

Electronics Engineering

Computer Organization and Architecture

Comprehensive Theory

with Solved Examples and Practice Questions



MADE EASY
Publications



MADE EASY Publications Pvt. Ltd.

Corporate Office: 44-A/4, Kalu Sarai (Near Hauz Khas Metro Station), New Delhi-110016

E-mail: infomep@madeeasy.in

Contact: 011-45124660, 8860378007

Visit us at: www.madeeasypublications.org

Computer Organization and Architecture

Copyright ©, by MADE EASY Publications Pvt. Ltd.

All rights are reserved. No part of this publication may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photo-copying, recording or otherwise), without the prior written permission of the above mentioned publisher of this book.

First Edition: 2015

Second Edition: 2016

Third Edition: 2017

Fourth Edition: 2018

Fifth Edition: 2019

Sixth Edition : 2020

Seventh Edition : 2021

Eighth Edition : 2022

Contents

Computer Organization and Architecture

Chapter 1

Computer Organization 1

1.1 Computer Architecture Vs Computer Organization	1
1.2 Evolution of Digital Computers.....	2
1.3 Components of Computer Structure.....	2
1.4 CISC and RISC Architectures.....	3
1.5 Flynn's Classification of Processors	4
1.6 Control Unit.....	5
1.7 Control Unit Implementation	6
1.8 Main Memory Organisation.....	9
1.9 Associative Memory.....	12
1.10 Pipelining	20
1.11 Secondary Storage	24
1.12 Internal Fragmentation and External Fragmentation.....	31
1.13 Paging	32
<i>Student's Assignments</i>	36

Chapter 2

Data Structure..... 40

2.1 Scope.....	40
2.2 Flow Control in 'C'.....	43
2.3 Evaluation of function.....	54
2.4 Pointers.....	56
2.5 Array.....	58
2.6 Stack.....	62
2.7 Expression Evaluation and Syntax Parsing.....	64
2.8 Evaluation of an Infix Expression.....	65
2.9 Evaluation of Prefix Expression	66

2.10 Postfix Evaluation.....	67
2.11 Infix to Postfix Conversion	69
2.12 Linked Lists	70
2.13 Queue.....	71
2.14 Tree Traversals	72
2.15 Binary Search Tree.....	73
2.16 Analysis Of Loops.....	81
2.17 Comparisons of Functions	86
2.18 Asymptotic Behaviour of Polynomials	87
<i>Student's Assignments</i>	90

Chapter 3

Operating System..... 93

3.1 Basics of Operating System.....	93
3.2 Process	98
3.3 CPU Scheduling.....	104
3.4 Memory Management	116
3.5 Virtual Memory	137
3.6 Basics of File	144
3.7 Protection versus Security	154

Chapter 4

Database Management System159

4.1 Introduction.....	159
4.2 Introduction to Database Design.....	159
4.3 Introduction to Database Design and Normalization.....	165
4.4 Introduction to Transaction.....	174
4.5 Introduction to Concurrency and Serializability.....	184



Computer Organization

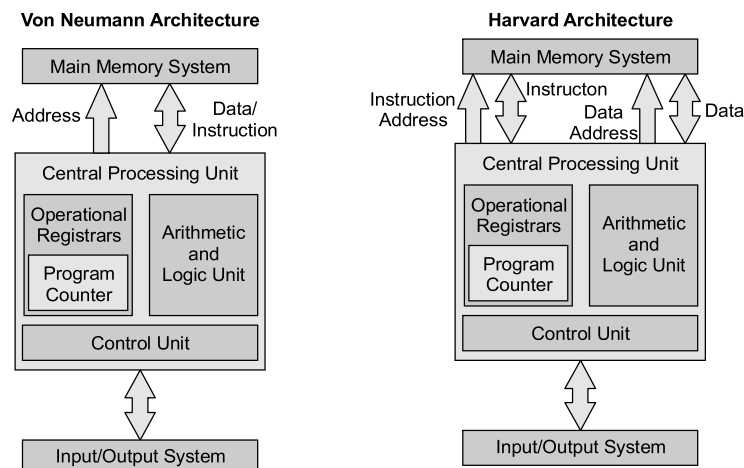
1.1 Computer Architecture Vs Computer Organization

Architecture and organization are independent; you can change the organization of a computer without changing its architecture.

1. The architecture indicates its hardware whereas the organization reveals its performance.
2. For designing a computer, its architecture is fixed first and then its organization is decided.

Computer Organization	Computer Architecture
<ul style="list-style-type: none"> • Computer organization deals with structural relationships that are not visible to the programmer (like clock frequency or the size of the physical memory). 	<ul style="list-style-type: none"> • Computer architecture deals with the functional behavior of a computer system as viewed by a programmer (like the size of a data type – 32 bits to an integer).
<ul style="list-style-type: none"> • A computer's organization expresses the realization of the architecture. 	<ul style="list-style-type: none"> • A computer's architecture is its abstract model and is the programmer's view in terms of instructions, addressing modes and registers.
<ul style="list-style-type: none"> • Organization describes how it does it. 	<ul style="list-style-type: none"> • Architecture describes what the computer does.

