Electronics Engineering

Microprocessors and Computer Organization

Comprehensive Theory

with Solved Examples and Practice Questions

MADE EASY Publications

MADE EASY Publications

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

Microprocessors and Computer Organization

First Edition: 2015
Second Edition: 2016
Third Edition: 2017
Fourth Edition: 2018

Contents

Microprocessors and Computer Organization

■■■

CHAPTER 1

Introduction to 8085 and Its Functional Organization

1.1 Introduction

The most important technological invention of modern times is the "microprocessor". A microprocessor is a multiple purpose programmable clock driven, register based electronic device that reads binary instructions from memory, accepts binary data as input and processing this data according to the instructions written in the memory. The microprocessor is capable of performing computing functions and making decisions to change the sequence of program execution. The microprocessor can be embedded in a larger system, and can function as the CPU of a computer called a microcomputer.

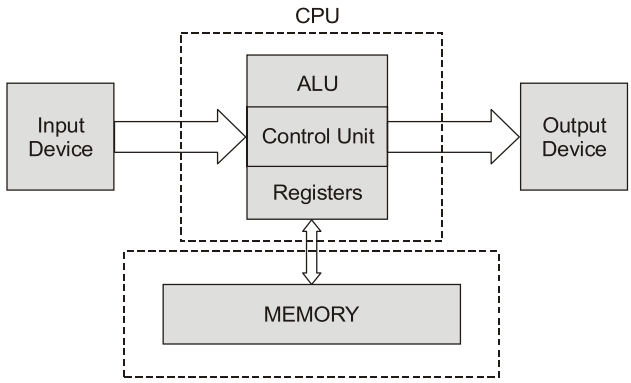


***Figure*-1.1 :** *Block diagram of microcomputer*

The Figure 1.1 shows the basic block diagram of a microcomputer which processes binary data and traditionally represented by four blocks i.e. CPU, memory, input device and output device.

Here, input device is a device that transfers information from outside world to the computer for example: Key board, mouse, webcam, microphone, scanner, electronic white boards, etc. The output device transfers information from computer to the outside world like monitor, printers (all types), speakers, headphones, projector, plotter, Braille embosser, LCD projection panel, computer output microfilm (COM) etc. Memory is an electronic medium that stores binary information.

Central Processing Unit (CPU) is the heart of computer systems. The microprocessors in any microcomputer act as a CPU. The CPU can be made up with ALU + CU + Registers. Where ALU is the group of circuits that performs arithmetic and logical operations. Control Unit (CU) is a group of circuits that provide timings and signals to all the operations in the computer and controls the data flow.

Microcontroller is a programmable device that includes microprocessor, memory and I/O signal lines on a single chip, fabricated using VLSI technology. Microcontrollers are also known as single chip microcomputers. They are mostly used to perform dedicated functions such as automatic control of equipment, machines and process in industries and consumer appliances.

System Bus

A bus is a group of wires/lines used to transfer data (bits) between components inside a computer or between computers. In most simple form, they are communication path used to carry the signals between microprocessor and peripherals.

The system bus of a microprocessor is of three types:

1. **Address Bus**
   - It is a group of lines that are used to send a memory address or a device address from the Microprocessor Unit (MPU) to the memory or the peripheral.
   - The address bus is always uni-directional i.e address always goes out of the microprocessor.
   - If the address line are 'n' for a MPU then its addressing capacity is 2^n.
2. **Data Bus**
   - It is group of lines used to transfer data between the microprocessor and peripherals and/or memory.
   - Data bus is always bi-directional.
3. **Control Bus**
   - Control bus provides signals to control the flow of data.

Do You Know: The internal architecture of the microprocessor unit depends on the data bus width, which is equal to the bit-capacity of the microprocessor.

1.2 History of Microprocessors

A brief review of certain microprocessors were given in the Table 1.1. Intel introduced its first 4-bit PMOS microprocessor 4004 in the year 1971. It has 16 pins, 640-bytes of memory addressing capability and 10 address lines. After this enhanced version of 4004, a 4-bit, Intel 4040 was developed. In 1972, Intel introduced its first 8-bit processor Intel 8008, which also uses PMOS technology. The PMOS technology processors were slow and not compatible with TTL logic. These microprocessors could not survive as general purpose microprocessor due to design limitations. In 1974, Intel introduced its more powerful and faster 8 bit NMOS microprocessor Intel 8080. These processors were faster and compatible with TTL logic. Intel 8085 followed 8080 microprocessor. The main limitations of 8 bit microprocessors tempted the designers to go for more powerful processors in terms of advanced architecture, more processing capability, larger memory addressing capability and more powerful instruction set. The Intel 8086 was the result, launched in 1978 . The technology used was HMOS, high speed and high performance MOS technology.

Microprocessor	Word length	Memory capacity
Intel 4004 (PMOS)	4-bit	640 B
Intel 8008	8-bit	16 kB
Intel 8080 (NMOS)	8-bit	64 kB
Intel 8085 (NMOS)	8-bit	64 kB
Intel 8086 (HMOS)	16-bit	1 MB
Intel 8088	8/16-bit	1 MB
Intel 80186	16-bit	1 MB
Intel 80286	16-bit	16 MB real, 4 GB virtual
Intel 80386	32-bit	4 GB real, 4 GB virtual
Intel 80486	32-bit	4 GB real, 64 TB virtual
Pentium-II	64-bit	64 GB real
Z-80	8-bit	64 kB
Z-800	8-bit	500 kB

Table 1.1 : *A brief review of various microprocessors*

NOTE: Most of the general purpose microprocessors used in the modern world computers are the family of 8086.

