

## **Computer Function & Computer System Level Hierarchy**

### **➤ Computer Function**

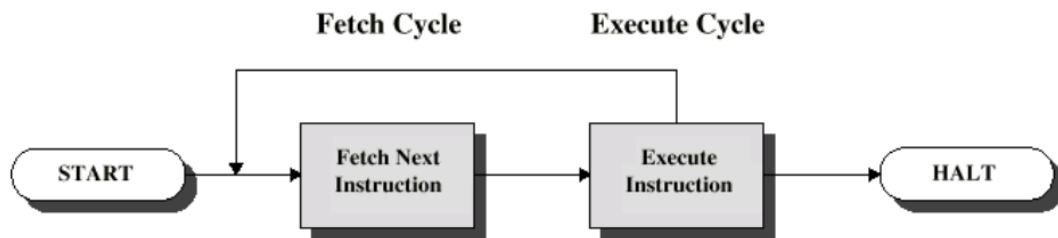
- The basic function performed by a computer is ***execution of a program***, which consists of a set of instructions stored in memory.
- The ***processor*** does the actual work by executing instructions specified in the program.
- In its simplest form, instruction processing consists of two steps:
  - The processor reads (fetches) instructions from memory one at a time.
  - Executes each instruction.

### **Program Execution**

- Program execution consists of repeating the process of instruction fetch and instruction execution.
- The instruction execution may involve several operations and depends on the nature of the instruction. **The processing required for a single instruction is called an *instruction cycle*.**
- The instruction cycle involves two steps which are referred to as the ***fetch cycle and the execute cycle***.

#### **❖ Program Execution Stop:** *Program execution halts only:*

- If the machine is turned off,
- Some sort of unrecoverable error occurs,
- Or a program instruction that halts the computer is encountered.



Instruction Cycle: Fetch and Execute

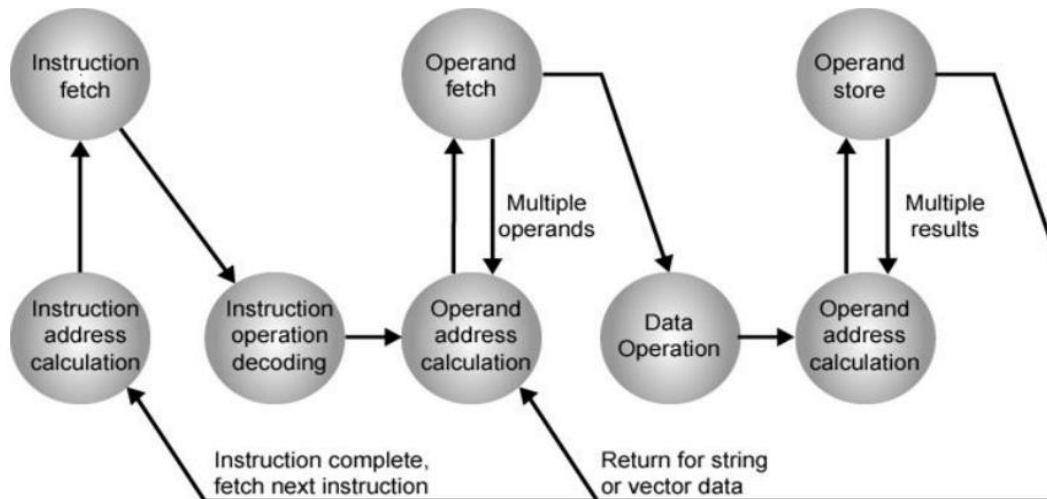
### ✓ Fetch Cycle

- Program Counter (PC) holds address of next instruction to fetch
- Processor fetches instruction from memory location pointed to by PC
- Increment PC
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions

### ✓ Execute Cycle

- Processor-memory
  - data transfer between CPU and main memory
- Processor-I/O
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
- Combination of above

## Instruction Cycle State Diagram



## Steps of the Instruction Cycle

- **Instruction address calculation (iac):** Calculate and determine the instruction address.
- **Instruction fetch (if):** Read instruction from its memory location into the processor.
- **Instruction operation decoding (iod):** Analyze instruction to determine type of operation to be performed and operand(s) to be used.
- **Operand address calculation (oac):** If the operation involves reference to an operand in memory or available via I/O, then determine the address of the operand.
- **Operand fetch (of):** Fetch the operand from memory or read it in from I/O.

