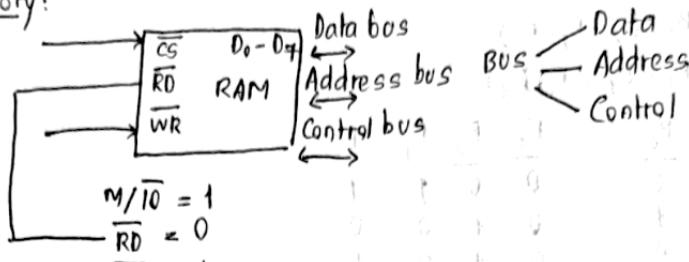


# COMPUTER ORGANIZATION AND ARCHITECTURE

## Basic Components of Computer:

- i). Input/Output devices
- v). Power supply
- ii). Processor
- vi). Mother board
- iii). Memory
- iv). Control unit

## Block diagram of Memory:



e.g.: 8 kb  $\Rightarrow 8k \times 8$   
 $\Rightarrow 2^3 \times 2^{10} \times 8 \text{ bit} \Rightarrow 2^{13} \times 8 \rightarrow \text{no. of registers}$

→ Registers: It has capacity of R/W data and also shifting data (Right/Left).  
 The building blocks of registers are Flip Flops and the building block of memory is register.

## CH: BASIC CONCEPTS OF COMPUTER EVOLUTION

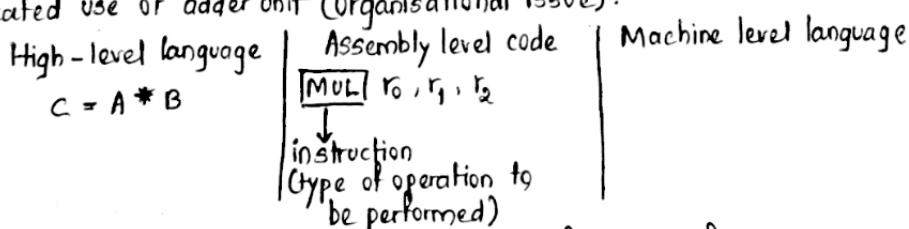
- Computer organization refers to the operational units and their interconnection that realise their architectural specification.
  - e.g: i). The memory technology used
  - ii). Interface b/w computer and peripheral.
  - iii). Control signals
- Computer architecture refers to those attributes of a system, that have a direct impact on the logical execution of a system.
  - e.g: i). memory addressing
  - ii). I/O mechanism
  - iii). datatypes being used
  - iv). The instruction

Each system has its own instruction set Architecture (ISA).

## Explanation of Computer Organization through an example:

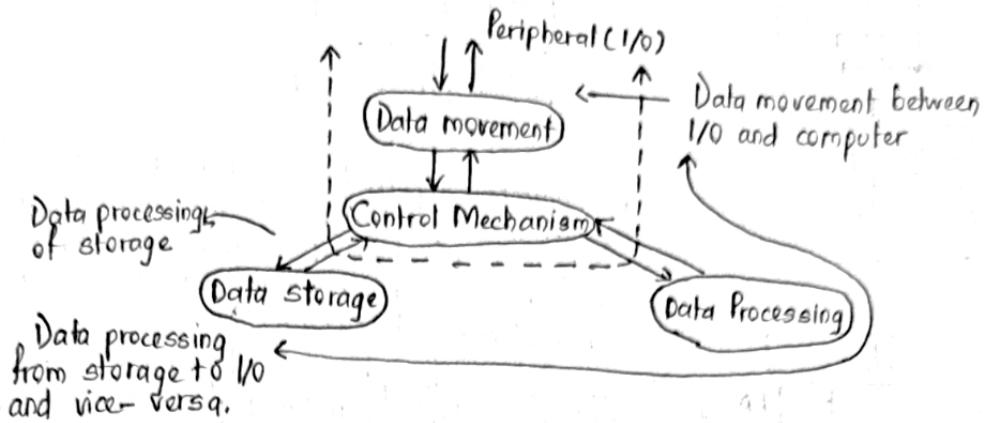
Multiplication of 2 numbers:

- i). We need to check whether computer have multiplication function or not (Architectural issue).
- ii). Whether the instruction will be implemented by MUL unit or by a mechanism that makes repeated use of adder unit (Organisational issue).



- A computer is a very complex system consisting of millions of small electronic components. The computer system is realised in hierachial structure. The hierachical nature of a complex system is essential for both design and description point of view.
- For design point of view, the designer need to deal with the particular level of the system at a particular time. At each level the designer is concerned with two things:
  - a). structure
  - b). function

- For description point of view, there are 2 ways to describe a complex system:
  - top-down approach
  - bottom-up approach
- The way in which components are interconnected, the function of each individual component is a part of the structure.
- There are 4 basic functions of a computer system:
  - Data processing
  - Data storage
  - Data Movement
  - Control mechanism



→ **Data Processing**: The computer must be able to process the data which may get a wide variety of forms (integer, float, double) and the range of processing. No. of bits: 8, 16, 32, 64

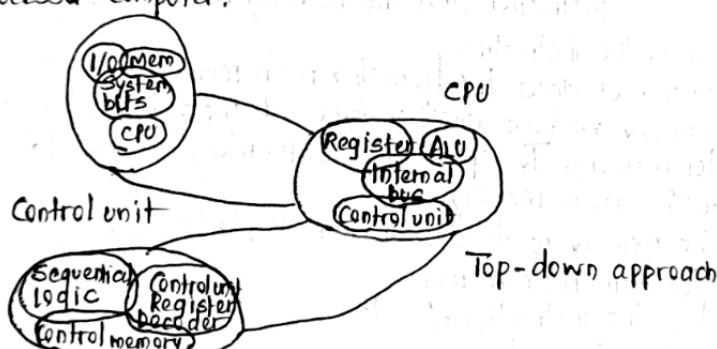
→ **Data Storage**: Computer stores the data temporarily or permanently.

