UNIT - IV

PARALLEL COMPUTER STRUCTURE

Parallel Processing

Parallel processing is a term used to denote a large class of techniques that are used to provide simultaneous data processing task for the purpose of increasing ~~[illegible]~~ computational speed of a computer system

Purpose:-

① To speed up the computer processing capability.

② Increase its throughput

Throughput:- The amount of processing that can be accomplished during a given interval of time.

Disadvantages:-

① Amount of hardware increases

② Increased cost

Classification of Parallel Processing

classification was introduced by M.J. Flynn considers the organization of a computer system by the number of instructions and data item that are manipulated simultaneously.

The sequence of instruction read from memory constitutes instruction stream

The operations performed on the data in the processor constitutes a data stream

* Flynn's classification divides computers into four groups (Architectural classification Schema)

a) SISD : Single Instruction Stream, Single Data Stream

b) SIMD : Single Instruction stream, Multiple Data Stream

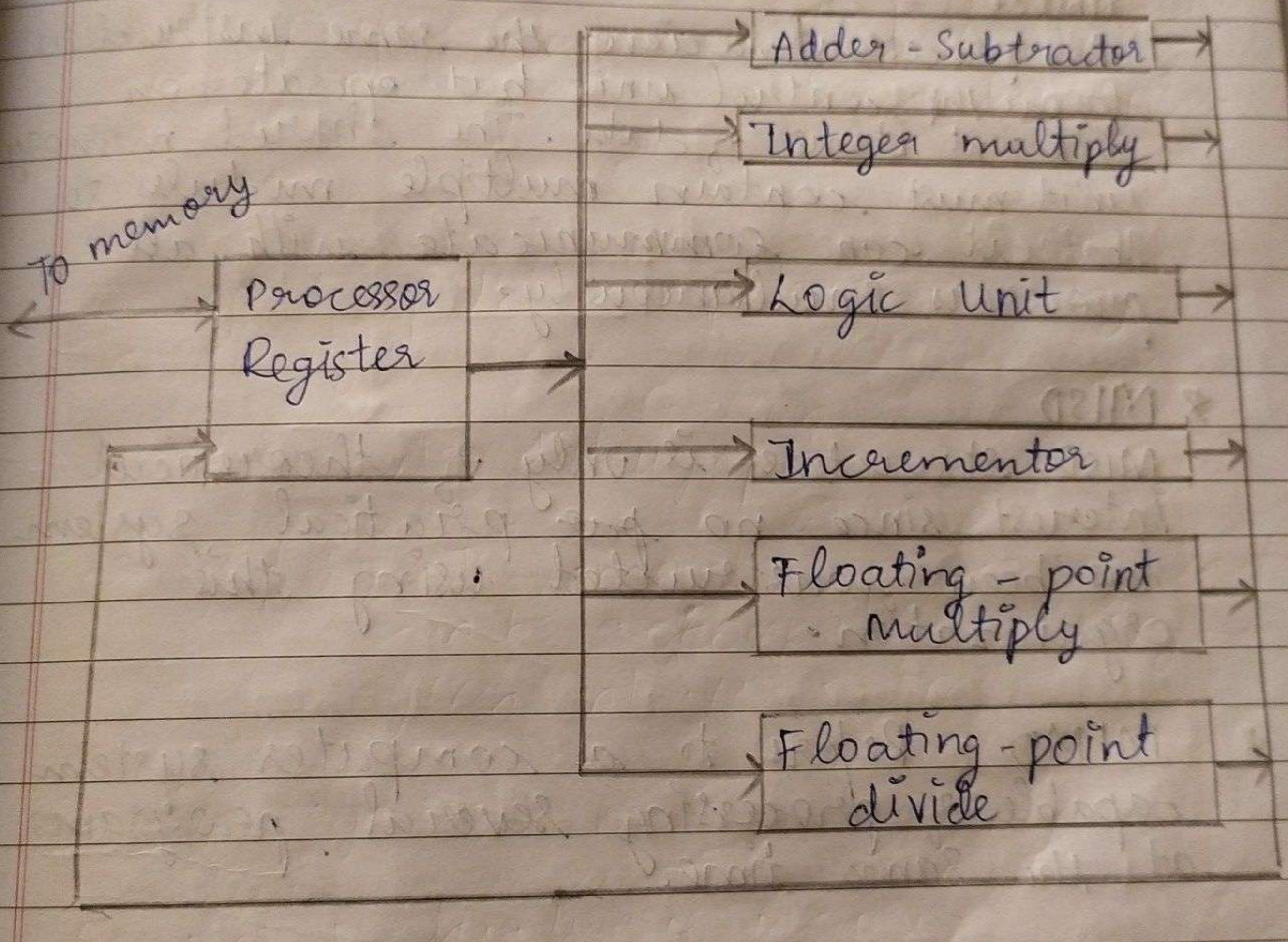
c) MISD : Multiple Instruction stream, Single Data Stream

d) MIMD : Multiple Instruction Stream, Multiple Data Stream

* SISD represents the organization of a single computer containing a control unit, a processor unit and a memory unit.
Instructions are executed sequentially and the system may or may not have

internal parallel processing capabilities. Parallel processing in this case is achieved by means of multiple functional unit or by pipeline processing

Processor with Multiple Functional Units



The adder and integer multiplier perform the arithmetic operations with integer numbers The floating point operations are separated into 3 circuits operating in parallel. The logic shift

All units are independent of each other so one number can be shifted while other number is being incremented.

* SIMD : Represents an organization that includes many processing units under the supervision of a common control unit.
All processors recieve the same instruction from the control unit but operate on different items of data. The shared memory unit must contain multiple modules so that it can communicate with all processors simultaneously.

