**DEPARTMENT OF COMPUTER SCIENCE**

# 17PCS02 – ADVANCED COMPUTER ARCHITECTURE

**YEAR : I SEM :I**

**Handled by**

**Dr.K.Shanmugavadivu**

**Core Course-II-17PCS02 ADVANCED COMPUTER ARCHITECTURE**

**UNIT-I**

Evolution of Computer systems – Parallelism in Unip rocessor Systems: Architecture,

Mechanisms – Parallel Computer Structures: Pipeline , Array, Multiprocessor.

**UNIT - II**

Linear Pipeline processors: Asynchronous and Synchronous Models – Non-linear Pipeline

Processors: Reservation and Latency Analysis – Coll ision-free scheduling – Instruction Pipeline

Design: Instruction Execution Phases – Mechanisms f or Instruction Pipelining – Arithmetic

Pipeline Design: Computer Arithmetic Principles – S tatic Arithmetic Pipelines – Multifunctional Arithmetic Pipelines - Superscalar Pipeline Design.

**UNIT- III**

SIMD Array Processor – SIMD Interconnection Network : Static vs Dynamic Network – Mesh connection Illiac Network- Tube interconnection Network. Associative Array Processing: Associative memory organisation.

**UNIT - IV**

Multiprocessor System Interconnects : Hierarchical Bus System - Crossbar Switch and Multiport Memory - Multistage and Combining Networks – Cache Coherence and Synchronization Mechanisms: The Cache Coherence Problem – Snoopy Bus Protocols – Directory-Based Protocols – Hardware Synchronizatio n Mechanisms – Message-Passing Mechanisms: Message-Routing Schemes – Deadlock and Virtual Channels – Flow Control Strategies – Multicast Routing Algorithms.

**UNIT - V**

Multiprocessor Operating Systems - Interprocessor Communication Mechanisms - Multiprocessor Scheduling Strategies.

**TEXT BOOKS:**

1. Kai Hwang, Faye A.Briggs, “Computer Architecture an d Parallel Processing,” McGraw-Hill, 1985.

2. Kai Hwang, “Advanced Computer Architecture,” McGraw -Hill International Editions, 2001.

**REFERENCE BOOKS:**

1. Grama, "An Introduction to Parallel Computing: Design and Analysis of Algorithms," 2nd Edition, Pearson, 2004.

2. Gita Alaghband, Harry Frederick Jordan, "Fundamentals of Parallel Processing," Prentice Hall, 2003.

3. Seyed H Roosta, "Parallel Processing and Parallel Algorithms: Theory and Computation," Springer Science & Business Madia, 1999

# UNIT – I

## Evolution of computer system

Each level of a **system evolution** is built on the previous, so that social **computing**emerges from personal **computing**, personal **computing** emerges from software, and software emerges from hardware. As **computing** evolves to higher **system** levels, so its design also changes, from technical to socio-technical design.

## Generations of computers

Each generation is defined by a significant technological development that changes fundamentally how computers operate – leading to more compact, less expensive, but more powerful, efficient and robust machines.

### 1940 – 1956: First Generation – Vacuum Tubes

These early computers used vacuum tubes as circuitry and magnetic drums for memory. As a result they were enormous, literally taking up entire rooms and costing a fortune to run. These were inefficient materials which generated a lot of heat, sucked huge electricity and subsequently generated a lot of heat which caused ongoing breakdowns.

These first generation computers relied on 'machine language' (which is the most basic programming language that can be understood by computers). These computers were limited to solving one problem at a time. Input was based on punched cards and paper tape. Output came out on print-outs. The two notable machines of this era were the UNIVAC and ENIAC machines – the UNIVAC is the first every commercial computer which was purchased in 1951 by a business – the US Census Bureau.

### 1956 – 1963: Second Generation – Transistors

The replacement of vacuum tubes by transistors saw the advent of the second generation of computing. Although first invented in 1947, transistors weren't used significantly in computers until the end of the 1950s. They were a big improvement over the vacuum tube, despite still subjecting computers to damaging levels of heat. However they were hugely superior to the vacuum tubes, making computers smaller, faster, cheaper and less heavy on electricity use. They still relied on punched card for input/printouts.

The language evolved from cryptic binary language to symbolic ('assembly') languages. This meant programmers could create instructions in words. About the same time high level programming languages were being developed (early versions of COBOL and FORTRAN). Transistor-driven machines were the first computers to store instructions into their memories – moving from magnetic drum to magnetic core 'technology'. The early versions of these machines were developed for the atomic energy industry.

