

# Advance Computer Architecture

## Unit-I

Part - I

### Basic concept

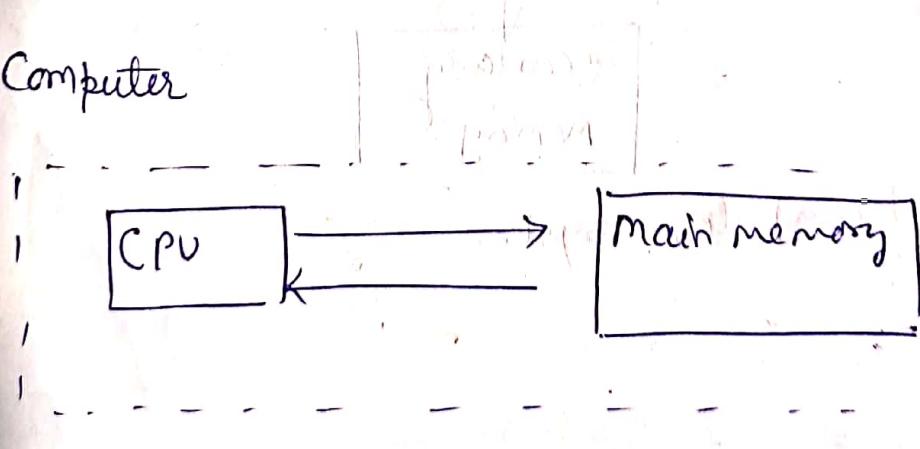
- ↳ Digital
- ↳ Interface
- ↳ I/P - O/P
- ↳ Microprocessor (cpu)
- ↳ fundamental of Computer Architecture

### What is computer :-

Page-1  
PDF 1st unit

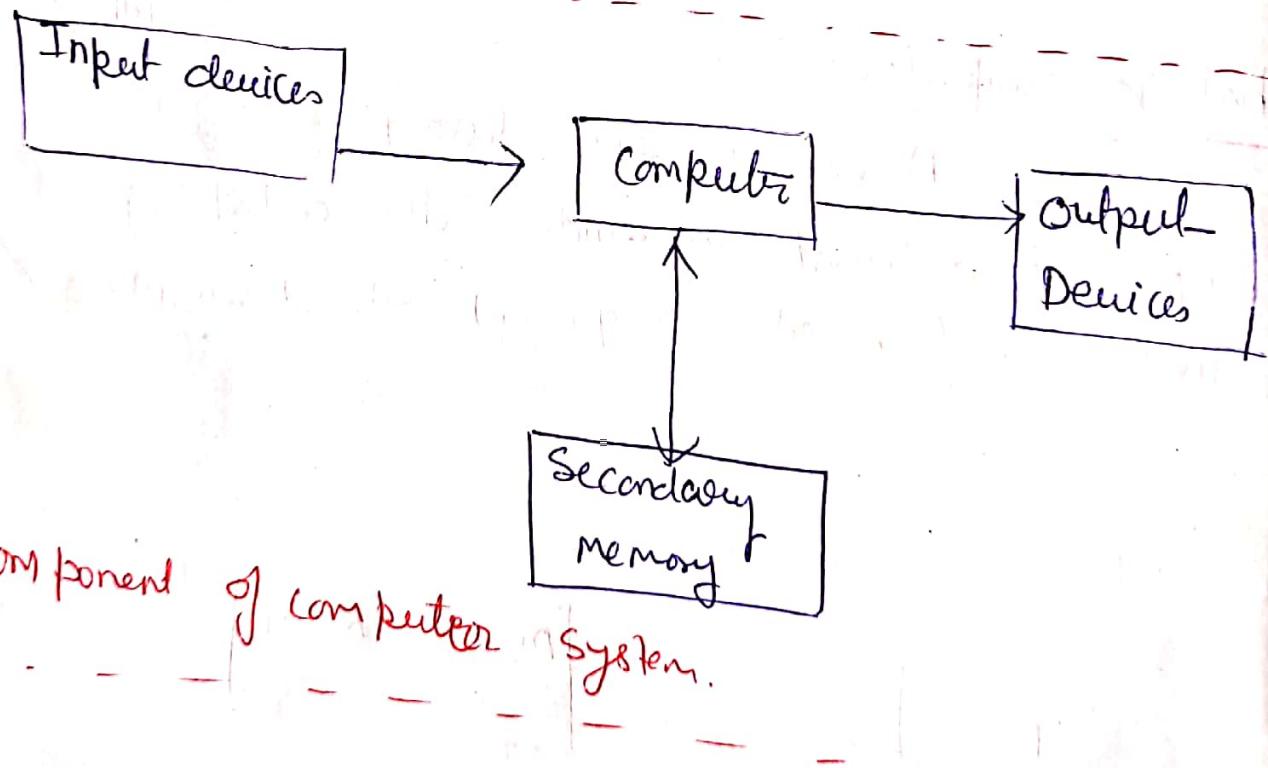
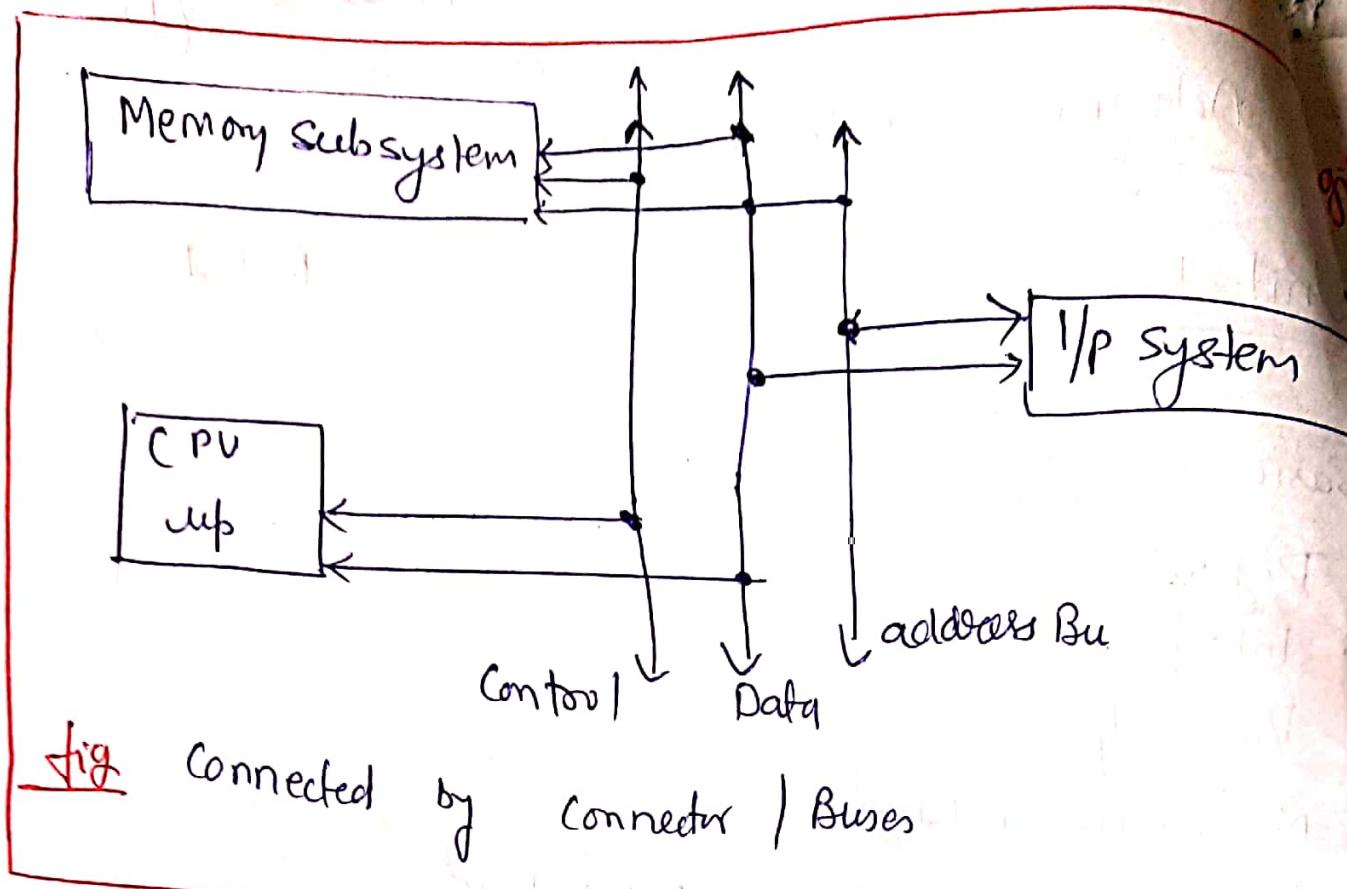
A computer is a data processing machine which is operated automatically under the control of a list of instruction (called a program) stored in main memory.

