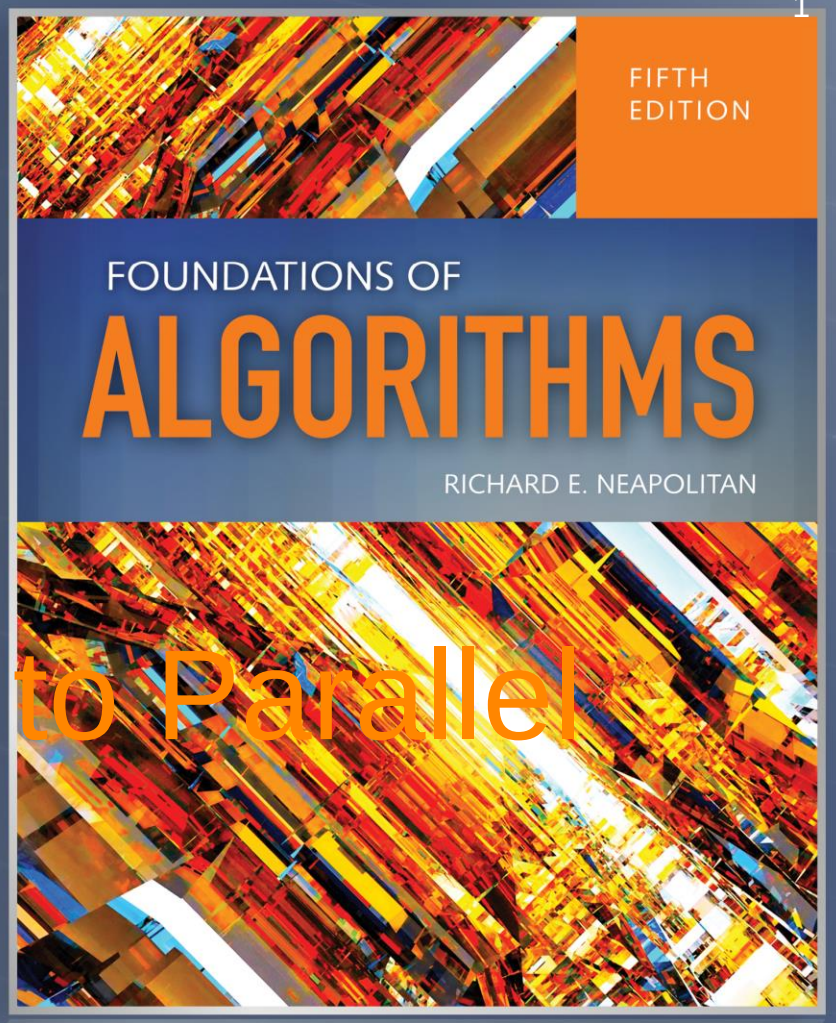


Introduction to Parallel Algorithms

Chapter 12

Objectives

- Explain the concept of parallel algorithms
- Explain the construction of parallel computers
- Define different parallel architectures
- Define shared memory and message passing communication models
- Define static and dynamic interconnection networks
- Define the PRAM model
- Define 4 models for concurrent memory access
- Define the CREW and CRCW PRAM Models

