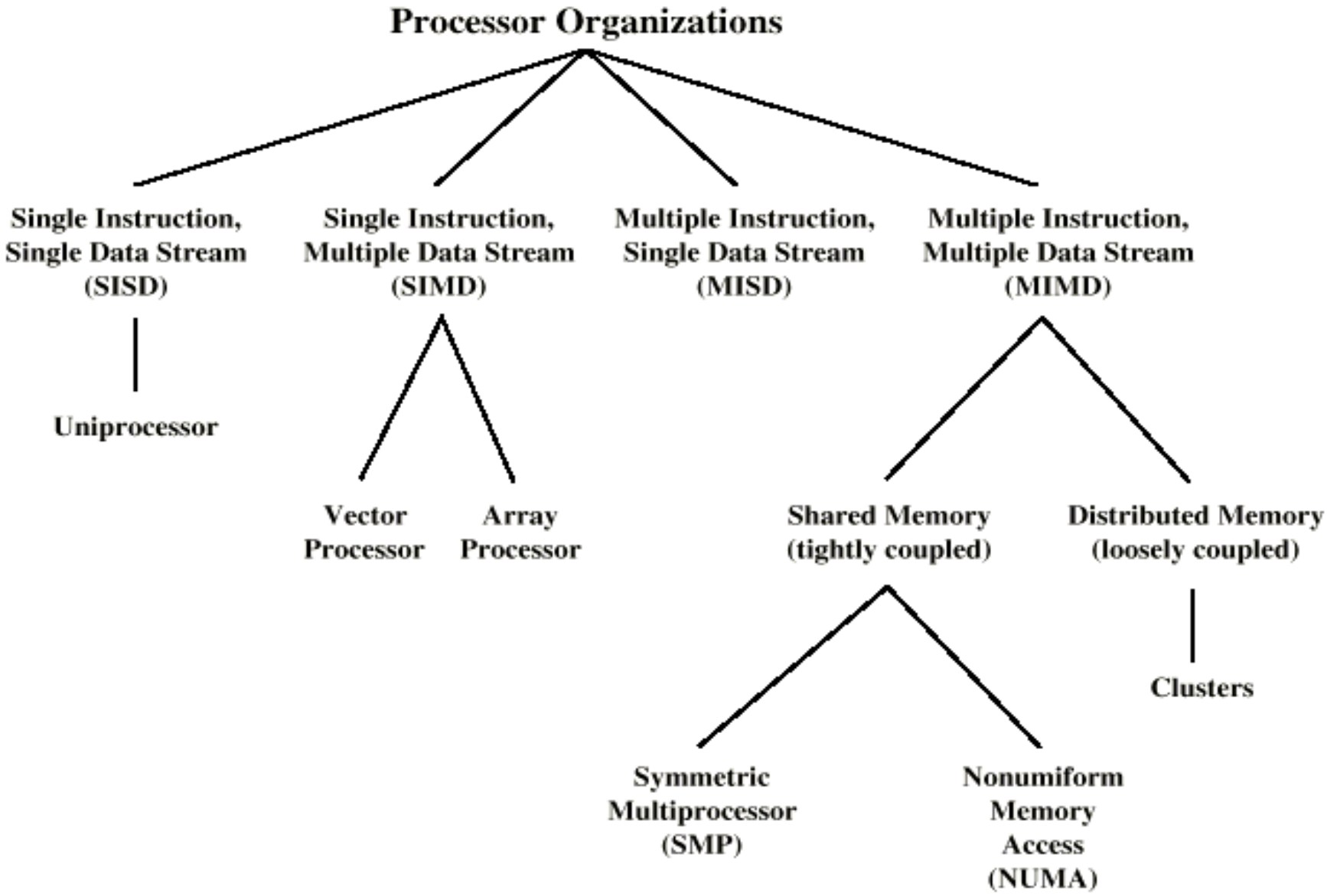
Multiple Instruction, Single Data Stream - MISD

- Sequence of data
- Transmitted to set of processors
- Each processor executes different instruction sequence
- Not clear if it has ever been implemented