1.3 Computer Language

- Scale of integration:
  - **SSI (Small Scale Integration):** The term refers to the technology used to fabricate discrete logic gates on a chip.
  - **MSI (Medium Scale Integration):** The process of designing few tens of gates on a single chip.
  - **LSI (Large Scale Integration):** The process of designing hundreds of gates on a single chip similarly terms VLSI (very large scale integration), ULSI (ultra large scale integration) are used to indicate the scale of integration.
- **Digital computer:** A programmable machine that process binary data. It is traditionally represented by five components: CPU, ALU, CU, memory, input and output.
- **Instruction:** a command in binary that is recognized and executed by the computer in order to accomplish a task. Some instructions are designed with one word, and some require multiple words.
- **Mnemonic:** a combination of letters to suggest the operation of an instruction.
- **Program:** a set of instructions written in a specific sequence for the computer to accomplish a given task.
- **Machine Language:** the binary medium of communication with a computer through a designed set of instructions specific to each computer.
- **Assembly Language:** a medium of communication with a computer in which programs are written in mnemonics. An assembly language is specific to a given computer.
- **Low-Level Language:** a medium of communication that is machine-dependent or specific to a given computer. The machine and the assembly languages of a computer are considered low-level languages. Programs written in these languages are not transferrable to different types of machines.
- **High-Level Language:** a medium of communication that is independent of a given computer. Programs are written in English-like words, and they can be executed on a machine using a written translator (a compiler or an interpreter).
- **Compiler:** a program that translates English-like words of a high-level language into the machine language of a computer. A compiler reads a given program, called a source code, in its entirety, and then translates the program into the machine language which is called an object code. (Ex. *C*, *C*++)
- **Interpreter:** a program that translates the English-like statements of a high-level language into the machine language of a computer. An interpreter translates one statement at a time from a source code to an object code. (Ex. BASIC)
- **Assembler:** a computer program that translates an assembly language program from mnemonics to the binary machine code of a computer and these machine codes are called object programme .

  Difference between compiler and interpreter: Interpreter reads one line at a time, converts it into object code, executes and then reads next line. Whereas compiler reads whole program at a time and convert it into the object code and then execute.
- **Bit:** a binary digit, 0 or 1.
- **Byte:** a group of eight bits.
- **Nibble:** a group of four bits.
- **Word:** a group of byte the computer recognizes and processes at a time.

Example - 1.1 Machine instructions are written using which of the following?

(a) Bits 0 and 1 only
(b) Digits 0 and 9 only
(c) Digits 0 and 9 and the capital alphabets *A* to *Z* only
(d) Digits 0 to 9, the capital alphabets *A* to *Z* and certain special characters

Solution : (a)

Machine instructions are written using bits 0 and 1 only.

Example - 1.2 Output of the assembler in machine code is referred to as

(a) Object program
(b) Source program
(c) Macroinstruction
(d) Symbolic addressing

Solution : (a)

Output of the assembler in machine code is referred to as object program.

Example - 1.3 Which one of the following statements is correct?

A micro-controller differs from a microprocessor in that it has

(a) Both on-chip memory and on-chip ports
(b) Only on-chip memory but not on-chip ports
(c) Only on-chip ports but not on-chip memory
(d) Neither on-chip memory nor on-chip ports

Solution : (a)

A micro-controller differs from a microprocessor in that has both on-chip memory and on-chip ports.

Example - 1.4 Assertion (A): Many programmers prefer assembly level programming to machine language programming.

Reason (R): It is possible to efficiently utilize the hardware of the computer in machine language programming.

(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 a correct explanation of A.
(c) A is true but R is false.
(d) A is false but R is true.

Solution : (b)

Many programmers prefer assembly level programing to machine language programming because assembly language is simple and easily understandable. So assertion is true. Also it is possible to efficiently utilize the hardware of the computer in machine language programming because the machine language is directly understood by microprocessor.

Application of Microprocessors

A few more applications of microprocessors are mentioned below:

- A microprocessor based stepping motor controller used for controlling several stepping motors in a pulsed Laser system. The motors are used to precisely align a set of mirrors used in this system.
- There are several other motor control applications reported in the literature, Lin (1977) describes one approach to motor speed control using an SCR chopper.

- A microprocessor controlled Railways Signalling Inter lock was developed to exhibit the applications of microprocessors in signalling. The system mirrors train positions in different blocks on a section and sends speed codes to each block. The speed codes are displayed and used by train drivers to control the speed.
- A patient surveillance system was designed using distributed processing.
- Microprocessors have been used in a variety of automation applications. Control of tester for surveillance checking the electronic functioning capability of a target detecting device (Frantz, 1977) is one of these. A microprocessor based blood gas analyzer has been developed by Margalith et al. (1977).