### Computer



### Computer System:-

A collection of Memory subsystem CPU and peripheral devices all the 3 component are interconnected by group of conductor or buses



## Digital System

A number of important discoveries and inventions led to the development of the digital microcomputer known today as the personal computer (PC).

Microprocessor include:-

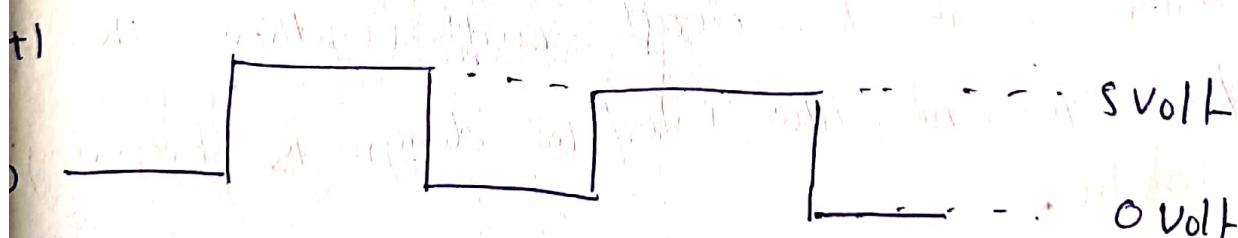
- ↳ Transistor, is essentially a solid state electronic switch.
- ↳ An integrated circuit is a semiconductor circuit that contain no. of transistors on a tiny piece or chip of Silicon.

A digital computer is a very flexible general-purpose m/c that can be applied to a wide and ever-increasing range of application.

Within Computer the Binary digit or bit are represent

5 Volts = logic level 1

0 Volts = logic level 0



0      1      0      1

Intel - 4004 processor

## S/w and H/w interaction

There are S/w designer and hardware designer - <sup>only</sup> two rather different breeds of designer with <sup>similar</sup> different backgrounds and way of working. Likewise the usability and ergonomics professions have developed with specialist in the two areas.

Until recently S/w interface design has largely focused on S/w for traditional computer. Whether it be a window, icon etc the designer isn't concerned with the physical design of computer monitor, keyboard or system unit. These h/w aspects being generic and standardised are rarely something an interface designer can influence anyway.

Today, however, there is a clear growth in the no. of products for which the h/w and s/w design must be dealt with together. The boundary b/w S/w and h/w interface design is beginning to dissolve.

Various examples exist already interactive ~~the~~ touch screen kiosks in shops and at airport mobile telephones which can be used for much more than just phone calls and messaging etc with more <sup>intuition</sup>.

③

isplay

Ensuring ease of use for these products will not only require traditional knowledge of human-computer interaction with SW interfaces but must be complemented with competence in the ergonomics of hardware interaction.

ref. PPT

Computer System :-

Architecture Vs organization

Architecture : refers to those attributes of a computer visible to a programmer or compiler writer ; eg instruction set, addressing techniques, I/O mechanisms.

Architecture concerned with the structure and behaviour of the Computer as seen by the user

It include

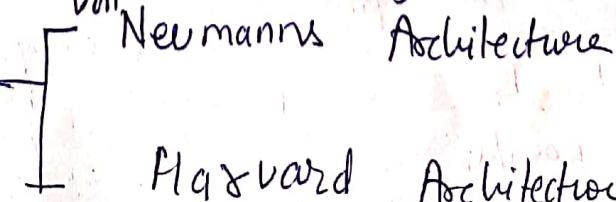
information

formats

Instruction set

Technique for addressing memory

Type of Architecture



# Architecture

## Von Neumann

## Harvard

Example :- [Desktop personal computer]

Example :- DSP [Digital Signal processor]

- ↳ It consists following components - ALU, RAM, CU, main machine interface I/O device.
- ↳ It uses physically separate storage and signal path way for their instruction and data.
- ↳ Due to integrated Ctl and miniaturization ALU + CU have been integrated onto the same chip and becoming an integrated part of comp's CPU.
- ↳ In which CPU can read instructions and data from memory at the same time lead double the memory bandwidth.
- ↳ Instruction are carried out sequentially, one instruction at a time.
- ↳ CPU - Both read & write data from memory is —————— not possible.

Organization :- Refers to how the features of a computer are implemented i.e. control signals are generated using the principle of finite state machine (FSM) or microprogrammery.

- Concerned with the way the h/w components operate and the way they are connected together to form the Computer system. The organization structure to verify that the computer part operate as intended.

### For Perspective of computer Architecture

- Processor Design
- I/P or O/P storage
- Memory Hierarchy
- Multiprocessor and N/W interconnection