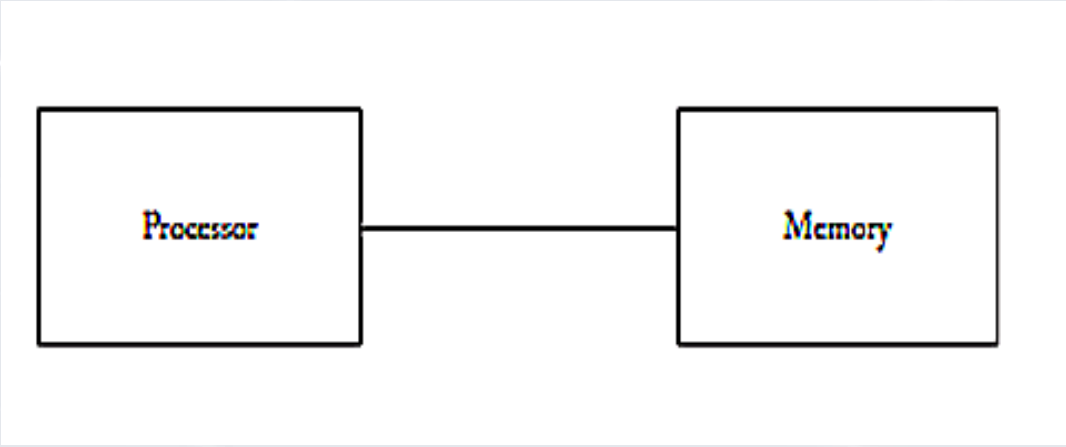
Concept

- Building a fence and need to dig 10 fence-post holes
- You can individually dig 10 holes in sequence
- Get 10 friends to dig them simultaneously (in parallel)
- By digging in parallel, job is done much faster

Traditional Sequential Computer

- One processor executing instructions in sequence
- vonNeumann Model
- Single Instruction Stream, Single Data Stream (SISD)
- Serial Computer

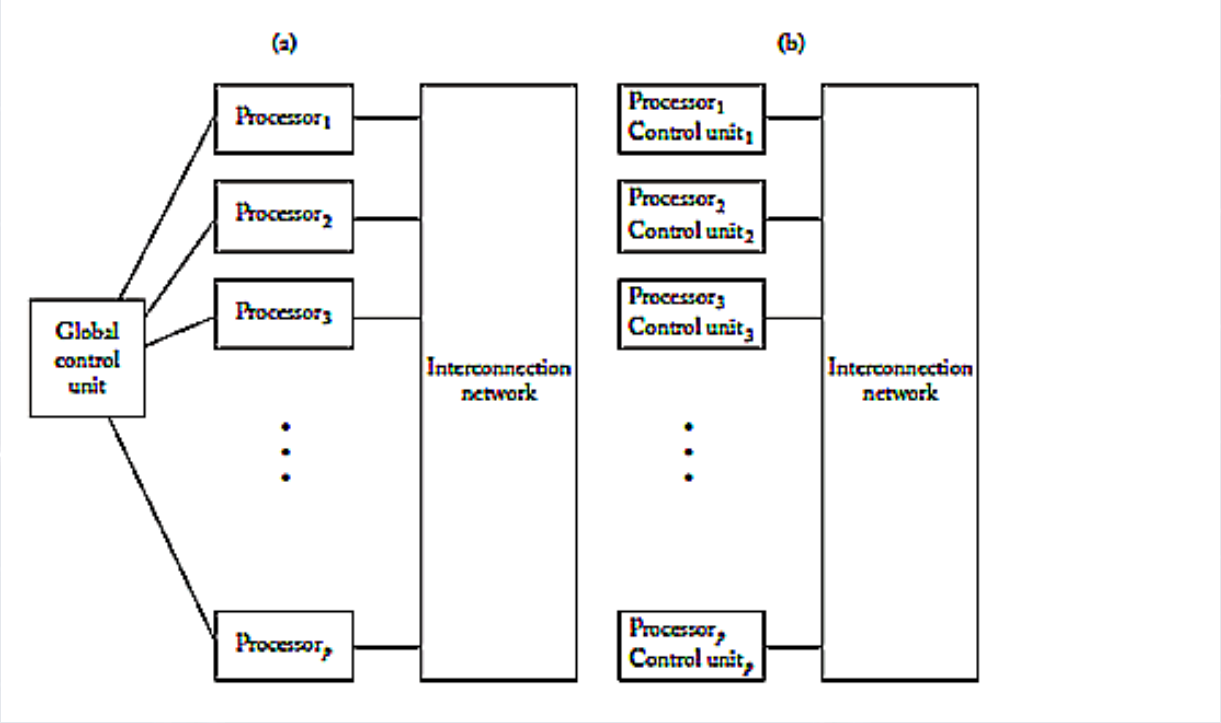
Figure 12.1



Construction of parallel computers can vary

- Control mechanism
- Address-space organization
- Interconnection network

Figure 12.3



Single Instruction Stream, Multiple Data Stream (SIMD)

- Each processor operates under control of centralized control unit
- Same instruction is executed synchronously by all processing units under control of central control unit
- Not all processors must execute an instruction in each cycle
- Suited for programs in which the same set of instructions is executed on different elements of a data set

Single Instruction Stream, Multiple Data Stream (SIMD)