1.4 Microprocessor Architecture

The process of data manipulation and communication is determined by the logic design of microprocessor, called the "Architecture". There are two types of architecture depending upon storage of program and data in memory:

- Von Neumann architecture of computers
- Harvard architecture of computers

Von Neumann Architecture

The idea of basic organization of a digital computer containing a CPU, a main memory, input and output device and secondary storage devices was given by John von Neumann in 1945. He introduced the "stored – program concept"-where the programs and data are stored in the same high speed memory unit. In Von Neumann architecture there is a program counter and instructions are executed in sequential manner. The MPU fetches one instruction of the program and executes it, then it goes to the next instruction. The speed of computer is limited by the speed at which the MPU can fetch the instructions and data from the memory and process them. Digital computers based on this principle are called control-flow or control driven computers.

Examples: Intel 8085 and Intel 8086

Harvard Architecture

The enhanced version of Von Neumann architecture is the Harvard architecture. It contains separate instruction memory and data memory. The instruction memory and data memory in Harvard architecture have separate data path , that eliminated the speed limitation of single bus architecture in a Von Neumann processor.

Examples: TMS 32010, Intel 8051, Intel's Pentium. etc.

1.5 The 8085 Microprocessor Pinout and Signals

The 8085A (8085) is an 8-bit microprocessor. The device has 40 pins, requires a +5 V single power supply, and can operate with a 3 MHz single-phase clock frequency. The 8085 is an advance version of 8080A. Its instruction set is compatible with that of the 8080A, it means that 8085A instruction set includes all the instruction of 8080A with some additional instructions.

Figure 1.2 (a) and (b) shows the 8085 pinout and simplified pinout of 8085 respectively.