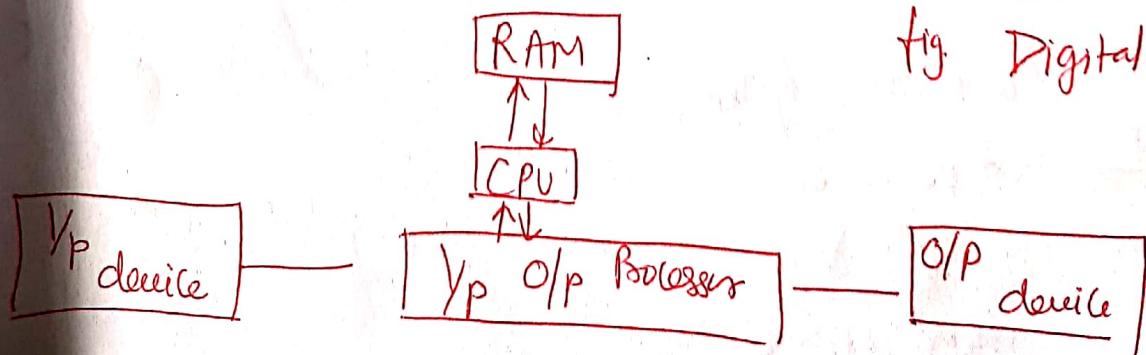


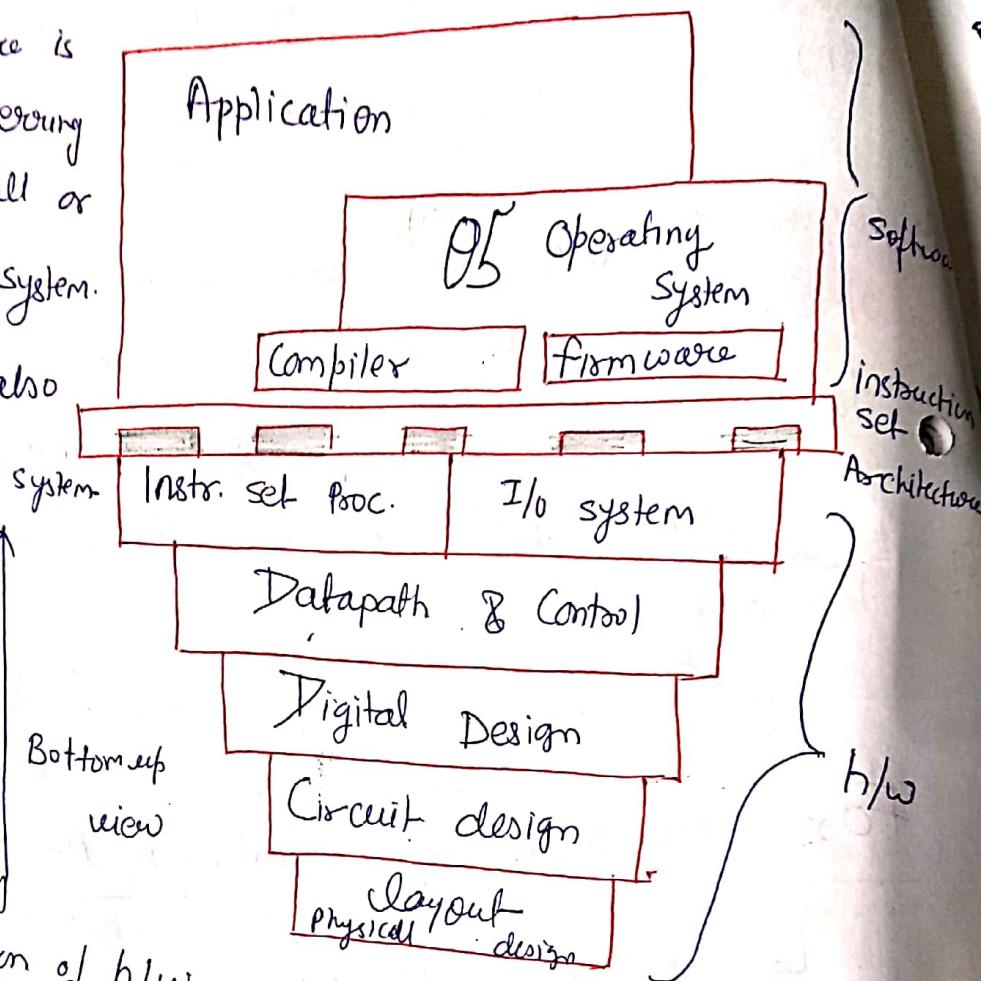
fig. Digital Computer

# Computer Architecture

## Architecture

Computer Architecture is a general term referring to the structure of all or part of a computer system.

The terms also cover the design of system. The term also covers the design of System SW such as OS as well as referring to the combination of h/w and the m/c on a computer N/W.



Computer Architecture refers to an entire structure and to the details needed to make it functional. Thus Computer architecture covers

- ↳ Computer system
- ↳ Microprocessors
- ↳ Circuit and
- ↳ System program

5

A digital system is an interconnection of digital h/w modules that can accomplish a specified information-processing task.

Digital systems vary in size and complexity from a few integrated circuit to a complex of interconnected and interacting digital computer.

Computer Architecture deal with the design of computers and with computer system and how they should be configured to satisfy overall system requirement.

Computer Architecture = Instruction set + Machine Organization

- ③ Technology always raises the bar for what could be done and change design's focus
- ④ Application usually derive capability and constraint  
eg embedded computing.
- ⑤ History always provides the starting point for innovation and filter out mistake

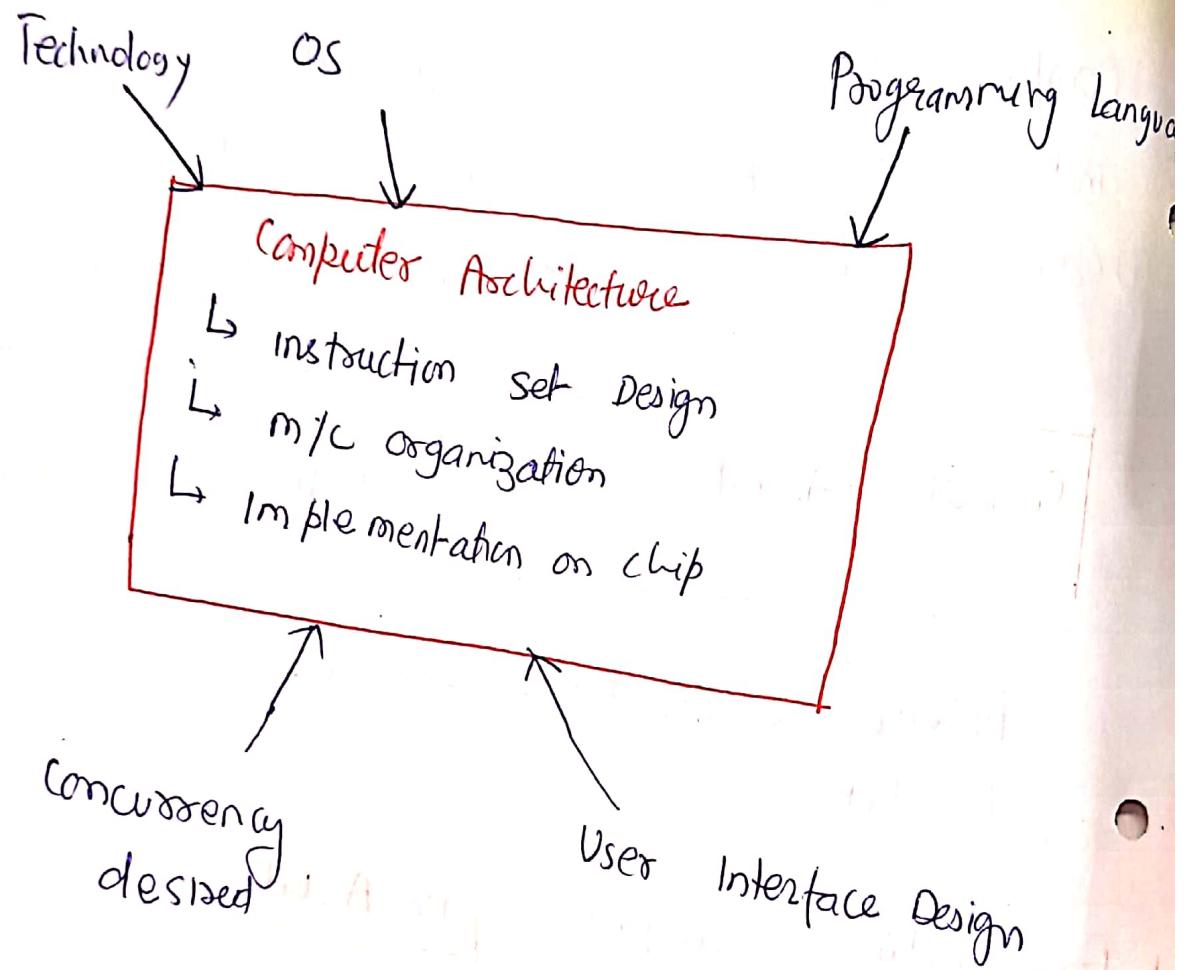


fig - forces on Computer Architecture

## Academic

## Commercial

Academic History

1945 - ENIAC - The world's 1<sup>st</sup> operation calculation

1946 ISA M/C - 10 times faster than ENIAC

Commercial History

1949 - 51 UNIVAC - 1 → \$1 million

1952 - 63 IBM 701, CDC 6600

1971 - 2006 microprocessor  
— mainframe  
— Super Computer  
— mini Super Computer  
— Work Station