Von Neumann Architecture Vs Harvard Architecture



1.2 Evolution of Digital Computers

First generation: Vacuum tube computers (1945~1953)

- Program and data reside in the same memory (stored program concepts: John von Neumann)
- Vacuum tubes were used to implement the functions (ALU & CU design)
- Magnetic core and magnetic tape storage devices are used.
- Using electronic vacuum tubes, as the switching components.
- Assembly level language is used

Second generation: Transistorized computers (1954~1965)

- Transistor were used to design ALU & CU
- High Level Language is used (FORTRAN)
- To convert HLL to MLL compiler were used
- Separate I/O processor were developed to operate in parallel with CPU, thus improving the performance
- Invention of the transistor which was faster, smaller and required considerably less power to operate

Third generation: Integrated circuit computers (1965~1980)

- IC technology improved
- Improved IC technology helped in designing low cost, high speed processor and memory modules
- Multiprogramming, pipelining concepts were incorporated
- DOS allowed efficient and coordinate operation of computer system with multiple users
- Cache and virtual memory concepts were developed
- More than one circuit on a single silicon chip became available.

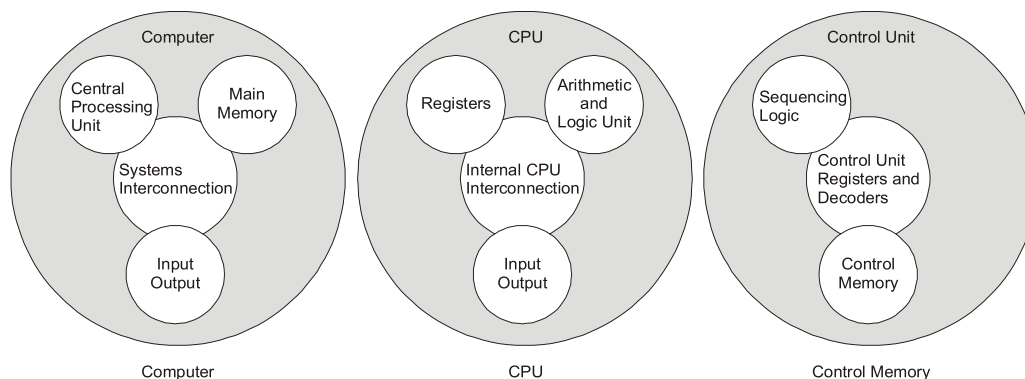
Fourth generation: Very large scale integrated (VLSI) computers (1980~2000)

- CPU termed as microprocessor
- INTEL, MOTOROLA, TEXAS, NATIONAL semiconductors started developing microprocessor
- Workstations, microprocessor (PC) & Notebook computers were developed
- Interconnection of different computer for better communication LAN, MAN, WAN
- Computational speed increased by 1000 times
- Specialized processors like Digital Signal Processor were also developed.

Fifth generation: System-on-chip (SOC) computers (2000~)

- E-Commerce, E- banking, home office
- ARM, AMD, INTEL, MOTOROLA
- High speed processor - GHz speed
- Because of submicron IC technology, more features were added in small size.

1.3 Components of Computer Structure



Computer Structure vs CPU Structure vs Control Unit

1. **Input Unit:** Computers can understand only machine language. Therefore for converting data from human language to machine language we use some special peripheral devices which are called input device.
Examples: Keyboard, Mouse, Joystick, etc.
2. **Output Unit:** After passing instructions for solving particular problem, the results came out from computer comes in machine language and this is very difficult to convert that results into human language. There are several peripheral devices which help us to convert the machine language data into human acceptable data. These devices are called output devices.
Examples: Monitor, Printer, LCD, LED etc.
3. **Memory Unit:** Which is used to store data in computer.
Memory unit performs the following functions
 - (a) Stores data and instructions required for processing.
 - (b) Stores the intermediate results obtain during processing.
 - (c) Stores final results before sending it to output unit.

Two class of storage units: (i) Primary Memory (ii) Secondary Memory
Two types of primary memory are RAM (Random Access Memory) and ROM (Read Only Memory). RAM is used to store data temporarily during the program execution. ROM is used to store data and program which is not going to change.
Secondary Memory is used for bulk storage or mass storage to store data permanently.
4. **CPU:** It is main unit of the computer system. It is responsible for carrying out computational task. The major structural components of a CPU are:
 - (a) *Control Unit (CU):* Controls the operation of the CPU and hence the computer.
 - (b) *Arithmetic and Logic Unit (ALU):* Performs computer's data processing functions.
 - (c) *Register:* Provides storage internal to the CPU.
 - (d) *CPU Interconnection:* communication among the control unit, ALU, and register.

1.4 CISC and RISC Architectures

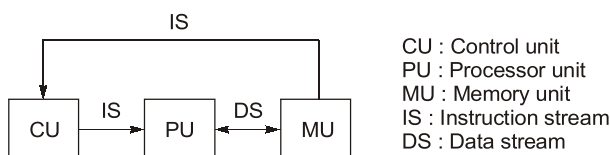
CISC (Complex Instruction Set Computers)	RISC (Reduced Instruction Set Computers)
• Large instruction set	• Compact instruction set
• Instruction formats are of different lengths	• Instruction formats are all of the same length
• Instructions perform both elementary and complex operations	• Instructions perform elementary operations
• Control unit is microprogrammed	• Control unit is simple and hardwired
• Not pipelined or less pipelined	• Pipelined
• Single register set	• Multiple register set
• Numerous memory addressing options for operands	• Compiler and IC developed simultaneously
• Emphasis on hardware	• Emphasis on software
• Includes multi-clock complex instructions	• Single-clock, reduced instruction only
• Memory-to-memory: "LOAD" and "STORE" incorporated in instructions	• Register to register: "LOAD" and "STORE" are independent instructions
• Small code sizes, high cycles per second	• Low cycles per second, large code sizes
• Transistors used for storing complex instructions	• Spends more transistors on memory registers
Examples of CISC processors: <ul style="list-style-type: none"> • VAX • PDP11 • Motorola 68000 family • Intel x86 architecture based processors. 	Examples of RISC processors <ul style="list-style-type: none"> • Apple iPods (custom ARM7TDMI SoC) • Apple iPhone (Samsung ARM1176JZF) • Nintendo Game Boy Advance (ARM7) • Sony Network Walkman (Sony in-house ARM based chip)

1.5 Flynn's Classification of Processors

1.5.1 Single Instruction Stream, Single Data Stream (SISD)

A computer with a single processor is called a Single Instruction Stream, Single Data Stream (SISD) Computer. It represents the organization of a single computer containing a control unit, a processor unit, and a memory unit. Instructions are executed sequentially and the system may or may not have internal parallel processing. Parallel processing may be achieved by means of a pipeline processing.

