# Computing Components

The base components of computer design and the way they are structured.

CHAPTER 5

# Chapter Goals

- List the components and their function in a von Neumann machine
- Describe the fetch-decode-execute cycle of the von Neumann machine
- Describe how computer memory is organized and accessed
- Name and describe different auxiliary storage devices
- Define three alternative parallel computer configurations

# Stored program computer

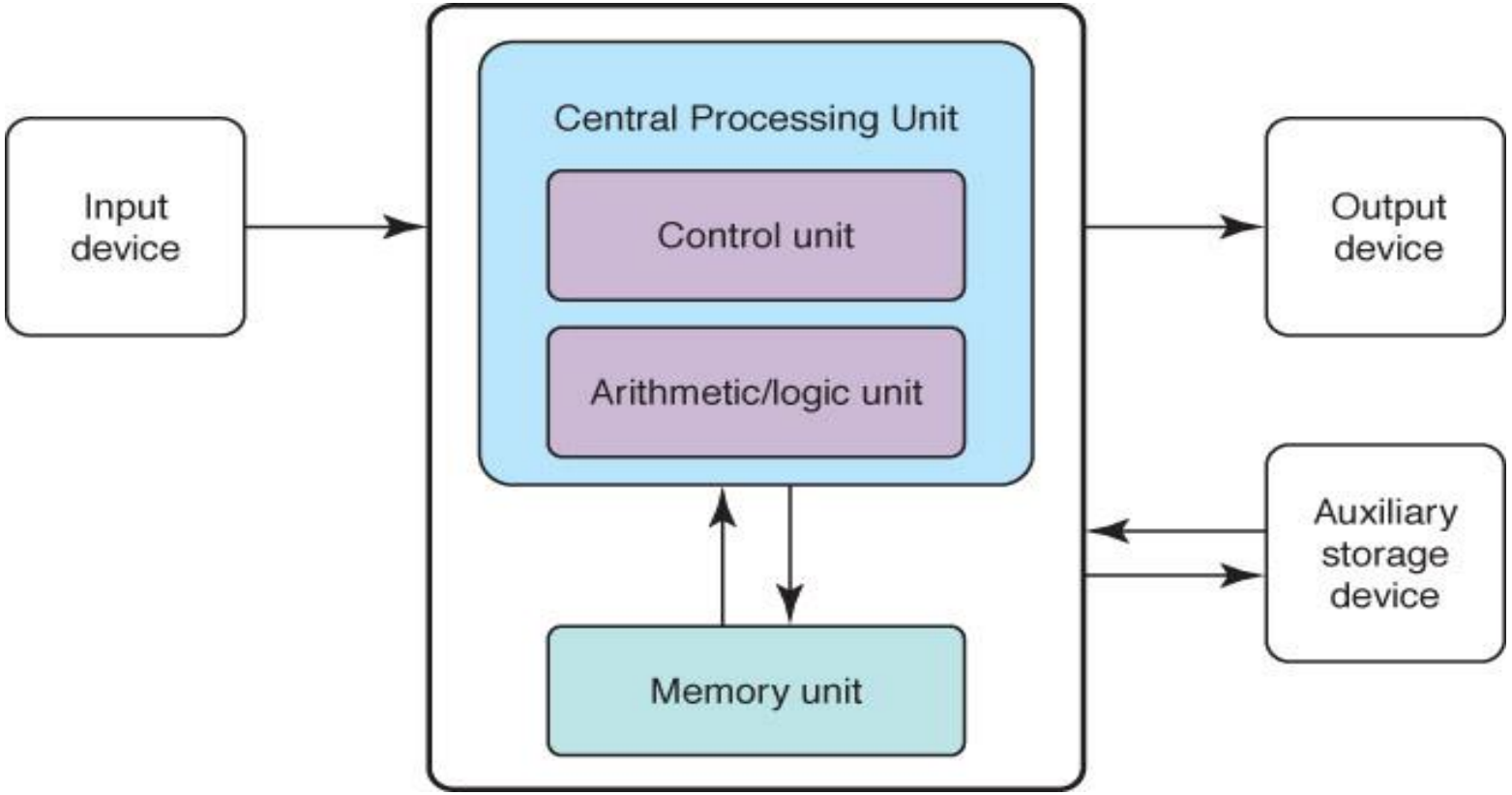
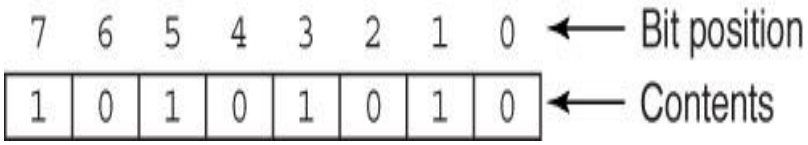



