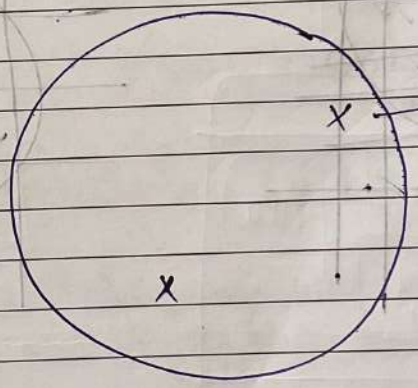
Advanced Computer Architecture (TCS-704)

Unit-I

* **Fabrication yield:**

$$\text{yield} = \frac{\text{Working chips}}{\text{chips on wafer}}$$

Fabrication yield is the percentage of chips at the end that we get to sell. So the yield is the no. of working chips we get at the end divided by the total number of chips that we actually had on a wafer.



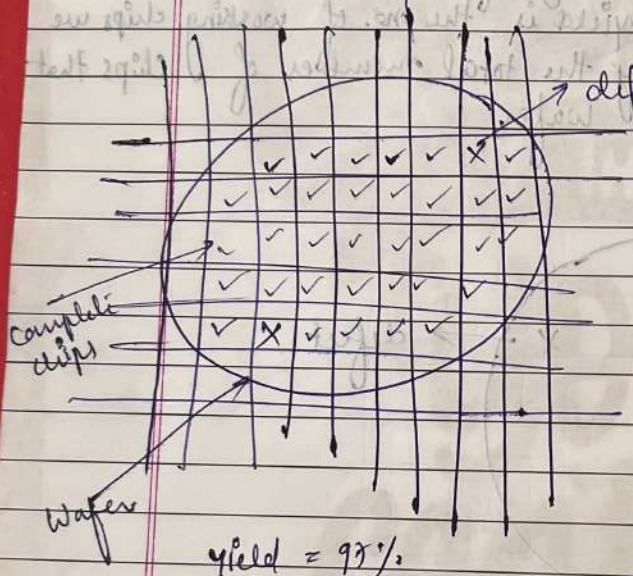
A wafer consists of small & few spots called defects. It is either because silicon was impure to begin with or because there was something wrong with the process at that place so in the end, once we get the manufacture wafer pretty much through a given manufacture process there will be some no. of expected defects per wafer. So let say this wafer has two defects.

Let say that ~~this~~ defect we divide this wafer into a lot of

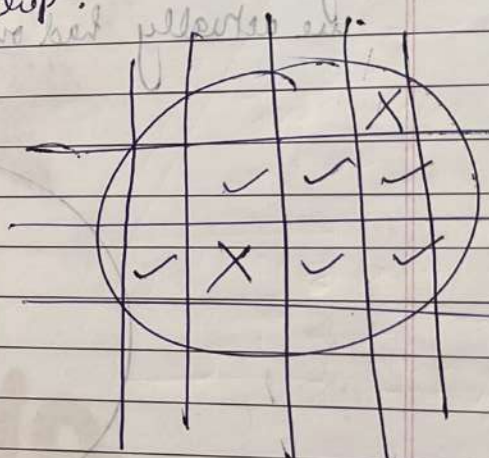
small chips. So, note that some of the chips are not really complete because of the wafer roundness. We can only use the complete ones. So this is a good chip. So we got 62 working chips and 2 bad chips.

So our yield is $\frac{62}{64}$ which is about 97%.

Now, let's start with the same wafer and the same defects but now let's have much larger chips. Now we have six working and 2 bad chips so our yield is 75%.



yield = 97%.