In such a computer a single stream of instructions and a single stream of data are accessed by the processing elements from the main memory, processed and the results are stored back in the main memory. SISD computer organization is shown in figure below.

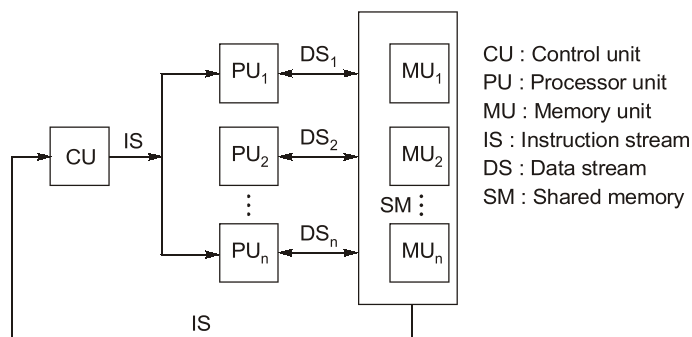


1.5.2 Single Instruction Stream, Multiple Data Stream (SIMD)

It represents an organization of computer which has multiple processors under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of the data. SIMD computers are used to solve many problems in science which require identical operations to be applied

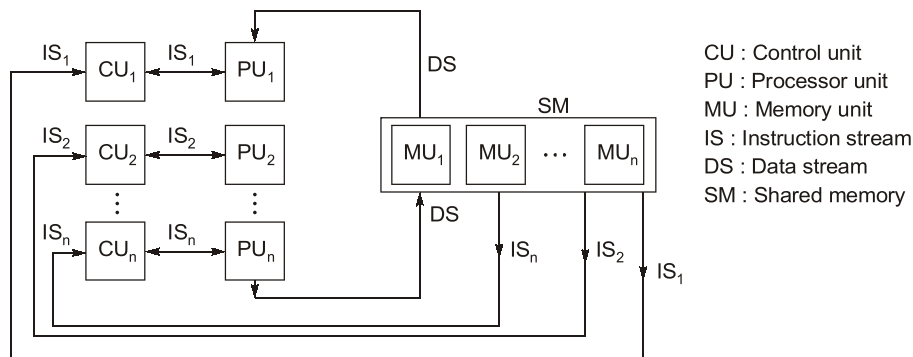
to different data set synchronously. Examples are added a set of matrices simultaneously, such as $\sum_i \sum_k (a_{ik} + a_{ik})$.

Such computers are known as array processors. SIMD computer organization is shown in figure below.



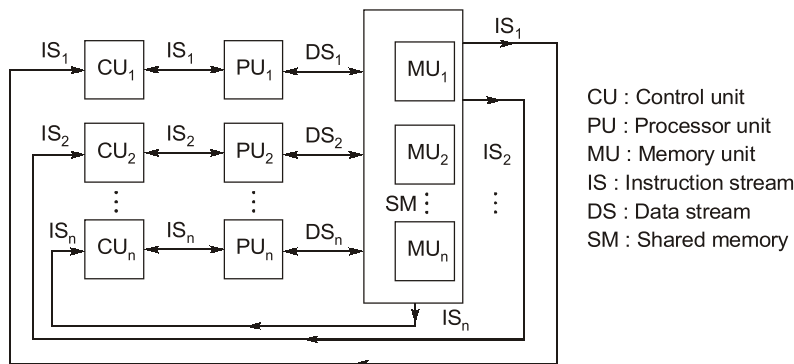
1.5.3 Multiple Instruction Stream, Single Data Stream (MISD)

It refers to the computer in which several instructions manipulate the same data stream concurrently. In the structure different processing element run different programs on the same data. This type of processor may be generalized using a 2-dimensional arrangement of processing element. Such a structure is known as systolic processor. MISD computer organization is shown in figure below.



1.5.4 Multiple Instruction Stream, Multiple Data Stream (MIMD)

MIMD computers are the general purpose parallel computers. Its organization refers to a computer system capable of processing several programs at a same time. MIMD systems include all multiprocessing systems. MIMD computer organization is shown in figure below.]



1.6 Control Unit

Control unit generates the signals for sequencing the operations in the datapath. It performs the task by repeatedly cycling through fetch-execute cycle steps:

- Read the instruction that's pointed to by the PC from memory and move it into the IR.
- Increment the PC.
- Decode the instruction in the IR.
- If the instruction has to read an operand from memory, calculate the operand's address (*effective address*) and read the operand from memory.
- Execute the current instruction from the IR.

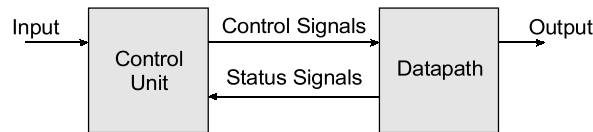
To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence.