Multiple Instruction, Multiple Data Stream- MIMD

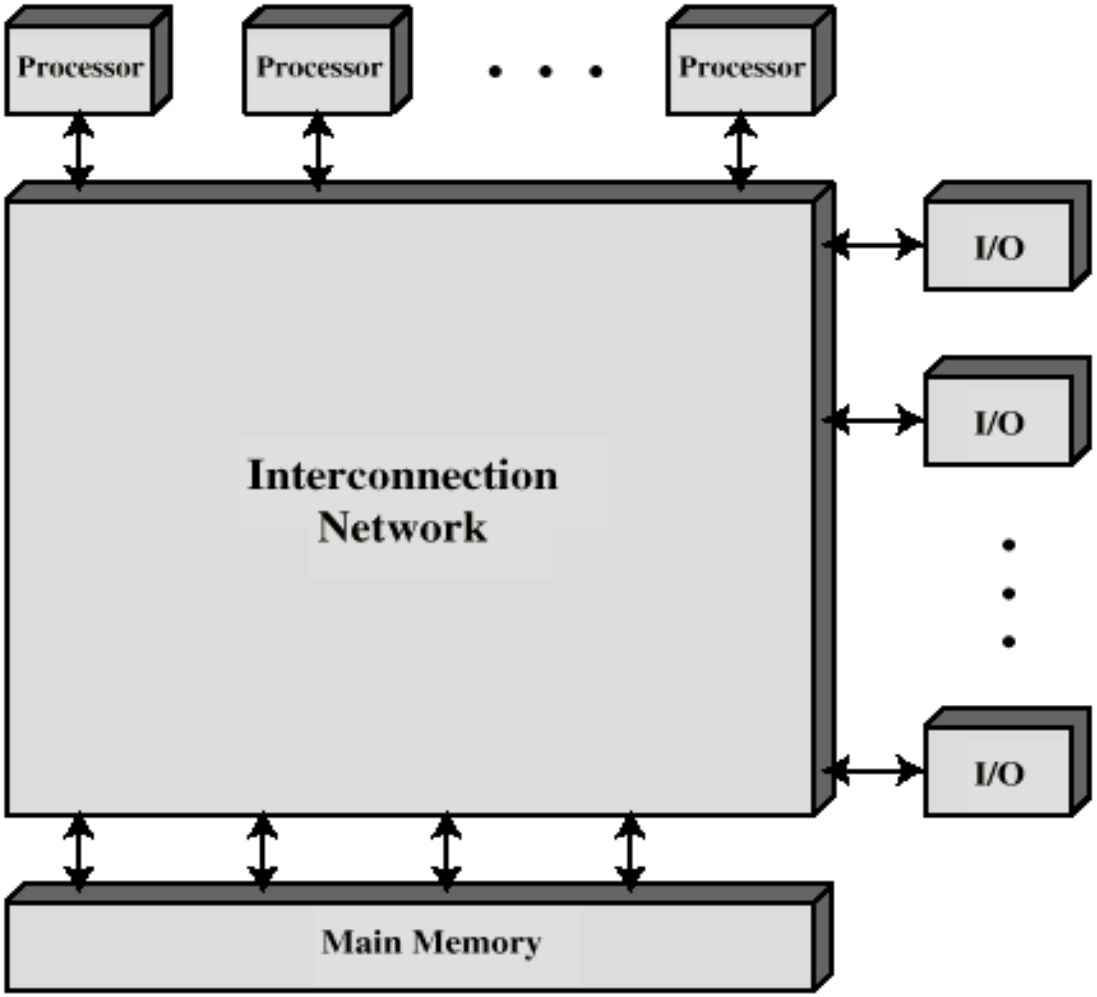
- Set of processors
- Simultaneously executes different instruction sequences
- Different sets of data
- Examples: SMPs, NUMA systems, and Clusters

