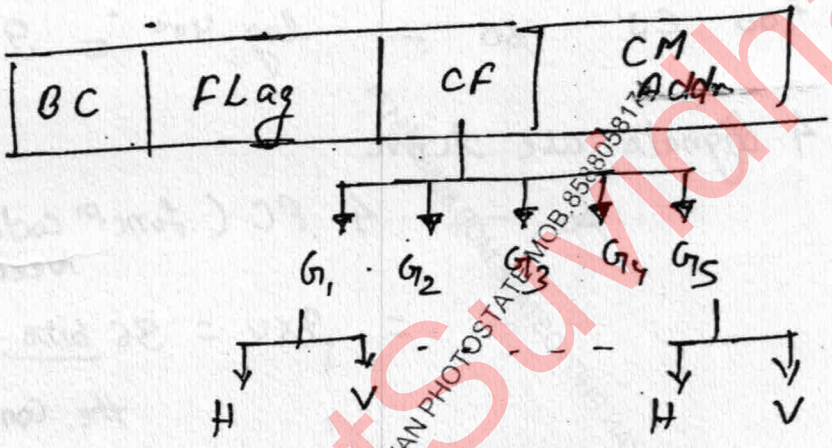
Que:- A Hypothetical CU. supports 5 groups of mutually exclusive control signals described below :-

Group	G_1	G_2	G_3	G_4	G_5
CS.	1	10	20	30	40

How many bits are saved by uping vertical over Horizontal Programming ?



bits required when the groups are Horizontal : $1 + 10 + 20 + 30 + 40$

$\Rightarrow$ 101 bits

bits req. when the groups are vertical : $\log_2 1 + \log_2 10 + \log_2 20 + \log_2 30 + \log_2 40$

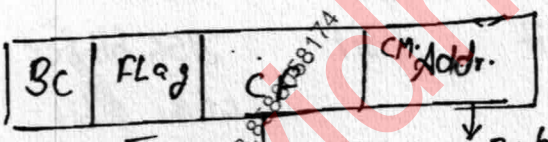
$= 1 + 4 + 5 + 5 + 6$

$= 21$ bits

# of bits saved $= 101 - 21$

$= 80$ bits

Que :- Control field of a micro-Instruction supports 2 groups of control signal, In which group1 indicate None/1 of a 400 control signals and group 2 indicates atmost 8 signals from the Remaining micro-Instruction designed with a unconditional jump with a Target address of 5 Hexa decimal digits. what is the size of a micro-Instruction.



20 bits → 5 Hexa digits
4 bits × 5 = 20 bits

G_1 (vertical) → None/1 → 400 CS's → 9 bits

G_2 → Atmost 8 CS's (more than 1) so Horizontal → 32 bits

8 bits

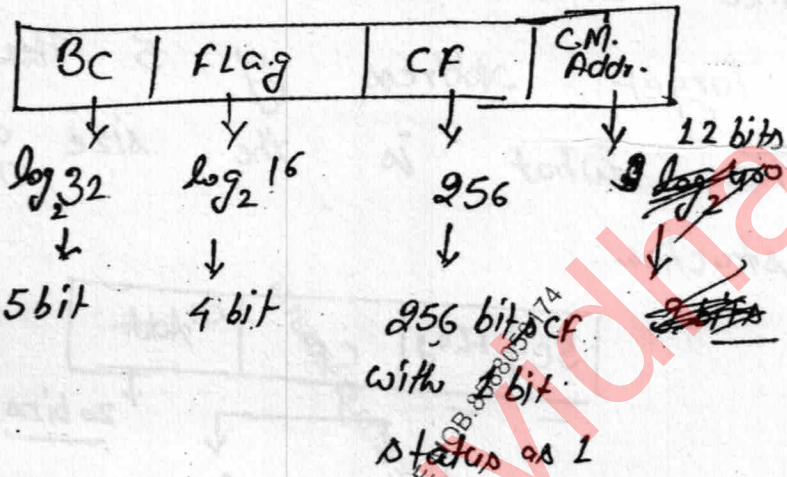
Instⁿ size : 9 + 8 + 20 = 37 bits.