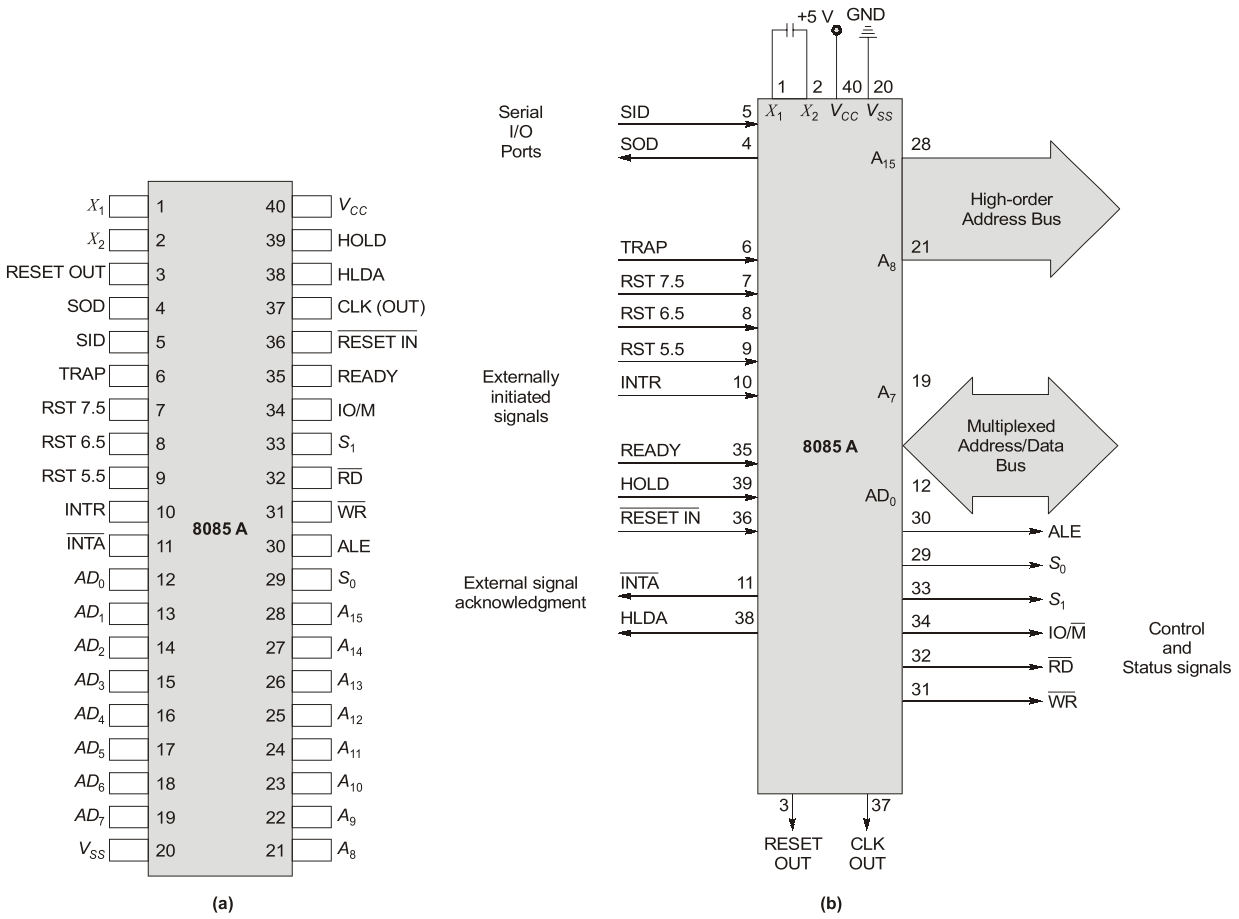


***Figure*-1.2 :** 8085 Pinout

Key Points of Microprocessors 8085

- It is manufactured from NMOS technology.
- It is upward compatible with 8080A.
- It is a 40 pin DIP (Dual in line Package) chip.
- It is a 8-bit processor.
- It has total 16 address lines with addressing capacity of 64 kB.
- It has 8 data bus lines which is the bit capacity of the microprocessor.
- Internal architecture of the 8085 depends on the bit capacity.
- Serial data transfer facility is provided by 8085 MPU.
- Low order address bus (AD_0 – AD_7) is multiplexed with data bus.
- High order address bus is not multiplexed with any other lines.
- Advantage of multiplexing lower order address with data lines is that the number of pins are reduced.
- To de-multiplex address from data ALE (Address Latch Enable) signal is used.
  ALE = 1, Address transfer to bus.
  ALE = 0, Data transfer to bus.
- Disadvantage of multiplexing is that speed will be reduced.
- It has on chip clock generation facility.

- It requires +5 V power supply for its operation.
- Only one ground pin is present.
- There are five hardware interrupts available for 8085.
- The crystal frequency of processor is 6 MHz and the clock frequency is 3.07 MHz (~3 MHz), which is approximately half the crystal frequency.
- The word length or bit capacity is 8.
- 8085 has 74 basic instructions with 246 opcodes.

Signals of 8085 Microprocessors

According to the above figure all the signals can be classified into six groups:

1. Address Bus signals
2. Data Bus signals
3. Control and Status signals
4. Power supply and frequency signals
5. Serial I/O ports
6. Externally initiated signals

Address Bus/Data Bus Signals

Address Bus Signals:

- Control pins: Pin 21 to 28.
- It is 16 bits of length.
- It is unidirectional bus.
- It is divided into two parts namely,
  Lower order address bus (AD_0 - AD_7) → is also called "Line number".
  Higher order address bus (A_8 - A_{15}) → is also called "Page Number".

Multiplexed Address/Data Bus Signals:

- Control pins: Pin 12 to 19.
- Its length is in 8-bit.
- It is a bidirectional bus.
- It is multiplexed with lower order address bus with lines (AD_0 - AD_7).
- To reducing the number of pins in microprocessor, databus is "Time Division Multiplexed" with address bus.

Control and Status Signals

Microprocessor 8085 has two control signals $\overline{RD}$ and $\overline{WR}$, three status signals IO/$\overline{M}$, S_1 and S_0 and one special purpose signal ALE.

- Control pins: pin31 and pin32
- Control signals: $\overline{WR}$ and $\overline{RD}$
- Status pins: pin34, pin33 and pin29
- Status signals: IO/$\overline{M}$, S_1 and S_0

$\overline{RD}$ (Read): It is an active low signal. When the signal is low on this pin, the microprocessor performs memory reading or I/O reading operation.

$\overline{WR}$ (Write): It is an active low signal. When the signal is low on this pin, the microprocessor performs memory writing or I/O writing operation.

IO/$\overline{M}$:

- This is the status signal used to differentiate between I/O and memory operations.
- When it is **HIGH** → an I/O operation performed.
- When it is **LOW** → a memory operation performed.

IO/$\overline{M}$	$\overline{RD}$	$\overline{WR}$	Description
0	0	1	Memory Read ($\overline{MEMR}$)
0	1	0	Memory Write ($\overline{MEMW}$)
1	0	1	IO Read ($\overline{IOR}$)
1	1	0	IO Write ($\overline{IOW}$)

***Table*-1.2 :** *Memory or IO operations based on Control Signals*

S_1 and S_0: These two status signals, similar to IO/$\overline{M}$, can identify various operations based on the combinations of S_1 and S_0.

S_1	S_0	Microprocessor Operation
0	0	Halt (no operation)
0	1	Write operation
1	0	Read operation
1	1	Opcode fetch (Reading instruction)

***Table*-1.3 :** *Processor operation based on status pins S_1 and S_0*

Machine Cycle	Status			Control signals
	IO/$\overline{M}$	S_1	S_0	
Opcode Fetch	0	1	1	$\overline{RD}$ = 0
Memory Read	0	1	0	$\overline{RD}$ = 0
Memory Write	0	0	1	$\overline{WR}$ = 0
I/O Read	1	1	0	$\overline{RD}$ = 0
I/O Write	1	0	1	$\overline{WR}$ = 0
Interrupt Acknowledge	1	1	1	$\overline{INTA}$ = 0
Halt	X	0	0	

***Table*-1.4**

ALE (Address Latch Enable): It is a special signal used to demultiplex the address bus and data bus. This is a positive going pulse generated every time the processor begins an operation (machine cycle) to latch the low-order address from the multiplexed bus and generate a separate set of eight address lines A_7 to A_0.

