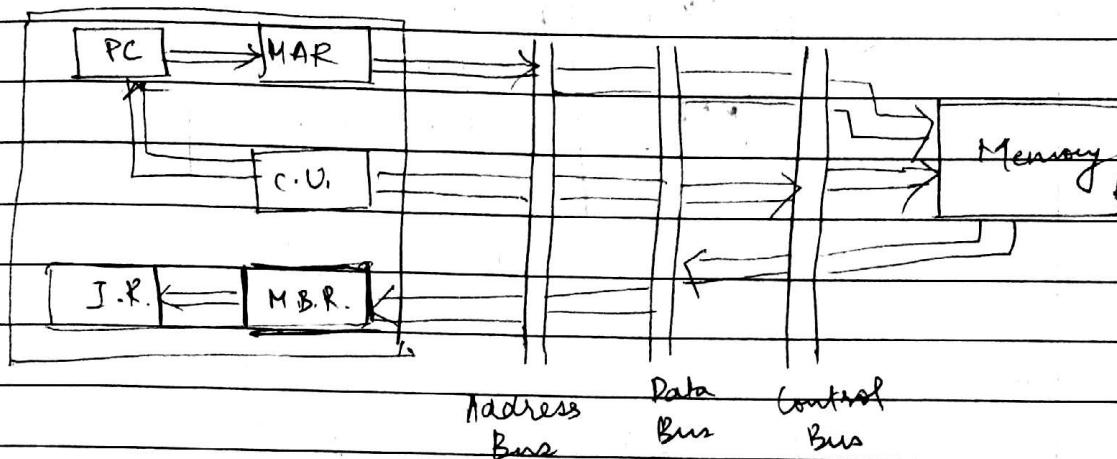


Data Flow

The exact seq. of events during an instruction cycle depends on the design of the processor.

Fetch cycle

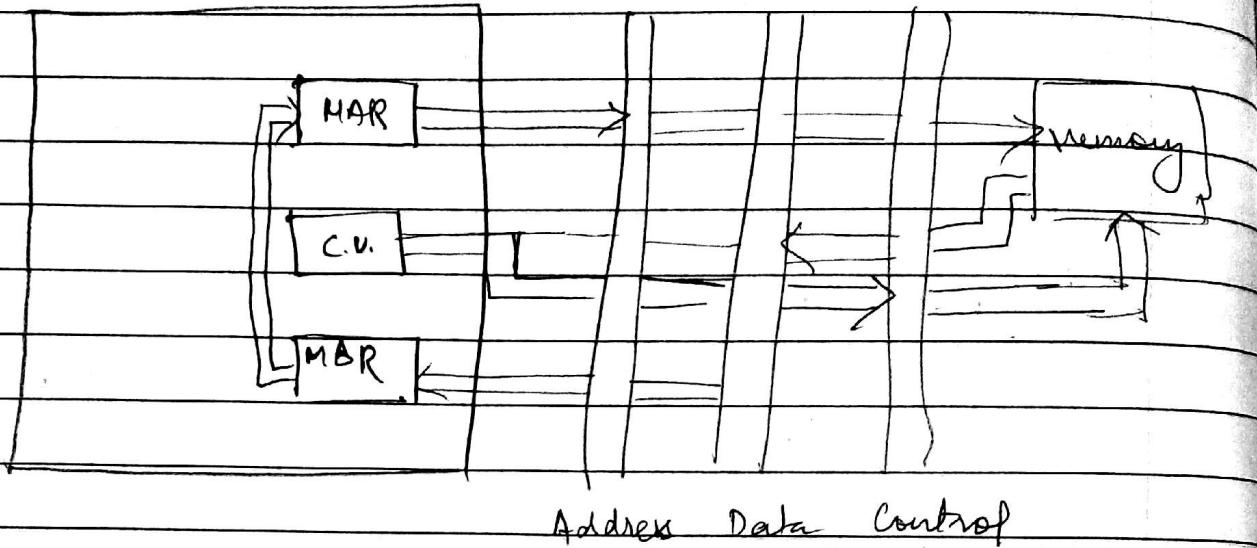


During the fetch cycle, an "inst" is read from the memory.

PC contains the addr. of the next instrⁿ to be fetched.

- 1) This add. is moved to MAR.
- 2) This add. is placed on the address bus.
- 3) C.U. sends a ^{memory} read signal to the control bus.
- 4) The result is placed on the data bus.
- 5) It is copied into M.B.R. from the data bus.
- 6) It is moved to IR.
- 7) Meanwhile, the PC is incremented by 1, preparatory for the next fetch.

Indirect cycle



Once the fetch cycle is over, the C.U. examines the contents of the I.R. to determine if it contains an operand specifier using indirect addressing. If so, an indirect cycle is performed.

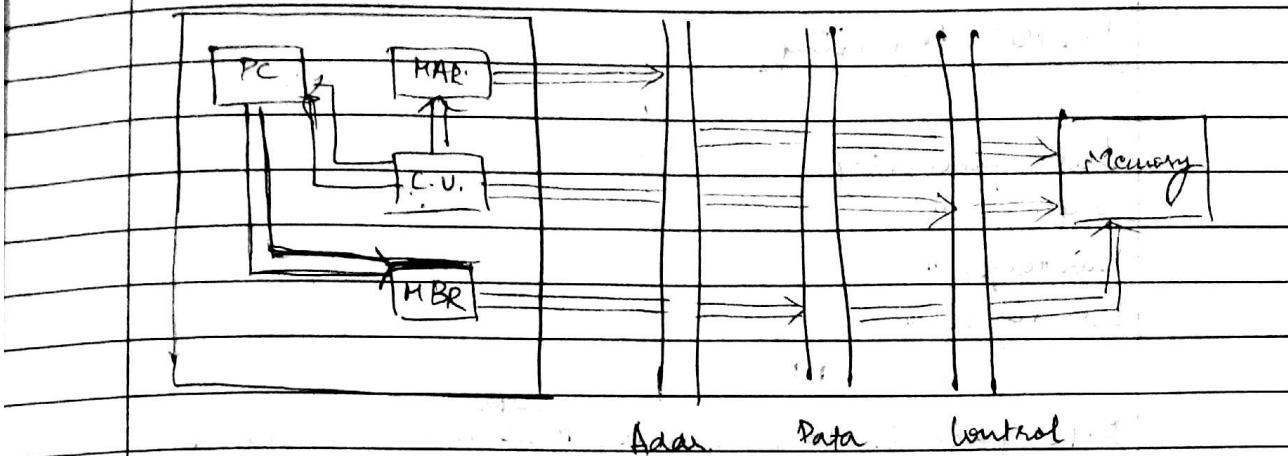
The rightmost n -bits of MBR which contain the address reference are transferred to MAR.