**1964 – 1971: Third Generation – Integrated Circuits**

By this phase, transistors were now being miniaturised and put on silicon chips (called semiconductors). This led to a massive increase in speed and efficiency of these machines.  These were the first computers where users interacted using keyboards and monitors which interfaced with an operating system, a significant leap up from the punch cards and printouts. This enabled these machines to run several applications at once using a central program which functioned to monitor memory.

As a result of these advances which again made machines cheaper and smaller, a new mass market of users emerged during the '60s.

**1972 – 2010: Fourth Generation – Microprocessors**

This revolution can be summed in one word: Intel. The chip-maker developed the Intel 4004 chip in 1971, which positioned all computer components (CPU, memory, input/output controls) onto a single chip. What filled a room in the 1940s now fit in the palm of the hand. The Intel chip housed thousands of integrated circuits. The year 1981 saw the first ever computer (IBM) specifically designed for home use and 1984 saw the

MacIntosh introduced by Apple. Microprocessors even moved beyond the realm of computers and into an increasing number of everyday products.

The increased power of these small computers meant they could be linked, creating networks. Which ultimately led to the development, birth and rapid evolution of the Internet. Other major advances during this period have been the Graphical user interface (GUI), the mouse and more recently the astounding advances in lap-top capability and hand-held devices.

**2010- : Fifth Generation – Artificial Intelligence**

Computer devices with artificial intelligence are still in development, but some of these technologies are beginning to emerge and be used such as voice recognition.

AI is a reality made possible by using parallel processing and superconductors. Leaning to the future, computers will be radically transformed again by quantum computation, molecular and nano technology.

The essence of fifth generation will be using these technologies to ultimately create machines which can process and respond to natural language, and have capability to learn and organise themselves.

**Trends Towards Parallel Processing**

From an application point of view, the mainstream of usage of computer is experiencing a trend of four ascending levels of sophistication: 1 • Data processing • Information processing • Knowledge processing • Intelligence processing With more and more data structures developed, many users are shifting to computer roles from pure data processing to information processing. A high degree of parallelism has been found at these levels. As the accumulated knowledge bases expanded rapidly in recent years, there grew a strong demand to use computers for knowledge processing. Intelligence is very difficult to create; its processing even more so. Todays computers are very fast and obedient and have many reliable memory cells to be qualified for datainformation-knowledge processing. Parallel processing is emerging as one of the key technology in area of modern computers. Parallel appears in various forms such as lookahead, vectorization concurrency, simultaneity, data parallelism, interleaving, overlapping, multiplicity, replication, multiprogramming, multithreading and distributed computing at different processing level.

There are four ascending levels. They are-

1) Data Processing
2) Information Processing
3) Knowledge Processing
4) Intelligence Processing

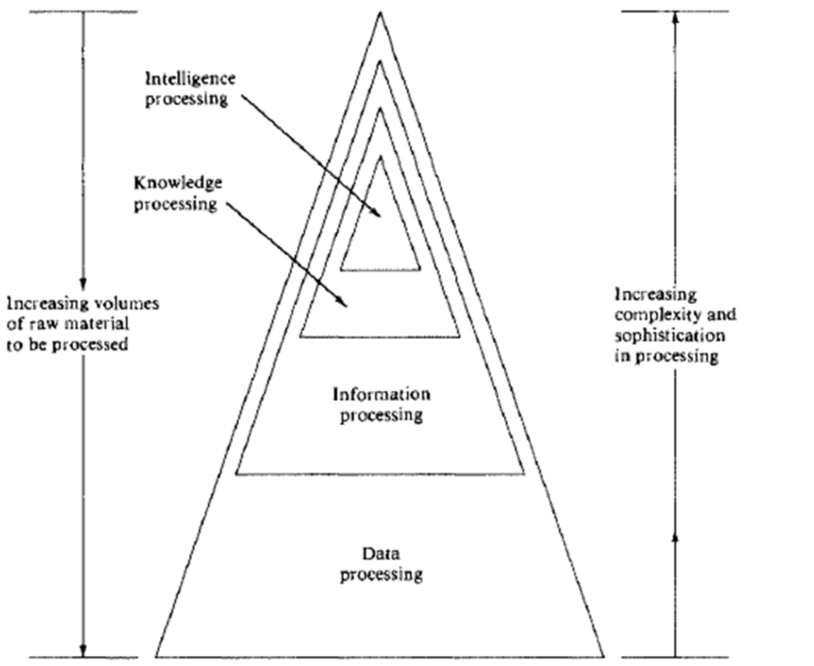

Intelligence processing
Knowledge processing
Increasing volumes of raw material to be processed
Information processing
Data processing
Increasing complexity and sophistication in processing