**Figure 5.1** **The von Neumann architecture**

# Memory

## Memory

A collection of cells, each with a unique physical address; both addresses and contents are in binary




| Address | Contents |
|---|---|
| 00000000 | 11100011 |
| 00000001 | 10101001 |
| : | : |
| . | . |
| 11111100 | 00000000 |
| 11111101 | 11111111 |
| 11111110 | 10101010 |
| 11111111 | 00110011 |

# Arithmetic/Logic Unit

- Performs basic arithmetic operations such as adding
- Performs logical operations such as AND, OR, and NOT
- Most modern ALUs have a small amount of special storage units called **registers**

# Input/Output Units

## Input Unit

A device through which data and programs from the outside world are entered into the computer;

## Output unit

A device through which results stored in the computer memory are made available to the outside world

# Control Unit

**Control unit**

The organizing force in the computer

**Instruction register** (IR)

Contains the instruction that is being executed

**Program counter** (PC)

Contains the address of the next instruction to be executed

**Central Processing Unit** (CPU)

ALU and the control unit.

# Flow of Information

## Bus

A set of wires that connect all major sections

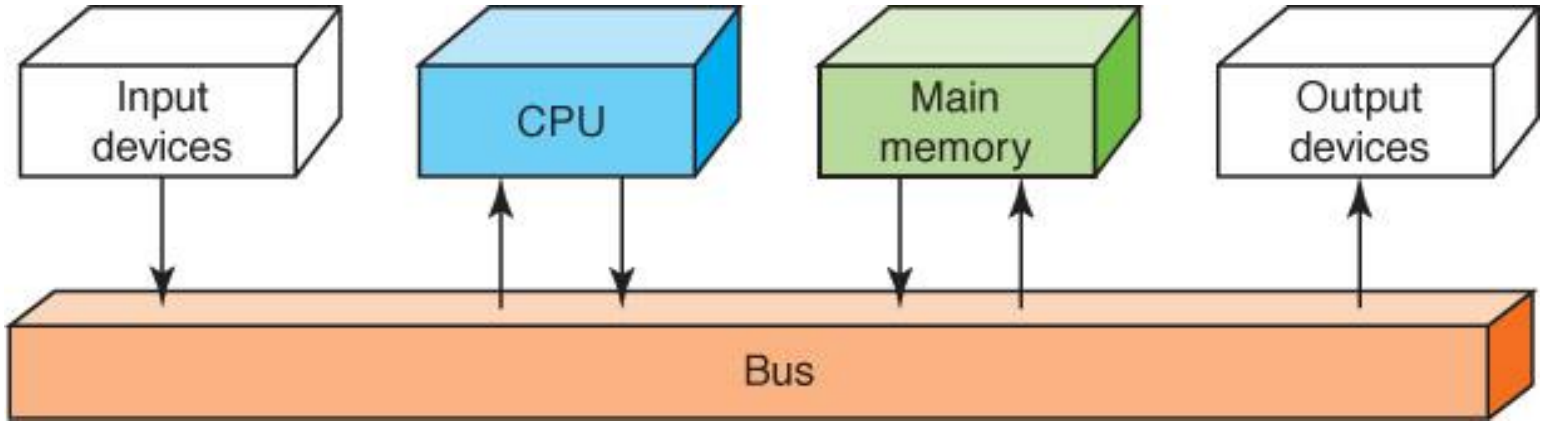



**Figure 5.2** **Data flow through a von Neumann architecture**

# The Fetch–Execute cycle

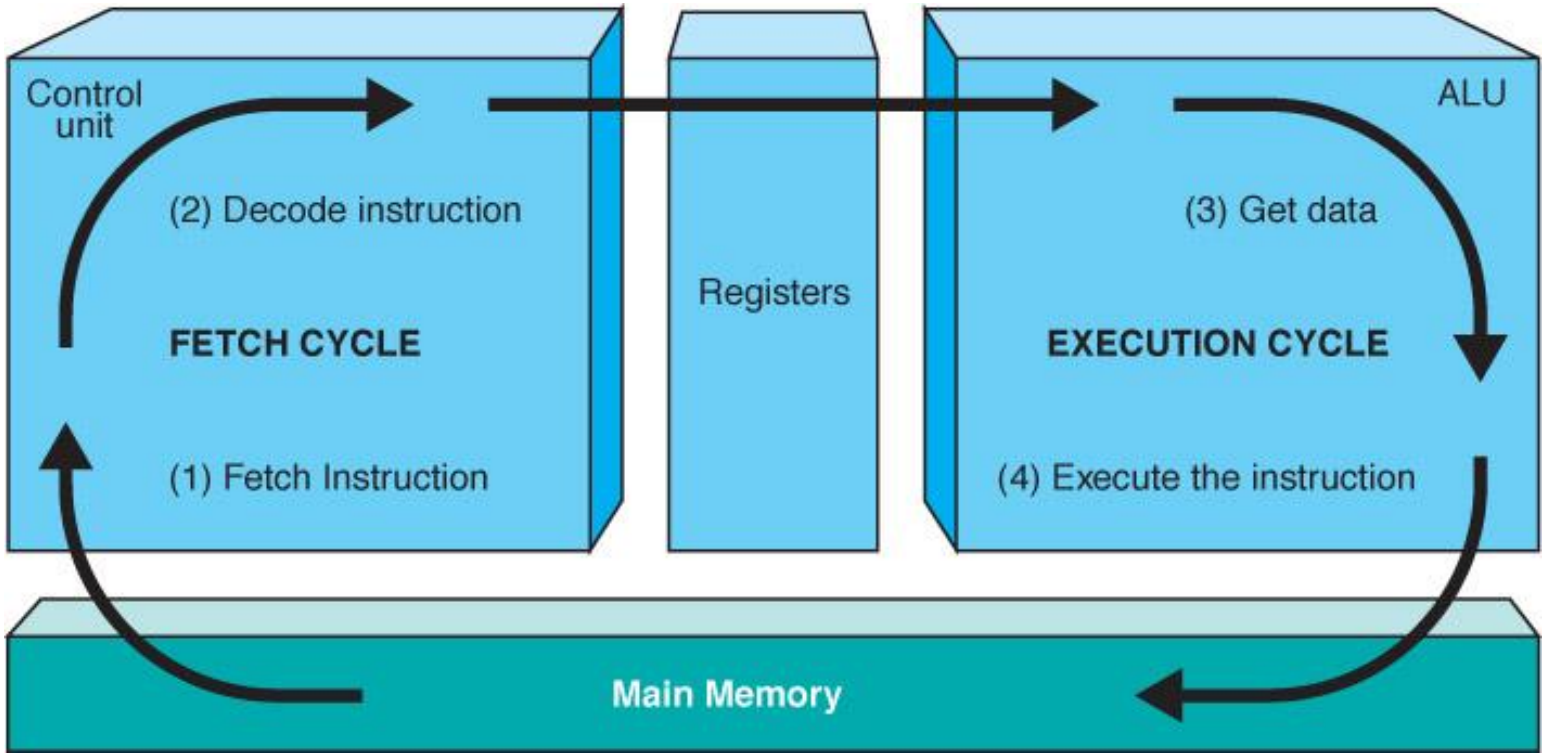



**Figure 5.3** **The Fetch-Execute Cycle**

# Example: The Fetch-Execute Cycle

Replace A by A+1. Branch if A<0 to instruction 00110:

| Instruction Register | Program Counter |
| --- | --- |
| If TRUE… | 00110 |

- 00110 …
- 01000 Get A
- 01010 Add 1 to A
- 01100 Test A<0
- 01110 If True, go to 00110
- 10000 …

| | |
| --- | --- |
| … | … |
| 00110 | … |
| 00111 | … |
| 01000 | Get |
| 01001 | A |
| 01010 | Add |
| 01011 | +1 |
| 01100 | Test |
| 01101 | <0 |
| 01110 | If TRUE |
| 01111 | branch 00110 |
| 10000 | … |
| … | … |