C.U. requests a mem. read to get the desired add. of the operand into ~~the~~ MBR.

Execute cycle

The execute cycle takes many forms. This cycle may involve transferring data among registers, read/write from memory / I/O, invocation of ALU.

Interrupt cycle

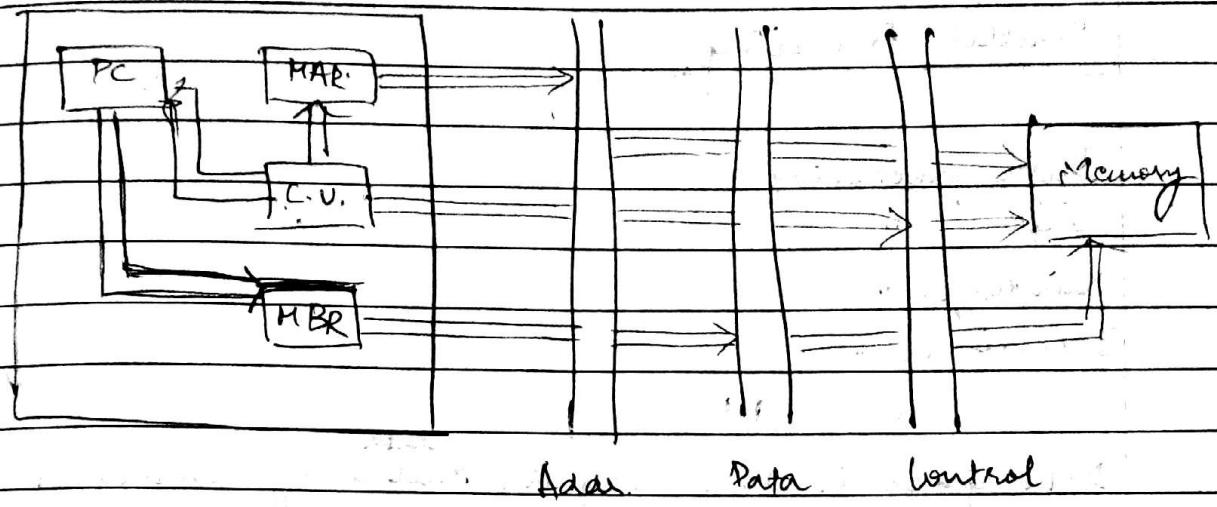


① The current contents of PC must be saved so that the processor can resume normal activity after the interrupt. The contents of PC are transferred to ~~the~~ MBR to be written into memory.

② The special mem. locaⁿ reserved for this purpose is loaded into MAR from C.U.

③ PC is loaded with the add. of the interrupt routine.

As a result, the next instⁿ cycle is a fetch cycle.



- ① The current contents of P.C. must be saved so that the processor can resume normal activity after the interrupt. The contents of PC are transferred to ~~the~~ MBR to be written into memory.
- ② The special mem. locaⁿ reserved for this purpose is loaded into MAR from C.U.
- ③ PC is loaded with the addr. of the interrupt routine.

As a result, the next instⁿ cycle is a fetch cycle.

Parallel Processing

Data processing

Info "

Knowledge "

Intelligence "

- Data space is the largest. Mutually ^{books} unrelated data objects in data space.

- Info is a "collc" of data objects that are related to some syntactic structure or relation

Info items form a subspace of data space

- Knowledge \rightarrow info + semantics, based on prev. exp.
~~note~~ subspace of info space

- Intelligence \Rightarrow collc" of knowledge, decision-making
innermost in Venn diagram

EL - analysing electrical ckt's

AGII - diagnosis of plant diseases

From an OS per., :-

1) Batch processing \rightarrow o/p in batches & is sequential

2) Multi-programming - multiple processes active in the same

3) Time sharing - Fixed/Variable time slices to multiple progs. ^{interval}

4) Multiprocessing - Multiple processors with shared mem. space,

$\frac{1}{2}$ peripherals under the ctrl of 2 integrated OS

\rightarrow interleaving of CPU & I/O

② & ③ have uniprocessor system to achieve concurrency.

Parallel processing in 4 programmatic levels:-

- 1) Job / prgm level
- 2) Task proc. lev
- 3) Inter-instrn "
- 4) Intra-instrn "