**Figure 1.2 The spaces of data, information, knowledge, and intelligence from the viewpoint of computer processing.**

-> The dataspace is the largest, including numeric numbers in various formats, character symbols and multidimensional; measures

-> Data objects are mutually unrelated. Huge amounts of data is being generated especially among the scientific, business and government sectors

-> An Information is a collection of data objects that are related by some syntactic relation. Therefore information forms a subspace of the dataspace

-> Knowledge consists of information items with some semantic meanings and thus, knowledge is the subspace of information space

-> Intelligence is the derived from a collection of knowledge items and innermost triangle in the Venn diagram

-> of today's computers many users are shifting to computer roles from pure data processing to information processing. A high degree of parallelism has been found at these levels.

**Basic Uniprocessor Architecture:**

A typically uniprocessor computer consists of the three major components the main memory, the CPU (Central Processing Unit) and the I/O(Input-Output) subsystem.

- There are two architectures of commercially available uniprocessor computers to show the relation between three subsystems

- The CPU contains the master controller of the VAX system
- There are 16, 32-bit general purpose register one of which is a Program Counter (PC).There is also a special CPU status register containing about the current state of the processor being executed
- The CPU contains an ALU with an optional Floating-point accelerator, and some local cache memory with an optional diagnostic memory
- The CPU can be intervened by the operator through the console connected to floppy disk
- The CPU, the main memory( 2^32 words of 32 bit each) and the I/O subsystem are all connected to a common bus, the synchronous backplane interconnection(SBI)

1. System Architecture of the supermini VAX-11/780 uniprocessor system

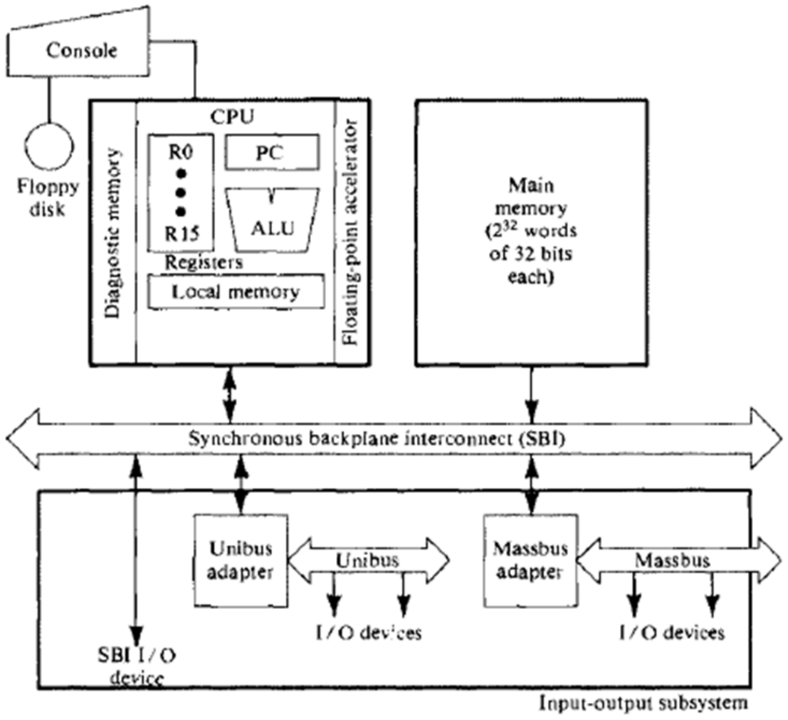


**Figure 1.3 The system architecture of the supermini VAX-11/780 uniprocessor system (Courtesy of Digital Equipment Corporation).**

- Through this bus, all I/O devices can communicate with each other with CPU or with the memory
- I/O devices can be connected directly to the SBI through the unibus and its controller or through a mass bus and its controller

System Architecture of the mainframe IBM system 370/model 168 uniprocessor computer