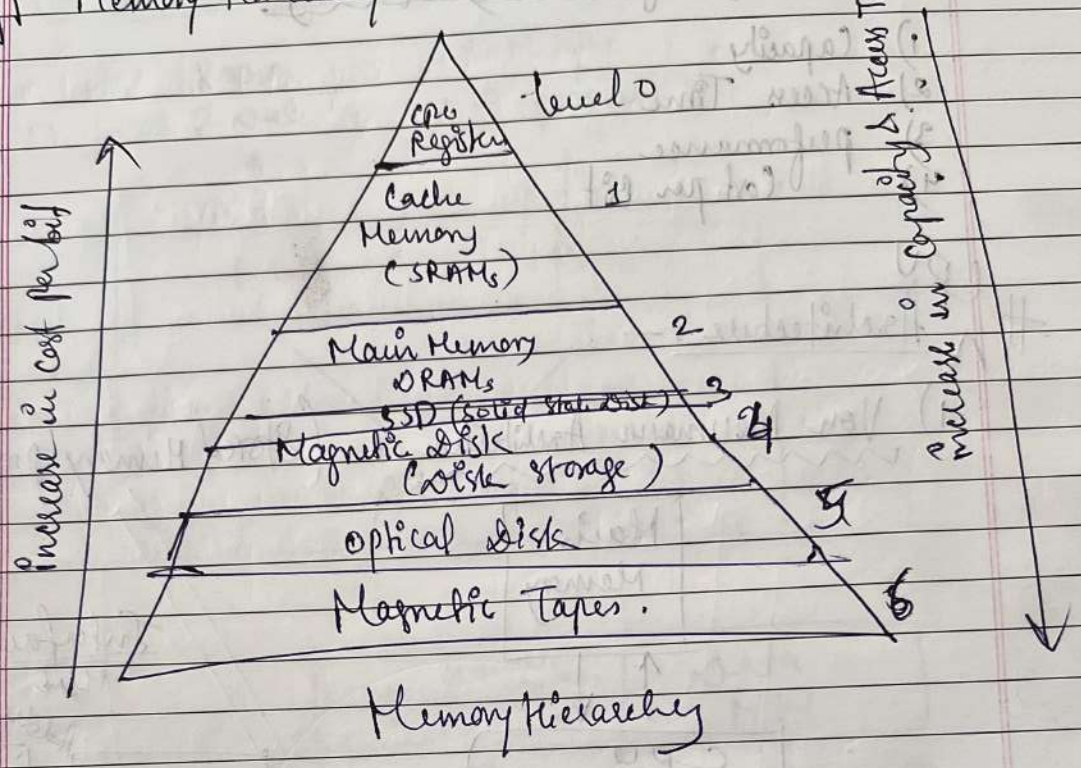
yield 75%.

Computer organization and architecture

Components of computer architecture

CPU — Memory — I/P devices — Storage devices (Hard drives)

Memory Hierarchy :-



Memory Hierarchy

1) Registers : capacity 16 - 64 bits . located in CPU.

2) Cache Memory : Store frequency used data & Instruction set that have been recently accessed from Main Memory.

3) Main Memory : RAM.
Larger storage capacity than cache but slower.
Store instructions that are currently in use by the CPU.

4) SSD and HDD - (non-volatile memory)
Store data and instructions not currently in use by CPU.

5) Magnetic disks - circular plates.

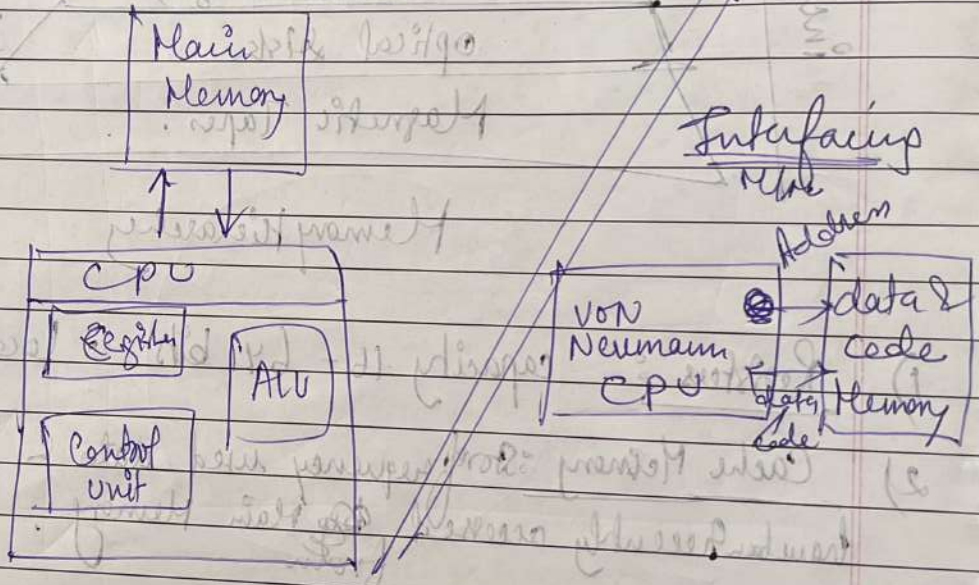
6) Magnetic tape - Magnetic recording device.