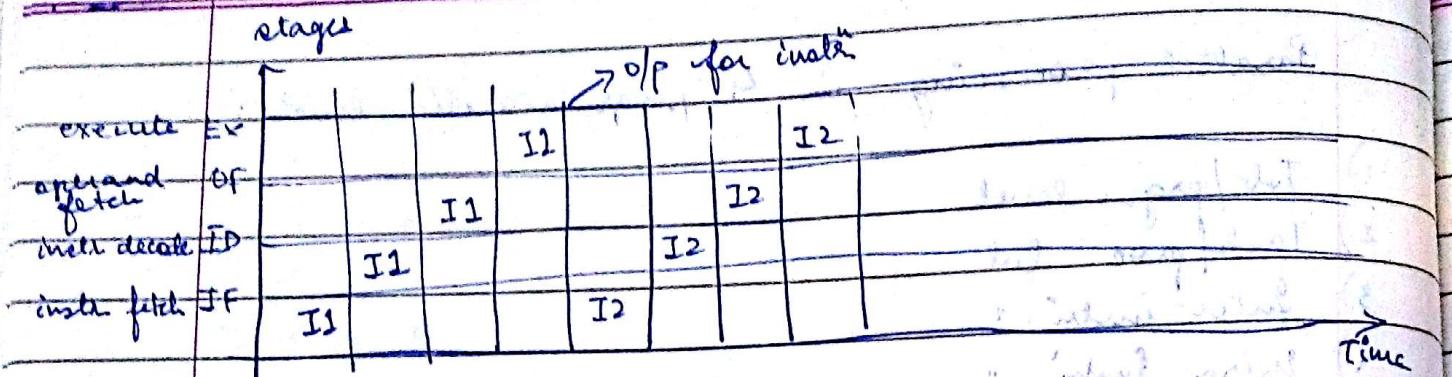
- ① Highest lvl of parallel processing \rightarrow multiple jobs
- ② Next highest level \rightarrow proc & tasks within the same pg.
 \rightarrow decomposition of a prg. into multiple tasks.
- ③ Exploit concurrency among multiple inst. Data dependency analysis is performed.
- ④ Faster & concurrent ops within each inst. implemented directly by h/w

* Parallel Computer

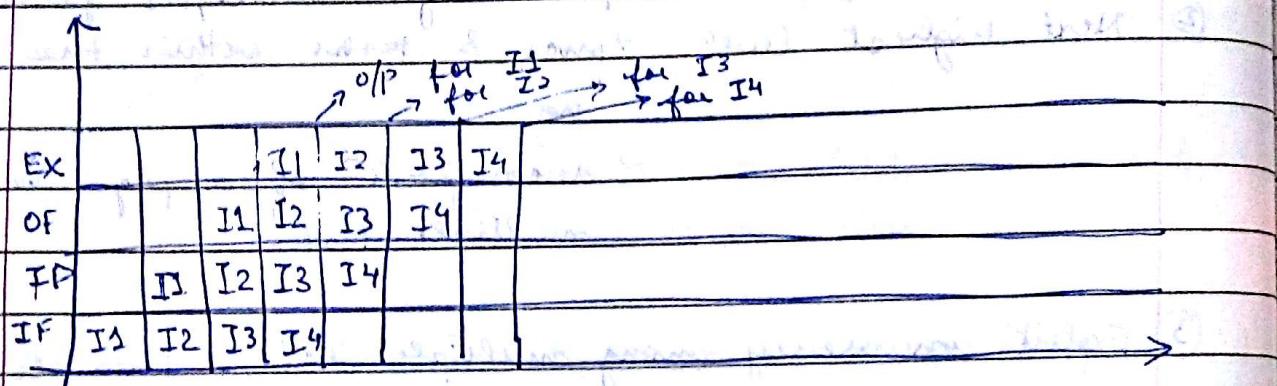
Parallel comp. are those systems that emphasize parallel processing. There are 3 basic config:-

- 1) Pipeline computers
- 2) Array processors
- 3) Multi-processor systems.