Functions of Control Unit

1. **Fetch and instruction sequencing:** Generates control signal to fetch instruction from memory and the sequence of operations involved in processing an instruction.
2. **Instruction interpretation and execution:**
 - Interpreting the operand addressing mode implied in the operation code and fetching the operands.
 - Sequencing the successive micro operations on the data path to execute the operation code specified in the instruction.

3. Interrupt processing: Process unmasked interrupts in the interrupt cycle as follows:

- Suspend execution of current program
- Save context
- Set PC to start address of interrupt handler routine
- Process interrupt
- Restore context and continue interrupted program

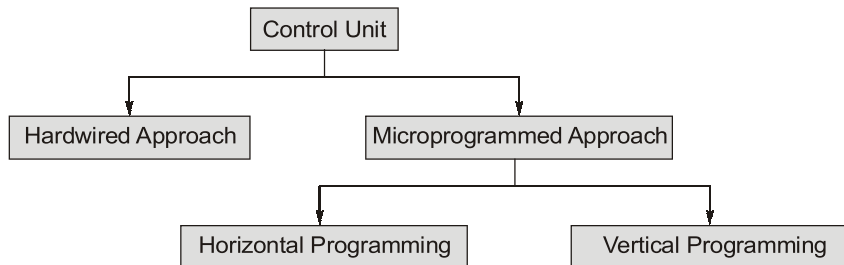
Control Unit and Datapath

Control Unit: It generates the signals for sequencing the micro-operations in the datapath based on status and input signals.

Datapath: It implements the micro-operations under control of the control unit using its functional units (registers, ALU, MUXes, Buses, etc.)

1.7 Control Unit Implementation

The main objective of control unit is to generate the control signal in proper sequence. Control unit is implemented in one of two ways either Hardwired control or Micro-programmed control.

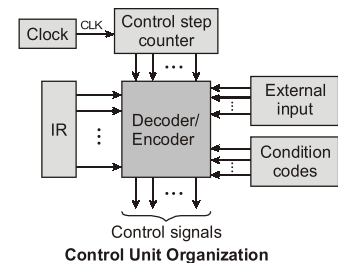
**1.7.1 Hardwire Control Unit**

Hardwire Control unit is made up of sequential and combinational circuits (Hardware) to generate the control signals.

Decoder/Encoder Block

It is a combinational circuit that generates the required control signals depending on the state of all its input.

- Step Decoder:** The decoder part of decoder/encoder part provide a separate signal line for each control step, or time slot in the control sequence.
- Instruction Decoder:** The output of the instructor decoder consists of a separate line for each machine instruction loaded in the IR, one of the output line INS_1 to INS_m is set to 1 and all other lines are set to 0.
- Encoder:** It is required to generate many control signals by the control unit. These are basically coming out from the encoder circuit of the control signal generator.
The control signals are: PC_{in} , PC_{out} , Z_{in} , Z_{out} , MAR_{in} , ADD, END, etc.



The encoder sends a reset signal after the end of an instruction and a stop signal to the sequencer after the last sequence. The encoder also sends count start signal to let the clock increment the counter during processing of an instruction.

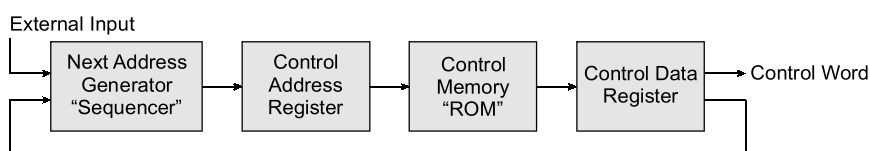
The advantage of hardwired control is that is very fast. The disadvantage is that the instruction set and the control logic are directly tied together by special circuits that are complex and difficult to design or modify.

The control signals are expressed as Sum-of-Product (SOP) expression and they are directly realized on the independent hardware.

1.7.2 Microprogrammed Control Unit

A control unit whose binary control variables are stored in memory is called a micro-programmed control unit. A control memory (Control storage) on the processor contains micro-programs that activate the necessary control signals whereas hardwired control unit generate control signals by sequential and combinational circuits.

- **Microinstruction:** The microinstruction specifies one or more micro-operations for the system. It contains a control word and a sequencing word.
- **Microprogram:** A sequence of microinstructions called a microprogram. Program stored in memory that generates all the required control signals to execute the instruction set correctly.



Control Memory (Control Storage): It is Memory unit in the micro-programmed control unit to store the micro-program. Each word in control memory contains within it a microinstruction.

Control Address Register: It specifies the address of the microinstruction in control memory.

Control Data Register: It holds the microinstruction read from memory.

The location of the next microinstruction may be the one next in sequence, or it may be locate somewhere else in the control memory. This reason it is necessary to use some bits of the present microinstruction to control the generation of the address of the next microinstruction.

The next address may also be a function of external input conditions. While the microoperations are being executed, the next address is computed in the next address generator circuit and then transferred into the control address register to read the next microinstruction.

Sequencing Word: Information needed to decide the next microinstruction address.

Next Address Generator (Sequencer or Microprogram Sequencer): It determines the microinstruction Address to be executed in the next clock cycle.