Characteristics of Memory Hierarchy:

1) Capacity
2) Access Time
3) Performance
4) Cost per bit.

CPU Architecture:-

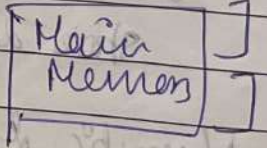
1) Von Neumann Architecture: (Stored Memory program)



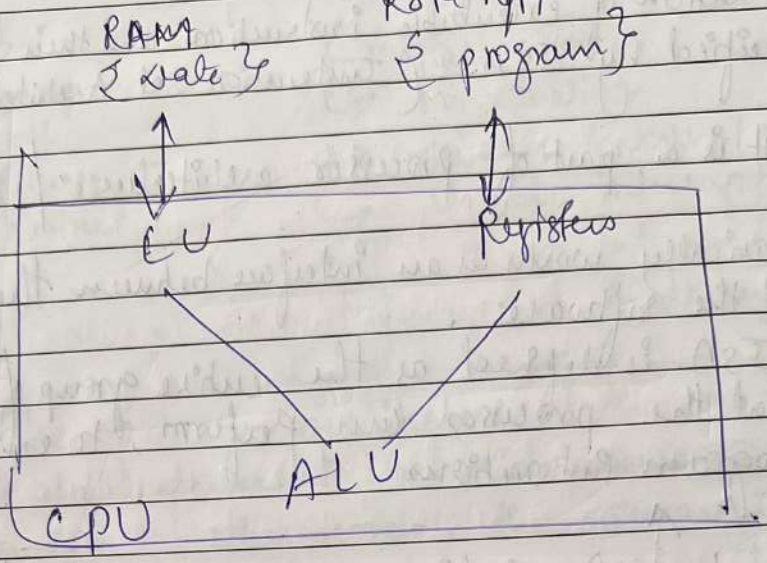
⇒ Data and program (set of instructions) are stored

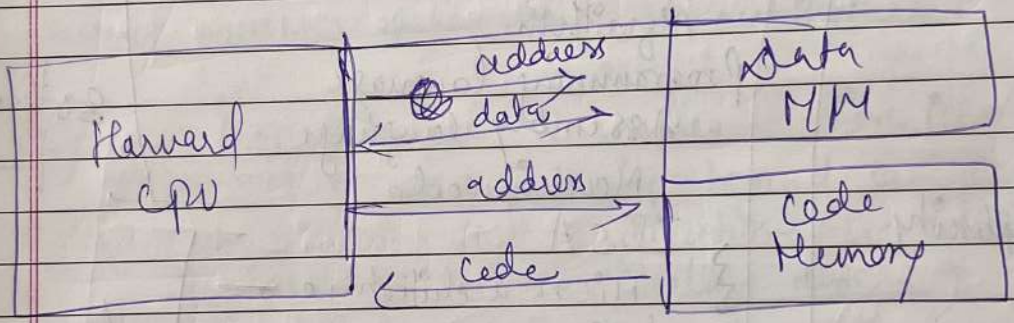
Data Eg: int a = 10
b = 20
c = a + b;

→ { Stored in Main M/M }



(2) Harvard architecture :—





Von-Neumann	Harvard.
→ ~~Mor~~ less complexity while interfacing	→ More
→ data & code cannot be accessed at same time (slow)	→ faster.

Computer architecture Design Principles.
↳ VON-Neumann architecture
↳ ISA
↳ Parallel processing.