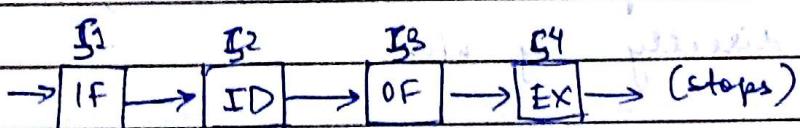
QUESTION



Space-time diag for non-pipelined processor



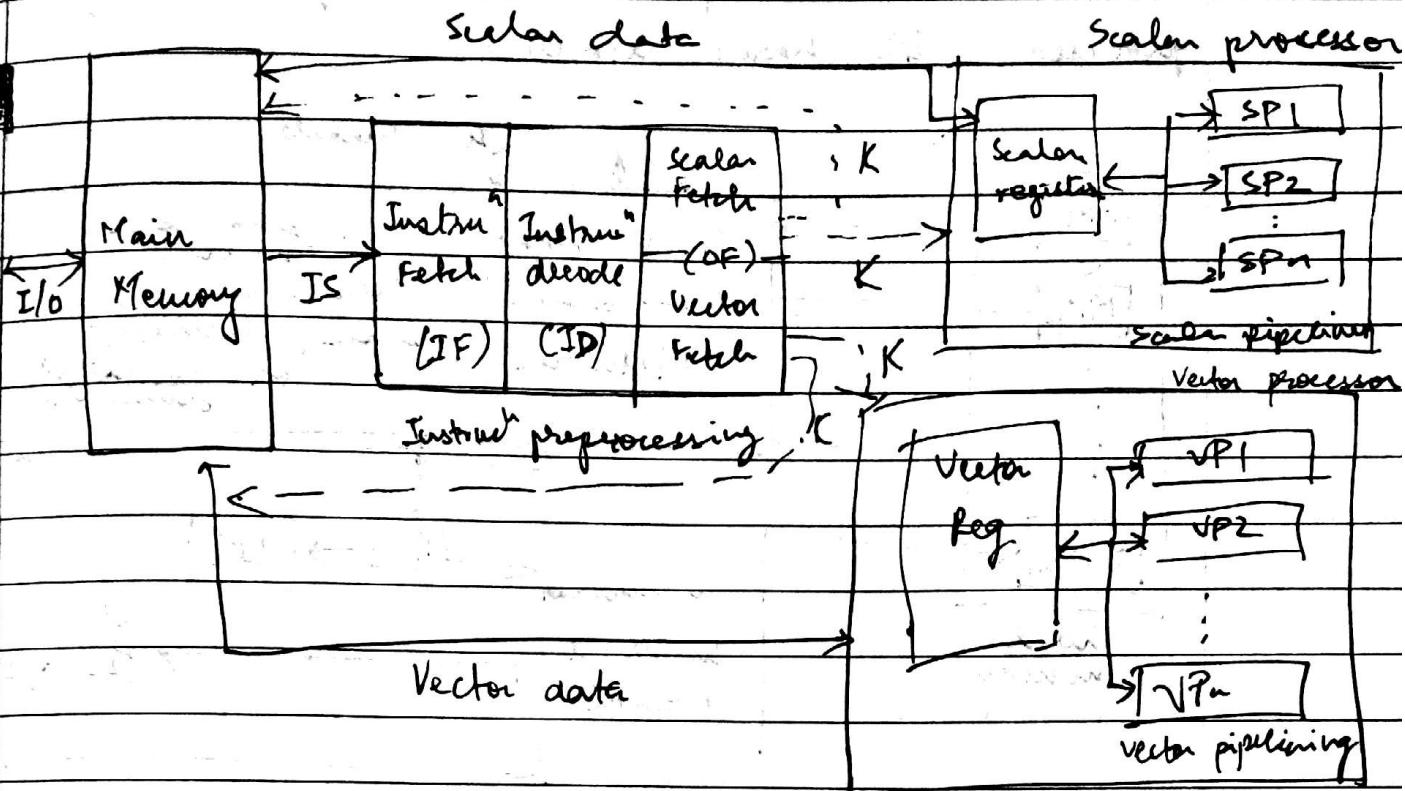
Pipelined processor



Pipeline computers perform overlapped computations to exploit temporal parallelism.

Temporal parallelism involves partitioning the processing tasks into a no. of steps which when applied sequentially to each unit of info. produce the same result as the original task.

Functional structure of a modern pipeline computer with scalar & vector capabilities

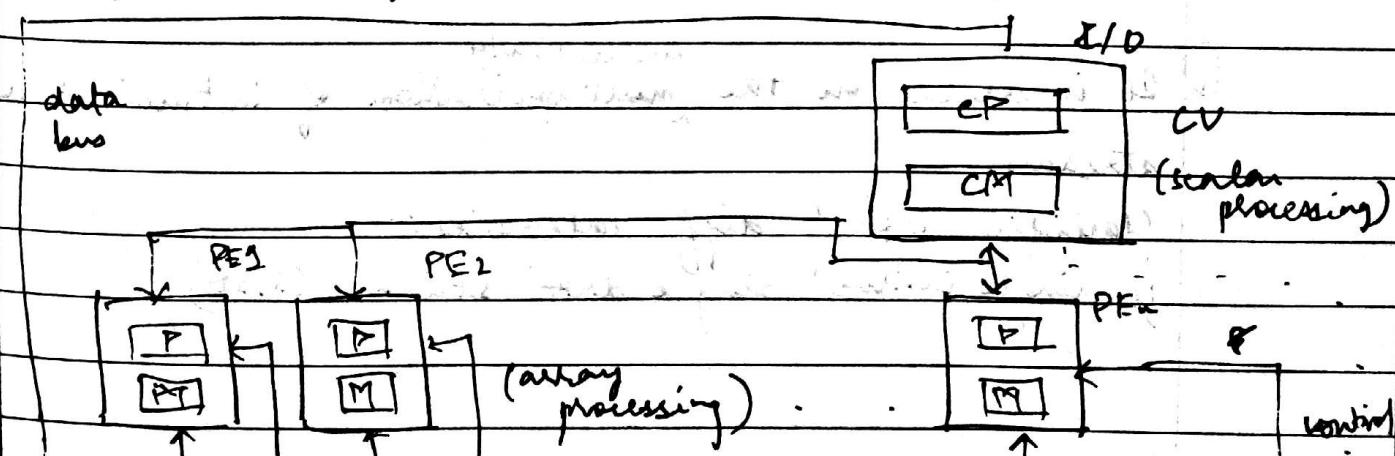


Vector registers are larger than scalar registers.

Array Computers

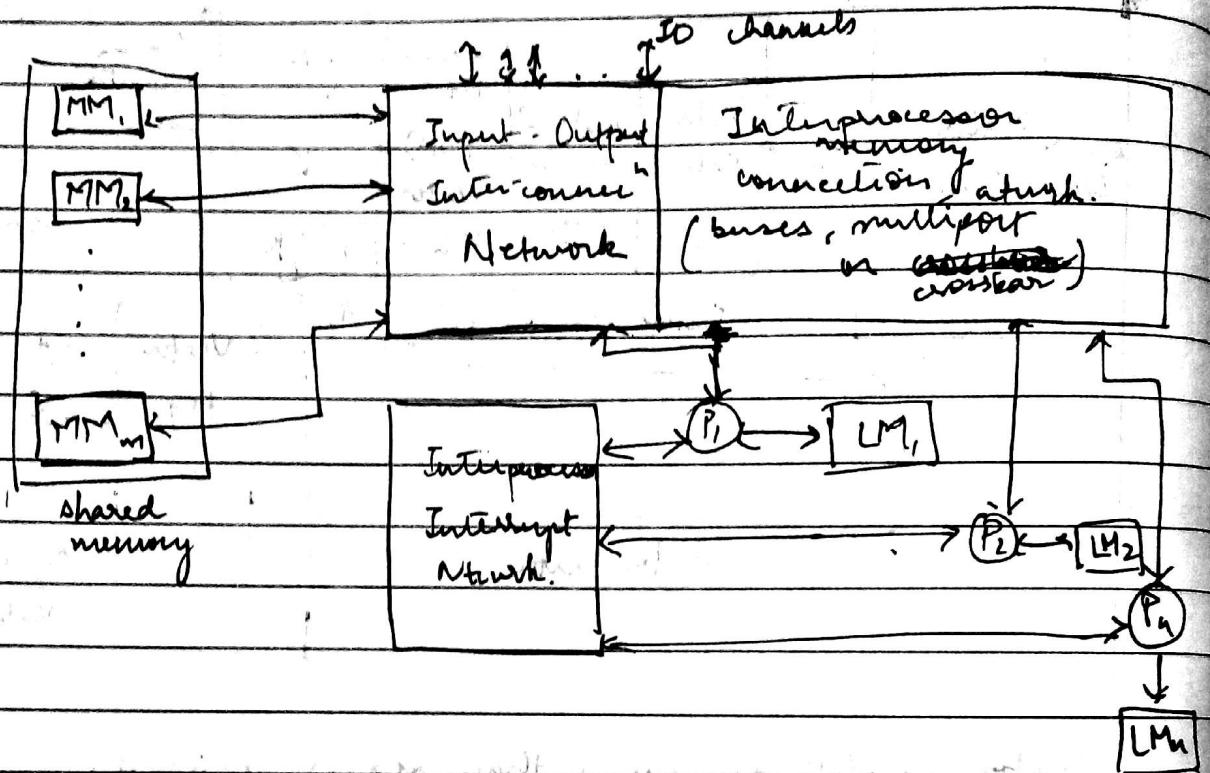
→ Uses multiple synchronized ALU to achieve spatial parallelism.