→ **Data Movement**: Computer must be able to move (transfer) the data between itself and peripheral.

→ **Control Mechanism**: There must be control of the above three functions. For this purpose, within a computer a control unit manages the resources.

→ **Structure**:

i. Single-processor Computer:



System Bus: i. Data Bus

ii. Address Bus

iii. Control Bus

. I/O: There are ports available between computer and external device.

. Mem (Memory): Used to store the data.

. CPU: Central Processing Unit controls the operation of computer and performs data processing function. It is referred to as processor.

Structural component of CPU are:

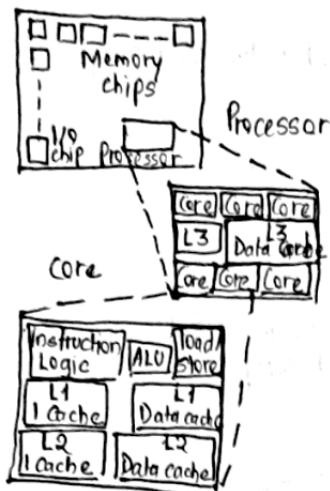
i. Control Unit: Controls the operation of CPU and hence the computer.

ii. ALU: Used to perform arithmetic and logical operations. It performs the computer's data processing function.

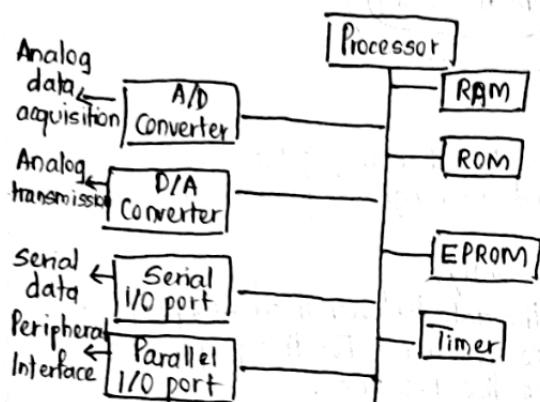
iii. Registers: Provide storage internal to CPU.

iv. Internal Bus: Used for interconnection between different units of a CPU.

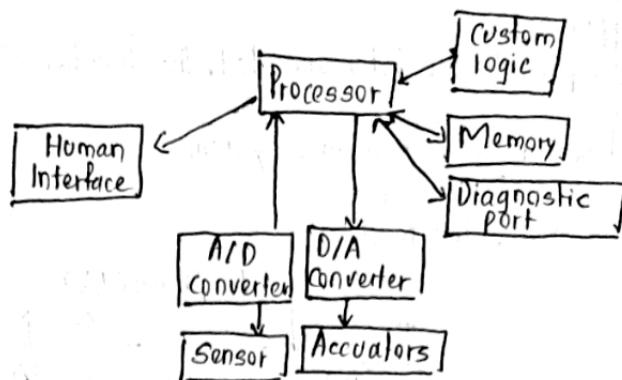
## ii). Multiprocessor Computer:



## Microcontroller:



## Embedded System:



D. Multicore computer have a PCB (Printed circuit board) which is a rigid flat board that holds and interconnects chips of different electronic components also known as Motherboard. The main element of motherboard is chips.

- Contains slots for processor chips which typically contains multiple individual parts for memory chips and I/O chips.
- The main functional unit of a processor is core. It contains 8 ports and L3 cache.
- The functional elements of a core are
  - Instruction logic responsible for fetching the instructions (reading the opcode from the memory location).
  - Decode the instruction to determine the type of instruction and the memory location of each operand.

- ALU performs the operation specified by the instruction.
- Load/store logic manages the transfer of data to/ from the main memory via cache. In between processor and main memory we have another type of memory called as cache memory. It is a smaller and faster memory. To speed up the memory access data is stored in the cache memory from the main memory. To increase the performance, cache memory is divided into multiple levels:

D. L1 cache: transfer the data to/ from main memory.

- L1 I-cache: Used to store the instructions/ code.
- L1 Data cache: Used to store the data.

→ Embedded system: The use of electronics (hardware) and software in a single system is called an embedded system.

e.g: digital camera, cell phone, microwave oven, home security system, etc.

- The organisation of embedded system consists of a processor, memory and a number of interfacing elements:
- Human Interface: It can be used as Robotic version
- Diagnostic port: It is required for diagnosing the system which helps in controlling the device.
- D/A / A/D: They convert analog data to digital for real time application.

→ Deeply Embedded system: Special embedded system that uses a microcontroller instead of processor. Not programmable as once the program logic for the system has been burned into the ROM if cannot be further modified.

Deeply embedded systems are dedicated single purpose device that takes input from the environment, perform the basic processing and produce the result to the environment.

→ Application Processor: i) Application processor are determined by the processor's ability to execute complex operating system programs such as linux, android, chrome, etc.  
ii) Used as general purpose processor  
e.g: processor embedded in a smartphone.

→ Dedicated Processor: i) Dedicated to one or small number of tasks required by the host device.  
e.g: processor embedded in washing machine.

### MicroProcessor

- i) Heart of the computer system
- ii) Based on Von Neumann architecture
- iii) It is just a processor, memory and I/O devices to be connected externally.
- iv) It has less number of registers so most of the operation are memory based.
- v) Power consumption more due to external component.

### Micro Controller

- i) Heart of embedded system
- ii) Based on Harvard architecture
- iii) It has a processor along with internal memory (unchipped RAM/ROM) and I/O ports.
- iv) More number of registers, hence most of the operations are register based.
- v) Power consumption is low.

