

UNIT-2General Purpose register

A. 8086 General Data Register (Register organisation) :-

8086 has a powerful registers known as general purpose register. All the 16 bit register, these registers also use at 8 bit registers, registers are used for holding data, variable and temporary storage of data.

Four types of registers -

1. General data registers

2. Segment registers

3. Pointer and Index registers.

4. Flag registers.

1. General data registers -

AX	AH	AL
BX	BH	BL
CX	CH	CL
DX	DH	DL

• AX is used as 16 bit accumulator.

AL is low 8 bit

AH is high 8 bit

- BX use as offset storage to form physical address.
- CX use as default counter in string and loop instruction.
- DX use as operands and destination instructions.

2- 8086 Segment Register - 8086 has a powerful registers known as general purpose register, segment register is second type registers. In which 1 megabyte memory which 8086 addresses. Each Segment contains 64 Kbytes.

CS	Code segment register use for addressing memory location.
SS	Stack segment register use addressing stack segment of memory.
DS	Data segment register use data segment of memory.
ES	Extra segment register also use data of memory.

3- Pointer and Index registers contains offsets with a particular segment -

SP	JP, SP, BP, offset.
BP	
SI	SI use for store offset of source data in data segment.
DI	DI use for store offset of destination data in data segment.
JP	destination data in data segment.

4- Flag Register -

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	0	0	J	T	S	Z	x	A	C	X	P	x

S = Sign flag = If result is -ve this flag is set if it is denoted by NSB.

Z = Zero flag = If previous instructions is zero this flag is set otherwise.

P = Parity flag = If lower byte of result contain even no. of 1's this flag is set.

Cy = Carry flag = If carry produce in NSB this flag is set.

T = Trap flag = If process in single step execution mode this flag is set.

I = Interrupt flag = If Maskable interrupt detect by CPU this flag is set.

D = Direction flag = this flag is 0. String is process in auto increment mode, and 1 in auto decrement mode.

AC = auxiliary carry flag = this flag is set if carry from lower nibble (bit)

O = Overflow flag = this flag is set if result is large than a destination register.

* Stack Structure of 8086 or 8088 :-

Stack is sequence set of data which

PUSH and POP instruction work.

Stack is used to save useful data.

At the starting of subroutine all the register contain main program push into the stack one by one.

How we can calculate stack top address

Stack pointer register 16 BIT register

contain offset address in stage segment.

It is 64 byte memory location.

SS stack segment contain base address of stack segment in memory.

L7- SS stack segment and SP make a address together stack top.

Stacks :- A data structure in which last item inserted is taken out first LIFO (Last In First Out).

- Only one item is inserted / Push at a time on top of stack.
- only one item is deleted / Pop at a time from top of stack.

	4		4		5	4	3
	3		3		1	3	
	2 →		2		2	2	[Push =
qor →	8	1	(if qor > top)		3	1	top + 1]
Top = 1	0	4	0 ← Top	4	8	0	

Stack is empty

$\text{Pop} = \{\text{top} = \text{top} - 1\}$.

	4		4
1	3 ← top	2	3
2	2 →	1	2
3	1 ← top	0	1 → top = -1
4	0		0

Empty

* Push operation -

1. Stack overflow ?

If $\text{Top} = \text{max_stack}$, write overflow and exist.

2. Read item.

3. Set $\text{top} = \text{top} + 1$

4. Set $\text{stack}[\text{top}] = \text{item}$

5. Exit

Initial	3	2 ← top → overflow,	3	2 ← top
	2	1 exit	2	1
	1	0	1	0

* Pop operation -

1. Stack overflow ?

If $\text{Top} = -1$ then write underflow and exit.

2. Repeat steps 3 to 5 until $\text{Top} \geq 0$

3. Set Item = Stack [Top]

4. Set Top = Top - 1

5. Write deleted Item.

6. Exit.

Item = Stack [2], Stack [1].