# A classification of computer architectures

- ↳ Von Neumann machine
- ↳ Non Von Neumann machine

⇒ We call a computer Von Neumann machine if it meets the following criteria :-

- It has three basic hardware subsystem
  - ↳ CPU
  - ↳ main memory
  - ↳ I/O system
- It is a stored-program computer
- It carries out instructions sequentially
- It has, or at least appears to have, a single path b/w main memory and CU of CPU;  
This is often referred to as the Non Neumann bottleneck

The hardware

- of the Von Neumann m/c consists of
- A CPU (CU+ALU)
  - A main memory
  - I/O system

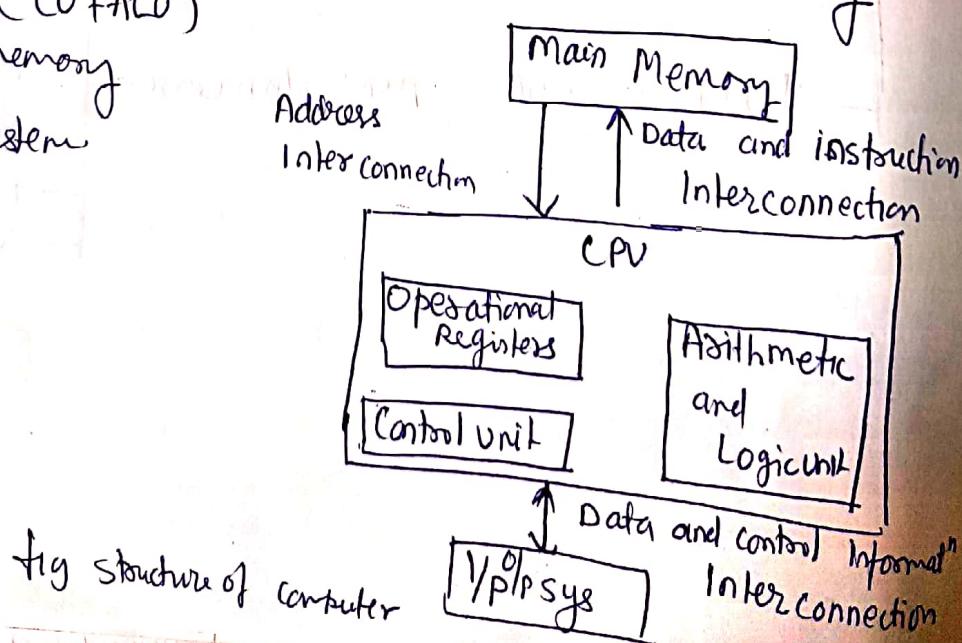


fig structure of computer

(7)

## Non-Von Neumann Machine : Flynn's Classification

- ↳ Single instruction stream, single data stream (SISD), Von Neumann m/c belong this category
- ↳ Single instruction stream, multiple data stream (SIMD)
- ↳ Multiple instruction stream, single data stream (MISD)
- ↳ Multiple instruction stream, multiple data stream (MIMD)

Multiprocessor m/c — SIMD & MIMD machine are multiprocessor m/c

- ↳ SIMD m/c mostly use global memory
- ↳ MIMD m/c can use local memory as well as global one

The following scenario are possible

- ↳ Processors share global memory without any local memory.
- ↳ Processors share global memory with local memory.
- ↳ Processors share no global memory.

## Classification of Computer Architecture design

↳ ALU

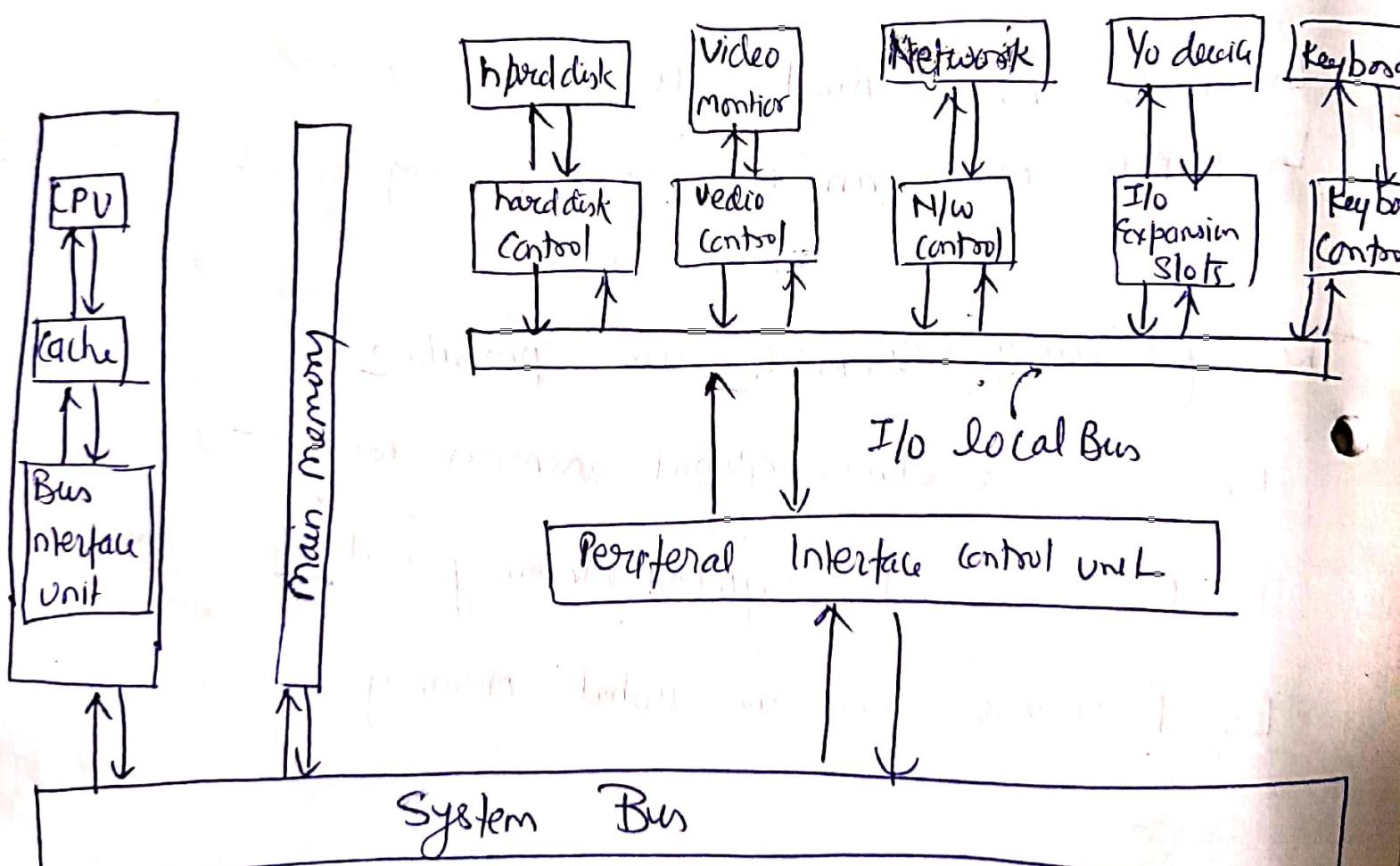
↳ memory

↳ I/O

↳ Out

Boxless Architecture, RISC and CISC

Basic organization.