** Que :- A CPU Contain Instⁿ set size of 400. Each Instⁿ takes 10 cycles to complete. Hardware contain 256 CS's. Implemented using Horizontal programming. micro Instⁿ designed with conditional jump coding. In the CPU 16 Flags are Present and 32 Branch conditions are used.

what is the size of CAR and COR Registers when 1-Address format is used to control the Branch logic.

Instⁿ set size = 400

Control signals = 956



Default 1 CS's Active

Instⁿ in CPU = 400

cycles / Instⁿ = 10

{# uopⁿ / Instⁿ} 2

∴ Total # u operations in the CPU = 400 × 10 = 4000 uopˢ

⇓

Programmed in Control m/y

∴ CM size = 4000 CW / u Instⁿ / u opˢ

So Addr. size = $\log_2 4000$

= $\log_2 2^{12}$ = 12 bit (CAR)

(COR)

uCW size = 277 bits

Que:- A Hypothetical Control unit uses the following control Expression to generate the Xin control signal.

$$Xin = T_1 + T_2(I_2 + I_3) + T_3 + T_4(I_1 + I_4)$$

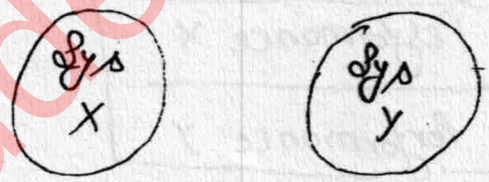
During the Execution of I_4 Inst^{n}, in which micro-operation Xin signal is ~~disabling~~ disabled in the Bape Hardware:

Ans:- In "T_2".

"Performance Analysis"

* performance is a Indirect measurement, depends on the Execution time.
* Amount of the time required to complete the Program is called as Execution time.

eg:



{Task} {Task}

$ET_x = 5ns$ $ET_y = 10ns$

$$P \propto \frac{1}{ET}$$

Execution time means CPU time

i.e. CPU time = # Seconds / Program

i.e. CPU time = # Inst^{n}/Program * # Cycles/Inst^{n} * # Seconds/Cycle

Inst^{n}/Program → Inst^{n} Count (IC)

Cycles/Inst^{n} → CPI

Seconds/Cycle → Cycle time

$$\text{CPU time} = IC * CPI * \text{Cycle time}$$

* In the Program different Instructions takes different cycles to complete therefore Program execution time is calculated as :

$$\text{CPU time} = \sum (IC_i * CPI_i) \text{ Cycle time}$$

i : Type of Inst^{n}

** "Speedup (S) factor" is used to measure the performance gain of a CPU i.e.

$$S = \frac{\text{Performance X}}{\text{Performance Y}}$$

$$S = \frac{\frac{1}{ET_x}}{\frac{1}{ET_y}}$$

$$S = \frac{ET_y}{ET_x}$$

$S = n$ ⟶ Some constant

{ System "X" will comp in time faster than System "y". }

* In the Processor design different Quantative principle are present, used to improve the performance. One of the Principle states that improve the performance within a cost & price limit while making common case fast, called as "Amdhal's Law"

* Amdhal's Law states that improve the performance by enhance the part of a System. that is measured using the Speed up (S).