- **Data operation (do):** Perform the operation indicated in the instruction.
- **Operand store (os):** Write the result into memory or out to I/O.

### ➤ Computer System Level Hierarchy

**Computer System Level Hierarchy** is the combination of different levels that connects the computer with the user. It also describes how the computational activities are performed on the computer and it shows all the elements used in different levels of system. Computer System Level Hierarchy consists of seven levels:

Level 6	User	Executable Programs
Level 5	High Level Language	C++ , Java
Level 4	Assembly Language	Assembly Code
Level 3	System Software	Operating System
Level 2	Machine	Instruction Set Architecture
Level 1	Control	Microcode
Level 0	Digital Logic	Circuits , Gates

- **Level-0:**

It is related to digital logic. Digital logic is the basis for digital computing and provides a fundamental understanding of how circuits and hardware communicate within a computer. It consists of various circuits and gates etc.

- **Level-1:**

This level is related to control. Control is the level where microcode is used in the system. Control units are included in this level of the computer system.

- **Level-2:**

This level consists of machines. Different types of hardware are used in the computer system to perform different types of activities. It contains instruction set architecture.

- **Level-3:**

System software is a part of this level. System software is of various types. System software mainly helps in operating the process and it establishes the connection between hardware and user interface. It may consist operating system, library code, etc.

- **Level-4:**

Assembly language is the next level of the computer system. The machine understands only the assembly language and hence in order, all the high-level languages are changed in the assembly language. Assembly code is written for it.

- **Level-5:**

This level of the system contains high-level language. High-level language consists of C++, Java, FORTRAN, and many other languages. This is the language in which the user gives the command.

- **Level-6:**

This is the last level of the computer system hierarchy. This consists of users and executable programs.

## ➤ The Abstraction Levels in Hierarchical Computer Architecture

- ✓ **Hardware level:** This is the lowest level in the hierarchy and includes all the *physical components* of the computer system such as the CPU, memory, storage devices, input/output devices, and other peripherals.
- ✓ **Firmware level:** This level includes the *software that is stored in non-volatile memory*, such as the BIOS or firmware on other devices, and is responsible for initializing and controlling the hardware.
- ✓ **Operating system level:** This level includes the *software that manages the resources* of the computer system, provides a user interface, and runs application programs. Examples of operating systems include Windows, macOS, and Linux.
- ✓ **Application level:** This level includes the *software applications* that run on the operating system and perform specific tasks such as word processing, spreadsheet calculations, or playing games.

## ➤ Advantages of the computer system level hierarchy

- **Modularity:** The hierarchical approach to organizing computer systems allows for greater modularity, which can make it easier to develop, test, and maintain individual components.
- **Standardization:** The use of standard interfaces and protocols between layers of the hierarchy can make it easier to integrate new components into a system and to reuse existing components in new contexts.

- **Abstraction:** Each layer of the hierarchy can provide a higher level of abstraction, allowing programmers and users to interact with the system at a higher level of complexity without having to deal with the details of lower-level components.
- **Scalability:** The hierarchical approach can make it easier to scale a system to handle larger workloads by adding more resources or components at different levels of the hierarchy.
- **Interoperability:** The different levels in the hierarchy are designed to work together seamlessly, allowing for software applications to run on different hardware platforms and operating systems.
- **Security:** The different levels in the hierarchy can be isolated from each other, providing a layered approach to security and reducing the risk of security breaches.

### ➤ **Disadvantages of the computer system level hierarchy:**

- ✓ **Complexity and Overhead:** The hierarchical approach can make it difficult for users and developers to understand and navigate the system, especially as the number of components and layers increases.
- ✓ **Dependencies:** Changes to one layer of the hierarchy can have ripple effects on other layers, which can make it difficult to modify or update a system without causing unintended consequences.
- ✓ **Inefficiency:** In some cases, the hierarchical approach can lead to inefficiencies in the use of system resources, as components at different levels may not be optimized to work together efficiently.