* MISD
MISD structure is only of theoretical interest since no ~~pa~~ pratical system has been constructed using this organization.

* MIMD : Refers to a computer system capable of processing several programs at the same time.

PIPELINE COMPUTERS

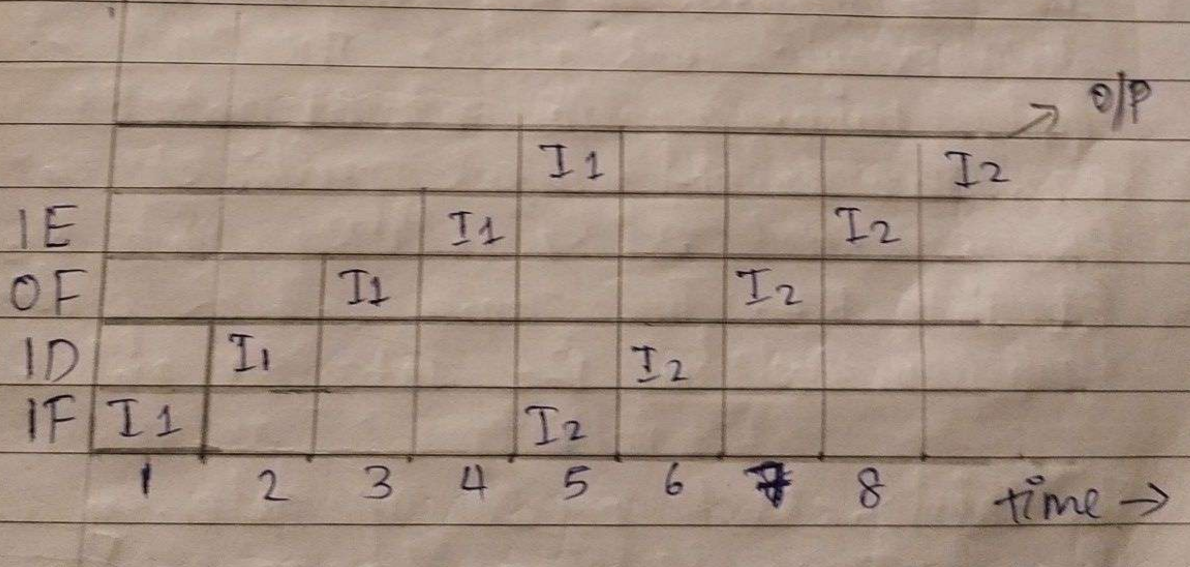
The process of executing an instruction involves four steps
1) Instruction Fetch (I.F)
2) Instruction Decoding (ID)
3) Operand Fetch (OF)

4) Instruction Execution (IE)

In a non-pipeline computer these 4 steps are being completed before the next instruction can be issued.

In a pipeline computer, successive instruction are executed in overlapped fashion. An instruction cycle consists of multiple pipeline cycles. A pipeline cycle can be equal to the time of the delays of the slowest stage.

The operation of all stages is synchronized by a common clock control. The CPU bound instruction the execution face can be further partioned into



Non-pipeline process

					O/P	O/P	O/P	
IE				I1	I2	I3		
OF			I1	I2	I3			
ID		I1	I2	I3				
IF	I1	I2	I3					
	1	2	3	4	5	6	7	8

Pipeline process

Multiple stage arithmetic logic pipeline

Module - 4

PARALLEL COMPUTER STRUCTURE

Parallel Processing

Parallel processing is a term used to denote a large class of techniques that are used to provide simaltainous data processing task for the purpose of increasing communational speed of a computer system.

Purpose

a) To speed up the computer processing capability.

b) Increase its through put

Through put

The amount of processing that can be accomplish during a given interval of time.

Disadvantages

(.) Amount of hardware increases

(.) Increased cost.

Classification of Parallel Processing

Classification introduced by M.J Flynn considers the organization of a computer system by the number of instructions and data items that are manipulated simultaneously.

(i) The sequence of instruction read from memory constitutes Instruction stream

The operations performed on the data in the processor constitutes an data stream

Imp * Flynn's Classification Divides Computers into 4 groups (architectural Classification Scheme)

a) SISD : Single Instruction stream, Single Data stream.

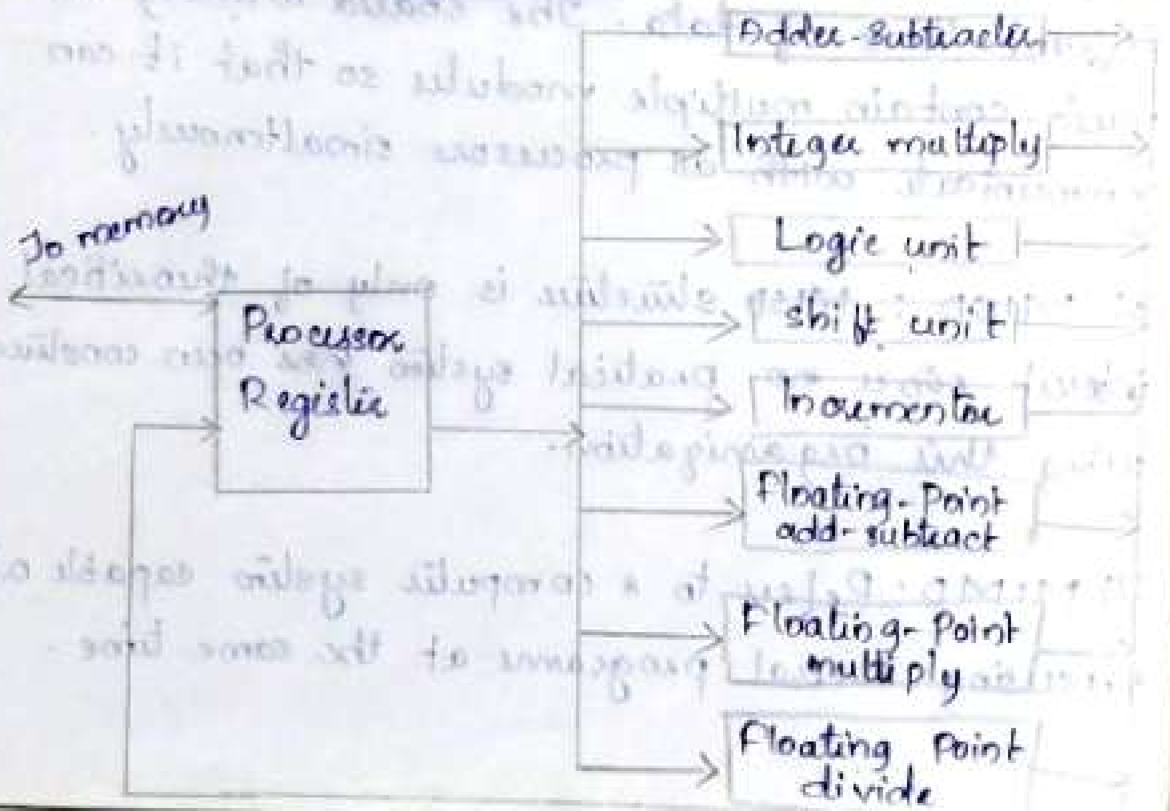
b) SIMD : Single Instruction stream, Multiple data stream