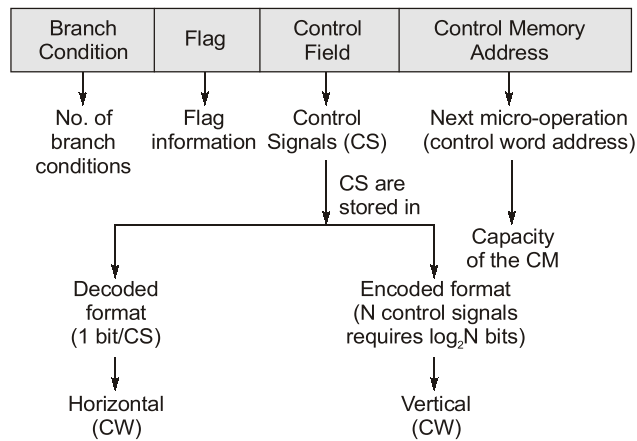
The address of the next microinstruction can be specified several ways, depending on the sequencer inputs as follows:

- (i) Incrementing the control address register by one,
- (ii) Loading into the control address register by an address from control memory,
- (iii) Transferring an external address, or
- (iv) Loading an initial address to start the control operations.

Determining the address of the next microinstruction depends on one of the following:

- In-line Sequencing
- Branch
- Conditional Branch
- Subroutine
- Loop
- Instruction op-code mapping

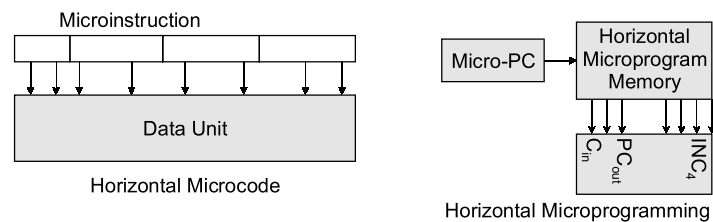
Control word: The control variables at any given time can be represented by a string of 1's and 0's called a control word. It has all the control information required for one clock cycle. Which can be programmed to perform various operations on the component of the system.



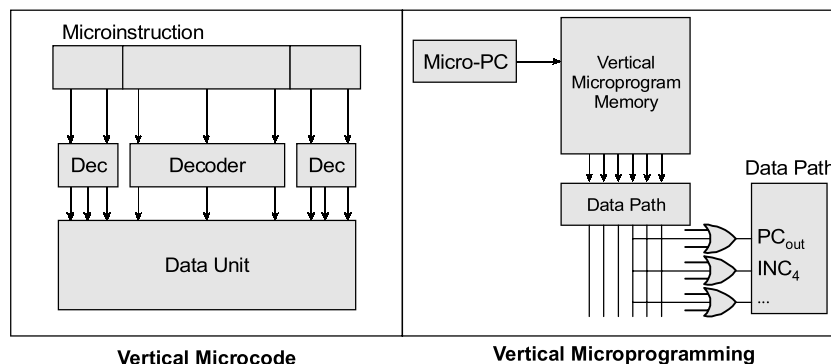
1.7.3 Types of Microinstructions

1. Horizontal Microprogramming:

- There are no intermediate decoders and the control word bits are directly connected to their destination.
- Each bit in the control word is directly connected to some control signal.
- The total number of bits in the control word is equal to the total number of control signals in the CPU.
- Each Microinstruction specifies many different microoperations to be performed in parallel.
- All control signals directly in micro-code
- Due to lot of signals, many bits in micro-instruction.



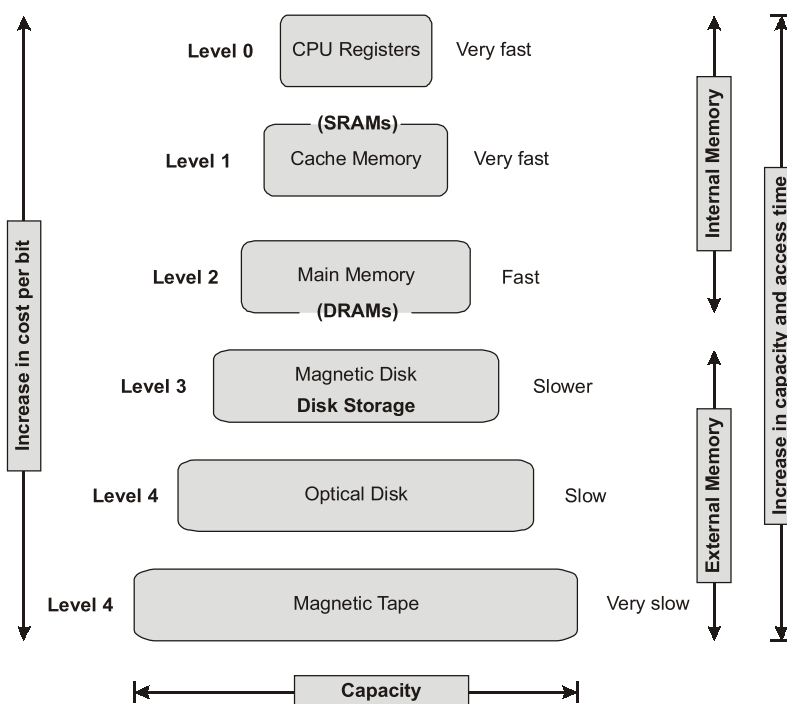
2. Vertical Microprogramming



- Vertical microcode schemes employ an extra level of decoding to reduce the control word width.
- From an n bit control word we may have 2^n bit signal values.
- It takes less space but may be slower
- Actions need to be decoded to signals at execution time
- Each Microinstruction specifies single or few microoperations to be performed.

1.8 Main Memory Organisation

The memory hierarchy was developed based on a program behavior known as locality of references. Memory references are generated by the CPU for either instruction or data access. These accesses tend to be clustered in certain regions in time, space, and ordering.