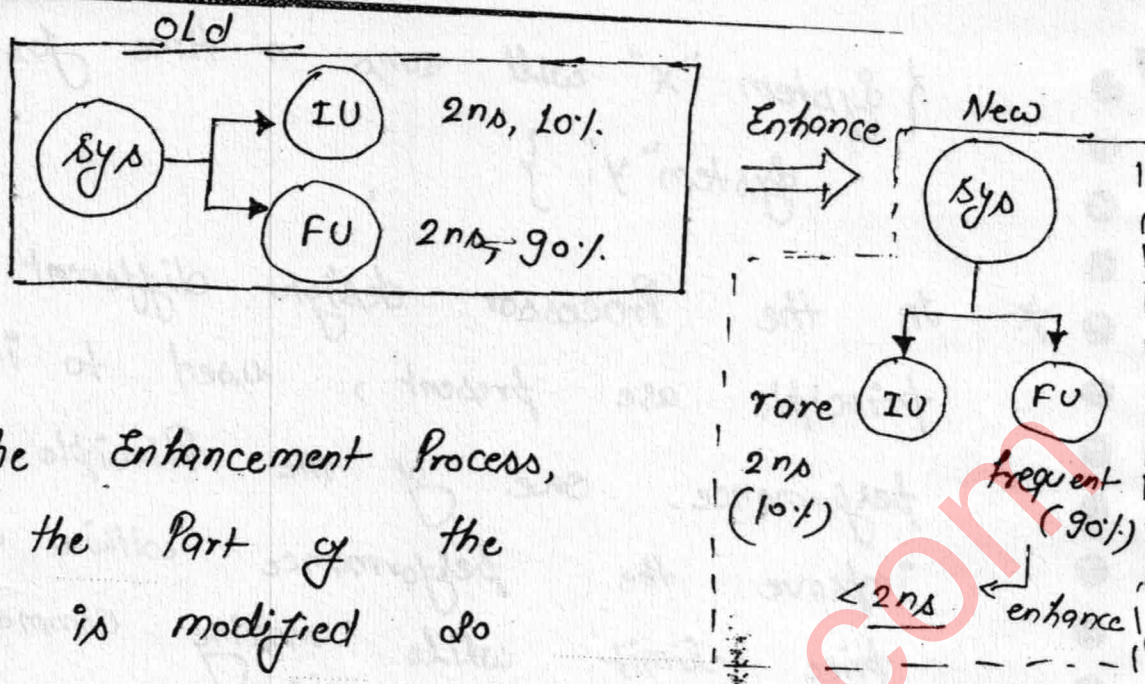
$$\therefore S_{overall} = \frac{\text{Performace of A System with Enhance (New)}}{\text{Performace of A System without Enhance (old)}}$$

$$\therefore S_{overall} = \frac{\frac{1}{ET_{New}}}{\frac{1}{ET_{old}}}$$

**

$$\therefore \boxed{S_{overall} = \frac{ET_{old}}{ET_{New}}} \quad \text{---(1)}$$

eg:



★ In the Enhancement Process, only the Part of the System is modified so New System contain 2 Portions :-

① unEnhanced Portion

② Enhanced Portion

therefore,

$$ET_{New} = E.T. \text{ of unEnhanced Portion} + E.T. \text{ of Enhanced Portion} \quad \text{— ②}$$

★ To calculate the ET_{New}, two Parameters are required :-

(1) fraction enhance (F)

(2) Speed up enhance (S)

★ "Fraction enhance" indicates that, How much part of the System is modified. i.e.

F : Enhanced Portion

(1-F) : unenhanced Portion

★ Speedup enhance indicates that, How much performance is improved in the Enhanced part i.e.

$$S = \frac{\text{Performance of New 'F'}}{\text{Performance of Old 'F'}}$$

$$S = \frac{\frac{1}{\text{ET of New 'F'}}}{\frac{1}{\text{ET of Old 'F'}}}$$

$$S = \frac{\text{E.T. of old 'F'}}{\text{E.T. of New 'F'}}$$

$$\therefore \text{E.T. of New F} = \frac{\text{E.T. of old 'F'}}{S}$$

:- Substitute the Above values in equation - ②

★★

$$ET_{New} = \text{E.T. of old '(1-F)'} + \frac{\text{E.T. of old 'F'}}{S}$$

:- Substitute the Above value in equation - ①

★★

$$S_{overall} = \frac{\text{E.T. old}}{\text{E.T. of old '(1-F)'} + \frac{\text{E.T. of old 'F'}}{S}}$$

★ To cover the complete system, let up consider the relative data i.e.

$$\text{System} = 100\%$$

$$S_{overall} = \frac{100\%}{(100\% - F) + \frac{F}{S}}$$

★★★

$$S_{overall} = \frac{1}{(1-F) + \frac{F}{S}} = \left[(1-F) + \frac{F}{S}\right]^{-1}$$

★ when $E.T._{old} = 1$ then,

ET_{New} always < 1 therefore,

$\therefore$ $S_{overall}$ always > 1

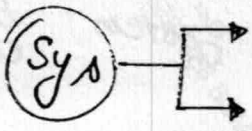
★ when the system contain multiple enhancement then performance gain is calculated as:

$$S_{overall} = \left[(1 - \Sigma F_i) + \Sigma \frac{F_i}{s_i}\right]^{-1}$$

i : # of enhancements.