c) MISD: Multiple Instruction stream, Single Data stream

d) MIMD : Multiple Instruction Stream, Multiple Data Streams

(a) SISD represents the organization of a single computer containing a control unit, a processor unit and a memory unit. Instructions are executed sequently and the system may or may not have internal parallel processing capabilities. Parallel processing in this case is acheived by means of multiple functional unit or by pipeline processing.

Processor with Multiple Functional Units



The adder and integer multiplier perform the arithematic operations with integer numbers. The floating point operations are separated into 3 circuits operating in parallel. The logic shift and incrementer increment can . All units are independent of each other so one number can be shifted while other number is being incremented.

(b) SIMD: Represents an organization that includes many processing units under the supervision of a common control unit. All processors recieve the same instruction from the control unit but operate on different items of data. The shared memory unit must contain multiple modules so that it can communicate with all processors simaltenously.

(c) MISD: MISD structure is only of theoritical interest since no pratical system has been construct using this organization.

(d) MIMD: Refers to a computer system capable of processing several programs at the same time.

12.03.19

Module - 5
Pipelining & Vector Processing

Pipelining

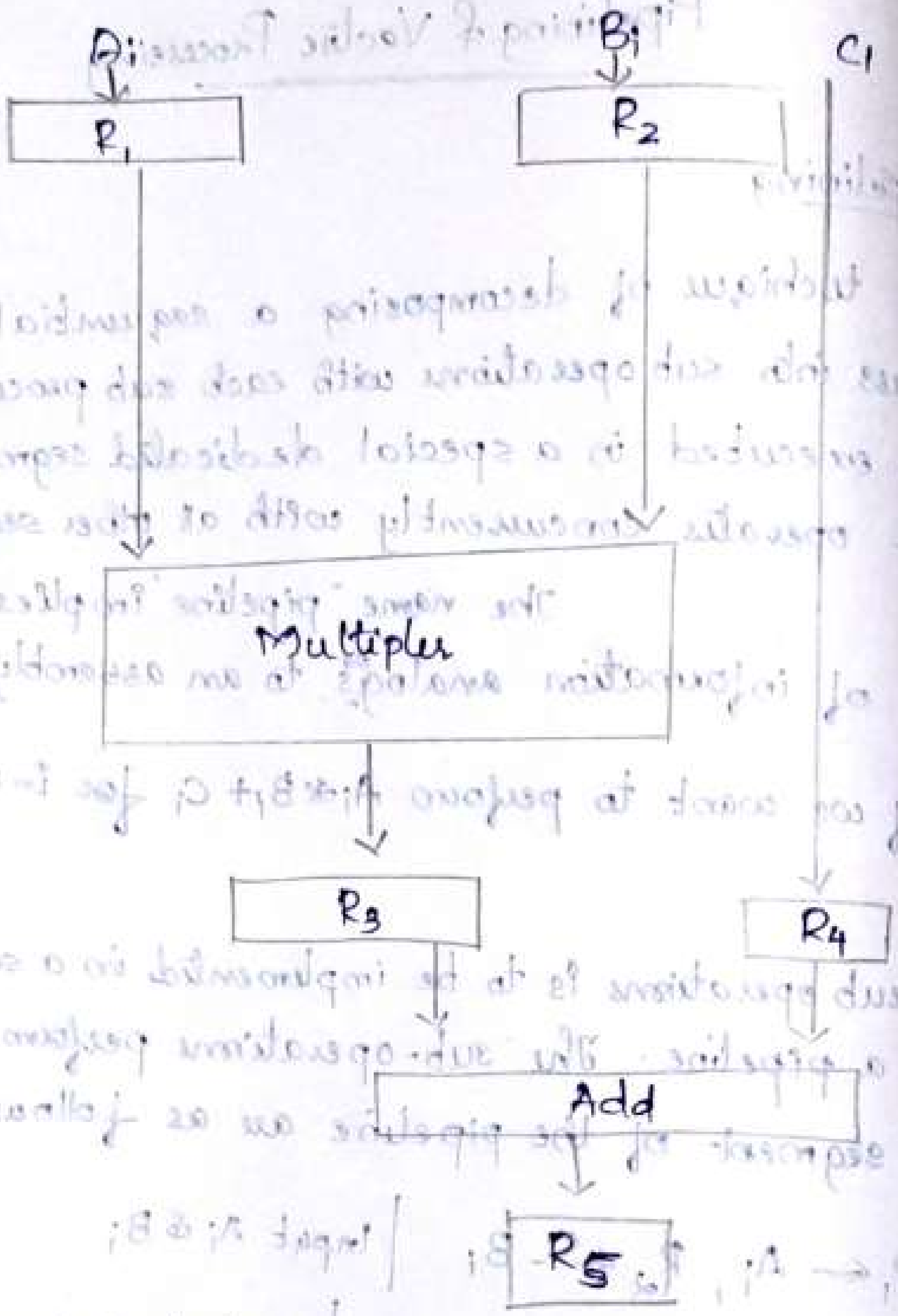
The techique of decomposing a sequential process into sub operations with each sub process being executed in a special dedicated segment that operates concurrently with all other segments.