1.8.1 Types of Memory based on Access

1. **Serial Access Memory:** The system must search the storage device from the beginning of the memory address until it finds the required piece of data. Memory device which supports such access is called a Sequential Access Memory or Serial Access Memory.
Example: Magnetic tape.
2. **Direct Access Memory:** Direct access memory or Random Access Memory, refers to condition in which a system can go directly to the information that the user wants. Memory device which supports such access is called a Direct Access Memory.
Example: Magnetic disk and optical disks.

1.8.2 Memory Access Methods

1. **Sequential Access:** In this method, the memory is accessed in a specific linear sequential manner. For example, if fourth record (collection of data) stored in a sequential access memory needs to be accessed, the first three records must be skipped. Thus, the access time in this type of memory depends on the location of the data. Magnetic disks, magnetic tapes and optical memories the CD-ROM use this method.

2. **Random Access:** In this mode of access, any location of the memory can be accessed randomly. In other words, the access to any location is not related with its physical location and is independent of other locations. For random access, a separate mechanism is therefore each location. Semiconductor memories (RAM, ROM) are this type.
3. **Direct Access:** This method is basically the combination of previous two methods. Memory devices such as magnetic hard disks contain many rotating storage tracks. If each track has its own read/write head, the tracks can be accessed randomly, but access within each track is sequential. In this case the access is semi-random or direct. The access time depends on both the memory organization and the characteristic of storage technology.
4. **Associative Access:** This is a special type of random access method that enables one to make a comparison of desired bit locations within a word for a specific match and to do this for all words simultaneously. Thus, based on a portion of a word's content, word is retrieved rather than its address. Cache memory uses this type of access mode.

Memory or Primary Memory (Core Memory/Store/Storage)

The memory stores the instructions and data for an executing program. Memory is characterized by the smallest addressable unit as one of the following.

- **Byte addressable:** Smallest unit is an 8-bit byte.
- **Word addressable:** Smallest unit is a word, usually 16 or 32 bits in length.

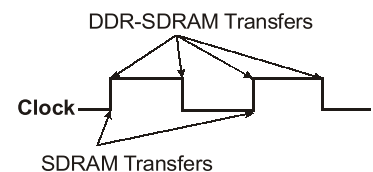
Most modern computers are byte addressable, facilitating access to character data. Logically, computer memory should be considered as an array. The index into this array is called the **address** or “**memory address**”.

There are two types of primary memories: RAM and ROM

1.8.3 Random Access Memory (RAM)

RAM is Read/write memory, Random access, and data is temporarily stored. RAM is further classified into two types:

1. **DRAM (Dynamic Random Access Memory):** It tends to lose its contents, even when powered. Special “refresh circuitry” must be provided.
SDRAM (Synchronous DRAM): It is DRAM that is designed to work with a **Synchronous Bus**, one with a clock signal. The memory bus clock is driven by the CPU system clock, but it is always slower.
 In **SDRAM**, the memory transfers take place on a timing dictated by the memory bus clock rate. This memory bus clock is always based on the system clock. In “plain” SDRAM, the transfers all take place on the rising edge of the memory bus clock. In **DDR SDRAM** (Double Data Rate Synchronous DRAM), the transfers take place on both the rising and falling clock edges.
 “Plain” SDRAM makes a transfer every cycle of the memory bus. DDR-SDRAM makes two transfers for every cycle of the memory bus, one on the rising edge of the clock cycle and another one on the falling edge of the clock cycle.
2. **SRAM (Static Random Access Memory):** It will keep its contents as long as it is powered. Compared to DRAM, SRAM is faster, more expensive, physically larger (fewer memory bits per square millimeter).



1.8.4 Read Only Memory (ROM)

ROM is Read only memory, Random access, and Data is permanently stored. ROM is further classified as following :

1. **MROM (Masked ROM):** The contents of the memory are set at manufacture and cannot be changed without destroying the chip.
2. **PROM (Programmable ROM):** The contents of the chip are set by a special device called a "PROM Programmer". Once programmed the contents are fixed.
3. **EPROM (Erasable and Programmable ROM):** It is same as a PROM, but that the contents can be erased using UV light and reprogrammed by the PROM Programmer.
4. **EEPROM (Electrically EPROM):** The contents can be erased electrically and reprogrammed by the PROM Programmer.

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-Access Memory (RAM)	Read-Write Memory	Electrically, byte-level	Electrically	Volatile
Read-Only Memory (ROM)	Read-Only Memory	Not possible	Masks	Non-volatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-Mostly Memory	UV light, Chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

1.8.5 RAM Vs ROM

	RAM	ROM
Accessibility	The information stored in the RAM is easily accessed because it communicates directly with the processor	The processor cannot directly access the information. Hence, the information will be transferred into the RAM and then it gets executed by the processor to access the ROM information.
Volatility	Volatile in nature, Data is stored as long as the power supply is switched on. Data will be erased if the computer crashes or is turned off.	Non-volatile in nature, Data is stored even the power supply is switched off. Data is retained even if the computer crashes or is turned off.
Storage	Data is temporary It is only there as long as the computer is on and it can be changed	Data is permanent It can never be changed Contents are remain same
Speed	The accessing speed of RAM is faster, it assist the processor to boost up the speed	Speed is slower compared to RAM, ROM cannot boost up the processor speed
Data Preserving	Electricity supply is needed in RAM to flow to preserving information	Electricity supply is not needed in ROM to flow to preserving information
Structure	The RAM is an chip, which is in the rectangle form and is inserted over the mother board of the computer	ROMs are generally the optical drivers, which are made of magnetic tapes.
Cost	The price of RAMs are comparatively high	The price of ROM's are comparatively low
Chip size	Physically size of RAM chip is larger than ROM chip	Physically size of ROM chip is smaller than RAM chip.
Category	Read-write memory Data can be written to or read from.	Read-only memory Data can only be read User cannot make any changes to the information

Let the main memory contains n blocks (which require $\log_2(n)$ bits of physical address) and cache contains m blocks. A given memory block can be mapped into one and only cache line (block). Here, n/m different blocks of memory can be mapped (at different times) to a cache block. Each cache block has a tag saying which block of memory is currently present in it, each cache block also contain a valid bit to ensure whether a memory block is in the cache block currently.