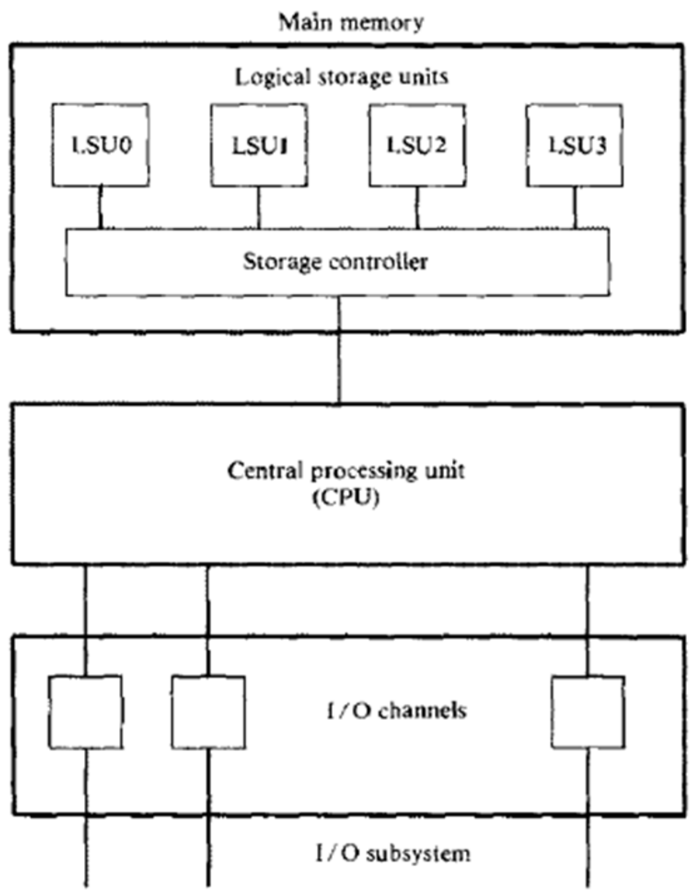


**Figure 1.4 The system architecture of the mainframe IBM System 370/Model 168 uniprocessor computer (Courtesy of International Business Machines Corp.).**

- The CPU contains the instruction decoding and execution units as well as cache
- Main memory is divided into four units referred to as logical storage units (LSU) that are four ways interleaved
- The storage controller provides multiport Connections between the CPU and the four
- LSU's

- Peripherals are connected to the system via high speed I/O channels which operate asynchronously with the CPU

- Parallelism in Multiprocessor Systems
- Parallel processing systems achieve parallelism by having more than one processor performing tasks simultaneously. Since multiprocessor systems are more complicated than uniprocessor systems, there are many different ways to organize the processors and memory, so a researcher, Michael J. Flynn proposed a classification based on the flow of instructions and data within the computer called Flynn's classification

**Parallel Processing Mechanisms**

Parallel processing is the process of breaking down program instruction by the computer and running it through a number of different processors. However in a uniprocessor this isn't possible. So we have a number of different mechanisms to make a uniprocessor system behave like a multiprocessor system. The different ways it is possible is:

1. Multiplicity of functional units
2. Parallelism and pipelining within the CPU
3. Overlapped CPU and I/O operations
4. Use of a hierarchical memory system
5. Balancing of subsystem bandwidths
6. Multiprogramming and time sharing

**Multiplicity of functional units**

- Early computers
- one ALU that perform one operation at a time.
- Slow process
- Multiple and specialized functional units.
- operate in parallel.
- IBM 360/91
- two parallel execution units (fixed and floating point arithmetic)
- CDC-6600 → 10 functional units