The name "pipeline" implies a flow of information analogous to an assembly line.

ex: If we want to perform $A_i * B_i + C_i$ for $i = 1, 2, 3 \ldots 7$.

Each sub operations is to be implemented in a segment with a pipeline. The sub-operations performed in each segment of the pipeline are as follows:

$R_1 \leftarrow A_i,\ R_2 \leftarrow B_i$	Input A_i & B_i
$R_3 \leftarrow R_1 * R_2,\ R_4 \leftarrow C_i$	Multiply & input C_i
$R_5 \leftarrow R_3 + R_4$	Add C_i to the product.



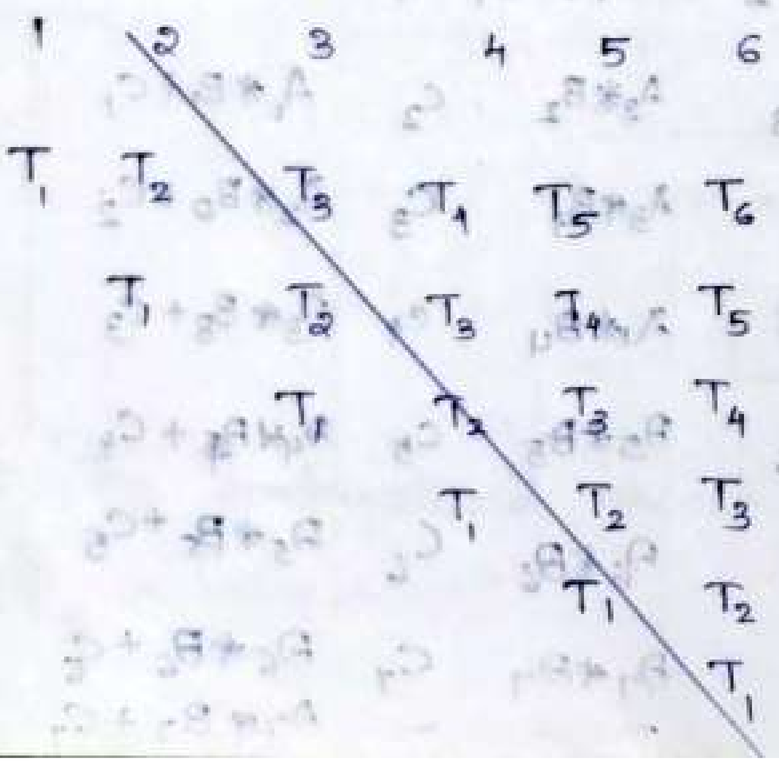
The combinational circuit is shown in the figure above.

The 5 registers are loded with new data every clock pulse. The first clock pluse ~~[illegible]~~ transfers A_1 and B_1 into R_1 and R_2. The second clock pluse transfers the product of R_1 and R_2 into R_3 and C_1 into R_4. The same clock pluse transfers A_2 and B_2 into R_1 and R_2. The third clock pluse operate on all three segments simaltenously. It takes 3 clock pluses to fill up the pipe and retrieve the first output from R_5.

clock pluse number	Segment 1		Segment 2		Segment 3
	R_1	R_2	R_3	R_4	R_5
1	A_1	B_1	–	–	–
2	A_2	B_2	$A_1 * B_1$	C_1	–
3	A_3	B_3	$A_2 * B_2$	C_2	$A_1 * B_1 + C_1$
4	A_4	B_4	$A_3 * B_3$	C_3	$A_2 * B_2 + C_2$
5	A_5	B_5	$A_4 * B_4$	C_4	$A_3 * B_3 + C_3$
6	A_6	B_6	$A_5 * B_5$	C_5	$A_4 * A_5 + C_4$
7	A_7	B_7	$A_6 * B_6$	C_6	$A_5 * B_5 + C_5$
8	–	–	$A_7 * A_7$	C_7	$A_6 * B_6 + C_6$
9	–	–	–	–	$A_7 * B_7 + C_7$

General Considerations

Any operations that can be decomposed into a sequence of sub operation of same complexity can be implemented by a pipeline processor. The behaviour of a pipeline can be illustrated with space-time diagram. This is a diagram that shows the segment utilization as a function of time. The horizontal access display the timing clock cycles and the vertical access displays the segment number. The diagram shows 6 tasks T_1, through T_6 executed in 4 segments. Intillay T_1 is handled by segment 1 The first task T_1 is completed after 4th clock cycle



1	2	3	4	5	6	7	8	9
T_1	T_2	T_3	T_4	T_5	T_6			
	T_1	T_2	T_3	T_4	T_5	T_6		
		T_1	T_2	T_3	T_4	T_5	T_6	
			T_1	T_2	T_3	T_4	T_5	T_6