- Number of bits in the tag = $\log(n/m)$
- Number of sets (or blocks) in the Cache = m
- Number of bits to identify the correct Block = $\log(m)$
- Number of Sets in cache = Number of Blocks in cache
- Each set contain only one block, so in direct cache mapping set is also called as block.
- INDEX is used to select the memory block.
- TAG is used to select the cache block from main memory set
- Select location within block using block offset.
- TAG + INDEX = Block Address

Example - 1.1

A 32 bits byte address Direct-mapped cache defined as:

Cache size = 2^n block, n bits used for index

Block size = 2^m block, m bits for word between block, 2 used for byte part of address

Size of tag field : $32 - (m + n + 2)$

One valid bit field is used in cache.

Find the direct mapped cache size (in bits).

Solution:

Total number of bits in direct-mapped cache = $2^n \times (\text{block size} + \text{tag size} + \text{valid field size})$

Block size = 2^m words

1 bit valid field is needed.

$$\begin{aligned}\text{Cache size} &= 2^n \times (2^m \times 32 + 32 - n - m - 2) + 1 \text{ bits} \\ &= 2^n \times (2^m \times 32 + 31 - n - m) \text{ bits}\end{aligned}$$

Example - 1.2

How many total bits are required for a direct-mapped cache with 16 kB of data, 1 bit field for valid and 4-word blocks, assuming a 32-bit address?

Solution:

16 kB = 2^{12} words of data, 4 words block size (= 2^2).

So, there are 1024 blocks (= 2^{10})

$$\text{Tag} = 32 - 10 - 2 - 2 \text{ bits}$$

Cache Entry size = 128 bits of data + tag bits + valid bit.

$$\begin{aligned}\text{Cache size} &= 2^{10} \times (4 \times 32 + (32 - 10 - 2 - 2) + 1) \\ &= 2^{10} \times 147 = 147 \text{ K bits}\end{aligned}$$

1.9.4 Fully Associative Cache

- It is also called as m -way set associative cache, where m is number of blocks and all blocks fit into one set.
- Instead of using a cache index, compare the tags of all cache entries in parallel
- Because no bit field in the address specifies a line number the cache size is not determined by the address size.

Solution:

1	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
1	1	1	1	3	2	1	5	2	1	6	2	5	6	6	1	3	6	1	2
	2	2	3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4
		3	2	1	5	2	1	6	2	5	6	3	1	3	6	1	2	4	3
x	x	x			x			x		x		x	x				x	x	x

Number of page faults = 11.

Summary

- **Computer:** A device that accepts input, processes data, stores data, and produces output, all according to a series of stored instructions.
- **Hardware:** Includes the electronic and mechanical devices that process the data; refers to the computer as well as peripheral devices.
- **Software:** A computer program that tells the computer how to perform particular tasks.
- **Computer organization:** Interconnection of hardware to form the computer system
- **Computer architecture:** the structure and behaviour of the computer perceived by the user.
- **Input:** The data or raw information entered into a computer.
- **Data:** Refers to the symbols that represent facts, objects, or ideas.
- **Information:** The results of the computer storing data as bits and bytes; the words, numbers, sounds, and graphics.
- **Output:** Consists of the processing results produced by a computer.
- **Main Memory:** Area of the computer that temporarily holds data waiting to be processed, stored, or output. *Example:* Cache and Main memory
- **Secondary Storage:** Area of the computer that holds data on a permanent basis when it is not immediately needed for processing. *Example:* Disk, Floppy, etc.

**Student's
Assignments**

- Q.1** What does CISC and RISC means?
- common instruction set controller and rare instruction set controller
 - complex instruction set controller and reduced instruction set controller
 - compiled instruction set source code and recompiled instruction source code
 - none of the above
- Q.2** A 32-bit address bus allows access to a memory of capacity
- 64 Mb
 - 16 Mb
 - 1 Gb
 - 4 Gb
- Q.3** The system bus is made up of
- data bus
 - data bus and address bus
 - data bus and control bus
 - data bus, control bus and address bus
- Q.4** Which of the following is not involved in a memory write operation?
- MAR
 - PC
 - MDR
 - data bus
- Q.5** The read/write line
- belongs to the data bus
 - belongs to the control bus
 - belongs to the address bus
 - CPU bus