CHAPTER 7

Computer Organization

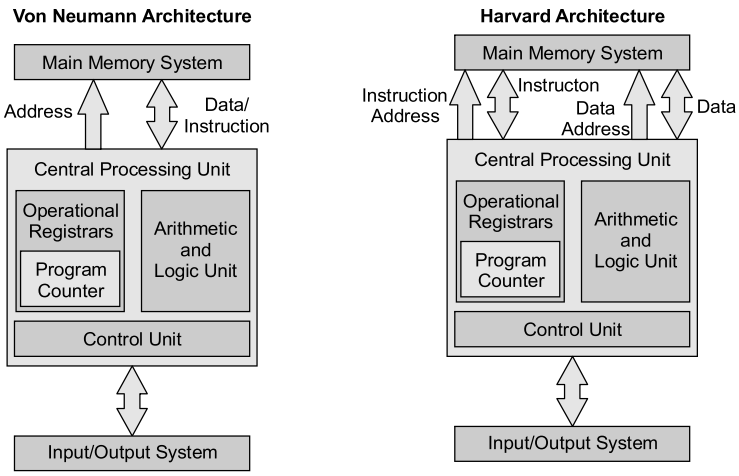
7.1 Computer Architecture Vs Computer Organization

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
• Computer organization deals with structural relationships that are not visible to the programmer (like clock frequency or the size of the physical memory).	• 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).
• A computer's organization expresses the realization of the architecture.	• A computer's architecture is its abstract model and is the programmer's view in terms of instructions, addressing modes and registers.
• Organization describes how it does it.	• Architecture describes what the computer does.

Von Neumann Architecture Vs Harvard Architecture



7.2 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.

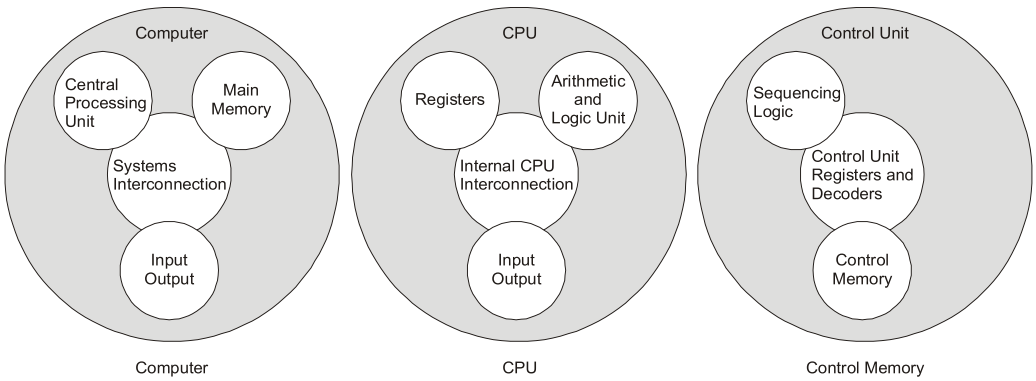
Fourth generation: Very large scale integrated (VLSI) computers (1980~2000)

- 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.

7.3 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.

7.4 CISC and RISC Architectures

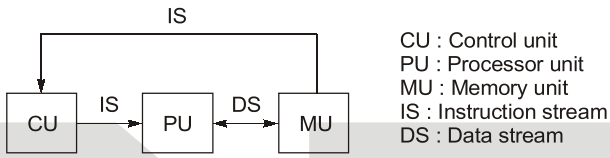
CISC (Complex Instruction Set Computers)	RISC (Reduced Instruction Set Computers)
• Large instruction set	• Compact instruction set
• Instruction formats are of different lengths	• Instruction formats are all of the same length
• Instructions perform both elementary and complex operations	• Instructions perform elementary operations
• Control unit is micro□ ĵrogrammed	• 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: • VAX • PDP?11 • Motorola 68000 family • Intel x86 architecture based processors.	**Examples** of RISC processors • Apple iPods (custom ARM7TDMI SoC) • Apple iPhone (Samsung ARM1176JZF) • Nintendo Game Boy Advance (ARM7) • Sony Network Walkman (Sony in?house ARM based chip)

7.5 Flynn's Classification of Processors

7.5.1 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.

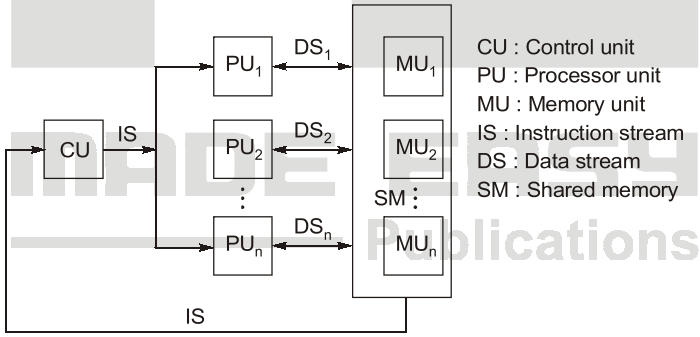
In such a computer a single stream of instructions and a single stream of data are accessed by the processing elements from the main memory, processed and the results are stored back in the main memory. SISD computer organization is shown in figure below.