Que:- Consider a computer system used in the Scientific Application area, Application Program refers the Integer unit and Floating Point unit during the Execution. Floating point unit is Enhanced then it runs 2 times faster but in the Application Program only the 10% of the Instn are floating point. what is the performance gain?

Sol?:-



Floating Point → 10% → F (Rare) given:

Integer → 90% → (1-F) (frequent)

Speedup 2 ns.

$$\text{Speedup}_{overall} = \left[(1-F) + \frac{F}{S} \right]^{-1}$$

$$= \left[(1-0.1) + \frac{0.1}{2} \right]^{-1}$$

$$= 1.05.$$

★ Note:- In the above Enhancement, Rare component is modified in the system therefore there is no considerable performance gain. So corrective Enhancement is frequent case modification i.e. Enhance frequent case. i.e. Intger

$$S_{overall} = \left[(1-f) + \frac{f}{S} \right]^{-1}$$

$$= \left[(1-0.9) + \frac{0.9}{2} \right]^{-1}$$

$$= \underline{1.81}$$

Que :- In a Memory design, cache memory is 14 times faster than main-memory and it is referred by the CPU 80% of the time. How much performance is improved in the system by using the cache-memory.

Soln :-

$S = 14 \leftarrow$ speed up.

Sys → cache – 80% → f

main – 20% ← (1-f)

$$S_{overall} = \left[(1-0.8) + \frac{0.8}{14} \right]^{-1}$$

$$= \left[(0.2) + \frac{0.8}{14} \right]^{-1}$$

$$= \underline{3.89}$$

Que:- Three Enhancements are proposed for a New Architecture with the following freedoms :

$s_1 = 30$

$s_2 = 20$

$s_3 = 15.$

Enhancement 1 & 2 each usable of 25% of the time. and Enhancement 3 will be used 45% of the time. then what is the Performance gain ?

Soln :-

$F_1 = 0.25, \quad F_2 = 0.25$

$F_3 = 0.45$

$$S_{overall} = \left[1 - (F_1 + F_2 + F_3) + \frac{F_1}{S_1} + \frac{F_2}{S_2} + \frac{F_3}{S_3} \right]^{-1}$$

$$= \left[1 - (0.25 + 0.25 + 0.45) + \frac{0.25}{30} + \frac{0.25}{20} + \frac{0.45}{15} \right]^{-1}$$

$$= \left[0.05 + \frac{0.25}{30} + \frac{0.25}{20} + \frac{0.45}{15} \right]^{-1}$$

$$= 9.95$$

Que:- Consider a computer System used in the Application Area to Process the digital Signals. In the System 40% computations are parallelized to reduce the Execution time

that Portion of the System speedup is 4. what is the performance gain in the System?

Soln:-

$F = 40\%$

$S = 4$

$$S_{overall} = \left[(1-F) + \frac{F}{S}\right]^{-1}$$

$$= \underline{1.42}.$$

Que:- Consider 4.2 ns clock cycle processor, Consumes 8 cycles for Load and Store Instructions, 6 cycles for ALU Instⁿ. 3 cycles for Branch Instⁿ. Relative frequencies of these Instructions are 30%, 50%, and 20% respectively.

a) what is the Performance (In MIPS)

↓

million Instⁿ/sec.

b) If the CPU is enhanced to make the Average CPI as 1, In this Process clock cycle time is increased upto 40% then what is the Performance gain.

8 cycles for Load & store

6 cycles for ALU

3 cycles for Branch.