- Q.6** _____ is a piece of hardware that executes a set of machine-language instructions.
(a) controller (b) bus
(c) processor (d) motherboard
- Q.7** Given below are some statements associated with the registers of a CPU. Identify the false statement.
(a) The program counter holds the memory address of the instruction in execution.
(b) Only opcode is transferred to the control unit.
(c) An instruction in the instruction register consists of the opcode and the operand.
(d) The value of the program counter is incremented by 1 once its value has been read to the memory address register.
- Q.8** In Flynn's classification of computers, the vector and array classes of machines belong to
(a) Single instruction/single data category
(b) Single instruction/multiple data category
(c) Multiple instruction/single data category
(d) Multiple instruction/multiple data category
- Q.9** The following are four statements regarding what a CPU with only a set of 32 bit registers can perform.
1. Hold and operate on 32 bit integers
2. Hold and operate on 16 bit integers
3. Hold and operate on 64 bit floating point arithmetic
4. Hold and operate on 16 bit UNICODE characters
Which of the following is true about such a CPU?
(a) all are true (b) 1,2 and 3 only
(c) 1,2 and 4 only (d) 1,3 and 4 only
- Q.10** The following are four statements about Reduced Instruction Set Computer (RISC) architectures.
1. The typical RISC machine instruction set is small, and is usually a subset of a CISC instruction set.
2. No arithmetic or logical instruction can refer to the memory directly.
3. A comparatively large number of user registers are available.
4. Instructions can be easily decoded through hard-wired control units.
Which of the above statements is true?
(a) 1 and 3 only (b) 1,3 and 4 only
(c) 1, 2 and 3 only (d) All of these
- Q.11** The word length of a CPU is defined as
(a) the maximum addressable memory size
(b) the width of a CPU register (integer or float point)
(c) the width of the address bus
(d) the number of general purpose CPU registers
- Q.12** Which of the following statements is false about CISC architectures?
(a) CISC machine instructions may include complex addressing modes, which require many clock cycles to carry out.
(b) CISC control units are typically micro-programmed, allowing the instruction set to be more flexible.
(c) In the CISC instruction set, all arithmetic/logic instructions must be register based.
(d) CISC architectures may perform better in network centric applications than RISC.
- Q.13** Which one is required while establishing the communication link between CPU and peripherals?
(a) synchronization mechanism
(b) conversion of signal values
(c) operating modes
(d) all of the above
- Q.14** What will be average cost per bit for a system with main memory of 1024 cost, 100 units and secondary memory of 4096 cost, 10 units.
(a) 35.7 (b) 28.0
(c) 82.0 (d) insufficient data
- Q.15** Consider a Disk I/O transfer, in which 1500 bytes are to be transferred, but number of bytes on a track is 1000, and rotation speed of disk is 1500 rps but the average time required to move the disk arm to the required track is 15 ms, then what will be total access time?
(a) 16.33 ms (b) 15.33 ms
(c) 16.33 μ s (d) 15.33 μ s
- Q.16** A disc drive has a rotational speed of 3600 rpm, an average seek time of 10 ms, 64 sectors per track and 512 bytes of data per sector. What is the average time to access the entire data of a 16 kbytes file stored sequentially on the disk?
(a) 18.85 ms (b) 10 ms
(c) 27.15 ms (d) 9 ms

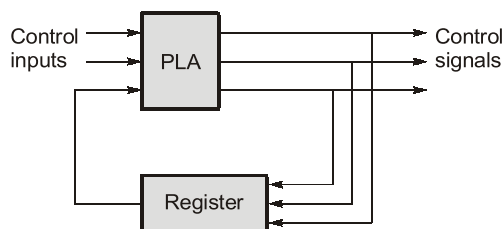
Q.31 Horizontal microinstruction have which of the following attributes?

1. Short formats
 2. Limited ability to express parallel micro operations
 3. Considerable encoding of the control information
- (a) 1 and 2 (b) 2 and 3
(c) 1, 2 and 3 (d) None of these

Q.32 Asynchronous data transfer between two independent units requires that control signals be transmitted between the communicating units to indicate the time at which data is being transmitted, way of achieving this _____.

- (a) Is by means of a strobe pulse from one of the unit
(b) Is by means of handshaking
(c) Both (a) and (b)
(d) None of these

Q.33 Consider the following figure



The above circuit represents _____.

1. Hard-wired control unit
 2. Micro programmed control unit
- (a) Only 1 (b) Only 2
(c) Both 1 and 2 (d) Either 1 or 2

Q.34 Pipelining improves CPU performance due to

- (a) reduced memory access time
- (b) increased clock speed
- (c) the introduction of parallelism
- (d) additional functional units

Q.35 Which of the following statements is false with regard to instruction pipelining?

- (a) The basic goal of instruction pipelining is to achieve a CPI of 1.
- (b) Instruction set architectures having simple registry/register addressing modes can be easily pipelined than those having complex addressing modes.

- (c) By the use of hardware special features and compiler design techniques, pipeline hazards can be removed.
- (d) The basic goal of instruction pipelining is to achieve a CPI of more than 1.

Q.36 Assertion (A): Reduced Instruction Set Computers (RISC) use pipelined control unit.

Reason (R): Pipelining reduces memory requirements of programs.

- (a) Both A and R are true and R is the correct explanation of A
- (b) Both A and R are true but R is NOT the correct explanation of A
- (c) A is true but R is false
- (d) A is false but R is true

Q.37 Match **List-I** with **List-II** and select the correct answer using the codes given below the lists:

List-I

- A. Pipelined ALU
- B. Simpler compiler
- C. Separate data and instruction caches
- D. Lesser cycles per instruction

List-II

1. RISC
2. CISC
3. Mixed
4. Mixed RISC-CISC

Codes:

	A	B	C	D
(a)	3	2	3	1
(b)	1	2	3	3
(c)	3	3	2	1
(d)	3	3	3	1

Answer Key :

1. (d) 2. (d) 3. (d) 4. (b) 5. (b)
6. (c) 7. (a) 8. (b) 9. (c) 10. (d)
11. (b) 12. (c) 13. (d) 14. (b) 15. (a)
16. (c) 17. (b) 18. (d) 19. (b) 20. (c)
21. (c) 22. (a) 23. (c) 24. (c) 25. (a)
26. (b) 27. (a) 28. (d) 29. (a) 30. (d)
31. (d) 32. (c) 33. (c) 34. (c) 35. (d)
36. (a) 37. (a)

◆◆◆◆◆