→ IOT (Internet of Things): i) The interconnection of smart devices ranging from appliances to tiny sensors through Internet.

- ii) Internet supports the interconnection of billions of industrial and personal objects through cloud system.
- iii) IOT is primarily driven by deeply embedded systems.
- iv) Four generations of Internet reached the highest level of development known as IOT.

i) IT (Information Technology): PC, servers, routers, etc. bought as IT devices by the IT people.

ii) OT (Operational Technology): Machines/appliances embedded with IT built by non-IT companies e.g: medical machineries

iii) PT (Personal Technology): Smartphones, tablet, e-book used as IT devices bought by customers that uses wireless connectivity.

iv) ST (Sensor Technology): Single purpose devices bought by consumers, IT, OT and PT form a single larger system known as IOT.

### Cloud Computing:

i) Definition: i) There is an increasingly trend in many organisations to store substantial portion or even all the information to an internet connected infrastructure known as cloud computing.

ii) Cloud computing is the delivery of computing services which includes servers, storage, database and networking all over the network to provide faster innovation, flexible resources and economy of scale with minimal charges.

ii) Benefits: i) It provides professional network management.

ii) It provides professional security management.

iii) It can store data and share it with others.

iii) Types: i) Public cloud computing: public clouds sell the services to anyone on the Internet.

ii) Private cloud is proprietary network or a data centre that supply the hosted services to limited number of users.

iv) Cloud Service Provider: A Cloud service provider maintain the data storage resources available over the network or the internet. To use these resources the customer can pay the rent for a portion of resources. Virtually cloud services are offered using one of the 3 models:

- a). IaaS: Service model that delivers computer infrastructure to support various applications.  
It provides underlying OS, virtual machine, server hardware, networking, etc.
- b). PaaS: Provides a platform and environment to allow developers to build applications and services over the internet.
- c). SaaS: It is a way of delivering services and applications provided by the CSP over the internet.

## LH - PERFORMANCE ISSUES

- The main objective of designing a computer is the performance and capacity of those systems must increase continuously. The cost of the computer system dropped by the processor. Now days, laptops are able to perform larger computation due to a chip (IC) called a microprocessor. So most of the desktop/laptop have microprocessor that supports variety of applications such as image processing, video conferencing, etc.
- So to improve the performance of the system is equivalent to improve the performance of the processor. The performance can be improved by increasing the number of transistors in the processor IC.
- Other techniques are used to improve the performance of the processor. Performance can be improved by improving the speed of the microprocessor.

D. Pipeline Execution: It enables the processor to work simultaneously on multiple instructions by performing different phases (fetch - decode - execute) for each of the multiple instructions in the same time.

i). Processor moves data and instructions in a conceptual pipe with all the stages of pipe processing simultaneously.

ii). Processor analyse the instructions which depend on other instruction results and data to create an optimised schedule of instruction.

iii). Branch Prediction: Processor looks ahead in the instruction node fetched from memory and which branches of the instruction are likely to be processed next to the pipe.

iv). Superscalar Execution: This is the ability to issue more than one instruction in each processor clock cycle.

v). Speculative Execution: Using Branch prediction and data flow analysis some processors speculatively execute instructions ahead of their actual appearance in the program execution.

→ Performance Balance:

- The main difficulty in designing a system is that different components operate at different speeds. e.g: DRAM and I/O devices are slower than processor.
- It is necessary to adjust the organisation and architecture to compensate for the mismatch amount of capabilities of different components. So instead of raw performance of individual component overall balance of the system is more important for performance balance. That is why a number of benchmarks are used to compare the system components.

To overcome the imbalance b/w memory and processor the approaches used are:

- Increase the number of bits that are retrieved at one time.
- Reduce the frequency of memory access by incorporating an efficient cache.
- Increase the interconnected bandwidth b/w the processor and memory by higher speed buses.
- Change the DRAM Interface.

To improve the ~~chip~~ organisation & architecture:

- Increase the speed by fundamentally reducing the logic gate size to accommodate more number of logic gates.
- By increasing the clock rate
- By reducing the propagation delay.
- Increase the size and speed of the cache. This could be achieved by dedicating a part of processor chip:

a). By reducing the access time of cache.

- Change the processor organisation and architecture. This could be achieved by:
  - Increasing the speed of executing instruction.
  - By using the concept of parallel processing.

→ Problems associated with increasing clock speed and logic gates:

#### D. Power dissipation:

Generation of more heat due to increase in density of logic gates and clock speed which in turn increases the power dissipation.

D. RC Delay: The speed at which the electrons flow limited by the resistance and capacitance of the wire, i.e. RC product. Delay increases as RC product increases.

→ Amdahl's Law: Amdahl's law deals with the potential speed of a program execution using multiple processor compared to a single processor.

$$\text{Speed up} = \frac{\text{Performance of a system using multiprocessor}}{\text{Performance of a system using single processor}}$$

$$\text{Performance of any system} \propto \frac{1}{\text{Execution time of program}}$$

- Consider a program is running in a single processor system. Then  $T$  = total execution time of a program using single processor.

$f$  = fraction of time involved for the code to be executed parallelly.

$1-f$  = fraction of time involved for the code to be executed sequentially.