Taxonomy of Parallel Processor Architectures

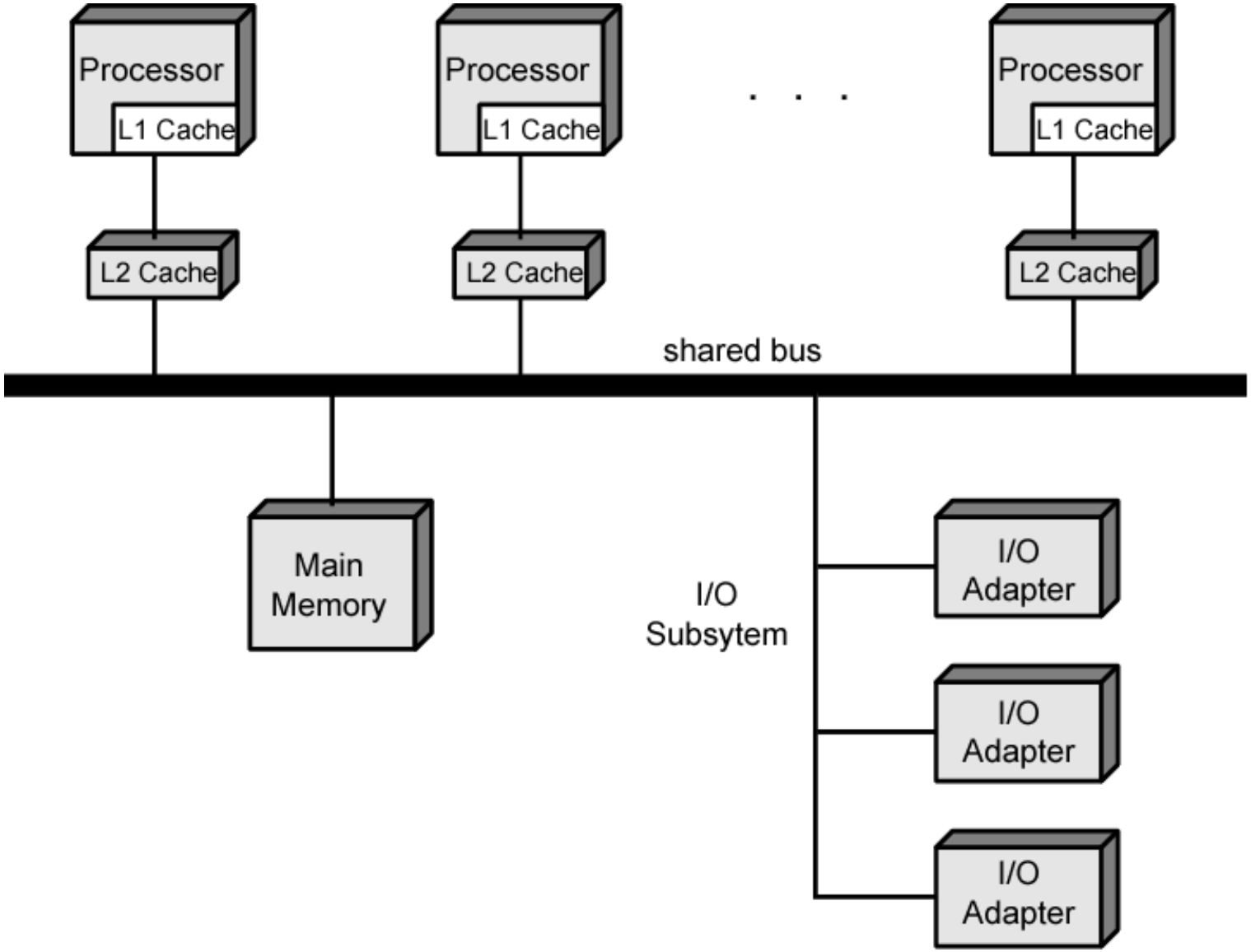


Block Diagram of Tightly Coupled Multiprocessor

- Processors share memory
- Communicate via that shared memory



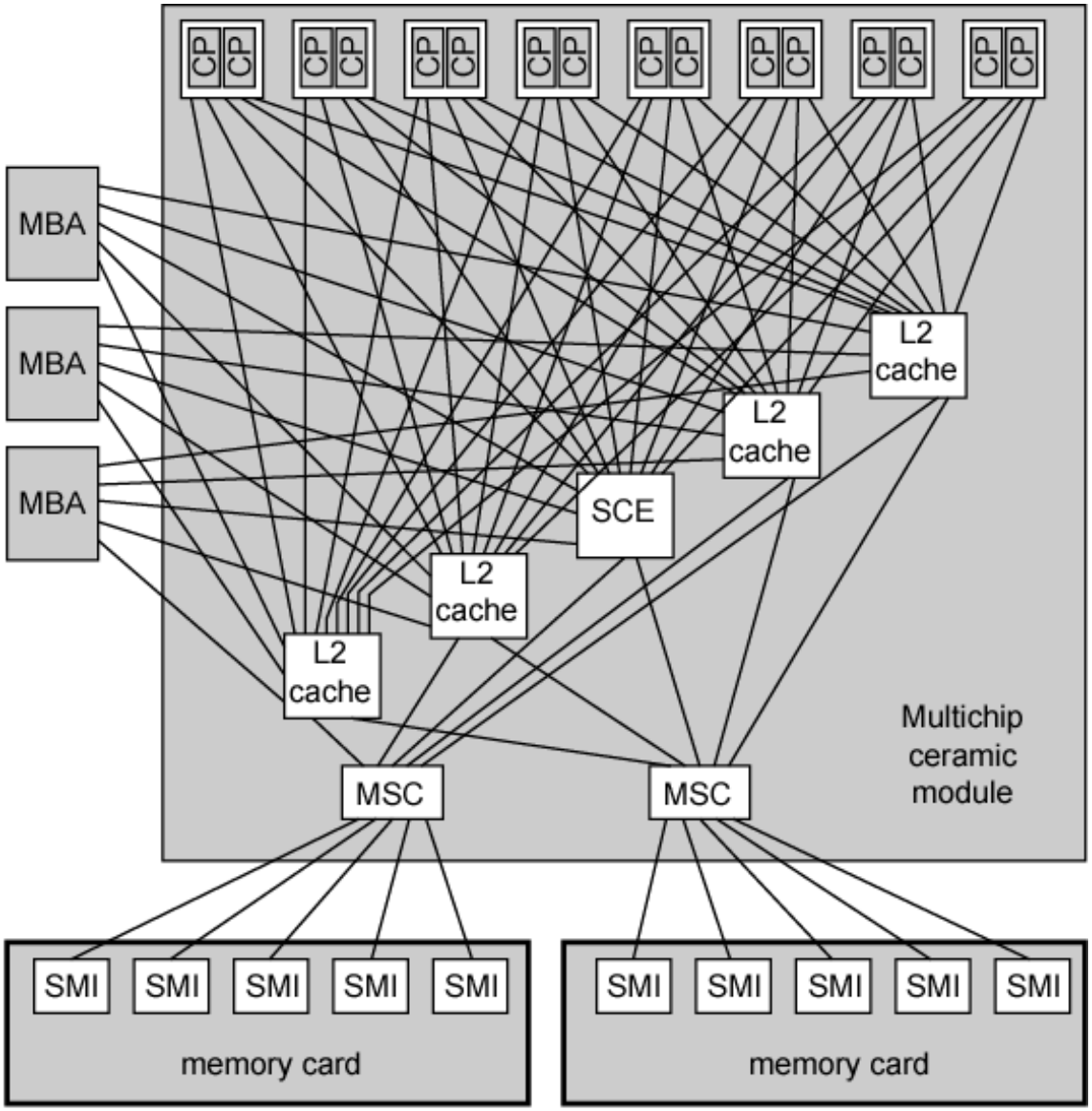
Symmetric Multiprocessor Organization



Symmetric Multiprocessors

- 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

IBM z990 Multiprocessor Structure



CP = central processor
MBA = memory bus adapter
MSC = main store control
SCE = system control element
SMI = synchronous memory interface