- Data parallel algorithms
- Best suited to parallel algorithms requiring synchronization

Disadvantage of SIMD

- Processors cannot execute different instructions in the same cycle
- If x = y then execute instructions A else execute instructions B
- Processors finding x != y must remain idle while those finding x = y execute instruction A
- Processors finding x = y must remain idle while those finding x!=y execute instruction B

Multiple Instruction Stream, Multiple Data Stream Architecture - MIMD

- Each processor operates under the control of its own control unit
- Store both OS and Program at each processor

Shared-Address Space Architecture

- Hardware provides read/write access by all processors to shared address space
- Communication is via the modification of data
- Additional mechanisms must be in place to ensure data integrity

Message Passing Architecture

- Each processor has its own private memory accessible only to that processor
- Processors communicate by passing messages

Interconnection Networks

- Static
- Dynamic

Static Interconnection Networks

- Direct links between processors
- Completely connected network costly
- Star Connected
  - One processor central
  - Processor has a link only to the central processor
  - Message sent to other processors by routing through central

Static Interconnection Networks

- Bounded degree network
  - Degree d – each processor linked to at most d other processors
  - Messages routed
- (d+1) dimensional hypercube
  - Each processor in a d-dim hypercube is linked to one processor in another d-dim hypercube
  - Given processor linked to the processor occupying corresponding position in 2^{nd} hypercube

Figure 12.6

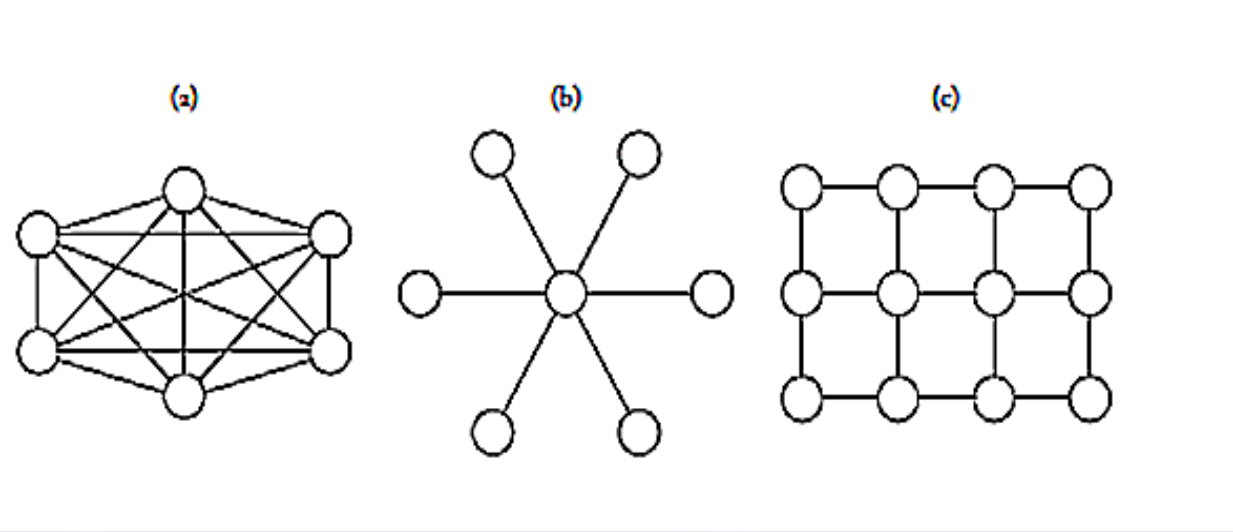
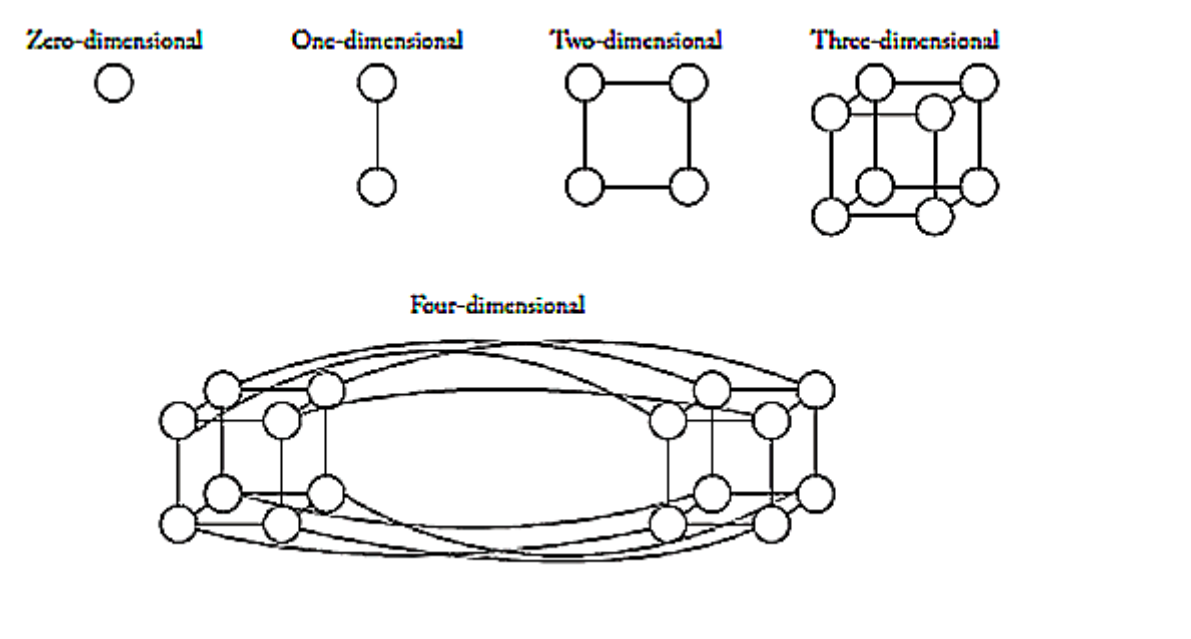