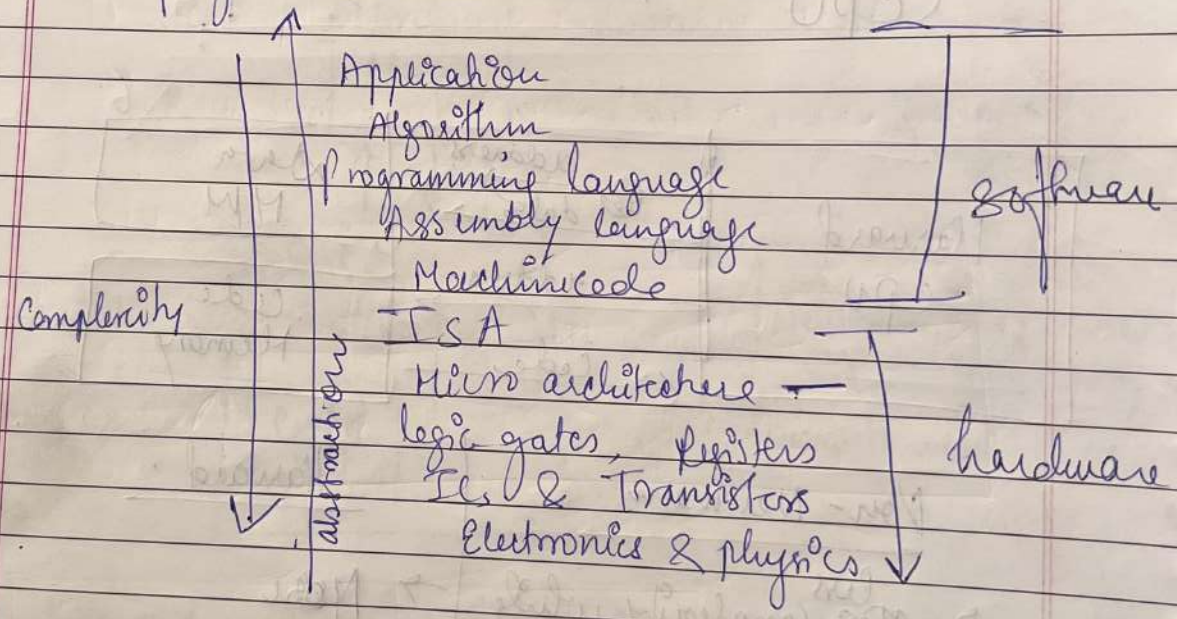
ISA :

↳ Collection of executable instructions and their format are specified by ISA (Instruction set Architectures).

↳ It is a part of processor architecture / cpu architecture.

↳ basically works as an interface between the hardware and the software.

↳ ISA is defined as the entire group of commands that the processor can perform, to execute the program instructions.



8085 Instruction Set

- Data transfer
  - ↳ load Register
  - ↳ Memory to register
  - ↳ Register to Register
  - ↳ device to Acc.
- Arithmetic operations
  - ↳ Addition
  - ↳ Subtraction
  - ↳ Increment
  - ↳ decrement
- Logical operations
  - → AND → OR complement → XOR Rotate Compare

DATE / /
PAGE

Branching

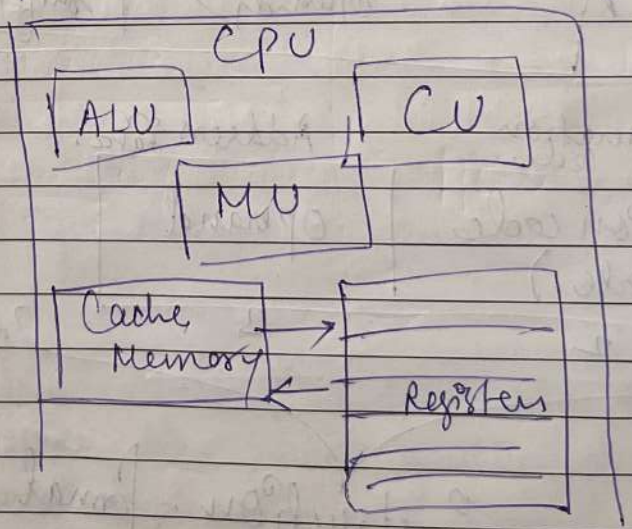
- ↳ Jump
- ↳ Call
- ↳ Return
- ↳ Restart.

Machine control.

- ↳ Halt process
- ↳ No operation
- ↳ Disable interrupt
- ↳ Unable interrupt
- ↳ Read interrupt.
- ↳ Set Interrupt.

The processor architecture at the hardware level is also referred to as Microprocessor architecture. In other words, the microarchitecture is the hardware circuitry of the processor chip that implements one particular instruction set architecture.

↳ Microarchitecture examples - Core i3 / core i5 etc.
(The chip manufacturing brand can decide how to implement the instruction set architecture into the processor microarchitecture design).