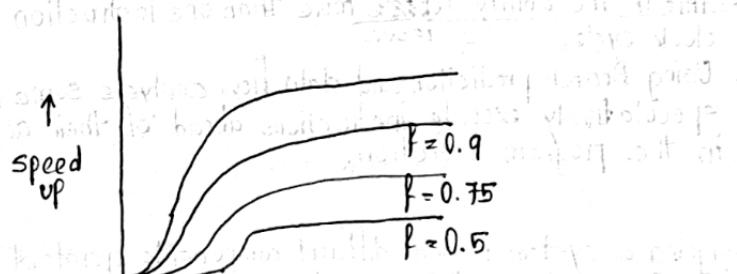
$N$  = No. of processors.

$$\Rightarrow \text{Speed up} = \frac{fT + (1-f)T}{(1-f)T + \frac{fT}{N}} = \frac{fT + T - fT}{T(1-f + \frac{f}{N})} = \frac{1}{1-f + \frac{f}{N}}$$

i). If  $f \rightarrow$  very small (0), then the use of parallel processing has little effect.

ii). When  $N \rightarrow \infty$ , then the speed up is bounded by a factor of  $\frac{1}{1-f}$ .

Applications of Amdahl's law: Parallel processing and Load balancing



No. of processors →  
Amdahl's law using multiple processors

i).  $\xleftarrow{(1-f)T} \xrightarrow{fT} \text{Single processor}$

ii).  $\xleftarrow{(1-f)T} \xleftarrow{\frac{fT}{N}} \text{Multiple processor}$

$$= (1-f)T + \frac{fT}{N} = (1-f + \frac{f}{N})T$$

$$\Rightarrow \text{total execution time: } [1 - f(1 - \frac{1}{N})]T$$

for multiprocessor

• Amdahl's law can be generalised to design any technical improvement.

Consider any enhancement to a feature that results in a speed up. The speed up can be expressed as:

$$\text{Speed up} = \frac{\text{Performance after enhancement}}{\text{Performance before enhancement}} = \frac{(1-F') + F'}{(1-F) + F}$$

$F'$  → Enhancement factor

$s'$  → Improvement factor

### Little's law:

According to the queuing theorem, for a steady state system, if  $\lambda$  is the average rate of items per unit time,  $w$  is the average time spent by each item, and  $L$  is the items in the system at any given time is defined by Little's law as

$$L = \lambda w$$

e.g: You have a book store with 10 visitors arriving at every hour and it takes them 30 minutes to find the book they want, pay and leave the shop. It means that you will have five customers in the shop at any given time.

Having 5 book store for more space to accommodate all the customers and not to hire more sellers.

Q: What will be the overall speed up if  $N = 10\%$  and  $f = 0.9$ ?

$$\text{Speed up} = \frac{1}{1 - 0.9 + \frac{0.9}{10}} = \frac{1}{0.1 + 0.09}$$

$$= \frac{1}{0.19} = 5.26$$

Q: Let a program have 40% of its code enhanced to yield what is the factor of improvement?

Soln:  $f' = 0.4, s' = 4.3$  system speed up 4.3 times faster.

$$\Rightarrow s' = \frac{(1-f') + f'}{s} \Rightarrow 4.3 = \frac{1}{0.6 + \frac{0.4}{s}} = 4.3$$

Q: What fraction of the execution time involves the code that is parallel to achieve an overall speed up of 2.25? Assuming 15 parallel processors.

$$\text{Soln: } N = 15$$

$$\text{Speed up} = 2.25 \Rightarrow 2.25 = \frac{1}{1 - f + f} \Rightarrow \frac{1}{15 - 14f} = 2.25$$

$$\Rightarrow \frac{15}{15 - 14f} = 2.25 \Rightarrow 2.25(15 - 14f) = 15$$

$$\Rightarrow f = 0.59$$

Q: Let a program have 40% of its code enhanced to run 2.3 times faster. What will be the overall system speed up?

$$\text{Soln: Speed up} = \frac{1}{(1-f') + \frac{f'}{s}}, f' = 0.4, s' = 0.3 \Rightarrow 2.3 = \frac{1}{(1-0.4) + \frac{0.4}{s}} \Rightarrow s = 1.29$$

Q: A doctor in a hospital observed that on an average 6 patients per hour arrive and there are typically 3 patients in the hospital. What is the average range of time each person spends in the hospital?

$$\text{Soln: } L = 3, \lambda = 6$$

$$W = \frac{L}{\lambda} = \frac{3}{6} = 0.5 \text{ hours}$$

Q: The owner of a shop observed that on an average 18 customers per hour arrive and there are 8 customers in the shop at any given time. What is the average length of time that each customer spends in the shop?

$$\text{Soln: } L = 8, \lambda = 18$$

$$W = \frac{L}{\lambda} = \frac{8}{18} = 0.44 \text{ hour}$$

i) The performance depends upon:

ii) clock rate / clock speed

iii) Execution time

iv) MIPS (Million Instructions per second) rate

v) MFLOPS (Million Floating point Instructions per second) rate.

vi) Clock speed: Operations performed by a processor such as fetching an instruction, decoding an instruction or performing arithmetic operations are governed by a system clock.

vii) The speed of a processor is detected by pulse frequency generated by the system clock and measured in Hz. Most of the instructions require multiple clock cycles to execute.

- ii). The rate at which clock pulses is generated is called clock rate.  
 iii). Every system is driven by a clock with frequency  $f$  or equivalently, a clock cycle time  $T = 1/f$ .

e.g: addition of two 16-bit numbers using 8086 processor:  
 The clock frequency of 8086 microprocessor is 5 MHz.