# Processing

## Architecture

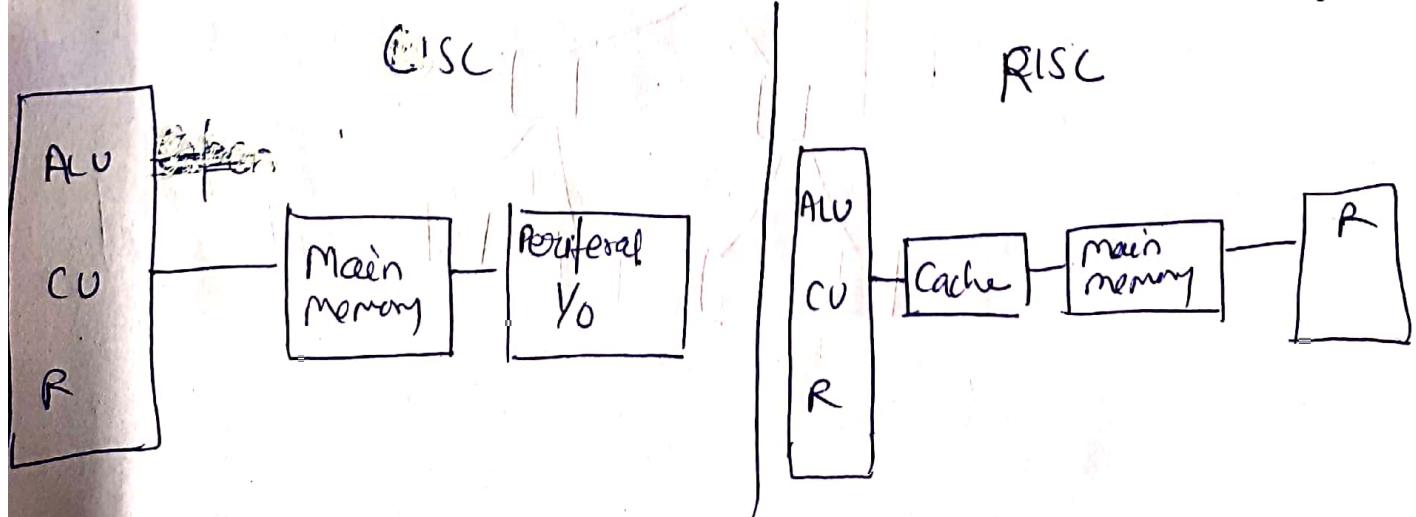
RISC

CISC

Early computers used only simple instruction because the cost of electronics capable of carrying out complex instruction was high. Complex instruction save time because they make it unnecessary for the computer to retrieve additional instruction.

The computer that combine several instructions into single operation are called complex instruction set computer (CISC).

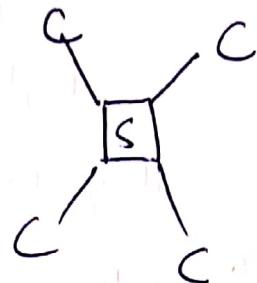
RISC designed are especially fast at the numerical computations required in science graphics and engg. application. CISC design are commonly used for nonnumerical computations because they provide special instruction sets for handling char. data such as text in a word processing program.



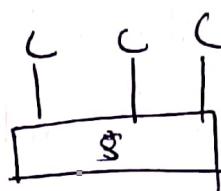
# N/W Architecture

## Topology

Star



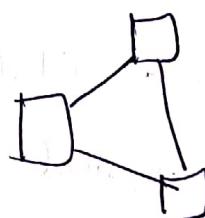
Bus



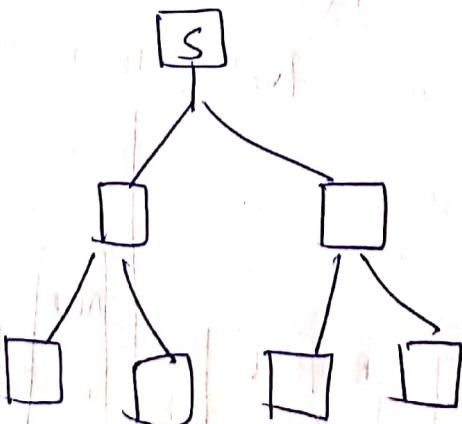
Ring



mesh



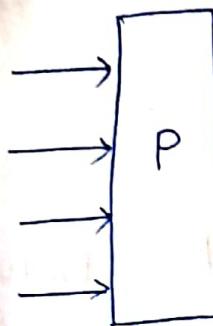
tree



# Parallel Processing

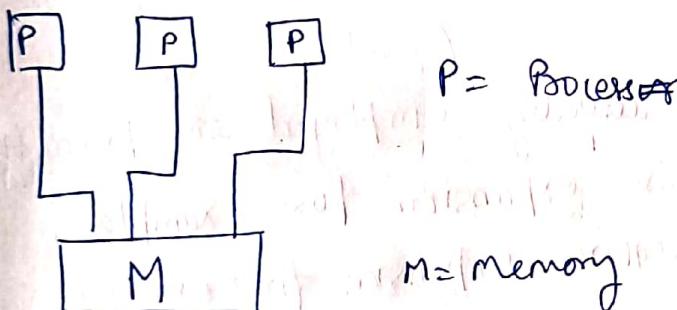
(9)

- 1) Shared memory
- 2) Distributed Memory



Super Computer

① shared Memory

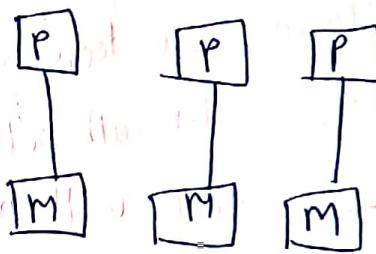


P = Processor

M = Memory

②

Distributed memory



In which process has  
separate memory

Which process share same  
memory

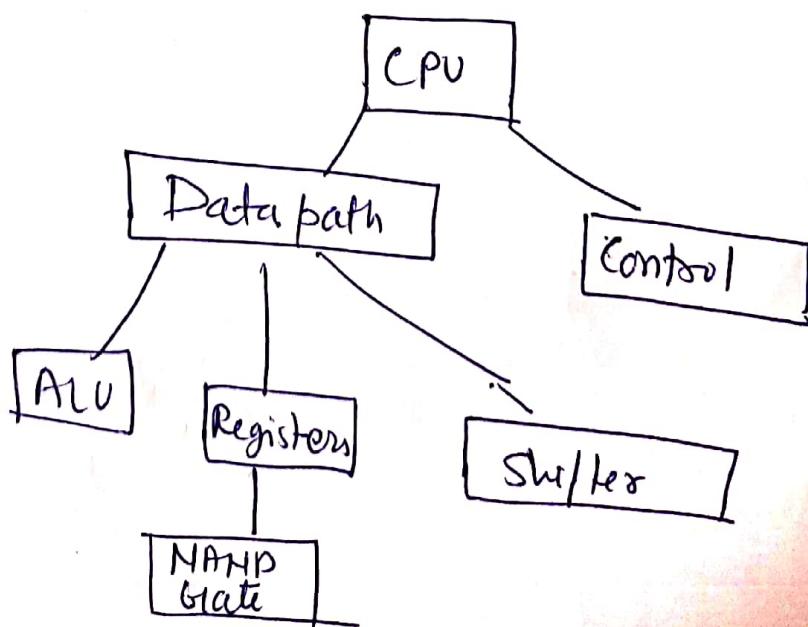
## Open and closed architectures

The CPU of a computer is connected to memory and to the outside world by means of either an open or a closed architecture.

An open architecture can be expanded after the system has been built, usually by adding extra circuitry such as a new microprocessor computer chip connected to the manufacturer.

Closed architectures are usually employed on specialized computers that will not require expansion for example, computers that control microwave ovens.

## The Design process



## Issues in Design of Computer Architecture

The following aspects of design are to be kept in mind.  
in design of computer Architecture

- ① Co-ordination of many levels of abstraction
- ② Design is to be done under a rapidly changing set of forces
- ③ Measurement, and evaluation will be required by so many authorities and designer have to satisfy them



The following factors keep on pressurising the designer of S/w from time to time.