# RAM and ROM

**Random Access Memory** (RAM)

Memory in which each location can be accessed and changed

**Read Only Memory** (ROM)

Memory in which each location can be accessed but *not* changed

RAM is volatile, ROM is not

# Magnetic Disks

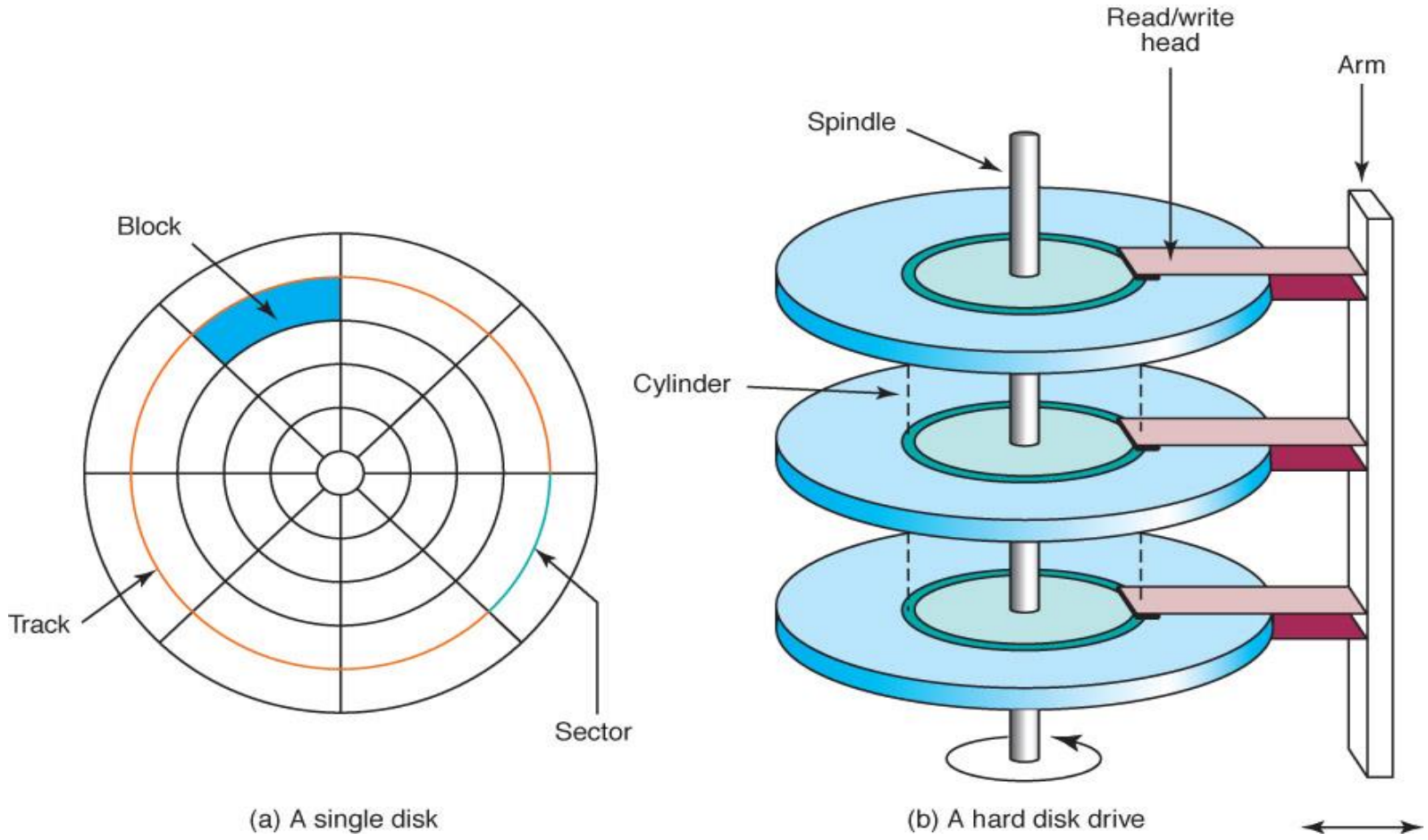



**Figure 5.5** **The organization of a magnetic disk**

# Magnetic Disks

Seek time

Time it takes for read/write head to be over right track

Latency

Time it takes for sector to be in position

Access time

Total time required to obtain data

Transfer rate

Rate at which data is transferred to internal memory.