Figure 12.7



Dynamic Interconnection Networks

- Processors connected to memory through set of switching elements
- Crossbar switching network – p processors connected to m memory banks
- Connection of one processor to a given memory bank does not block connection of another processor to another memory bank
- Number of banks at least as large as number of processors so at a given time, each processor can access at least one memory bank

Dynamic Interconnection Networks

- Each processor allowed to access every word in memory but cannot send a direct message to ay of the other processors

Pram Model

- Parallel Random Access Machine
- Theoretical model for parallel machines
- p processors having uniform access to a large shared memory
- Processors share a common clock but may execute different instructions in the same cycle
- Synchronous MIMD computer
- Multiple processors may try to simultaneously read/write the same memory location

4 Models for Concurrent Memory Accesses

- Exclusive-read, exclusive-write (EREW)
  - Concurrent reads/writes not allowed
  - Weakest PRAM model – minimum concurrency
- Exclusive-read, concurrent-write (ERCW)
  - Simultaneous write operations allowed but not simultaneous read operations
- Concurrent-read, exclusive write (CREW)
  - Simultaneous read operations allowed but not simultaneous write operations

4 Models for Concurrent Memory Accesses

- Concurrent-read, concurrent-write (CRCW)
  - Both simultaneous read/write operations allowed

CREW PRAM Model Conventions/Assumptions

- One version of algorithm written and executed simultaneously by all processors
- Each processor needs its own index (P1, P2, . . .)
- p = index of this processor
  - Returns the index of a processor
- Variable can either be shared or in a processor's private memory

CREW Pram Model Conventions/ Assumptions

- All algorithms data-parallel operating on different elements of a data set
  - Data set stored in shared memory
  - Read from shared to local
  - Write from local to shared
  - Never compare two elements in shared memory – always read to private and compare in private
  - Only direct comparisons are to variables representing size of input
- All processors read during a given step at the same time

CREW Pram Model Conventions/ Assumptions

- All processors that write during a given step write at the same time
- Assume as many processors needed are always available

CRCW PRAM Model

- Concurrent writes must be resolved when two processors try to write at the same memory location in the same step
- Most frequently used protocols for resolving conflict:
  - Common: concurrent writes allowed only if all the processors attempting to write same values
  - Arbitrary: Picks an arbitrary processor as the one allowed to write to the memory location

CRCW PRAM Model

- Priority: all processors are organized in a pre-defined priority list and only the one with the highest priority is allowed to write
- Sum: writes the sum of the quantities being written by the processors