Consider the case where k-segment pipeline with a clock cycle time 'tp' is used to execute 'n' task. The 1st task T_1 requires a time = ktp, to complete its operation. To complete 'n' task it requires $k+(n-1)$ clock cycles. By substituting the value of space diagram in this equation will give;

$$k+(n-1)$$

$$= 4+(6-1) = 4+5 = 9 \text{ clock cycles.}$$

The speed up of a pipeline processing over an equivalent non-pipeline processing is defined by the ratio as

$$S = \frac{nt_n}{(k+n-1)t_p}$$

where n = no: of task

Arithmatic Pipeline

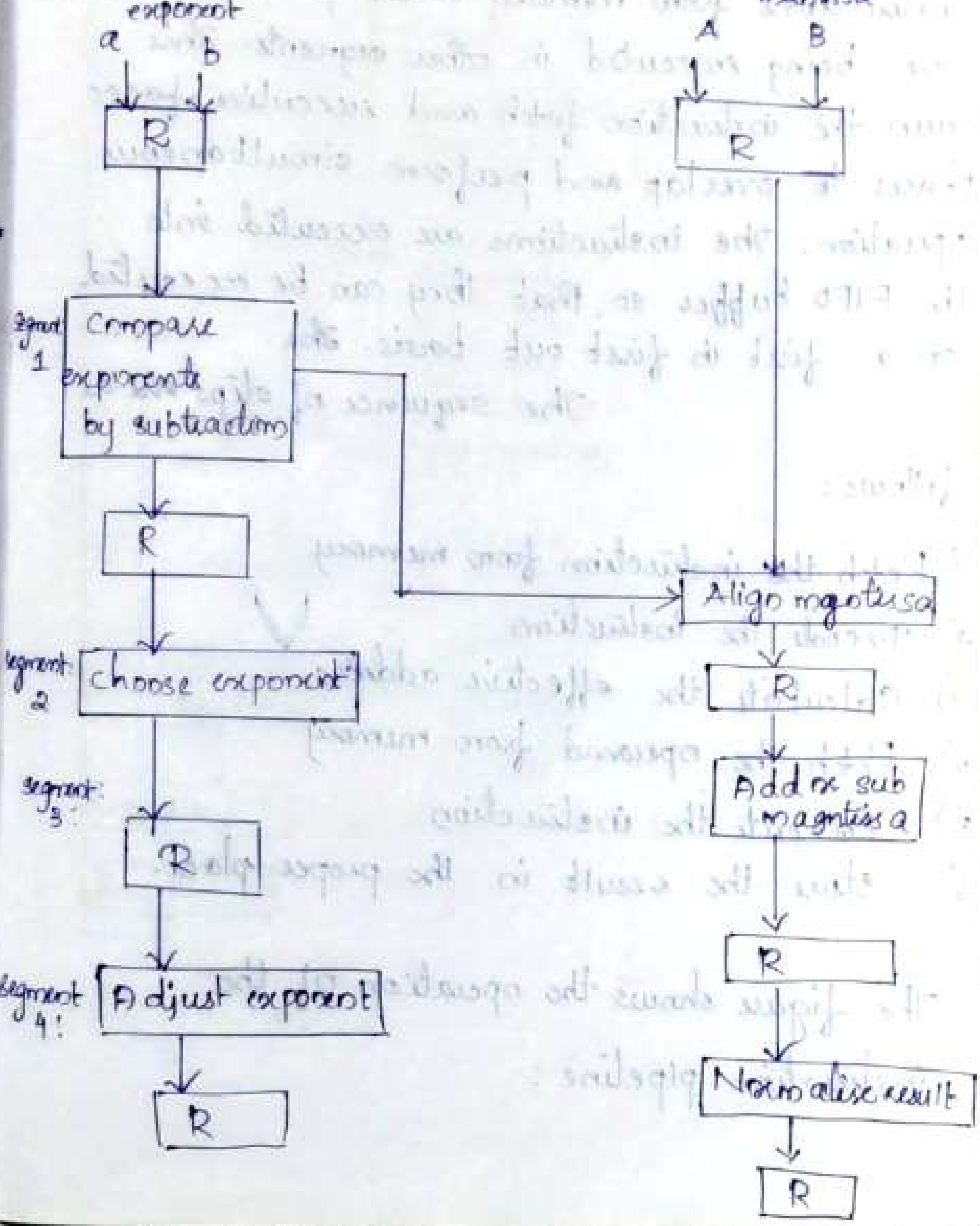
Pipeline arithmatic unit are used to implement ~~a~~ floating point operations, ~~mult~~ multiplication of fixed point number etc. The floating point addition and subtraction can be performed in 4 segments as shown in figure. The registers labelled R are placed btwn the segments to store the intermedi result. The sub-operations performed in the 4 segments are:

- (i) Compare the exponents
- (ii) Align the mantissa
- (iii) Add or subtract the mantissa
- (iv) Normalize the result.

ex: Consider the two floating point number $x = 0.9504 \times 10^3$ & $y = 0.8200 \times 10^2$. The two exponents are subtracted in the first segment to obtain $3-2=1$. The larger exponent 3 is used as the exponent of the result. The next segment shift the mantissa of Y to ~~once~~ to right to obtain $X = 0.9504 \times 10^3$

$Y = 0.0820 \times 10^3$

This aligns the 2 mantissa under the same exponent. the addition of the 2 mantissa's in segment 3 produces the sum is 1.0324×10^3