**Parallelism and pipelining within CPU**

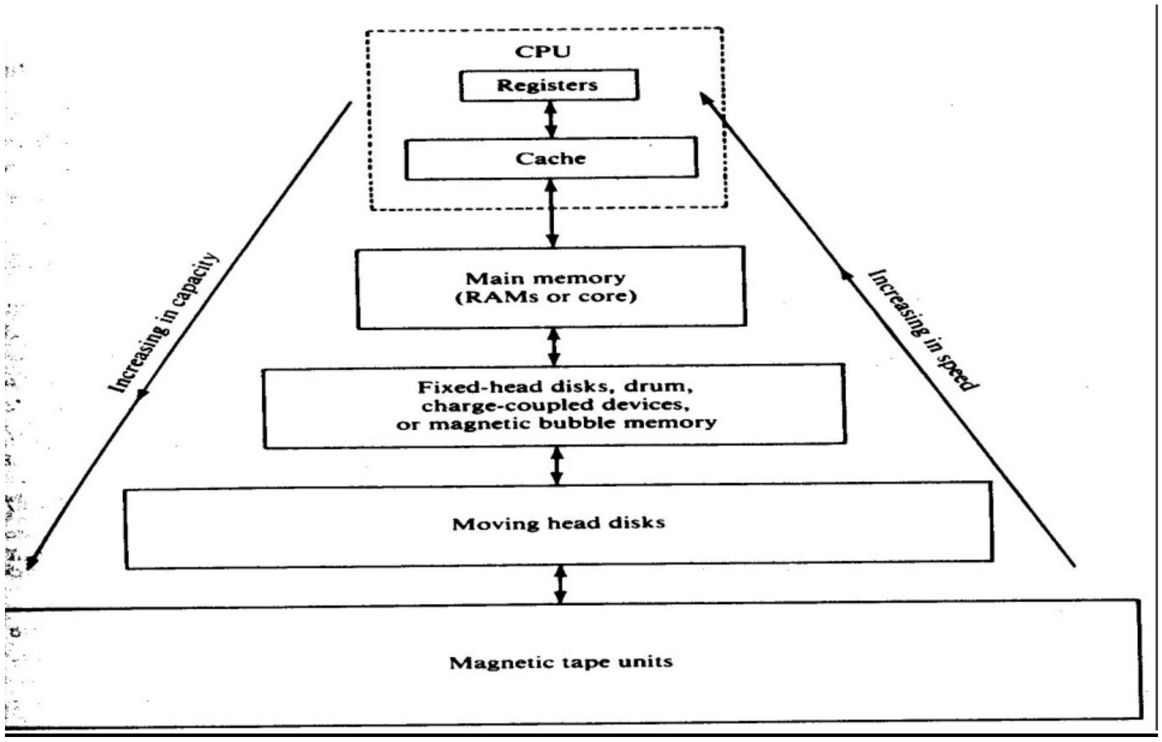
- Parallel Adders
  - bit serial adders.

  - carry-lookahead and carry save adders.
- Multiplier recoding and convergence division.
- Phases of instruction execution are pipelined
  - Instruction fetch, decode, operand fetch, arithmetic logic execution, store result.
- Instruction Prefetch and data buffering.

**Overlapped CPU and I/O operations**

- I/O operations performed simultaneously with CPU computations
  - separate I/O controllers, channels or I/O processors.
- DMA channels – cycle stealing.

**Use of a hierarchical memory system**


CPU
Registers
Cache
Main memory (RAMs or core)
Fixed-head disks, drum, charge-coupled devices, or magnetic bubble memory
Moving head disks
Magnetic tape units
Increasing in capacity
Increasing in speed

**Balancing of subsystem bandwidths:**

The bandwidth of a system is defined as the number of operation performed per unit time. In the case of a main memory system, the memory bandwidth is measured by the number of memory words that can be accessed per unit time.

In general, the CPU is the fastest unit in a computer, with a processor cycle of t p ; main memory cycle time of t m;and I/O devices average access time of t d , it is observed that:
td>tm>tp
$B_m=W/t_m$
Memory access conflicts may cause delayed access of some of the processor requests. Therefore, the utilized memory bandwidth $B^n$m.
$B^n{}_m = B/\sqrt{M}$
For the bandwidth of external memory and I/O devices, the average access rate per tape is 1 megabyte/ sec. So for 10 tapes it would be 10 megabytes/sec
. A modern magnetic tape unit has a data transfer rate around 1.5 megabytes/s.
The bandwidth of a processor is measured as the maximum CPU computation rate B,
p As in 160 megaflops for Cray-1 and 12.5 million instructions per second for IBM 370/168. In practice the utilized CPU rate is less than the bandwidth.
$B^m{}_p = R_w / T$
The following relationship has been observed between the bandwidths of major subsystems in a high performance uniprocessor:
$B_m \geq B_m^\mu \geq B_p \geq B_p^\eta \geq B_d$
This implies that the main memory has the highest bandwidth, since it must be updated by the CPU and I/O. Due to unbalanced speeds we need to match the processing power of the three subsystems.
<u>Bandwidth balancing between CPU and memory:</u>
The speed gap between CPU and the main memory can be closed up by using the fast cache memory between them. The cache should have an access time equal to processor time. A block of memory is moved from the main memory into the cache so that immediate instructions can be available most of the time from the cache. The cache

serves as a data buffer.