* Parallel processing :-

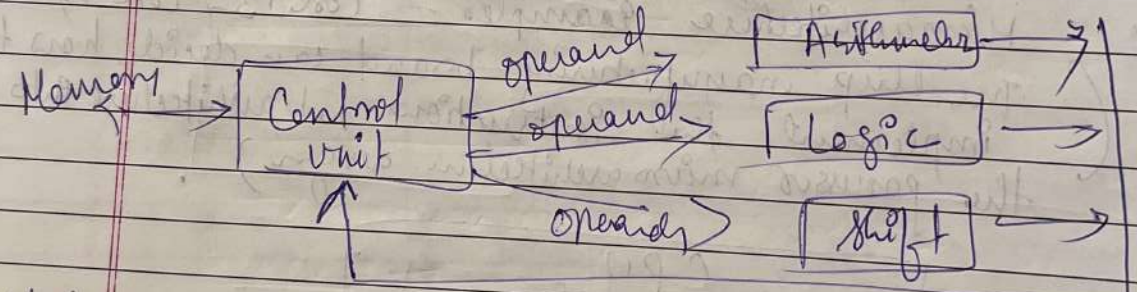
Instead of processing each instruction sequentially, we use a different technique called parallel processing :

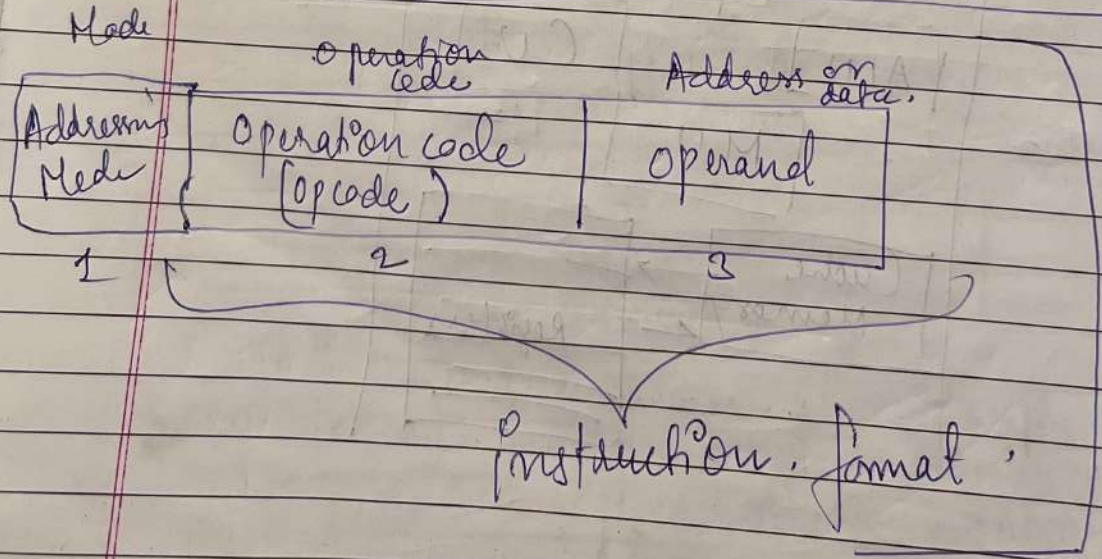
Parallel processing :— This technique enables a system to perform concurrent data processing to achieve faster execution time.

Ex: 1) While an instruction is being executed in the ALU the next instruction can be read from memory.

2) The system can have two or more ALUs and be able to execute two or more instructions at the same time.

3) The system may have two or more processors operating concurrently.





↳ device to ALU,

* Challenges in Computer Architecture :-

1) Affected by CPU, Memory, storage devices and bandwidth.
2) Requires More energy to function
3) Higher energy cost & environmental impacts.

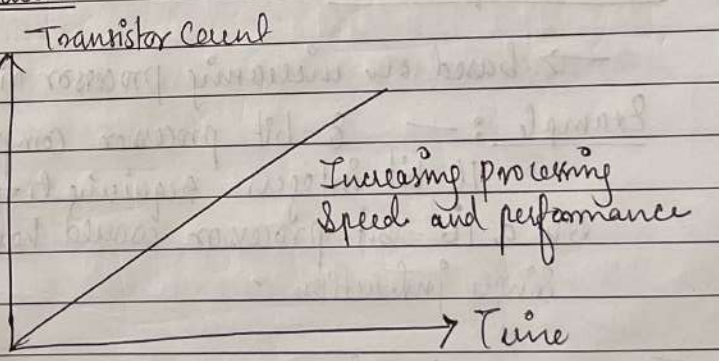
* Emerging Trends in Computer Architecture :-

1) ML & AI helps in data Analysis, Natural language processing & computer vision.

2) Quantum Computing uses quantum bits (qubits) to perform Computations.