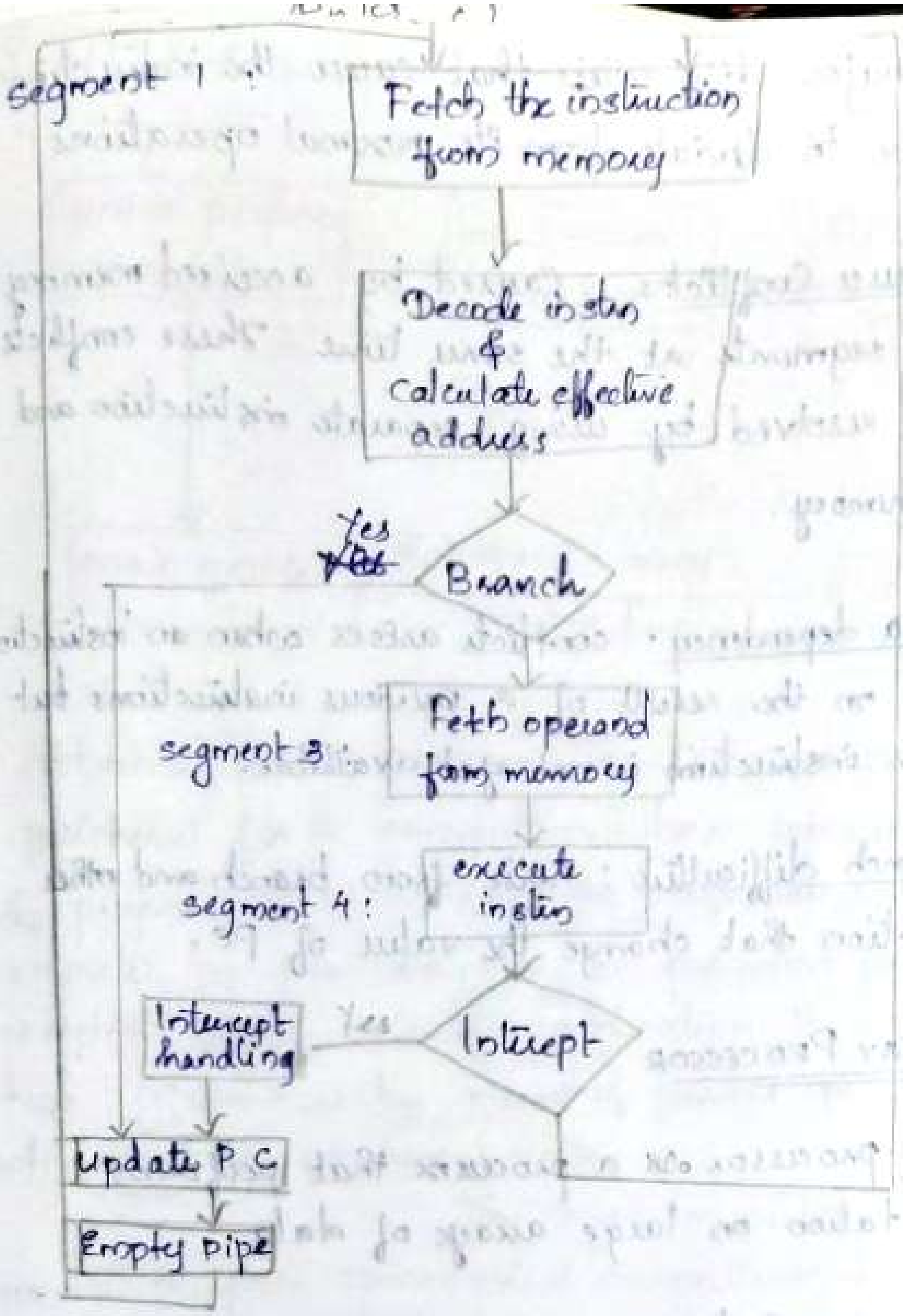
Instruction Pipelining

An instruction pipeline reads ~~consoeq~~ consecutive instructions from memory while previous instruction are being executed in other segments. This causes the instruction fetch and executive ~~faces~~ phases to overlap and perform simultaneous operation. The instructions are executed into the FIFO buffer so that they can be executed on a first in first out basis. ~~The~~

The sequence of steps are as follows:

1) Fetch the instruction from memory
2) Decode the instruction
3) Calculate the effective address
4) Fetch the operand from memory
5) Execute the instruction
6) store the result in the proper place.

The figure shows the operation of the instruction pipeline:


segment 1 :
Fetch the instruction from memory
Decode instrn & Calculate effective address
Yes ~~Yes~~
Branch
segment 3 :
Fetch operand from memory
segment 4 :
execute instrn
Interrupt handling
Yes
Interupt
Update P.C
Empty pipe

* 3 major difficulties that cause the instruction pipeline to deviate from its normal operations

a) <u>Resource Conflicts</u> : caused by accessed memory by 2 segments at the same time. These conflicts can be resolved by using separate instruction and data memory.

b) <u>Data dependency</u>: conflicts arises when an instruction depends on the result of a previous instructions but its this instruction is not yet available.

c) <u>Branch difficulties</u> : Arise from branch and other instruction that change the value of PC.