MOV AX, 4500H  
 MOV BX, 4300H  
 ADD AX, BX  
 HLT

Instruction	No. of clock cycle	Time for 1 clock cycle
MOV AX, 4500H	4	$T = 1/f = 1/5 \times 10^6 = 0.2 \mu s$
MOV BX, 4300H	4	$T = 4 \times 0.2 = 0.8 \mu s$
ADD AX, BX	2	$T = 4 \times 0.2 = 0.8 \mu s$ $T = 0.2 \times 2 = 0.4 \mu s$

$$\text{Total execution time: } 0.8 + 0.8 + 0.4 = 2.0 \mu s$$

$$\text{Execution time of program} = \frac{1}{f} \times \text{overall CPI} \times \text{clock cycle time}$$

- iii). MIPS rate: It is a common measure of performance for the processor and it evaluates the rate at which instructions are executed.

$$\text{MIPS rate} = \frac{IC}{T \times 10^6} = \frac{IC}{CPI \times 10^6}$$

- iv). MFLOPS rate: It is another performance measure that deals with the floating point operations only. They are commonly used in scientific calculations and in gaming applications.

$$\text{MFLOPS rate} = \frac{\text{no. of executed floating point operations in a program}}{\text{Execution time} \times 10^6}$$

- Q: Consider the execution of a program that results in the execution of 2 million instructions on a 400 MHz processor. The program consists of 4 main types of instructions. The instruction makes and the CPI of each instruction type is given below based on the result of a specific program pattern execution. Calculate the average CPI and the corresponding MIPS rate.

Instruction type	CPI	Instruction mix
Arithmetic and logic instruction	1	60
load/store with cache hit	2	18
Branch	4	10
Memory reference with cache miss	8	10

$$\text{SOP: avg CPI} = \frac{\sum_{i=1}^n (\text{CPI} \times I_i)}{I_c} = \frac{1 \times \frac{60}{100} + 2 \times \frac{18}{100} + 4 \times \frac{10}{100} + 8 \times \frac{10}{100}}{100/100} = 2.24$$

$$\text{MIPS rate} = \frac{400 \times 10^6}{2.24 \times 10^6} = 178.571 \text{ Instructions/s}$$

→ Mean:

$$i). \text{ Arithmetic mean: } AM = \frac{x_1 + x_2 + x_3 + \dots + x_n}{n} \quad AM > GM > HM \quad z = \text{no. of operators}$$

$$ii). \text{ Geometric mean: } GM = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \dots x_n} \quad R_i = \frac{z}{t_i} \quad \text{used in a program}$$

$$iii). \text{ Harmonic mean: } HM = \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \dots + \frac{1}{x_n}} \quad AM \text{ rate} = \frac{1}{n} \sum_{i=1}^n R_i \quad n = \text{no. of benchmarks program} \\ t_i = \text{execution time of program}$$

d) In the table given below a comparison b/w the performance of three computers (A, B and C) on the execution of two programs is listed:

	Comp A time(s)	Comp B time(s)	Comp C time(s)
Prog 1 $10^8$ FP operations	2.0	1.0	0.75
Prog 2 $10^8$ FP operations	0.75	2.0	4.0
Prog 3 Total execution $10^8$ FP operations time	2.75	3.0	4.75

Calculate:

- i) The arithmetic mean of time for each computer.
- ii) Inverse of total execution time.
- iii) MFLOPS rate for each computer, each program, taking into account that the execution of each program results in the execution of  $10^8$  FP operations.
- iv) The arithmetic mean of MFLOPS rate for each computer and for each program respectively.
- v) The harmonic mean of rates for each computer and for each program respectively.
- Soln: vi) AM for prog 1:  $\frac{2.0 + 1.0 + 0.75}{3} = 1.25 \text{ s}$

AM for prog 2:

$$\frac{0.75 + 2.0 + 4.0}{3} = 2.25 \text{ s}$$

vii) Comp A :  $\frac{1}{0.75} = 0.333 \text{ s}^{-1}$

Comp B :  $\frac{1}{2.0} = 0.5 \text{ s}^{-1}$

Comp C :  $\frac{1}{4.0} = 0.25 \text{ s}^{-1}$

viii) Comp A :  $\frac{10^8}{0.75 \times 10^6} = 33.33 \text{ MFLOPS}$

Comp B :  $\frac{10^8}{2.0 \times 10^6} = 50 \text{ MFLOPS}$

Comp C :  $\frac{10^8}{4.0 \times 10^6} = 25 \text{ MFLOPS}$

- Conclusion:
- i) From the AM of the execution times we can conclude that computer A is fastest, then B and C.
  - ii) We can relate the above bit with the total execution time given in the question.
  - iii) From the AM of the MFLOPS rate we can also conclude that A is fastest, but C is faster than B (discrepancy obtained in the result).
  - iv) We can conclude from the HM of the MFLOPS rate that A is fastest, then B, then C.

Q: Compare the system performance wrt AM rate and GM rate (all the results are normalised to computer A).

	Comp A time(s)	Comp B time(s)	Comp C time(s)
Prog 1	2.0	1.0	0.75
Prog 2	0.75	2.0	4.0

$$GM \text{ rate} = \left( \prod_{i=1}^n R_i \right)^{1/n}$$

Soln: Normalized table

	Comp A	Comp B	Comp C
Prog 1	$2/2 = 1.0$	$1/2 = 0.5$	$0.75/2 = 0.375$
Prog 2	$0.75/0.75 = 1.0$	$2/0.75 = 2.67$	$4/0.75 = 5.33$

AM	$1.0$	$1.585$	$2.8525$
	$(1+1)/2$	$(0.5+2.67)/2$	$(0.375+5.33)/2$

Normalised time	$(1 \times 1)^{1/2} = 1$	$(0.5 \times 2.67)^{1/2} = 1.15$	$(0.375 \times 5.33)^{1/2} = 1.41$
total execution time	2.75	3.0	4.75

- Conclusion:
- A is the fastest computer in terms of total execution time, AM rate and HM rate.
  - The AM rate of computer B is lower than computer C, whereas computer B is faster than C.
  - Looking at the HM rate, HM is preferred when calculating the execution rate because HM is inversely proportional to total execution time which is a desired property whereas AM execution rate is proportional to the sum of inverse of execution time.