SMP Advantages

- 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

Cache Coherence Problems

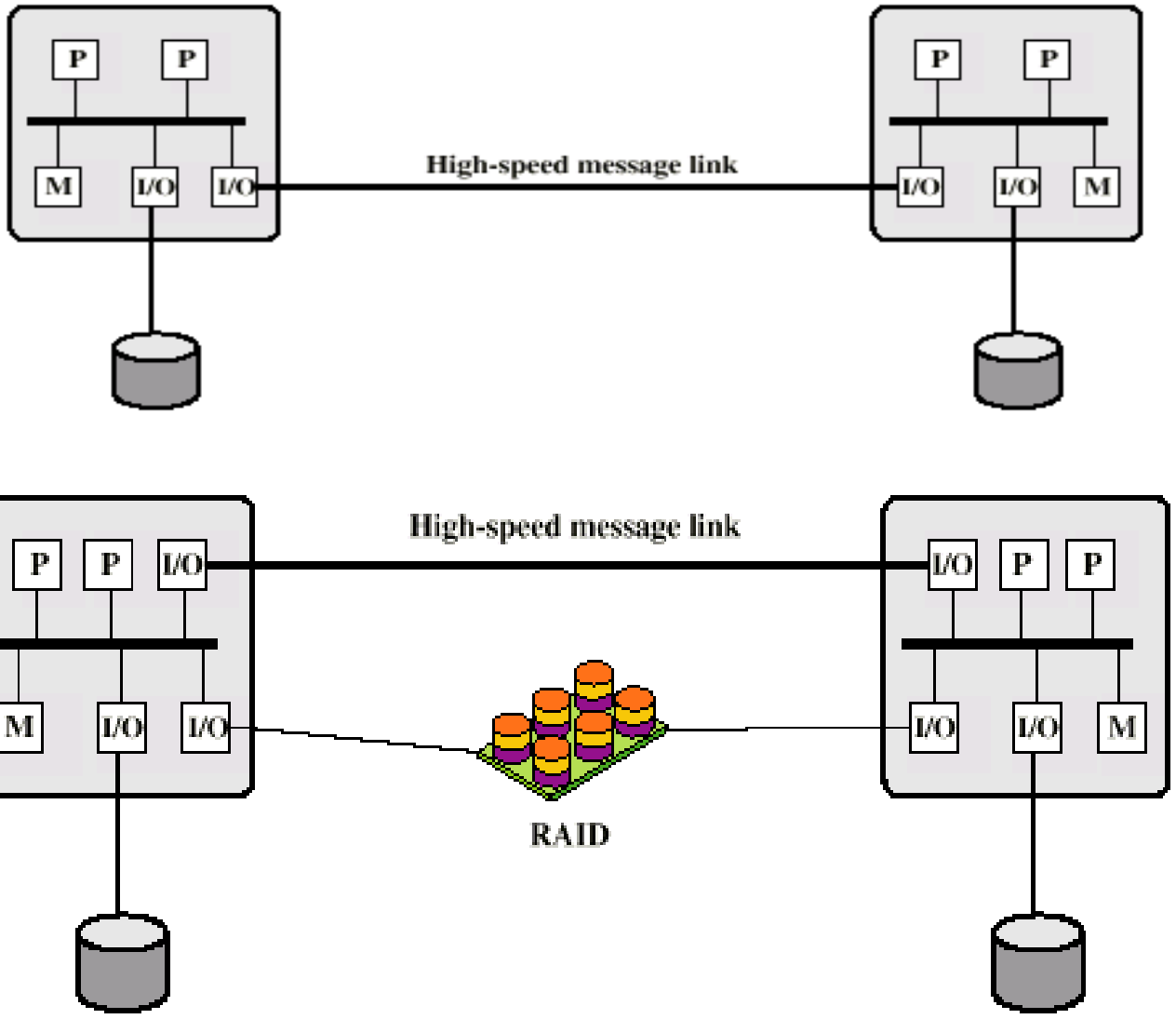
Popular solution - Snoopy Protocol

- Distribute cache coherence responsibility among cache controllers
- Cache recognizes that a line is shared
- Updates announced to other caches

Loosely Coupled - Clusters

- Collection of independent whole uniprocessors or SMPs
  - Usually called nodes
- Interconnected to form a cluster
- Working together as unified resource
  - Illusion of being one machine
- Communication via fixed path or network connections