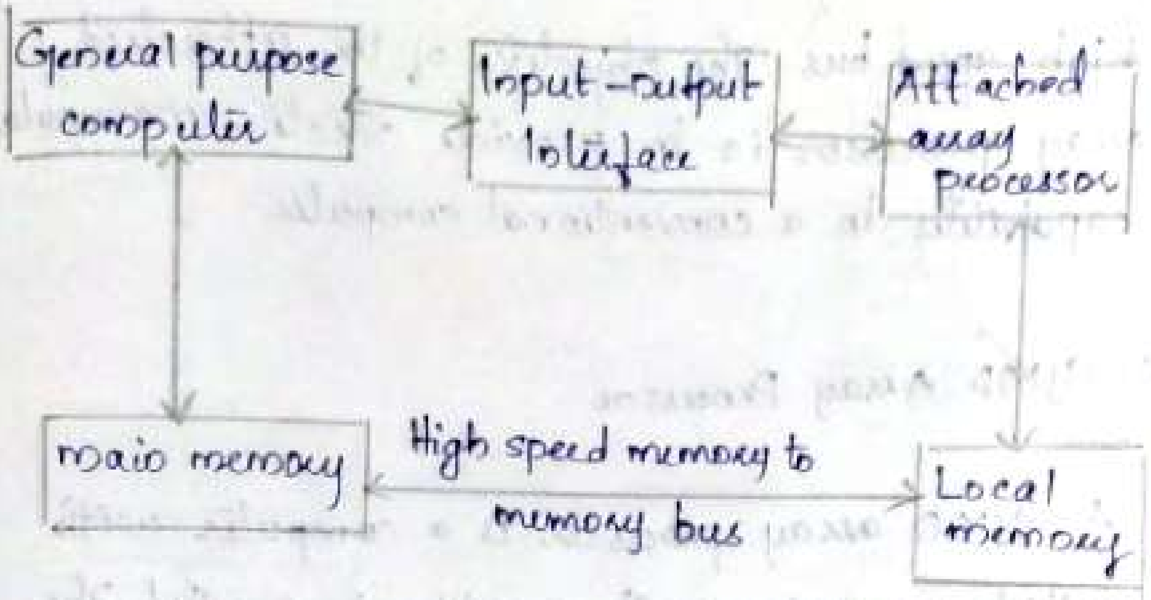
* <u>ARRAY PROCESSOR</u>

Array processor is a processor that performs computation on large arrays of data.

There are 2 types:

1) Attached array processor
2) SIMD array processor.

i) Attached array processor



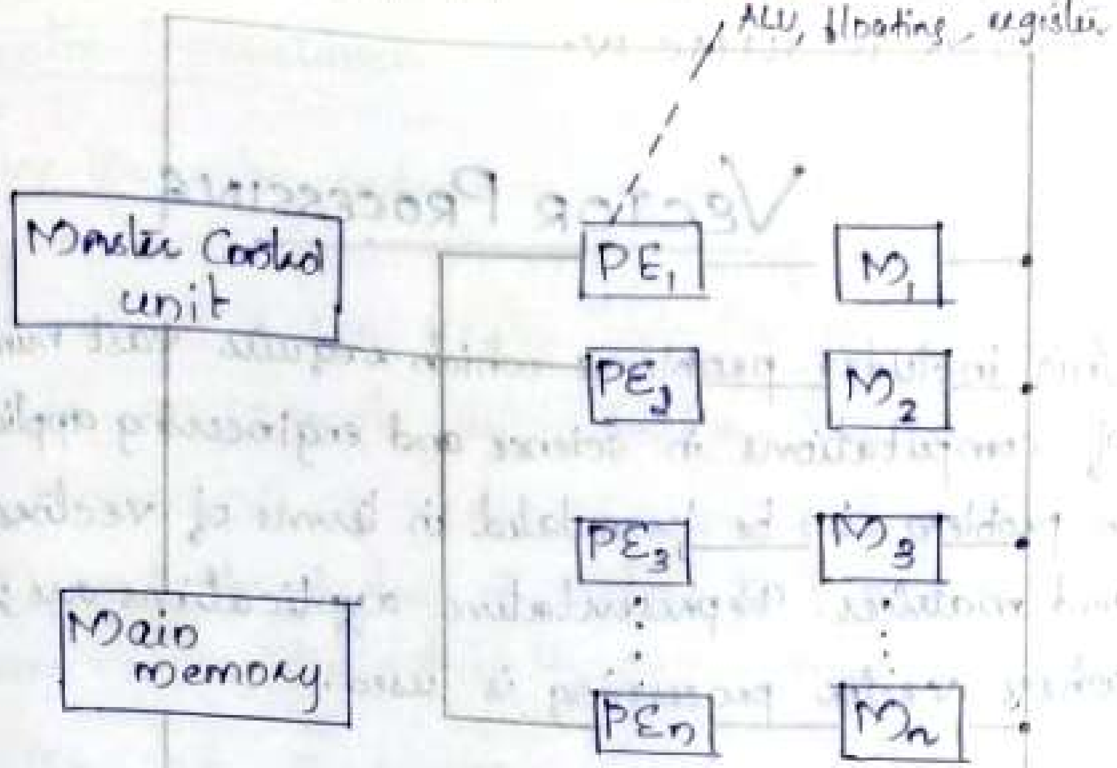
Attached array processor is designed as peripheral for a conventional host computer. The purpose is to enhance the performance of the computer by providing vector processing for complex ~~signif~~ scientific application. It achieves high performance by means of parallel processing with multiple functional units.

The host computer is a general purpose commerical computer and the attached processor is a back end machine. The array processor is connected through an input-output controller to the computer and computer considereds it as an external interface. The data for the attached processor are transfered from main memory to a local memory through high speed bus. The objective of the attached array processor is to provide vector manupulation capabilites to a conventional computer.

imp

2) SIMD Array Processor

A SIMD array processor is a computer with multiple processing unit operating in parallel. The processing unit are syncronchised to perform the same operation under the control of a common control unit, thus providing SIMD organization. The block diagram of an SIMD array processor is shown below:



It contains a set of identical processing elements (PEs). Each having a local memory M. Each processor element includes an ALU, floating point unit and working registers. The Master control unit controls the operations in the processor elements. The main memory is used for the storage of the programs. The function of the master control unit is to decode the instructions and determine how the instruction is to be executed. Each PE uses operands stored in its local memory.

The best known SIMD array

Processor is ILLIAC IV.

Vector Processing

This includes problems which require vast numbe of computations in science and engineering applicat the problem can be formulated in terms of vectors and matrices. Representative applications are; where vector processing is used.