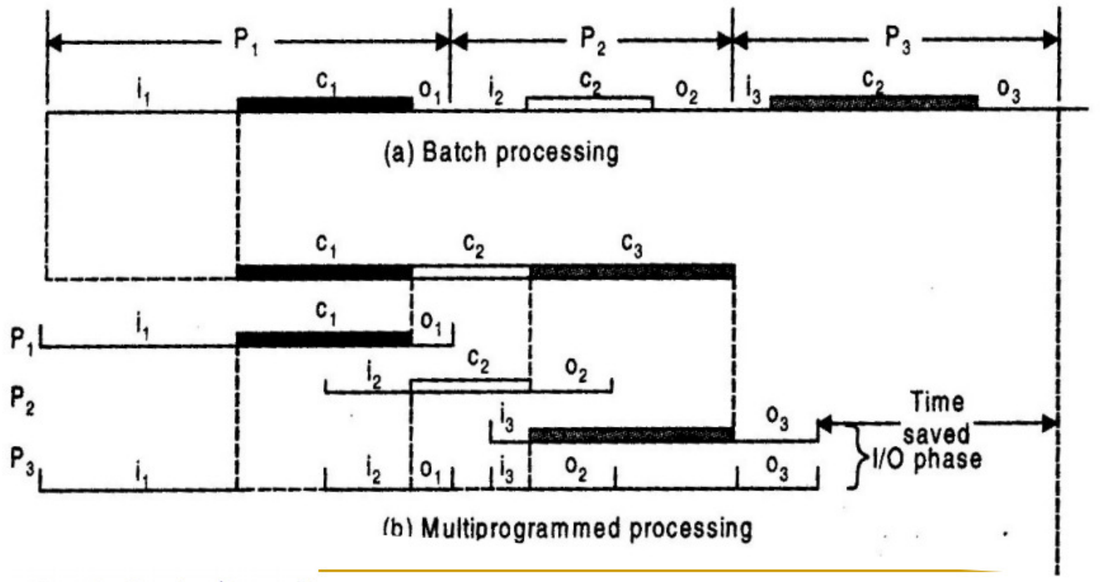
Bandwidth balancing between memory and I/O devices

Input-output channels with different speeds can be used between the slow I/O devices and the main memory. These I/O channels perform buffering and multiplexing functions to transfer the data from multiple disks into the main memory bu stealing cycles from the CPU. Furthermore, intelligent disk controllers or database machines can be used to filter out the irrelevant data just off the tracks of the disk. This filtering will alleviate the I/O channel saturation problem. The combined buffering, multiplexing and filtering operations thus can provide a faster, more effective data transfer rate, matching that of the memory.

In the ideal case, we wish to achieve a totally balanced system, in which the entire memory bandwidth matches the bandwidth sum of the processor and I/O devices that is $B^{p\mu} + B_d = B_m^{\mu}$

**Multiprogramming and time sharing**

- Batch processing
  - Sequential execution
- Multiprogramming
  - Interleaving of CPU and I/O operations among several programs
- Time sharing
  - Assign fixed or variable time slices to multiple programs



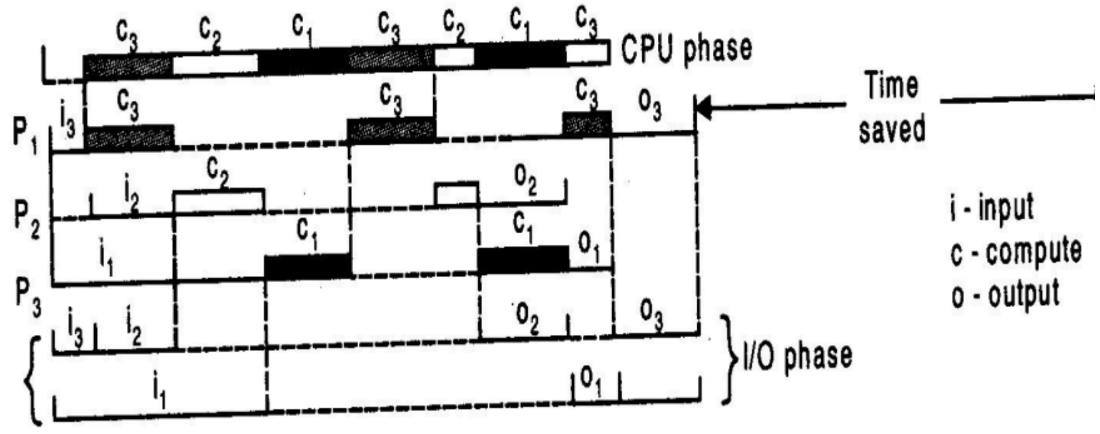


Fig. 1.8. Time Shared Processing