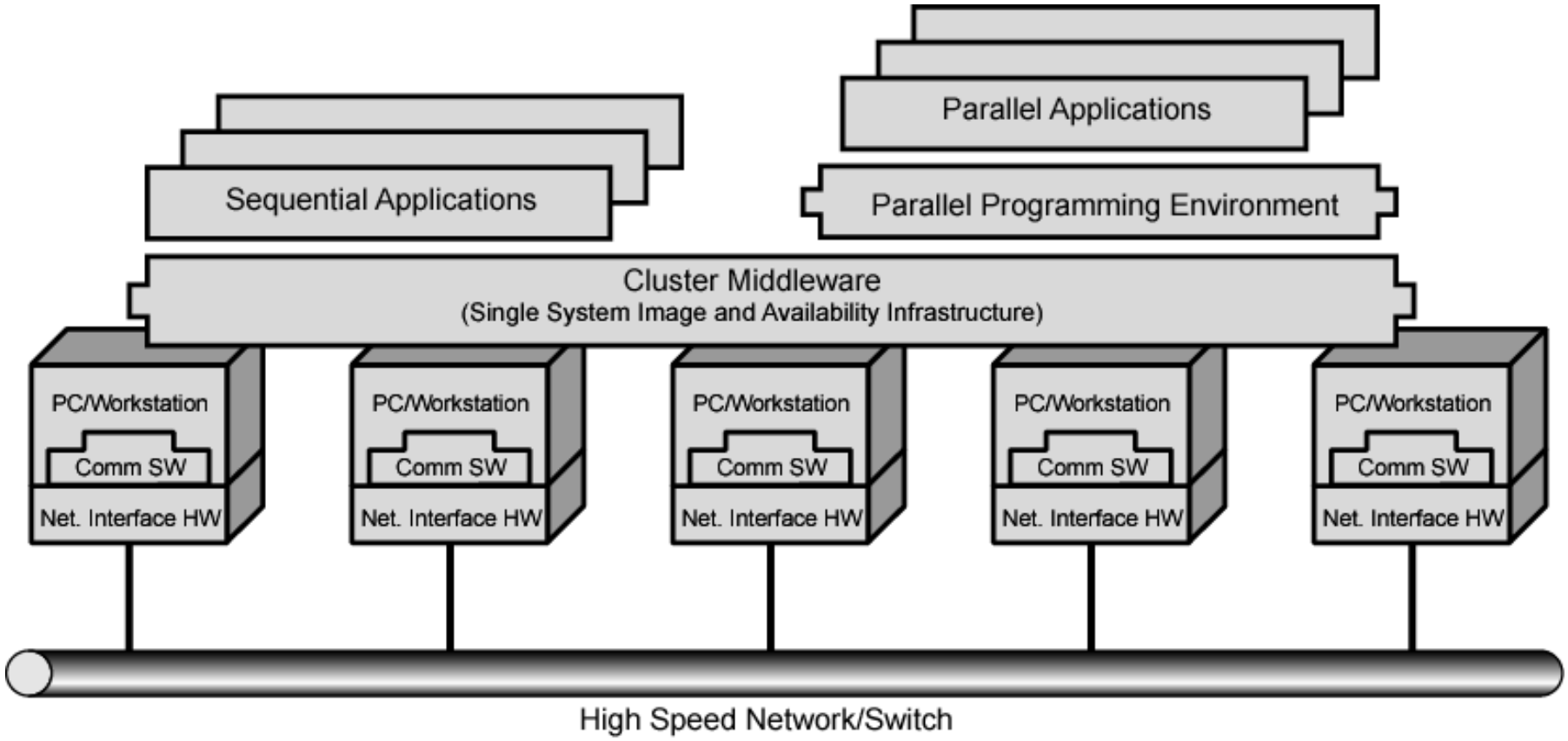
Cluster Configurations



Cluster Benefits

- Absolute scalability
- Incremental scalability
- High availability
- Superior price/performance

Cluster Computer Architecture



Cluster v. SMP

- Both provide multiprocessor support to high demand applications.
- Both available commercially
- SMP:
  - Easier to manage and control
  - Closer to single processor systems
    - Scheduling is main difference
    - Less physical space
    - Lower power consumption
- Clustering:
  - Superior incremental & absolute scalability
  - Less cost
  - Superior availability
    - Redundancy

Nonuniform Memory Access (NUMA) (Tightly coupled)

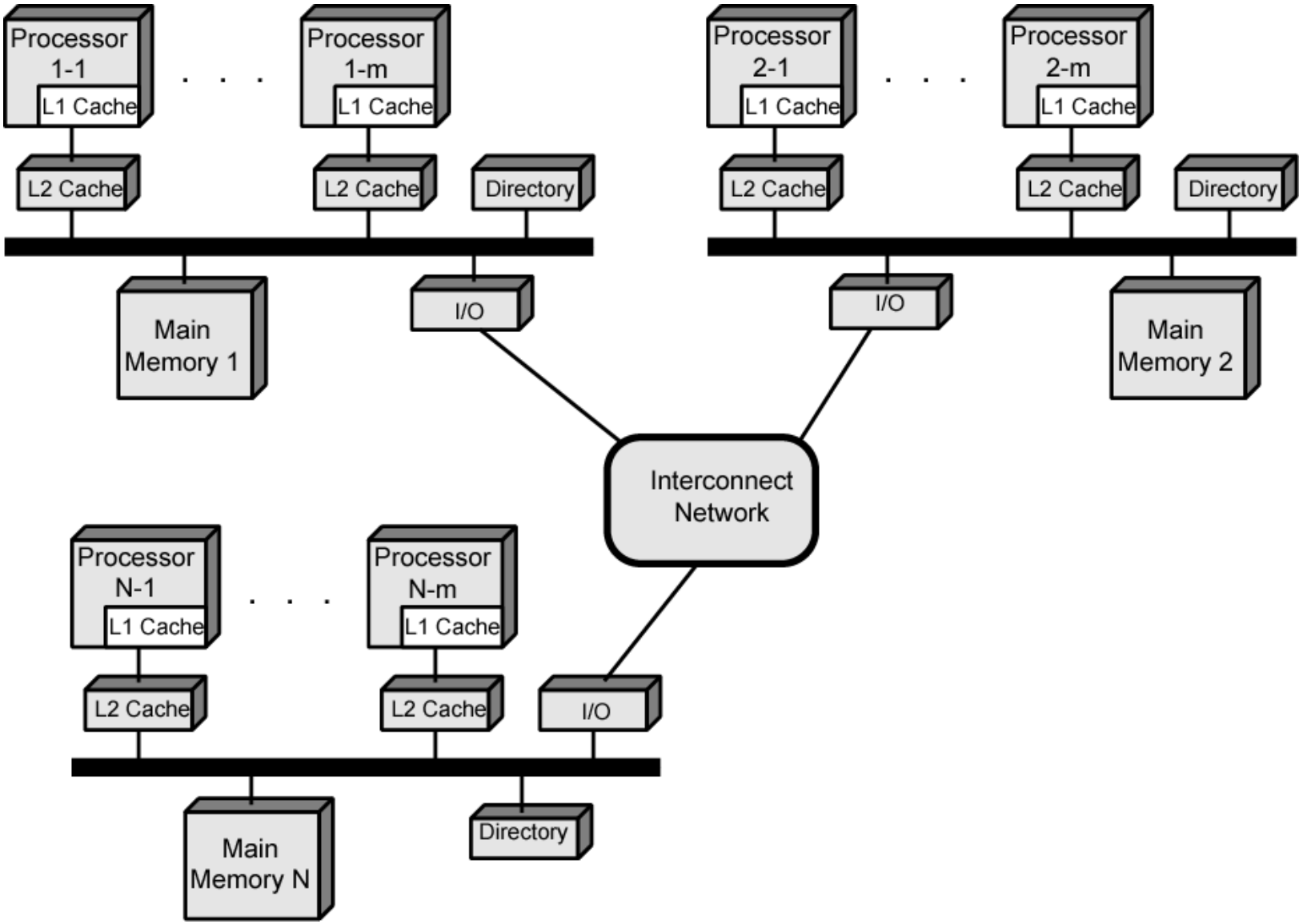
- Alternative to SMP & Clusters
- Nonuniform memory access
  - All processors have access to all parts of memory
    - Using load & store
  - Access time of processor differs depending on region of memory
    - Different processors access different regions of memory at different speeds
- Cache coherent NUMA ?
  - Cache coherence is maintained among the caches of the various processors
  - Significantly different from SMP and Clusters