1) Long-Range whether forecasting
2) Petroleum explorations
3) Seismic data analysis
4) Medical diagnosis
5) Aerodynamics & flight simulations.
6) Artifical Intelligence and expert system
7) Mapping the human genome
8) Image processing

Vector Operations

A vector is an ordered set of one-dimensional array of data items. A vector V of length 'n' is represented as a row vector $V = [V_1, V_2, \cdots V_n]$.
consider an example with Fortan Do loop.
The element V_i of vector V is ~~returned~~ written as V(I) and the index I refers to a memory address or register where the item number is stored.

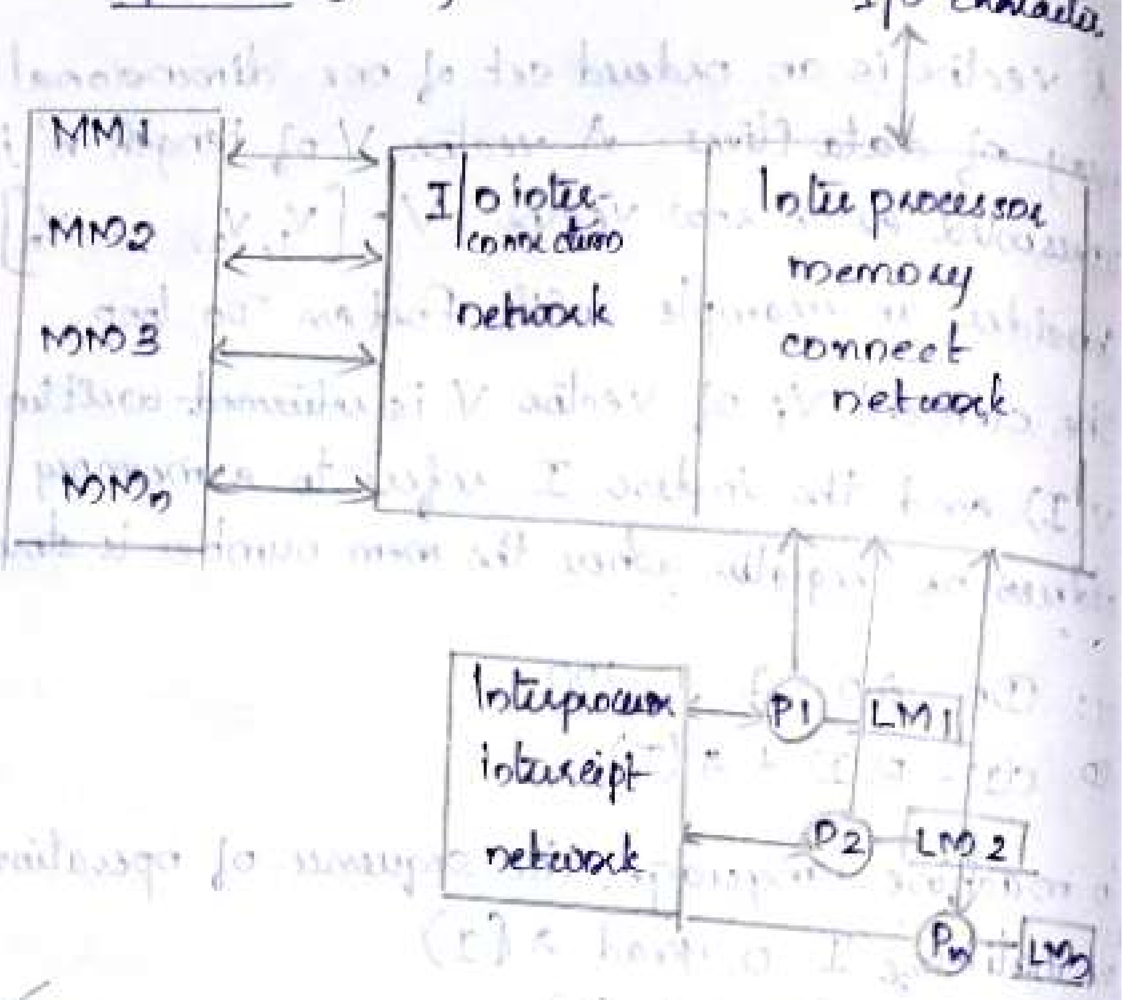
```
ex: Do 20 I = 1, 100
20  C(I) = B(I) + A(I).
```

In machine language, the sequence of operations are intialize I=0. Read A(I)

```
                Read B(I)
     20      Store C(I) = B(I) + A(I)
                  Increment I = I + 1
              If I ≤ 100 go to 20
                  continue
```

The single vector instruction is C(1:100) = B(1:00) + A (1:00)

Multiprocessing Systems


I/O channels
MM1
MM2
MM3
MMn
I/O inter-connection network
Inter processor memory connect network
Interprocessor interrupt network
P1
LM1
P2
LM2
Pn
LMn

A multiprocessor system is a interconnection of 2 or more processors with similar capabilites. All processors share access to common memory module I/O channels and peripheral devices. The entire system is controlled by single integrating operating system. Each processor has its own memory and private devices.

Inter processor communication can be done through an inter processor interupt network. Communication btwn processor and memory modules or I/o channels can be done through inter processor memory connection network or input/output interconnection network.

(i) Classifications of Multi-processor

Multi-processor are classified by the way their memory is organized. A multi-processor system with common shared memory is classified as shared memory or tightly coupled multi-processor. This does not include each processor having its own local memory. Most tightly coupled multi-processor provides a cache memory with each CPU, also there will be a global common memory that all CPU's can access.

Another model is distributed memory or loosely coupled system Each processor in a loosely coupled system has its own private local memory. The processors relay program and data two other processors in packets. Each packets contains an address, the data content and some error detection code. The packets are addressed to the specific processor or taken by the 1st available processor.