3) 3D Stacked Memory is a technology involving stacking Multiple layers of memory chips on top of each other to increase memory systems' storage density & bandwidth.

Moore's law :-



→ Moore's law principle states - that since the number of transistor on a silicon chips roughly doubles every two years, the performance and capabilities of computers will continue to increase while the price of computer decreases.

→ It is a prediction made by American Engineer Gordan Moore in 1965.

Example Intel Moore's law (processor)

1971 - Intel introduced the intel 4004 with a transistor Count of 2250
1974 - (Transistor 6000) (intel ~~8080~~ ~~4004~~)
1976 - (Transistor 6500) (intel 8085)
1978 - (Transistor 29000) (Intel 8086)
1980 - (Transistor 50000) (Intel 8051)
1982 - (Transistor 55000) Intel 80186
1985 - (Transistor 275000) (Intel 80386)

This came to be known as the Intel Moore's law.
It is evident that there have been increments in the transistor counts over the years with a period of two years.

Parallel Computing :-

Types of parallelisms :-

1) Bit-level parallelism :

→ based on increasing processor size.

Example :- 8-bit processor computing the sum of two 16 bit integers, requiring two instructions. But a 16-bit processor would have performed it in single instruction.

2) Instruction-level parallelism :-

A processor can only address less than one instruction for each clock cycle phase. These instructions can be re-ordered and grouped which are later on executed concurrently without affecting the result of the program.

3) Task parallelism :- Tasks parallelism employs the decomposition of a task into subtasks and then allocating each of the subtasks for execution. The processors perform the execution of subtasks concurrently.

4) Data level parallelism :-
Instructions from a single stream operate concurrently on several data.

Applications :-
1) Databases and Data Mining
2) Real time Simulation of systems
3) Advanced graphics, augmented reality and virtual reality.

Iron law of Performance :-

CPU Time = # of Instructions per program
× Cycles per Instruction
× Clock cycle Time

CPU Time = $\frac{\#\text{ of instructions}}{\text{program}} \cdot \frac{\#\text{ of cycle}}{\#\text{ of instructions}} \cdot \frac{\text{Seconds}}{\#\text{ of cycle}}$

CPU Time = $\frac{\text{How many seconds}}{\text{program}}$
= $\frac{\text{Seconds}}{\text{program}}$.

DATE / /
PAGE

Components of CPU Time and Computer Architecture :-

of Instruction per program } Affected by the Algorithm Compiler Instruction set

X

Cycles per Instructions } Instruction set processor design

X

Clock cycle Time } process design circuit design Transistor physics.

Amdahl's law :—

⇒ Amdahl's law is a formula used to find
— Maximum improvement possible by improving particular part of a system.

⇒ In Ilel computing, it is mainly used to predict the
— The theoretical max speed up for program processing using multiple processors.

⇒ It is related to speed up of Ilel computers
when a program run on Ilel computer then Computation may be → Serial
→ Parallel
→ both

⇒ There will be a certain part of a program need to be run sequentially

Consider Sequential fraction of program = F
Ilel Computation of program = (1 - F)

Speed up of Parallel computer

$S(P) \leq \frac{1}{f + \frac{1-f}{p}}$

⇒ Named after computer Scientist Gene Amdahl in 1967

Formula : $S = \frac{1}{(1-P) + \frac{P}{N}}$

S = Speed up of the system
P = Proportion of the system that can be improved
N = N is the Number of processors in the system.

Example If a system has a single bottleneck that occupies 20% of the total execution time and we add 4 more processors to the system, the speedup would be —

$$S = \frac{1}{(1-P) + \frac{P}{N}}$$

$$= \frac{1}{(1-0.2) + \left(\frac{0.2}{5}\right)}$$

$$= \frac{1}{0.8 + 0.04}$$

$$= \frac{1}{0.84}$$

$$= 1.19$$

Overall performance of the sys. would improve by about (19%) with addition of 4 processors

Lhadma's law :-

AMDAHL's law : Make common case fast
LHADMA's law : Do not mess up uncommon case too Badly.

Example

We can achieve an improvement of a factor of 2, on 90% of the execution time.

But at the cost of slowing down the the rest by 10x (10%)

$$\text{Speed up} = \frac{1}{\frac{0.1}{0.1} + \frac{0.9}{2}} = \frac{1}{1+0.45} = 0.7$$

or

$$\text{Speed up} = \frac{1}{\frac{10\%}{0.1} + \frac{90\%}{2}} = 0.7$$

overall slow down.