Motivation

- SMP has practical limit to number of processors
  - Bus traffic limits to between 16 and 64 processors
- In clusters each node has own memory
  - Apps do not see large global memory
  - Coherence maintained by software not hardware
- NUMA retains SMP flavour while giving large scale multiprocessing

- Objective is to maintain transparent system wide memory while permitting multiprocessor nodes, each with own bus or internal interconnection system

CC-NUMA Organization



NUMA Pros & Cons

- Possibly effective performance at higher levels of parallelism than one SMP
- Not very supportive of software changes
- Performance can breakdown if too much access to remote memory
  - —Can be avoided by:
    - L1 & L2 cache design reducing all memory access
      - + Need good temporal locality of software
- Not transparent
  - —Page allocation, process allocation and load balancing changes can be difficult
- Availability?

Multithreading

- Instruction stream divided into smaller streams (threads)
- Executed in parallel
- There are a wide variety of multithreading designs

Definitions of Threads and Processes

- Threads in multithreaded processors may or may not be same as software threads
- Process:
  - An instance of program running on computer
- Thread: dispatchable unit of work within process
  - Includes processor context (which includes the program counter and stack pointer) and data area for stack
  - Thread executes sequentially
  - Interruptible: processor can turn to another thread
- Thread switch
  - Switching processor between threads within same process
  - Typically less costly than process switch

Implicit and Explicit Multithreading

- All commercial processors and most experimental ones use explicit multithreading
  - Concurrently execute instructions from different explicit threads
  - Interleave instructions from different threads on shared pipelines or parallel execution on parallel pipelines
- Implicit multithreading is concurrent execution of multiple threads extracted from single sequential program
  - Implicit threads defined statically by compiler or dynamically by hardware