- ④ The designer should not think whatever she or he knows all techniques as technology keep on having a dramatic change in processor power availability

Tech

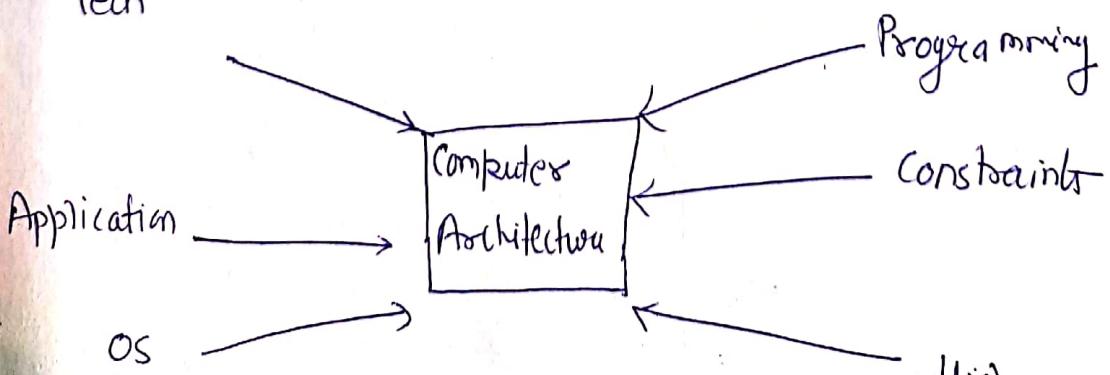


fig:- forces Affecting Design of Computer Architecture

Performance Metrics

It is a way by which we measure CPU performance

Performance metrics

Availability

Bandwidth

Throughput

Scalability

Instruction Path ( length and speed up )

Compression ration

Response time

Channel capacity

latency

Completion time

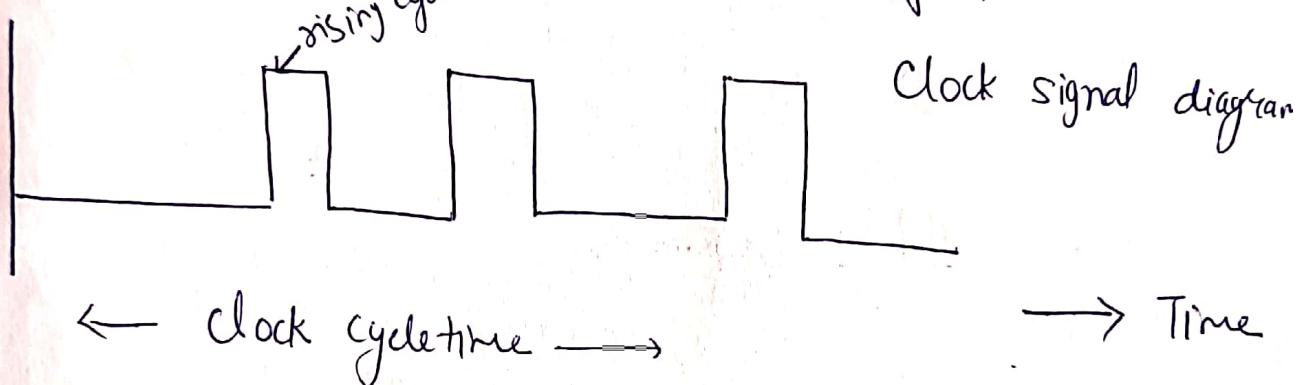
Service time

Relative efficiency

②  
Performance Analysis should help answering question such as how fast can a program be executed using a given computer?

For this we need to determine the time taken by a computer to execute a given job.

We define the clock cycle time as the time b/w two consecutive rising (trailing) edges of provide clock signal



Time taken en by CPU to execute a job

$$\boxed{\text{CPU time} = CC \times CT = CC/F}$$

where

CC = The no. of CPU clock cycle for executing a job  
to be the cycle count (CC)

CT - cycle time

clock frequency

$$f = 1/CT$$

(3)

Clock cycle allow counting unit computation because the storage of computation result is synchronized with rising (trailing) clock edge

It may be easier to count the no. of instruction executed in a given program as compared to counting the no. of CPU clock cycle needed for executing that program

The avg. no of clock cycles per instruction (CPI) executed in a given program as per compared to counting the no of CPU clock cycles needed for executing that program

The avg. no of clock cycle per instruction (CPI) has been used as an alternate performance measurement

(3)

Clock cycle allow counting unit computation because the storage of computation result is synchronized with rising (trailing) clock edge

may be easier to count the no. of instruction executed in given program as compared to counting the no. of CPU clock cycle needed for executing that program

~~The avg. no of clock cycles per instruction (CPI) executed in a given program as per compared to counting the no of CPU clock cycles needed for executing that program~~

• The avg. no of clock cycle per instruction (CPI) has been used as an alternate performance measurement

$$CPI = \frac{\text{CPU clock cycle for program}}{\text{Instruction Count}}$$

(4)

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{clock cycle time}$$

$$\boxed{\text{CPU time} = \frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock scale}}}$$

It is known that the instruction set of a given machine consist of a no. of instruction categories

- ALU [Simple assignment and arithmetic and logic instruction]
- load
- Store
- branch etc

→ In the case of that the CPI for each instruction category is known, the over all CPI can be computed as

$$\boxed{\text{CPI} = \frac{\sum_{i=1}^n (\text{CPI}_i \times I_i)}{\text{Instruction count}}} \quad \begin{matrix} \text{1.} \\ \text{---} \\ \frac{\Sigma}{I=1} \end{matrix}$$

Where  $I_i$  = the no. of times as instruction of type  $i$  is executed in the program.  $\text{CPI}_i$

$\text{CPI}_i$  = Avg no. of clock cycles needed to execute such instruction

(5)

implie CPI for a machine A. Assume CPU's clock  
rate = 200 MHz

Instruction category	Percentage of Occurrence	No. of cycles Per Instruction
ALU	38 %.	1
load store	15 %.	3
Branch	42 %.	4
Other	5 %.	5

Assuming the execution of 100 instruction, the overall CPI can be computed as:-

$$CPI_a = \frac{\sum CPI_i \times I_i}{\text{Instruction Count}} = \frac{38 \times 1 + 15 \times 3 + 42 \times 4 + 5 \times 5}{100} = 2.76 \text{ msec}$$

- CPI Reflects the organisation and the instruction set Architecture of the processor and
- Instruction count reflects the instruction set Architecture and compiler technology used.