In spatial parallelism, the component used to carryout the processing task is replicated so that each unit of info is processed by its own dedicated component.



Multi-processor system

Achieves asynchronous parallelism through a set of internal processes with shared resources



Synchronous parallelism

Functional design of MIMD multi-processor system

Architectural classification schemes

Flynn's classification

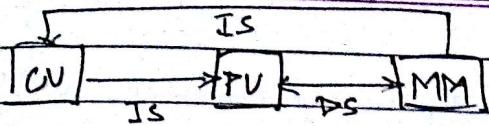
Flem's "

Hanlder's "

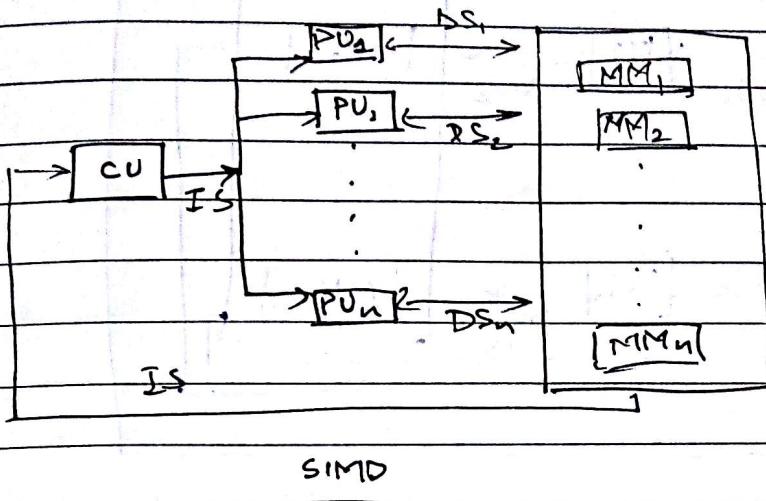
- ① It is based on the ~~multiplication~~ ^{multiplicity} of instruction & data streams.

Classified into 4 diff. categories:-

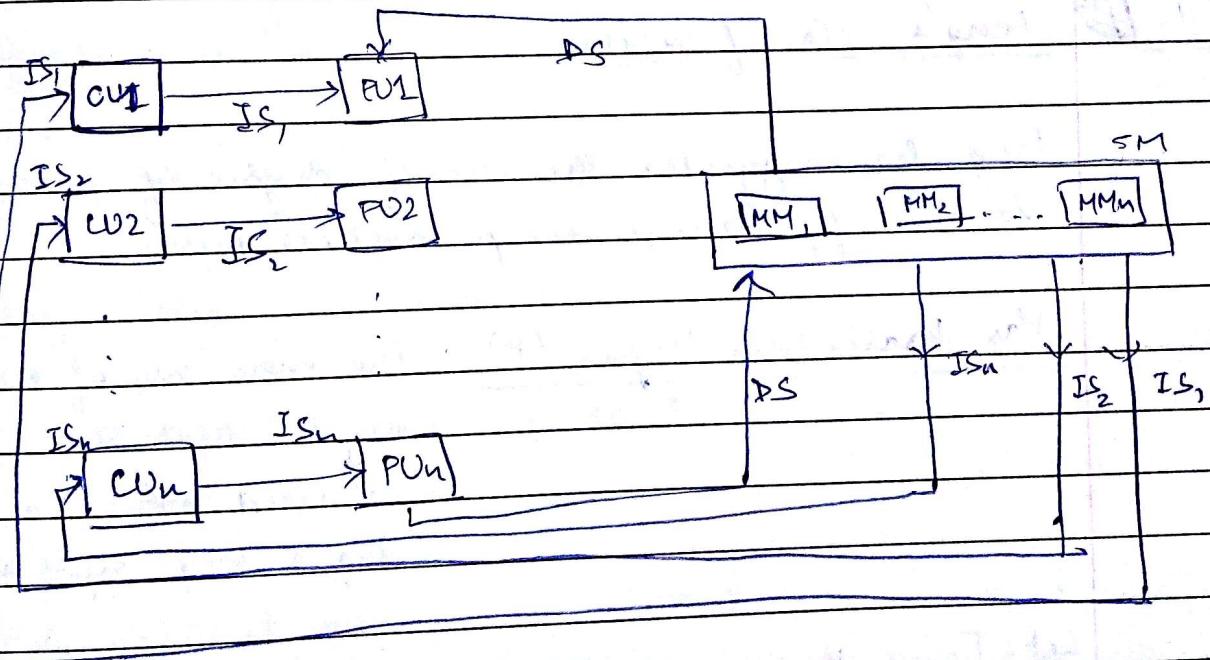
- 1) Single instruction single data stream (SISD)