CPU time = $(0.3 * 8) + (0.5) * 6 + (0.2) * 3$

$= 2.4 + 3.0 + 0.6$

Avg Inst^{n} E.T $= \Sigma (IC_i * CPI_i) *$ cycle time

$= (6.0) * 4.2\ ns$

$= 25.2\ ns$

1 Inst^{n} $\longrightarrow$ 25.2 ns

Inst^{n} in 1 sec. $\longrightarrow$ $\frac{1}{25.2\ ns} = \frac{1}{25.2 \times 10^{-9}}$ Inst/sec

$\frac{1000}{25.2} * 10^6$

$= 0.03968 * 10^9$

$= 39.68 * 10^6 = 36.68$ MIPS

b.) $ET_{New\ CPU} = \Sigma (IC_i * CPI_i) *$ cycle time

$= [(0.3 * 1) + (0.5 * 1) + (0.2 * 1)] * 4.2\ ns + [40\% * 4.2\ ns]$

$= 5.88\ ns$

Speedup $= \frac{E.T_{old}}{E.T_{New}} = \frac{25.2\ ns}{5.88\ ns} = 4.285$

Date
18/08/2017

"High Performanced CPU Design" :-

★ High Performanced CPU exploits the Con-currency.

★ Concurrency means two or more instructions execution at a time.

★ According to a Flynn's Classification, Computer Architecture is divided into 4 types :

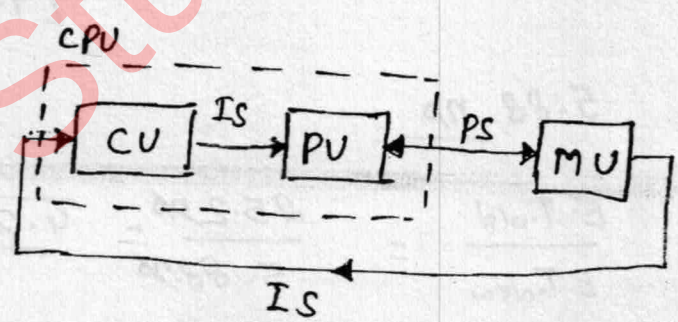
1.) SISD { Single Instruction Stream & Single data Stream }
2.) SIMD { Single Inst^{n} Stream and multiple data streams }
3.) MISD { multiple Inst^{n} Stream and Single data Stream }
4.) MIMD { multiple Inst^{n} Stream and multiple data streams }

Short forms :-

CPU { CU : Control unit
PU : Processing unit(ALU)
MU : memory unit

IS : Inst^{n} Stream
DS : Data Stream.

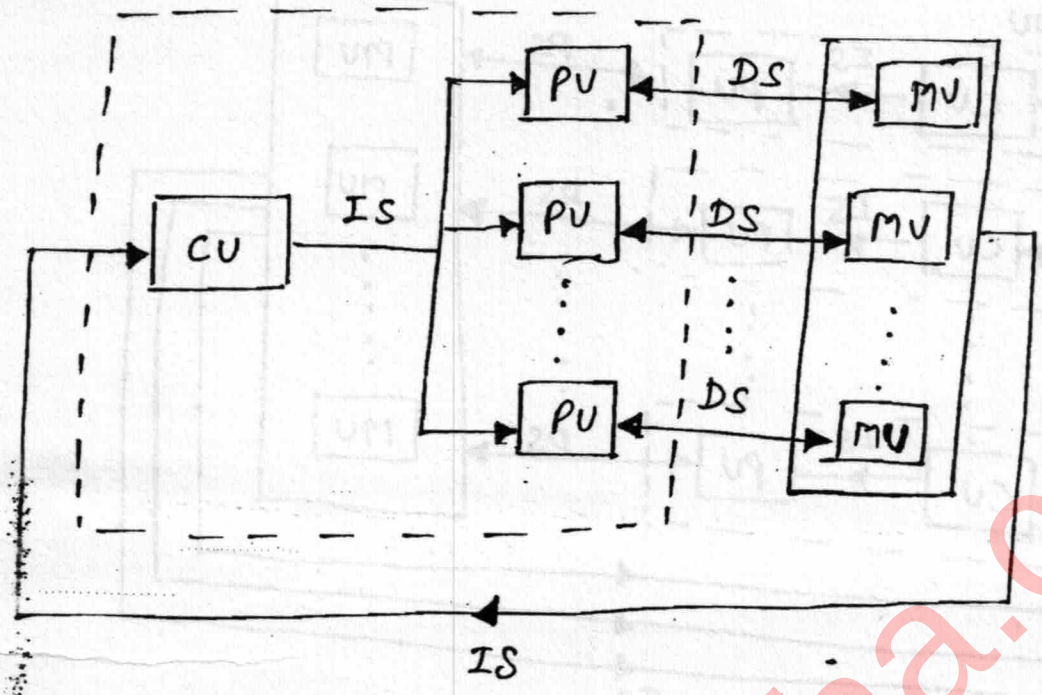
SISD :-



Implemented as a uni-processor System.