7.5.2 Single Instruction Stream, Multiple Data Stream (SIMD)

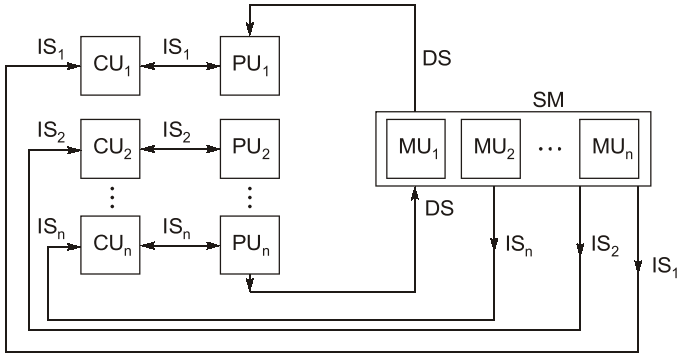
It represents an organization of computer which has multiple processors under the supervision of a common control unit. All processors receive the same instruction from the control unit but operate on different items of the data. SIMD computers are used to solve many problems in science which require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



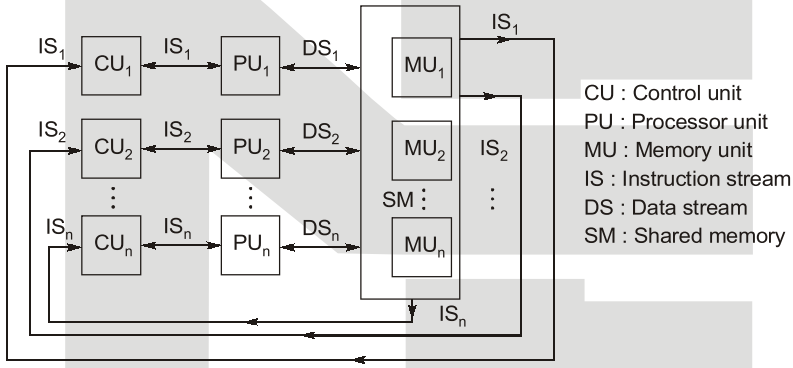
7.5.3 Multiple Instruction Stream, Single Data Stream (MISD)

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.



7.5.4 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.]



7.6 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.

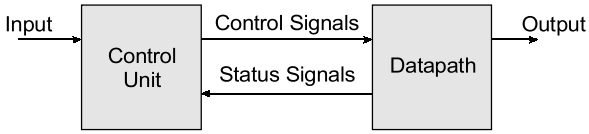
To execute an instruction, the control unit of the CPU must generate the required control signal in the proper sequence.

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

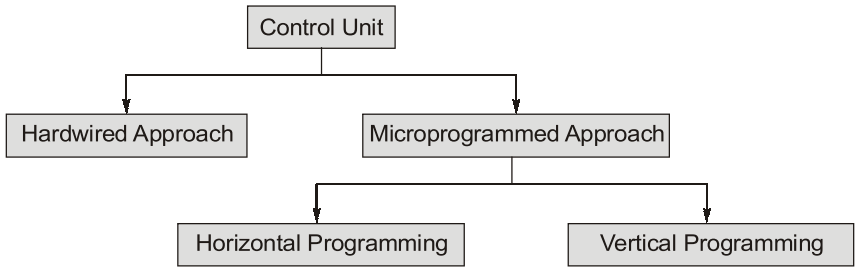


Control Unit: It generates the signals for sequencing the micro-operations in the datapath based on status and input signals.

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

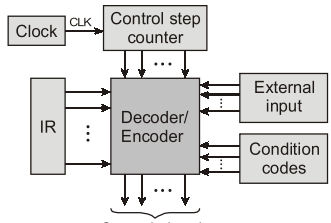
7.7 Control Unit Implementation

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.



7.7.1 Hardwire Control Unit

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



Control Unit Organization

Decoder/Encoder Block

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

(*i*) **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.

(*ii*) **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.

(*ii*) **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.

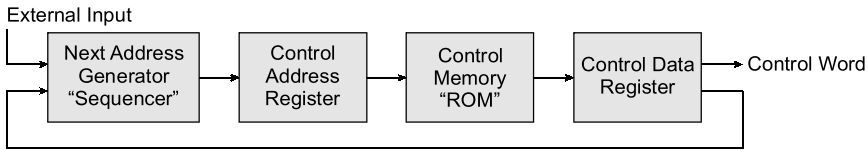
The advantage of hardwired control is that is very fast. The disadvantage is that the instruction set and the control logic are directly tied together by special circuits that are complex and difficult to design or modify.

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

7.7.2 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.

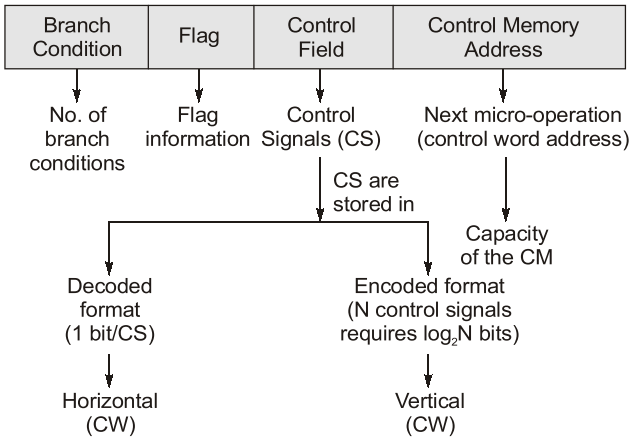
The address of the next microinstruction can be specified several ways, depending on the sequencer inputs as follows:

(*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

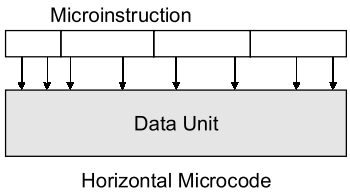
Control word: The control variables at any given time can be represented by a string of 1's and 0's called a control word. It has all the control information required for one clock cycle. Which can be programmed to perform various operations on the component of the system.