② Single instruction multiple datastream (SIMD)



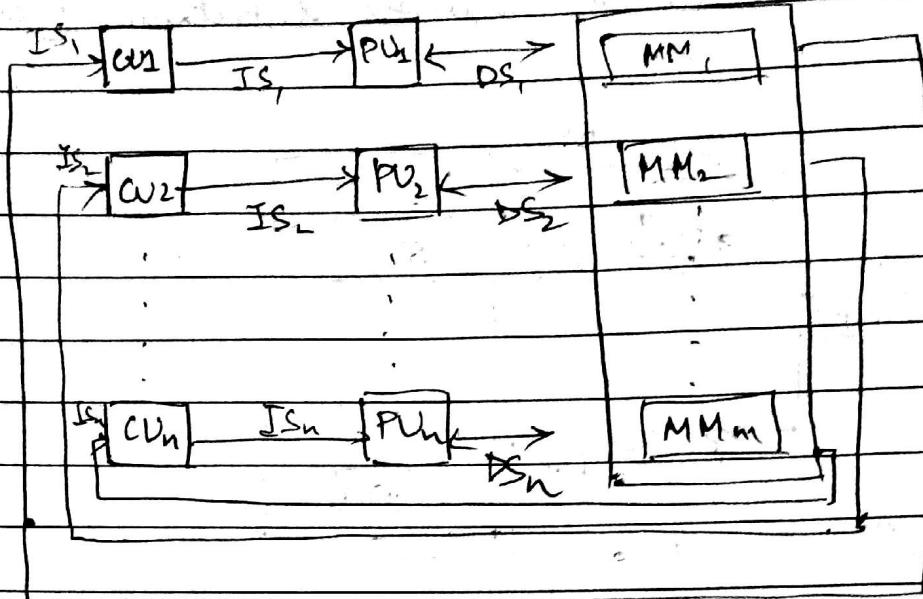
③ MISD - Multiple instances single data stream \Rightarrow



macropipe \Rightarrow array of processes

④ Multiple ~~inst~~ instruction multiple data stream (MIMD)

(in SIMD)



Processors can be tightly coupled (more interactions) or loosely coupled (less interaction)

II- Feng's classification

Feng has suggested the use of degree of parallelism to classify various comp. architecture.

Max. Parallelism Degree (P) : The max no. of binary digits that can be processed within a unit time by a comp. system.

Let P_i be the no. of bits that can be processed within the i th processor cycle.

Consider t processor cycles.

$$\text{The av. parallelism deg } P_a = \frac{\sum_{i=1}^t P_i}{t}$$

In general, $P_i \leq P$.

$$\text{Utilisation Rate } (\mu) = \frac{P_a}{P} = \frac{\sum_{i=1}^t P_i}{t \cdot P}$$

If the comp power of the processor is fully utilised, then we have $P_i = P$ for all i and $\mu = 1$.

Bit slice

A bit slice is a string of bits, one from each of the words at the same vertical bit position.

The max parallelism deg of a given comp system C is represented by $P(c) =$ The product of word length n & the bit slice length m.

$$P(c) = n \times m$$

4 diff processing methods:-

1) Word serial, bit serial (WSBS)

$$n=1, m=1$$

2) Word parallel, bit serial (WPBS)

This is called bit (bit-slice) processing bcoz m-bit slices are processed at a time.

$$n=1, m>1$$

3) Word serial, bit parallel (WSBP)

word slice processing because 1 word of m bits is processed at a time.

$$m > 1, m=1$$

(4) word parallel, bit parallel (WPBP)

Fully parallel processing in which array of $n \times m$ bits are processed at once.

$$n > 1, m > 1$$

Parallelism Vs Pipeline

Handler has proposed classification scheme for identifying the parallelism degree & pipelining degree built into the hardware str. of a comp.

He considers parallel pipeline processing at 3 subsystem levels

1) Processor Control Unit (PCU)

Each PCU corresponds to 1 processor or 1 CPU.

2) Arithmetic control Unit (ALU)

ALU is equivalent to the processing element.

3) Bit level circuit (BLC)

It corresponds to the combinational logical circuitry needed to perform 1 bit operation in the ALUs.

A comp. system can be characterised by a triple containing 6 independent entities as defined below
 $T(C) = \{K \times K', D \times D', W \times W'\}$ where K = no. of processors (PCUs), D = no. of ALUs on PEs under the control of 1 PCU, W = word length of an ALU or of a PE.

W' - no. of pipeline stages in all ALUs

D' - no. of ALUs that can be pipelined

K' - no. of PCU that can be pipelined

TI-ASC \rightarrow 1 controller controlling 4 arithmetic pipelines,
64-bit word lengths \in 8 stages

$$T(C) \langle K \times K', D \times D', W \times W' \rangle$$

$$K = 1$$

$$D = 4$$

$$W = 64$$

$$W' = 8$$

$$K' = 1$$

$$D' = 1$$

$$T(C) = \langle 1 \times 1, 4 \times 1, 64 \times 64 \rangle$$

$$\langle 1, 4, 64 \times 8 \rangle$$

whenever $K', D', W' = 1$, we drop it.