(6)

## MIPS

Million Instruction Per sec.

one alternative way to measure CPU performance is MIPS  
or million instruction per sec

$$\text{MIPS} = \frac{\text{Instruction count}}{\text{Execution time} \times 10^6} \quad \text{--- (1)} \quad | \text{ / find.}$$

Since

$$\text{Execution time} = \frac{\text{Instruction count} \times (\text{PI})}{\text{Clock Rate}} \quad | \text{ (total instruction)}$$

put the value

$$\text{MIPS} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6} \quad \text{--- (2)}$$

Since MIPS is a rate of operation per unit time

CPU performance can be specified as the inverse

of execution time with faster machine having  
a higher MIPS rating

(1)

Example 1 - Two M/C A and B

Clock rate of 200 MHz

Instruction Category	Percentage of occurrence		No of cycles per instruction		No. of ins. millions	
	(A)	(B)	(A)	(B)	(A)	(B)
38	35	1	1	8	10	
15	30	3	2	4	8	
42	15	4	3	2	2	
5	20	5	5	4	4	
					18	24

$$CPI_A = \frac{\sum_i^n CPI_i \times I_i}{\text{instruction count}}$$

$$= \frac{(38 \times 1) + (15 \times 3) + (42 \times 4) + (5 \times 5)}{100}$$

$$= 2.76$$

$$MIPS_A = \frac{\text{Clock Rate}}{CPI_A \times 10^6}$$

$$= \frac{200 \times 10^6}{2.76 \times 10^6}$$

$$= 70.24 \text{ Mips}$$

$$CPI_B = \frac{\sum_i^n CPI_i \times I_i}{\text{instruction count}}$$

$$= \frac{(35 \times 1) + (30 \times 2) + (15 \times 3) + (20 \times 5)}{100}$$

$$= 2.4$$

$$MIPS_B = \frac{\text{Clock rate}}{CPI_B \times 10^6}$$

$$= \frac{200 \times 10^6}{2.4 \times 10^6}$$

$$= 83.33 \text{ Mips}$$

task complete

## Performance in term of CPU Execution time

(8)

$$CPU_A = \frac{\text{Instruction count} \times CPI}{\text{clock cycle}}$$

$$= \frac{18 \times 10^6 \times 270}{200 \times 10^6}$$
$$= 0.24$$

$$CPU_B = \frac{\text{Instruction count} \times CPI}{\text{clock cycle}}$$

$$\frac{2.4 \times 10^6 \times 24}{200 \times 10^6}$$
$$= 0.28$$

$$CPU_B > CPU_A \quad \& \quad MIPS_B > MIPS_A$$

So it requires longer CPU Time to execute the same set of benchmark program.

## MFLOPS

(9)

### Performance matrix

It is another popular alternative to measure execution times is million floating point operation per sec<sup>k</sup> or MFLOPS (mega flops)

$$\text{MFLOPS} = \frac{\text{No. of floating point operation in a program}}{\text{Execution time} \times 10^6}$$

The MFLOPS rating is dependent on the machine and on the program, and same since MFLOPS are intended to measure floating point performance, they are not applicable outside that range.

for example:- Compiler has a MFLOPS rating of nearly zero no matter how fast the CPU is since compilers rarely use floating point arithmetic. When comparing the performance of different machines MFLOPS is not dependable b/w the set of floating point operation is not consistent across machines

Amdahl's law :- The performance gain that can be obtained by improving some portion of a computer. can be calculating using Amdahl's law

$$\text{Speed up} = \frac{\text{Performance new}}{\text{Performance old}}$$

or

$$= \frac{\text{Execution Time old}}{\text{Execution Time new}}$$

$$\therefore E = 1/p$$

$$\boxed{s_{\text{latency}}(s) = \frac{[1]}{[(1-p) + p/s]}}$$

## Derivation

A task whose resources are improved compared to an initial similar system can be split into 2 parts

## Tasks

1) → Part that don't benefit

2) → Part that Benefit

Ex A program that processes files from disk.

(a) part of the program that scan the directory of the disk

(b) the part passes each file to separate the thread for process

$T$  = Execution time of whole task before improvement

$\rho$  = portiona percentage of the execution time of task that Benefit

$(1-\rho)$  = The part that does not benefit

$$T = (1-\rho) T + \rho T \quad \rightarrow \textcircled{1}$$

if Speed up by factor  $S$

then

$$T_{(S)} = (1-\rho) T + \rho T / S \quad \rightarrow \textcircled{2}$$

$$\text{Slatency} = T / T_{(S)}$$

put the equation 1 & 2

$$\text{Slatency} = \frac{T}{(1-\rho) + \rho / S}$$

Ques.

- i) machine A runs a program in 10 second and machine B runs the same program in 15 second then compare the performance of A and B

## Necessity

Example If 30% of execution time may be due to the subject of speedup and improvement makes the effective part faster than find the overall speedup.

$$P = 30\% = 0.3$$

$$S = 2$$

$$\boxed{\text{Speed up} = \frac{1}{(1-P) + P/S}}$$

Ex for a serial program in two parts A and B,  $T_A = 3S$  and  $T_B = 1$  se.

If part B is made to run 5 times faster

$$T = \frac{T_B}{T_A f T_B} = \frac{1}{3} = 0.25$$

## Question

Consider a serial program in two parts (A and B) with execution time 3s and 1s respectively

(i) if part B is made to run 5 times faster

(ii) if part A is made to run 2 times faster

$$(A) S = 5 -$$

$$P = \frac{1}{4} = 0.25$$

$$\text{Speedup} = \frac{1}{1-0.25 + 0.25/5} = 1.25$$

$$(B) S = 2 - \quad P = 3/4 = 0.75$$

$$\text{Speedup} = \frac{1}{(1-0.75 + 0.75/2)} = 1.60$$

## Performance

Suppose a program takes 1 billions instruction to execute on a processor running at 3 GHz

Suppose also that 50% of the instruction executes in 3 clock cycle. 30% executes in 4 clock cycles and 20% executes in 5 clock cycles.

What is the execution time for the program or task

Given - No of instruction = 1 billion =  $10^9$

$$\text{Clock time} = \frac{1}{\text{clock rate}} = \frac{1}{3 \times 10^9} = 3.33 \times 10^{-10} \text{ sec}$$

CPU execution = No of instruction  $\times$  CPI  $\times$  clock time

value	frequency	Product
3	0.1	1.5
4	0.3	1.2
5	0.2	1.0

$$CPI = 3.7$$

$$\text{so CPU exec} = 10^9 \times 3.7 \times 0.1 \times 10^{-9}$$

CPU execution = 1.85 sec.

Ques- Suppose the processor in the previous example is redesigned so that all instruction that originally executed in 5 cycle now execute in 4 cycles. Due to changes in the circuitry the clock rate has to be decreased. With 1.9 GHz no change are made to the instruction set what is the overall percentage improvement.

1. slower clock rate implies worse performance the factor of the improvement

$$= \frac{1.9}{2.0}$$

② lower clock rate means higher throughput

value

$$3 \times 0.5 = 1.5$$

$$9 \times 0.3 = 1.2$$

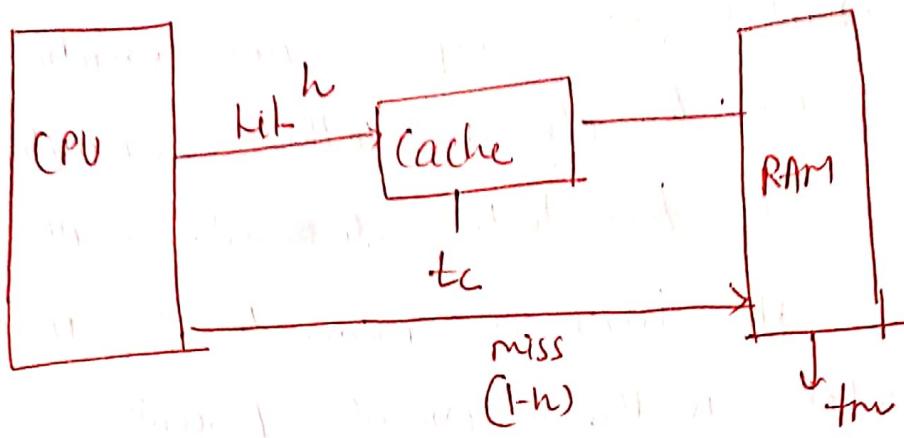
$$7 \times 0.2 = 1.4$$

$$\text{and } = \text{CPI} = 3.5$$

$$\text{Improvement factor} = \frac{\text{old}}{\text{new}} \times \frac{1.9}{2.0} = \frac{3.7 \times 1.9}{3.5 \times 2.0} = \frac{[ \text{old execution time} ]}{[ \text{new execution time} ]}$$

$$(1.064) (= 1.064)$$

## Cache Memory



- H is stand b/w RAM and CPU.
- H is also a RAM (Static Ram)
- H is faster Memory. No need different clk.
- It's contain most heavily used data

$$T_{eff} = h \cdot t_c + (1-h) [t_c + t_m]$$

One -

A cache is 10 Time faster than main memory and the cache can be used 90% of the time. How much Speed we gain by using cache