Q: Why GM is chosen over AM?

Soln: i). GM gives consistent result regardless of which system is used as reference.  
 ii). Distribution of performance ratios are better modeled by log (ln) normal distribution.  
 iii). GM is less biased (chances of less incorrect) than HM and AM.

Q: A benchmark program is run first on a 200 MHz and then on a 300 MHz processor. The executable program consists of 1 million instruction execution with the following instruction mix and clock count.

Instruction type	Instruction count	Clock cycle count
Integer Arithmetic	$4 \times 10^5$	1
Data Transfer	$3.5 \times 10^5$	2
Floating-point Instruction	$2 \times 10^5$	3
Control Transfer	$0.5 \times 10^5$	2

Determine the effective CPI, execution time and MIPS rate for both the cases.

$$\text{processor 1: } f_1 = 200 \text{ MHz} = 200 \times 10^6 \text{ Hz}$$

$$\text{processor 2: } f_2 = 300 \text{ MHz}$$

Soln: For  $f_1$ :

$$f = 200 \text{ MHz} = 200 \times 10^6 \text{ Hz}$$

$$\text{Overall CPI} = \frac{\sum \text{CPI}_i I_c}{I_c} = \frac{1 \times 4 \times 10^5 + 2 \times 3.5 \times 10^5 + 3 \times 2 \times 10^5 + 2 \times 0.5 \times 10^5}{2 \times 0.5 \times 10^5} = 10^6$$

$$Z = \frac{1}{f} = \frac{1}{200 \times 10^6} = 1.8 \times 10^{-6}$$

$$E = I_c \times \text{CPI} \times Z = 10^6 \times 1.8 \times \frac{1}{200 \times 10^6} = 0.009$$

$$\text{MIPS rate: } \frac{I_c}{T \times 10^6} = \frac{f}{\text{CPI} \times 10^6} = \frac{200 \times 10^6}{1.8 \times 10^6} = 111.11$$

A.8.2: For machine A:

No. of class  $n = 4$

Total no. of instruction:  $I_0 = I_c$

$$f = 400 \text{ MHz}$$

$$\text{CPI} = \frac{\sum \text{CPI}_i I_c}{I_c} = \frac{2 \times 0.5 \times 10^5 + 3 \times 0.15 \times 10^5 + 4 \times 0.15 \times 10^5 + 1 \times 0.45 \times 10^5}{2 \times 0.2 \times 10^5} = 2.45$$

$$\text{execution time} = I_c \times \text{CPI} \times Z = 10^5 \times 2.45 \times \frac{1}{400 \times 10^6}$$

$$= 0.6125 \times 10^{-3} \text{ s}$$

$$\text{MIPS rate} = \frac{f}{\text{CPI} \times 10^6} = \frac{400 \times 10^6}{2.45 \times 10^6} = 163.27$$

For machine B:

$$n = 4.5$$
$$I_c = 10$$
$$f = 400 \text{ MHz}$$
$$\text{CPI} = \frac{2 \times 0.1 \times 10^5 + 3 \times 0.1 \times 10^5 + 4 \times 0.15 \times 10^5 + 1 \times 0.65 \times 10^5}{10^5}$$
$$= 0.2 + 0.3 + 0.6 + 0.65 = 1.75$$

$$\text{execution time: } I_c \times \text{CPI} \times T$$
$$= \frac{10^5 \times 1.75 \times 1}{400 \times 10^6} = 0.4375 \times 10^{-3} \text{ s}$$

$$\text{MIPS rate} = \frac{400 \times 10^6}{1.75 \times 10^6} = 228.57$$

A. 2.3: i) VAX Machine:

$$\text{CPU time} = 12x$$

$$\text{clock freq} = 5 \text{ MHz}$$

$$\text{MIPS} = 1$$

$$\text{MIPS} = \frac{I_c}{\text{CPU execution time}}$$

$$I_c = \text{CPU time} \times \text{MIPS} \times 10^6$$
$$12 \times 10^6 \times 10^6 = 12 \times 10^6$$

IBM Machine:

$$\text{CPU time} = x$$

$$\text{clock freq} = 25 \text{ MHz}$$

$$\text{MIPS} = 18$$

Relative Instruction count:

$$I_c = 18 \times x \times 10^6$$

$$\frac{I_c(\text{VAX})}{I_c(\text{IBM})} = \frac{12 \times x \times 10^6}{18 \times x \times 10^6} = \frac{2}{3}$$

$$\text{i). MIPS} = \frac{f}{\text{CPI} \times 10^6} \Rightarrow \text{CPI} = \frac{f}{\text{MIPS} \times 10^6} = \frac{5 \times 10^6}{1 \times 10^6} = 5$$

$$\text{MIPS} = \frac{f}{\text{CPI} \times 10^6} \Rightarrow \text{CPI} = \frac{25 \times 10^6}{18 \times 10^6} = \frac{25}{18} = 1.389$$

	Comp A	Comp B	Comp C
Prog 1	50	20	10
Prog 2	100	200	40

$$\text{MIPS}_A = \text{i). } \frac{I_c}{T \times 10^6} = 0.2 \quad \text{MIPS}_C = \text{D. } 1$$

$$\text{ii). } 0.8$$

$$\text{MIPS}_B = \text{D. } 0.5$$

$$\text{ii). } 0.05$$

Relative speed up is defined as the measure of all the system work as compared to another system.

Relative speed up =

→ Multicore MIC & GPGPUs:

- Multicore MIC & GPGPUs:

- Multicore: The use of multiple processors on a single chip is called multicore.
  - MIC: If in a chip more than 50 cores are available then it is known as Many Integrated Core.
  - GPGPU: A chip with multiple general purpose processor + graphics processing unit and specialised cores for various processings are embedded in a single chip known as General Purpose Computing on GPUs.