Ex2 Control data 6600 has a CPU with an ALU that ^{has} 10 specialized b/w funcⁿ each of w= 60 bits. Up to 10 of these funcⁿ can be linked into a longer pipeline 10 peripheral I/O processors which can operate in parallel. Each I/O processor has 1 ALU with word length of 12 bits.

$$\begin{array}{l}
 \overbrace{W = 60}^{\text{I/O}} \\
 W' = 1 \\
 D = 1 \\
 D' = 10 \\
 K = 1 \\
 K' = 1
 \end{array}
 \quad
 \begin{array}{l}
 \overbrace{w=12}^{W'} \\
 W' =
 \end{array}$$

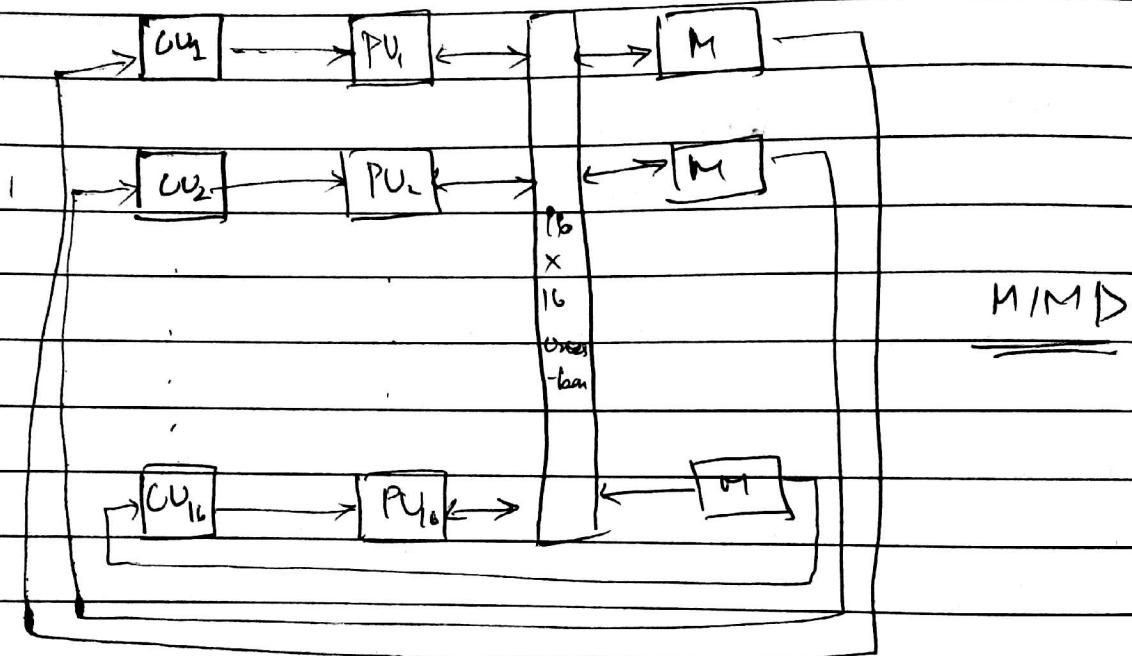
$$\begin{aligned}
 T(\text{CD 6600}) &= T(\underset{\text{central}}{\text{processor}}) \times T(\text{I/O processor}) \\
 &= \langle 1 \times 1, 1 \times 10, 60 \rangle \times \langle 10, 1, 12 \rangle \\
 &= \langle 1, 10, 60 \rangle \times \langle 10, 1, 12 \rangle
 \end{aligned}$$

Ex 3

C.mmp \rightarrow 16 PDP-11 minicomputers of a word length of 16 bits

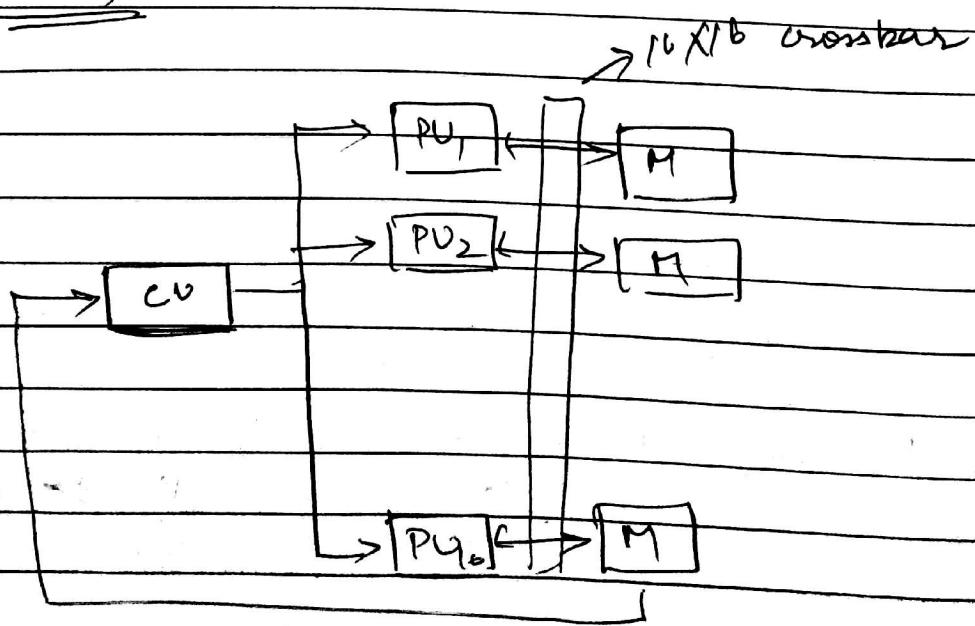
It can operate in 3 modes:

- MIMD
- SIMD
- MISD



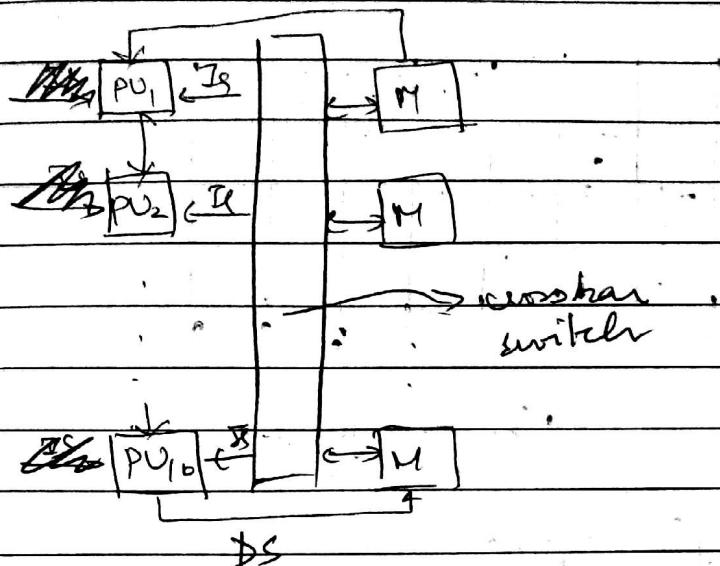
$$T(C.mmp) = \langle 16 \times 1, 1 \times 1, 16 \times 1 \rangle + \langle 1 \times 1, 1, 16 \rangle \\ + \langle 16 \times 16, 1, 16 \rangle$$