$M$  = Main memory access time

$C$  =  $M/10$  (Cache memory) access time

Total access time  $\rightarrow$ , hit miss

$$T_{eff} = \cancel{h} (h \cdot t_c) + (1-h) (t_c + t_m)$$

$$= \left( \frac{90}{100} \times \frac{M}{10} \right) + (1 - \frac{90}{100}) \left( \frac{M}{10} + M \right)$$

$$\text{Total access time} = \frac{9M}{10} + 1 \cdot \left( \frac{11}{10} M \right)$$

4

Saathi

Date \_\_\_\_\_ Benchmark

for marks:

Benchmark are programs that are used to evaluate the performance of computer for example

- ↳ kernel: These are small part of a real application
- ↳ Toy benchmarks: These are small programs typically used for education purpose
- ↳ Synthetic benchmarks: These are artificial programs built to match the profile and behaviors of real application
- ↳ Real benchmarks: These are real life application from various domains typically constructed to be portable across different systems while minimizing the impact of I/O on the Performance

or

A benchmark is the test that measure the performance of h/w s/w or computer. These tests can be used to help compare how well a product may do against other product. When comparing benchmarks, the higher the value of the result, the faster the component s/w or overall computer is.

(3)

Date 1/1/

Information

Saa

Date

(5) Yo benchmarks

(6) Database benchmarks

- Measure the throughput and response time of database management system

(7) Parallel benchmarks

- Used on m/c with multiple cores and/or processors or system consisting of multiple m/c

In Computer, a benchmark is the act of running a computer program, a set of programs or other operation in order to assess the relative performance of an object, normally by running a no. of standard test and trial against it.

The term benchmark is also commonly utilized for the purpose of elaborately designed benchmarking programs themselves.

Benchmarks provide a method of comparing the performance of various subsystem across different chip / system architecture.

## Purpose

As computer architecture advanced, it became more difficult to compare the performance of various computer system simply by looking at their specification.

Benchmark becomes particularly important in CPU design giving processor architecture the ability to measure and make tradeoffs in microarchitecture decision.

Ex

If a benchmark extracts the key algorithm of an application, it will contain the performance sensitive aspects of that application. Running this much smaller snippet on a cycle accurate simulator can give clues on how to improve performance.

### (1) Type of Bench mark

- (1) Real program
- word processing SW
- tool SW of CAD
- user application SW (MIS)

### (2) Component Benchmark / Micro benchmark

- Core routine consist of a relatively small and specific piece of code.
- measure performance of computer's basic component
- may be used for automatic detection of config parameter like no. of register, cache size, memory latency etc.

### (3) Kernel

- contain key code
- normally abstracted from actual program
- popular kernel - Linux, windows
- result are represented in MFlop/s

### (4) Synthetic Bench mark

These were the 1<sup>st</sup> general purpose industry standard computer Bench mark but they don't necessarily obtain high score on modern pipelined comp.

Instruction set  
(Instructions)

Execution time = no. of instruction  
(in cycles clock) per a program

$\begin{cases} \text{avg} \\ = \text{clock} \\ \text{cycles} \\ \text{per instruction} \end{cases}$

Suppose

Set

we have  
architecture

of 250 psec

Computer A has a clock cycle time  
and CPI of 2.0 for same program

Computer B

has CPI

of 1.2 for same program which

is faster

for

this program

ISA

LOAD

STOR

ADD

SUB

clock cycle time

$$= 250 \text{ pico sec}$$

$$= 250 \times 10^{-12} \text{ sec}$$

$$\text{CPI} = 2.0$$

(i)

$$\text{Execution time} = n * \cancel{\text{instructions}}$$

\* 2.0

$$* 250 \times 10^{-12}$$

$$\text{Execution time} = n * 250 \times 10^{-12}$$

B

$$\text{Clock cycle time} = 500 \times 10^{-12}$$

$$\text{CPI} = 1.2$$

$$\text{Execution time} = n * \cancel{1.2}$$

clock cycle

$$* 500 \times 10^{-12}$$

$$\text{Execution time} = n * 600 \times 10^{-12}$$

$$\frac{\text{Performance A}}{\text{Performance B}} = \frac{\text{Ex B}}{\text{Ex A}} =$$

$$\frac{n \times 600 \times 10^{-12}}{n \times 500 \times 10^{-12}}$$

$$\frac{P_A}{P_B} = 6/5 = 1.2$$

(10)

Better Definition is "distance travelled"

1 unit of computation (in distance) = 1 floating Point operation

Millions of floating point operations per sec. (MFlops)

$$= \frac{f}{T_e} \times 1000000$$

•  $T_e$  = executing time

$f$  = No of floating point instructions

Integer program = 0 MFLOPS

Example - Search, ~~sort~~ sorting, linking

(11)

If a computer A runs a program in 10 sec. and B runs the same program in 5 sec. then how much faster is B than A

$$E_A > E_B$$

$$\text{then } P_A < P_B$$

Because =   
 Performance =   
 Execution time

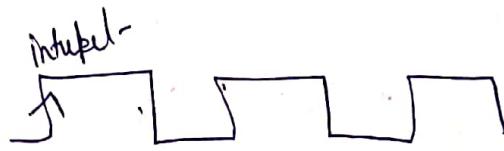
$$P_A = \frac{1}{10} \quad P_B = \frac{1}{5} \Rightarrow \frac{P_A}{P_B} = \frac{1}{2}$$

$$P_B = 2 P_A$$

B is 2 times faster than P\_A

Ex

clock cycle time



~~formula~~

$$\text{CPU execution time} = \text{CPU clock} * \text{clock cycle time}$$

Q

A program runs in 10 sec on computer A, which has a 2 GHz clock we are trying to help computer B which will run this program in 6 sec. The designer has determined that a substantial increase in clock rate is possible. But this increase will affect that rest of the CPU design causing computer B to require 1.2 times as many clock cycles as computer A for this program what clock rate should we tell the designer to

$$\text{clock frequency} = \frac{1}{\text{clock cycle time}} \quad (12)$$

Computer A

$$\text{Given} = \text{frequency} = 261 \text{ Hz}, \text{time} = 10 \text{ sec}$$

$$\text{Clock cycle time} = \frac{1}{261 \text{ Hz}} = 0.5 \text{ nsec}$$

$$\text{Computer} = (B) = 6 \text{ sec.}$$

$$\text{clock cycle time} = \frac{1}{\text{clock rate}} \quad (\text{increase frequency})$$

(A)

$$0.5 \rightarrow 1 \text{ clock (operation)} \\ 10 \text{ sec.} / 0.5 \text{ nsec} = 20 \times 10^9 \text{ clock cycle}$$

$$10 \text{ sec.} = \underline{20 \times 10^9 \text{ clock cycle}}$$

$$(B) \quad 1.2 \times 10^3 \text{ clock cycle.}$$

$$6 \text{ sec.} = 24 \times 10^9 \text{ clock cycle.}$$

$$1 \text{ clock cycle} = \underline{\underline{1 \text{ cycle}}}.$$

$$1 \text{ clock cycle} = \frac{1}{24 \times 10^9} \times 6 \text{ sec}$$

$$= 0.25 \text{ nsec}$$

$$\text{Rate} = \frac{1}{0.25 \text{ nsec.}}$$

$$= \frac{100}{0.25 \times 10^{-9} \text{ sec.}}$$

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