# Compact Disks

## CD

A compact disk that uses a laser to read information stored optically on a plastic disk; data is evenly distributed around track

- CD-ROM read-only memory
- CD-DA digital audio
- CD-WORM write once, read many
- CD - RW or RAM both read from and written to

## DVD

Digital Versatile Disk, used for storing audio and video in particular.

# Touch Screens

## Touch screen

A computer monitor that can respond to the user touching the screen with a stylus or finger

There are several types

- Resistive
- Capacitive
- Infrared
- Surface acoustic wave (SAW)

# Parallel processing

Idea is to speed up the execution of large jobs based on design rather than processor speed.

There are four approaches to this:

- Bit level

  Manipulate bigger words in each process;

- Instruction level

  Instructions done in parallel rather than sequentially,

  Superscalar is one CPU with multiple resources;

# Parallel processing

- Data Level

  Execute instruction blocks on multiple data sets at the same time,

  Different ALUs do the same instructions on multiple data sets – SIMD;

Task level

Distribute tasks between different processors.

# Synchronous processing

One approach to data level parallelism is to have multiple processors apply the same program to multiple data sets

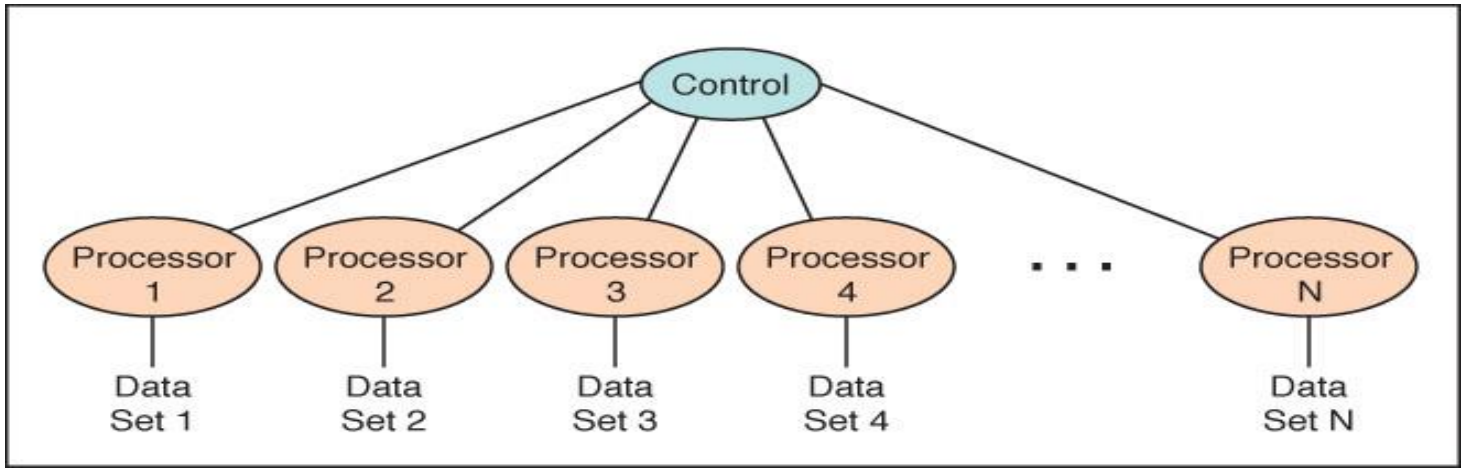



**Figure 5.7** **Processors in a synchronous computing environment**

# Pipelining

Arranges processors in tandem, where each processor contributes one part to an overall computation