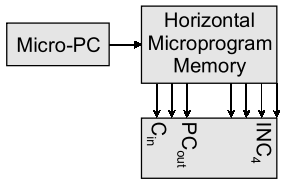
7.7.3 Types of Microinstructions

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.

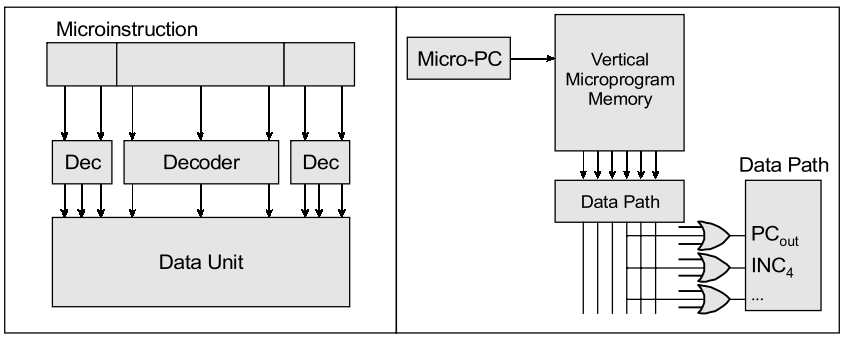


Horizontal Microcode



Horizontal Microprogramming

2. Vertical Microprogramming



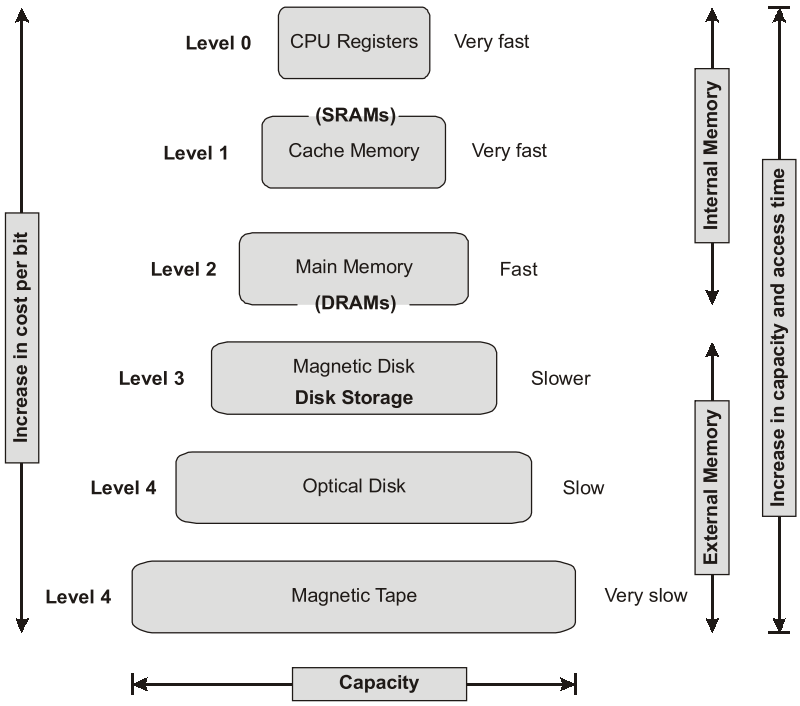
Vertical Microcode

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.

7.8 Main Memory Orgranisation

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.



7.8.1 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.

7.8.2 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.

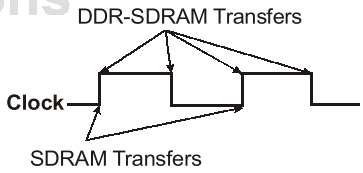
Most modern computers are byte addressable, facilitating access to character data. Logically, computer memory should be considered as an array. The index into this array is called the **address** or **"memory address"**.

There are two types of primary memories: RAM and ROM

7.8.3 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).

7.8.4 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.

<table>
<tr><th>Memory Type</th><th>Category</th><th>Erasure</th><th>Write Mechanism</th><th>Volatility</th></tr>
<tr><td>Random-Access Memory (RAM)</td><td>Read-Write Memory</td><td>Electrically, byte-level</td><td>Electrically</td><td>Volatile</td></tr>
<tr><td>Read-Only Memory (ROM)</td><td rowspan="2">Read-Only Memory</td><td rowspan="2">Not possible</td><td>Masks</td><td rowspan="4">Non-volatile</td></tr>
<tr><td>Programmable ROM (PROM)</td><td rowspan="3">Electrically</td></tr>
<tr><td>Erasable PROM (EPROM)</td><td rowspan="3">Read-Mostly Memory</td><td>UV light, Chip-level</td></tr>
<tr><td>Electrically Erasable PROM (EEPROM)</td><td>Electrically, byte-level</td></tr>
<tr><td>Flash memory</td><td>Electrically, block-level</td></tr>
</table>

7.8.5 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 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 retain 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

7.8.6 Characteristic of Memory

- **Capacity:** It is the global volume of information (in bits) that the memory can store.
- **Access time:** It is the time interval between the read/write request and the availability of the data.
- **Cycle time:** It is the minimum time interval between two successive accesses.
- **Throughput:** It is the volume of information exchanged per unit of time, expressed in bits per second.
- **Non-volatility:** It characterizes the ability of a memory to store data when it is not being supplied with electricity.