eg :- 8085 up
8086 up

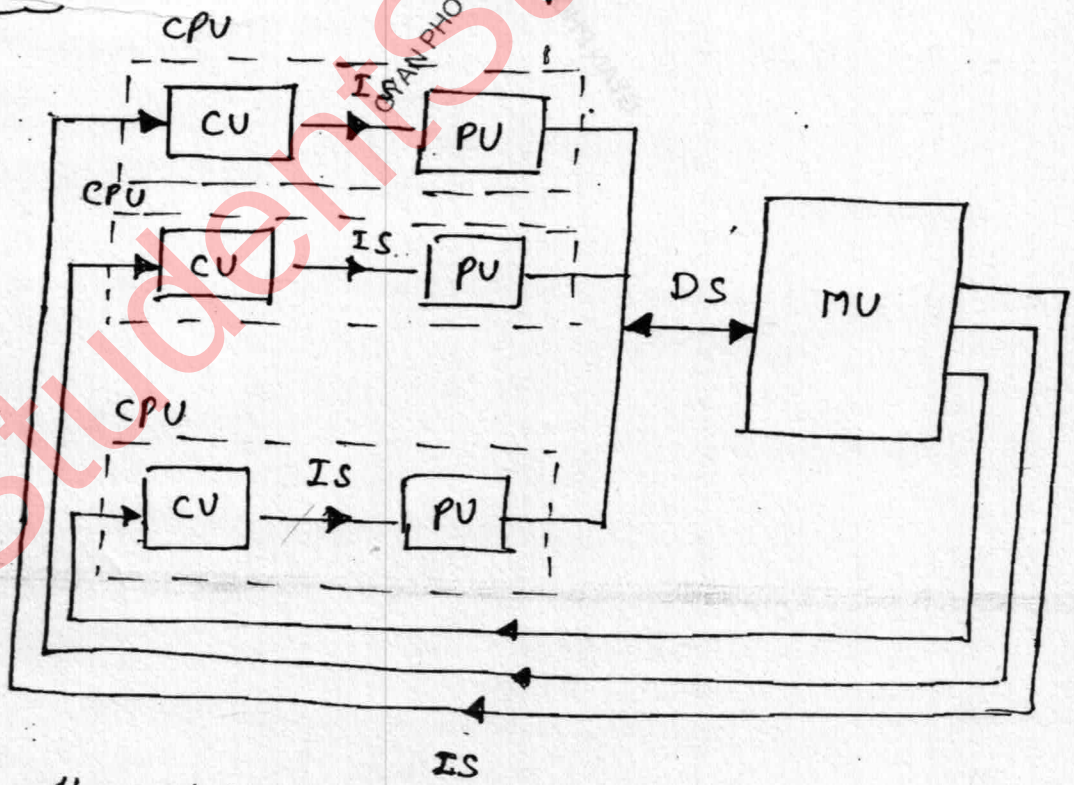
SIMD :-



Implemented as an Array Processor System

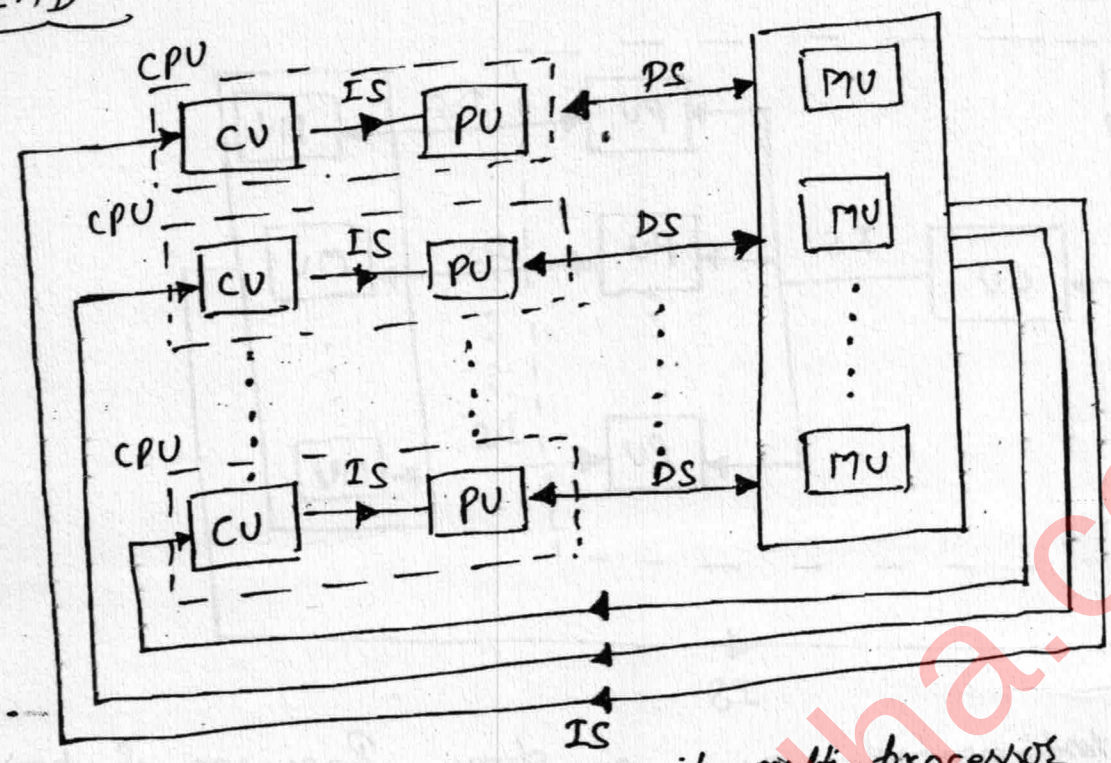
eg:- Staran Processor
PEPE Processor

MISD :-



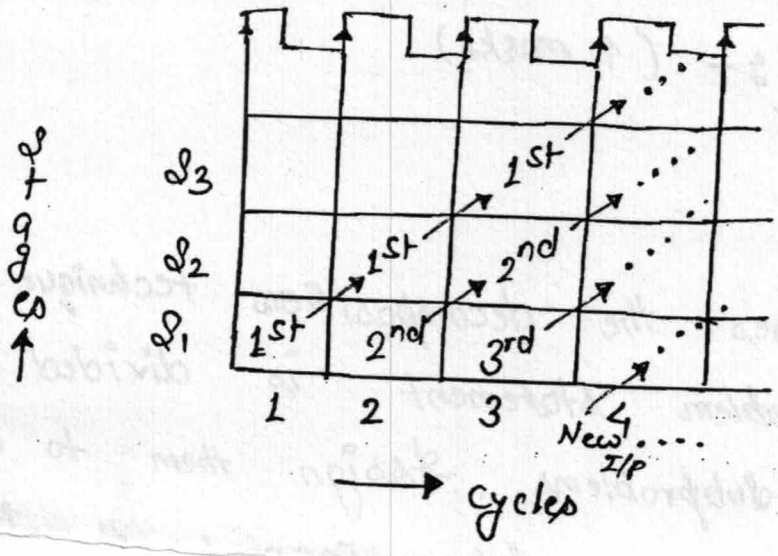
:- It contain multiprocessors, but only one Processor is in the upc. It is not yet implemented.

MIMD :-



:- Implemented as a ~~unit~~ multi-processor system.

eg:- Cray Processor
Cyber Processor.



★ Successful characteristic of a ~~the~~ Pipeline is, In Every New cycle, New input must be inserted to pipeline therefore CPI = 1

Design :-

→ Pipeline has two ends :-

i) Input End.

ii) Output End.

★ Between these ends multiple pipes are inter-Connected to satisfy the objective of a Pipeline. Each Pipe in the Pipeline is called as "stage" or "segment."

2.) Interface Register is used between the stages to hold the intermediate output. It is also name as buffer or latch or pipeline Registers.

3.) In the Pipeline design, all the stages along with a interface Registes is controlled by the "common clock" so clock adjustment