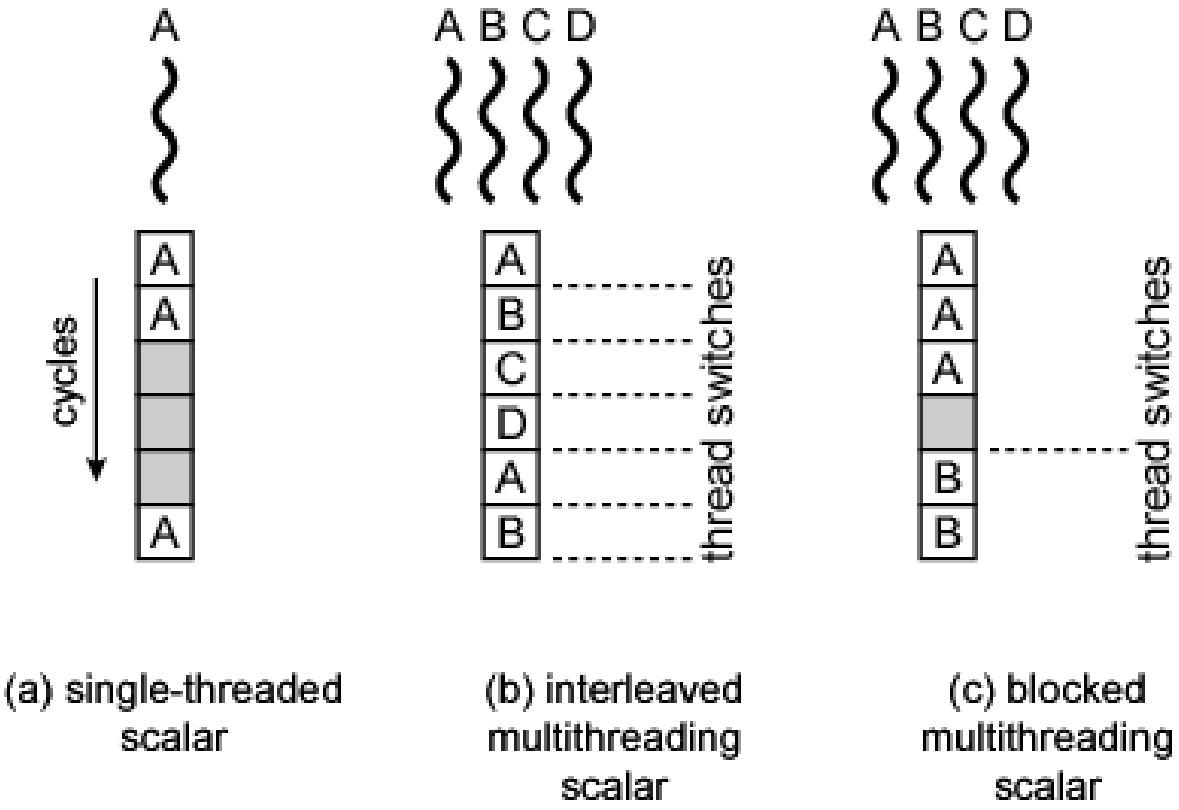
Approaches to Explicit Multithreading

- Interleaved
  - Fine-grained
  - Processor deals with two or more thread contexts at a time
  - Switching thread at each clock cycle
  - If thread is blocked it is skipped
- Blocked
  - Coarse-grained
  - Thread executed until event causes delay
  - E.g. cache miss
  - Effective on in-order processor
  - Avoids pipeline stall
- Simultaneous (SMT)
  - Instructions simultaneously issued from multiple threads to execution units of superscalar processor
- Chip multiprocessing
  - Processor is replicated on a single chip
  - Each processor handles separate threads

Scalar Processor Approaches

- Single-threaded scalar
  - Simple pipeline
  - No multithreading
- Interleaved multithreaded scalar
  - Easiest multithreading to implement
  - Switch threads at each clock cycle
  - Pipeline stages kept close to fully occupied
  - Hardware needs to switch thread context between cycles
- Blocked multithreaded scalar
  - Thread executed until latency event occurs
  - Would stop pipeline
  - Processor switches to another thread

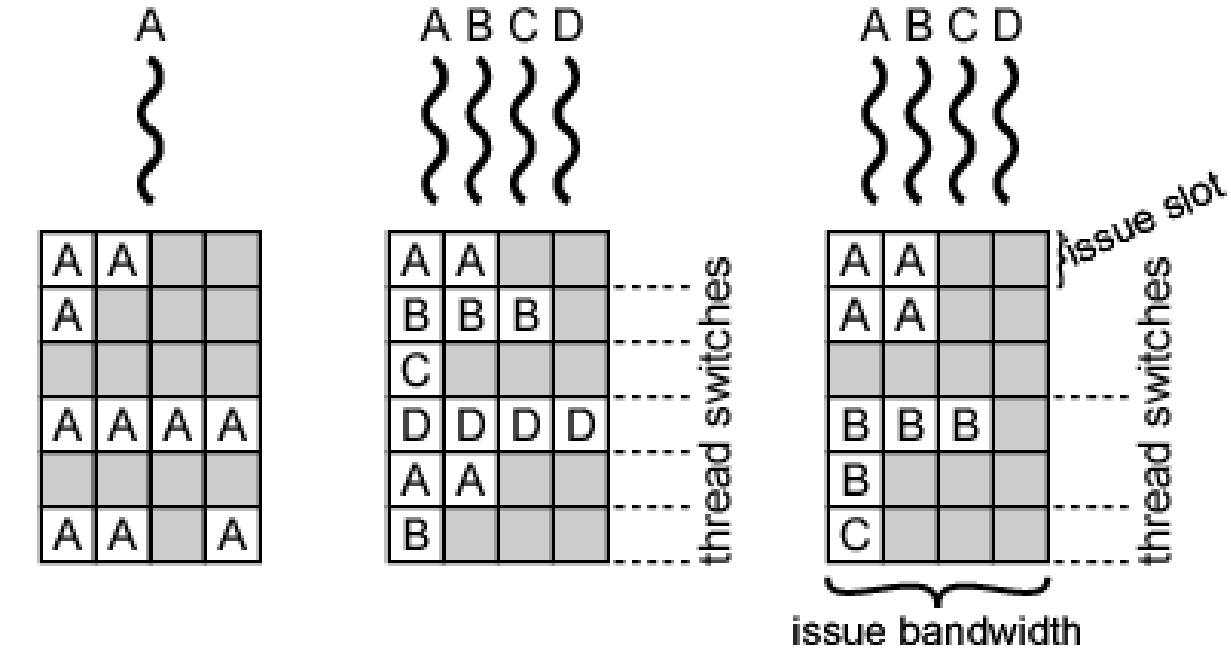
Scalar Diagrams



Multiple Instruction Issue Processors

- Superscalar
  - No multithreading
- Interleaved multithreading superscalar:
  - Each cycle, as many instructions as possible issued from single thread
  - Delays due to thread switches eliminated
  - Number of instructions issued in cycle limited by dependencies
- Blocked multithreaded superscalar
  - Instructions from one thread
  - Blocked multithreading used