7.9 Associative Memory

- It is also known as content addressable memory(CAM) or associative storage or associative array.
- It is a special type of memory that is optimized for performing searches through data, as opposed to providing a simple direct access to the data based on an address.
- It is a hardware search engines, a special type of computer memory used in certain very high searching applications.
- It is composed of conventional semiconductor memory (usually SRAM) with added comparison circuitry that enable a search operation to complete in a single clock cycle.

Where is associative memory used?

We are only using this associative memory in memory allocation format and it is widely used in database management systems, etc.

Advantages of Associative Memory: This is suitable for parallel searches. It is also used where search time needs to be short. Associative memory is often used to speed up databases, in neural networks and in the page tables used by the virtual memory of modern computers.

Disadvantages of Associative Memory: It is expensive than RAM, as each cell must have storage capability and logical circuits for matching its content with external argument.

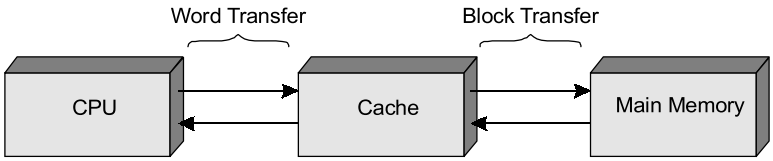
Associative Memory Vs Random Access Memory (RAM)

- In RAM, the user supplies a memory address and the RAM returns the data word stored at that address.
- In associative memory, the user supplies a data word and the associative memory searches its entire memory to see if that data word is stored anywhere in it.
- If the data word is found, the associative memory returns a list of one or more storage addresses where the word was found.
- Hardware of associative memory allows operations to occur in a single-clock cycle, as opposed to the much greater time required for an algorithmic based search through traditional RAM.
- It is a special type of memory that is optimized for performing searches through data, as opposed to providing a simple direct access to the data based on an address as in RAM.

7.9.1 Cache Memory

Cache is a random access memory used by the central processing unit (CPU) to reduce the average time to access memory. Cache memory stores instructions that are repeatedly required to run programs, for improving overall system speed as cache memory is designed to accelerate the memory function. The cache is a smaller, faster memory which stores copies of the data from the most frequently used main memory locations.

The advantage of cache memory is that the CPU does not have to use the motherboard 's system bus for data transfer, enabling process of data transfer is processed much more faster by avoiding bottleneck created by system bus.



When the computer read from or write to a location in main memory, it first checks whether a copy of that data is in the cache. If so, the processor immediately reads from or writes to the cache, which is much faster than reading from or writing to main memory.

- **Principle of Locality:** Program access a relatively small portion of the address space at any instant of time.
  (a) *Temporal Locality (Locality in Time):* If an item is referenced, it will tend to be referenced again soon.
  (b) *Spatial Locality (Locality in Space):* If an item is referenced, items whose addresses are close by tend to be referenced soon.
- **Cache Hit:** If data is present in the cache then it is called as cache hit.
- **Hit Rate:** It is the fraction of memory access found in the cache.
- **Hit Time:** It is the time to access the cache which consists of Cache access time and time to determine hit.
- **Cache Miss:** If data is not present in the cache then it is called as cache miss.
  Miss Rate = 1 – (Hit Rate)
  Miss Penalty = Time to replace a block in the cache + Time to deliver the block to the processor.
- **Average Access Time:** = Hit Time x (1 – Miss Rate) + Miss Penalty x Miss Rate

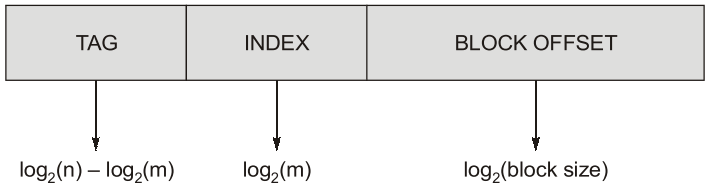
7.9.2 Elements of Cache Design

- Cache Size: It is the amount of main memory data that cache can hold.
- Block Size: It is the number of words (bytes) grouped into a single unit called a block.
- Mapping Function: Direct mapped, Associative and Set associative mapping.
- Replacement Algorithm: Lease Recently Used Algorithm.
- Write Policy: Write Through and Write Back.

There are three methods in block placement: Direct Mapped Cache, Fully Associative Mapped Cache and Set-Associative Mapped Cache.

7.9.3 Direct Mapped Cache (1-way Set Associative Cache)

The memory address is divided into three parts in direct mapping: **TAG**, **INDEX** (also called as SET/ BLOCK/Set Index/Block Index/Line), and **Block offset** (also called as offset/word).



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*-7.1** 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 x (block size + tag size + valid field size)

Block size = 2^m words

1 bit valid field is needed.

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

***Example*-7.2** 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})

$$\text{Tag} = 32 - 10 - 2 - 2 \text{ bits}$$
$$\text{Cache Entry size} = 128 \text{ bits of data + tag bits + valid bit.}$$
$$\text{Cache size} = 2^{10} \text{ x } (4 \text{ x } 32 + (32 - 10 - 2 - 2) + 1$$
$$= 2^{10} \text{ x } 147 = 147 \text{ K bits}$$

7.9.4 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.