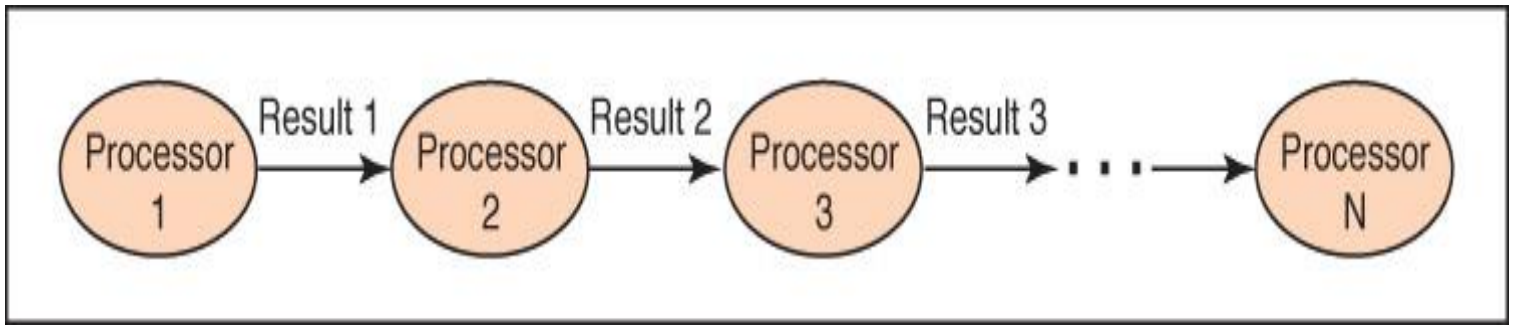



**Figure 5.8** **Processors in a pipeline**

# Independent Processing with Shared Memory

Communicate through shared memory

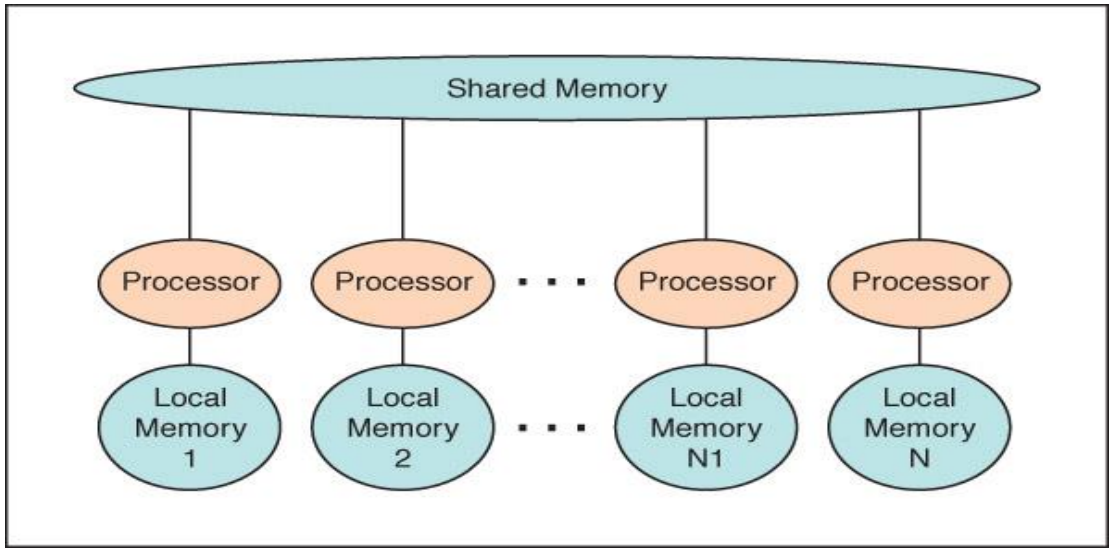



**Figure 5.9** **Shared memory configuration of processors**

# South Africa and HPC

- South Africa has a well-equipped central resource with high end computers – top 500.
- Centre for High Performance Computing

  www.chpc.ac.za

  - SUN Constellation System: 2000 8-core processing units, 400 terabytes, state of the art water cooling system;
  - Training courses;
  - Technical support;
  - Access through high speed links;
  - Flagship projects – Computational Imaging and Remote Sensing (Wits).