SIMD

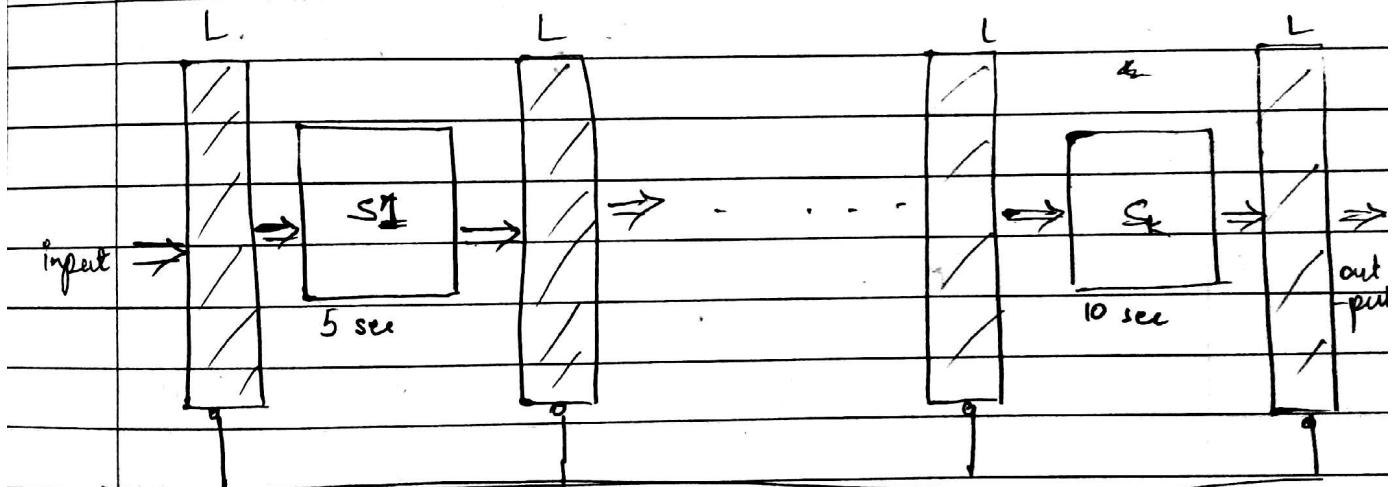


~~IF~~

MISD DS



Principles of linear pipelining



S_i: ith stage

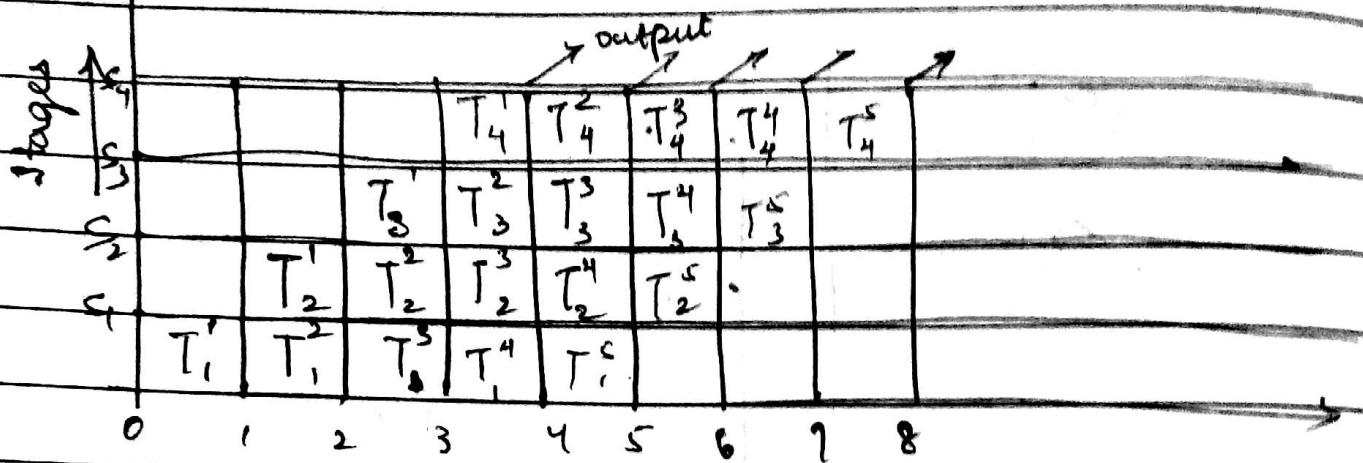
L: latch

C: clock

- Q. Draw a space-time diagram by taking 5 tasks & each task has 4 stages.

$$n=5$$

$$k=4$$



Time (in cycles)

T_{ij} : jth stage / subtask in
ith task

A linear precedence relation task T_j cannot start until all the earlier subtasks $\{T_i\}$ for all $i < j$ finish.

Parameter Analysis of Linear Pipeline

Clock Period (T) : The logic circuitry in each stage S_i has a time delay denoted by T_i .

Let T_e be the time delay of each interface latch.

$$\begin{aligned} \text{Clock period} \Rightarrow T &= \max \{ T_i \}_{i=1}^k + T_e \\ &= T_m + T_e \end{aligned}$$

The reciprocal of clock period is called freq.

$$f = \frac{1}{T}$$