→ SPEC Benchmark:

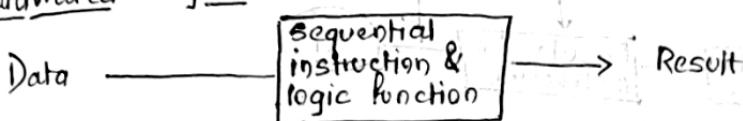
- The common need in industry, academic and research communities for generally accepted computer performance development has led to the development of standardised benchmark suits.
  - A benchmark suit is a collection of programs defined in a high level language that together attempt to provide a representative case of a computer in a particular application on system programming area.
  - The best known such collection of benchmark suits is defined and maintained by a corporation known as SPEC.

## (H: A TOP LEVEL VIEW OF COMPUTER FUNCTION & INTERCONNECTION)

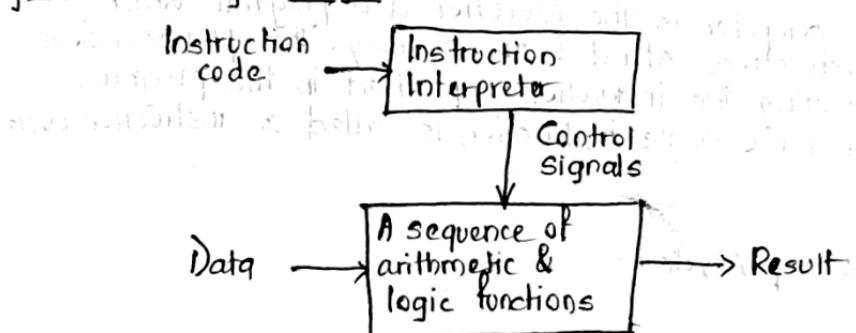
### Introduction:

- Computer components:
  - Most of the computers are based on Von Neumann architecture.
  - The architecture is named on 3 key factors/concepts:
    - i) Data and programs are stored in a single R/W memory.
    - ii) The contents of the memory are addressed by their location.
    - iii) Sequential execution of a program when it is not explicitly modified.
- Two types of configurations used for interconnected computer components:
  - At the top level a computer consists of CPU, memory and I/O components with one or more modules of each type.
  - These components are interconnected in some fashion to achieve the basic function of a computer which is to execute the program. Thus at a top level, we can characterise a computer system by describing
    - i) the external behaviour of each component.
    - ii) the interconnection structure and function required to manage the use of interconnection structure.
- For this reason the top level view of structure and function is important.

### i). Hardwared Program:

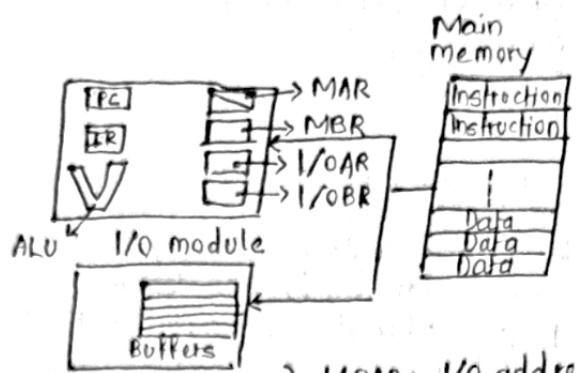


### ii). Programming using Software:



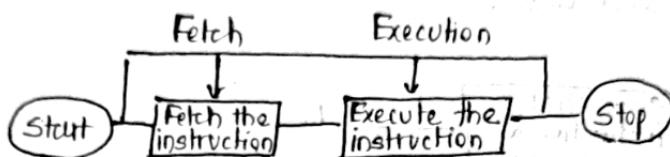
- There are various ways to connect a set of basic logic components to store data and perform arithmetic and logic operations.
- When a specific configuration of logic components are designed to perform a particular computation and the process of connecting various components in the desired configuration for a particular computation is called a program.
- The resulting program in term of hardware is called hardware program. In this the system accepts the data and produce the result.
- Instead of rewriting the hardware for each new program, a general purpose configuration might be used which consist of a set of hardware that performs various arithmetic and logical functions. With the general purpose hardware the system accepts various instructions code, interprets the code and generates necessary control signals. Along with the control signals, the system accepts data and produce result for various programs.
- Each code in the instruction and parts of the hardware interpret each instruction and generate the control signals.

## → Computer Function:



- i). PC: Program counter
- ii). IR: Instruction register
- iii). MAR: Memory Address register
- iv). MBR: Memory Buffer register
- v). I/OAR: I/O address register
- vi). I/O BR: I/O buffer register

- Type of operations performed by a computer by alternating an instruction:
  - i). Processor-Memory
  - ii). Data storage
  - iii). Processor-I/O
  - iv). Control
- 2 phases/cycles of Instruction execution:
  - i). Fetch cycle
  - ii). Execution cycle



## → Instruction cycle:

Definition: The basic function of a computer is the execution of a program which consists of a set of instructions stored in the memory. The processor does the actual work by executing the instructions specified in the program. The processing required for a single instruction is called an instruction cycle.