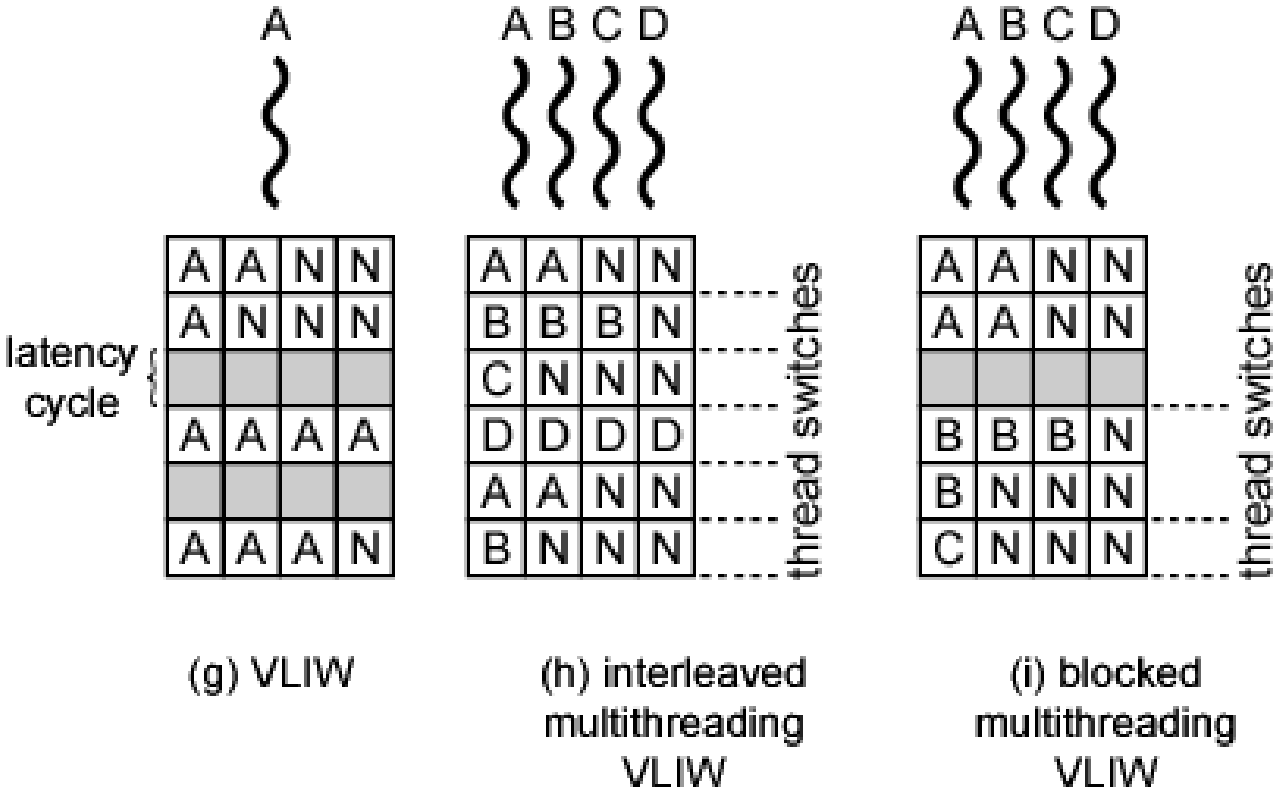
Multiple Instruction Issue Diagram



Multiple Instruction Issue Processors

- Very long instruction word (VLIW)
  - E.g. IA-64
  - Multiple instructions in single word
  - Typically constructed by compiler
  - Operations may be executed in parallel in same word
  - May pad with no-ops
- Interleaved multithreading VLIW
  - Similar efficiencies to interleaved multithreading on superscalar architecture
- Blocked multithreaded VLIW
  - Similar efficiencies to blocked multithreading on superscalar architecture

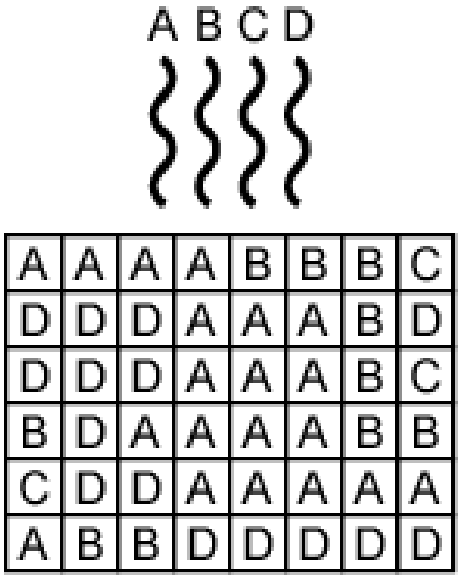
Multiple Instruction Issue Diagram



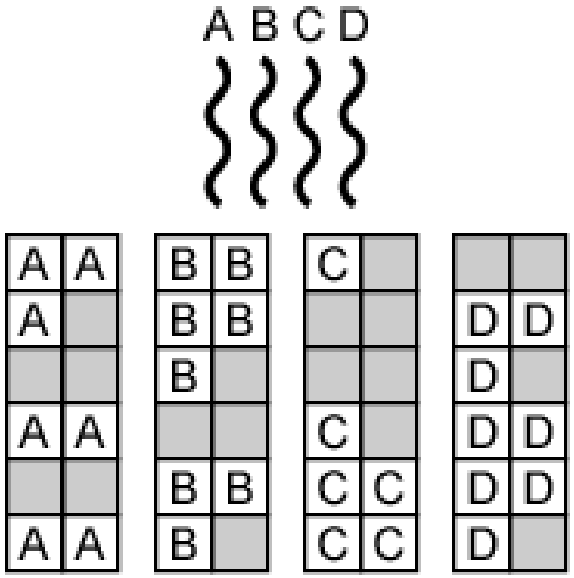
Parallel, Simultaneous Execution of Multiple Threads

- Simultaneous multithreading
  - Issue multiple instructions at a time
  - One thread may fill all horizontal slots
  - Instructions from two or more threads may be issued
  - With enough threads, can issue maximum number of instructions on each cycle
- Chip multiprocessor
  - Multiple processors
  - Each has two-issue superscalar processor
  - Each processor is assigned thread
    - Can issue up to two instructions per cycle per thread

Parallel Diagram



(j) simultaneous multithreading (SMT)



(k) chip multiprocessor

Examples

- Some Pentium 4 (single processor)
  - Intel calls it hyperthreading
  - SMT with support for two threads
  - Single multithreaded processor, logically two processors
- IBM Power5
  - High-end PowerPC
  - Combines chip multiprocessing with SMT
  - Chip has two separate processors
  - Each supporting two threads concurrently using SMT