- Each Instruction cycle consists of 2 steps/cycles:
    - i). Fetch cycle
    - ii). Execute cycle
  - Consider a hypothetical machine which has only one machine called as accumulator. Both data & instruction are 16-bit long. Each M/L is capable of storing 16-bit data.
- Instruction Format:
- |        |         |
|--------|---------|
| Opcode | Address |
|--------|---------|
- 14 bit 12 bit
- Data Format:
- |      |           |
|------|-----------|
| Sign | Magnitude |
|------|-----------|
- 1 bit 15 bit
- 4 bit of instructions used for opcode.  
 $2^4$  total possible opcode
  - 12 bit of instructions used for address.  
 $2^{12} = 4096$  words of memory ( $000_H - FFF_H$ )
  - Example of opcode:
    - i). 0001: load AC from memory
    - ii). 0010: store AC to memory
    - iii). 0101: Add AC to memory content
    - iv). 0011: Load AC from I/O
    - v). 0111: Store AC to I/O

d: Using the hypothetical machine (1As machine) add the content of M/L 940H & 941H and store the result in 942H. Show the execution program.

Program

M/L  
300H  
301H  
302H

Opcode

Mnemonics

LOAD M (940H)  
ADD AC, M(941H)  
STORE M (942H)

PC points to the memory location of the instruction which is going to be fetched.

The opcode from the memory location comes to IR for decoding.

MAR specifies the address in the memory for the next read/write.

MBR contains the data to be written into the memory or read from the memory.

I/O AR specifies the address of I/O devices.

I/O BR stores the data to be written into output device or read from input device.

~~The basic function of a coreop~~

From the figure, In each instruction cycle, processor fetch the instruction from the memory location.

i). In most of the processors, PC holds the address of the instruction to be fetched next.

ii). After fetching the instruction, PC is incremented by 1, 2 or 4 (depending on the type of processor) automatically.

iii). The fetched instruction is loaded into the IR for decoding.

iv). After decoding the processor performs the required action (execute cycle). This action

v). may fall into 4 categories:

a). processor-memory: Data transfer b/w memory and processor.

b). processor-I/O: Data transfer b/w processor and I/O.

c). Data Processing: Perform arithmetic and logical operations.

d). Control: An instruction may specify the sequence of execution to be altered.

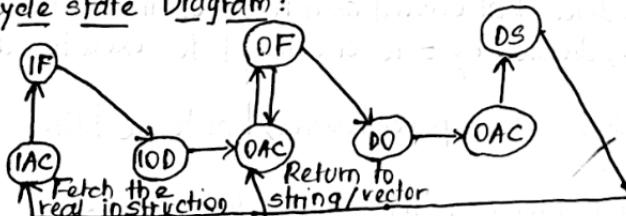
<u>Sol<sup>n</sup>:</u>	<u>M/L</u>	<u>Opcode</u>	<u>Mnemonics</u>
	300H	1940H	Fetch { LOAD M (940H)
	301H	5941H	cycle { ADD AC, M(941H) } → execution cycle
	302H	2942H	STORE M (942H) }

0000 1001 0100 0000  
1      9      4      0

0010 1001 0100 0010  
2      9      4      2

0101 1001 0100 0001  
5      9      4      1

→ Instruction cycle state Diagram:



→ Interrupt: Interrupt is a signal generated by external device request or by software to the processor for the execution of another program, and needs the attention of the processor.

i). maskable: Interrupt which can be ignored

ii). non-maskable: Cannot be ignored and processor has to respond.

## → Types of Interrupt:

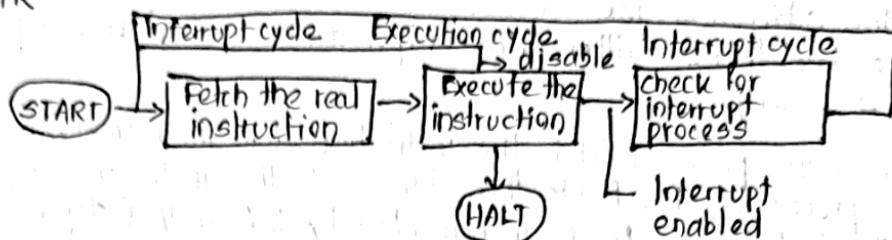
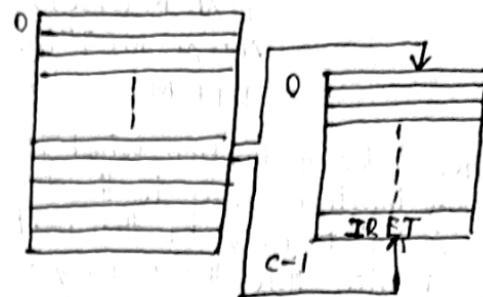
- Maskable interrupt
- Non-maskable interrupt

## → Classes of Interrupt:

- Program
- Hardware Failure
- Timer
- I/O

DMA  
INTR

## → Interrupt Handler/Subroutine:



IAC: Instruction address calculation. Determine the address of the next instruction to be executed and store in PC.

IF: Instruction fetch. Reads the opcode/instruction from the memory location pointed at PC value.

IOD: Analyze the instruction to determine the type of operation to be performed and operand to be used (basing on the value of IR).  
Instruction operation decoding

OAC: Operand Address calculation. If operation involves memory or I/O reference then determines the address of the operand.

OF: Operand Fetch. Fetch/read the data/operand from memory location or I/O.

DO: Data Operation. Perform operation indicated in the instruction.

OS: Operand store. Write the result in the memory or I/O.

- Operation consists of 3 instruction cycles, each contains a fetch and execution cycle.
- More details of the instruction cycle are represented by a diagram called as instruction cycle state diagram.
- For any given instruction cycle, some states are null or some states may be visited more than once.

i) Program: Generated by some error condition that occurred as a result of instruction execution such as arithmetic overflow, division by zero or attempt to execute an illegal instruction.

ii) Hardware Failure: Generated by a failure such as power failure, hardware failure or memory parity error.

iii) Timer: Generated by a timer within the processor. This allows the processor to perform certain function on a regular basis (linear calculation).

iv) I/O: Generated by an I/O controller. I/O devices can't send their request to the processor for the execution of another program.