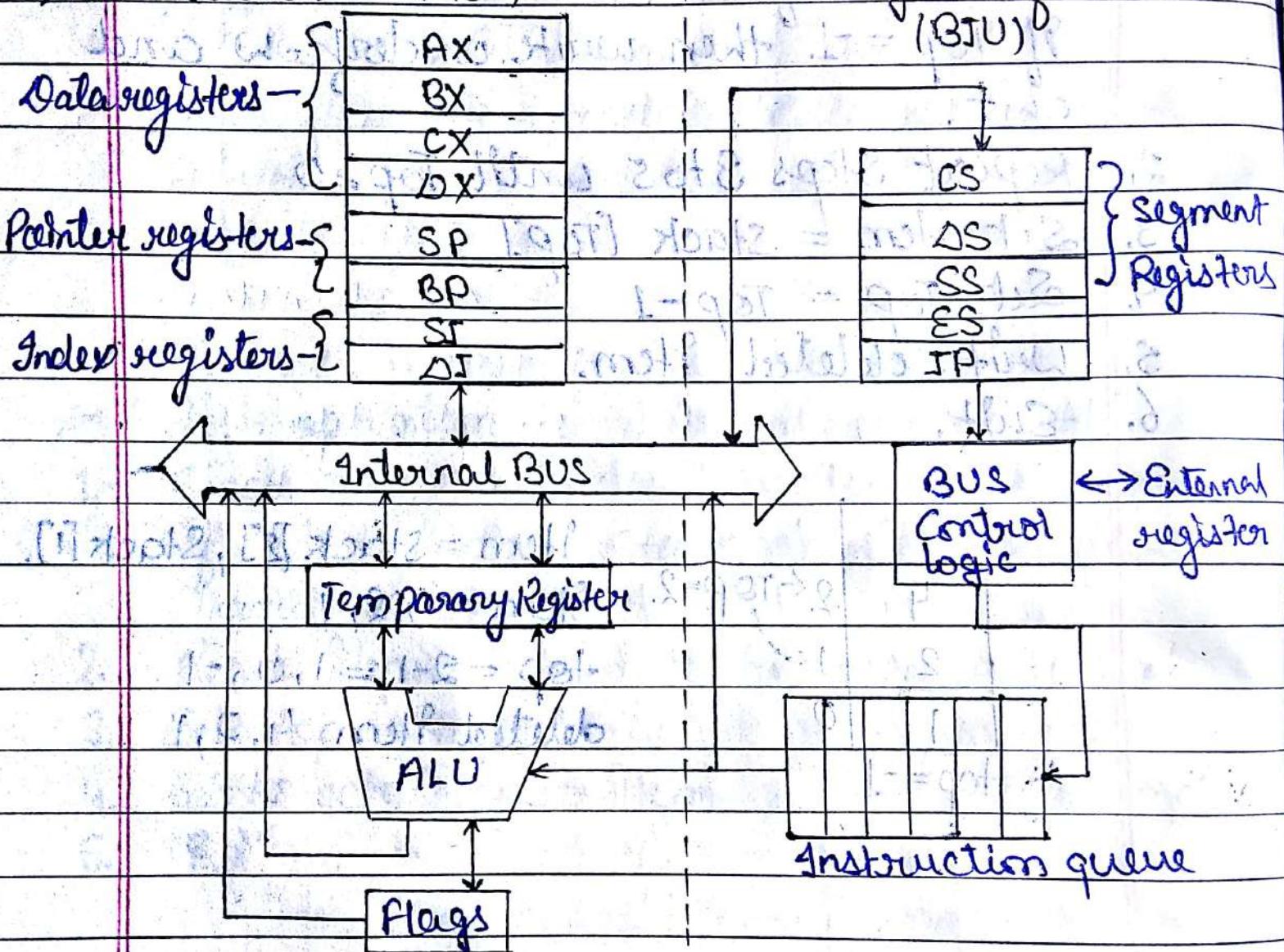
$2 \leftarrow \text{TOP} = 2$. Item = 4, 2, 1

$\text{top} = 2-1 = 1, 0, -1$

deleted Item 4, 2, 1

$\text{top} = -1$

A. Execution unit (EU)



Internal Architecture of 8086 Microprocessor

* Data buses:-

They provide a path for moving data among system modules.

- These lines are collectively referred as data Bus.
- The number of lines in Data Bus is called width of the data bus.
Like, 16, 32, 64, 128 or more.
- Each line carries 1 bit at a time, so, the no. of lines in data bus determine how many bits can be transferred parallelly.
- Data bus width determines overall system performance.

* Address Lines -

Specify where to store the data from where to retrieve the data (present on the data bus).

- Collectively, address lines are known as Address Bus.
- Address Bus width determines the memory capacity \rightarrow Total no. of unique addresses / locations.
- The same address lines can refer to I/O parts.

- 1- Designate high order bits to specify the module that we want to access.
- 2- Low order bits specify location within memory or a particular part.

CONTROL LINES

They are used to control the access to the Address Bus and Data Bus and monitor their use.

Why?

They are required because A.B. and D.B. are shared between components and a mechanism to control their use is required.

How?

Generates Control Signals, Timing Information, Command Info.

- Timing signals tell validity of data and address on DB and AB.
- Specify the operations to be performed.

- Memory Write : Data \rightarrow Memory
- Memory Read : Memory \rightarrow Data Bus
- I/O Write : Data Bus \rightarrow I/O Port
- I/O Read : I/O Port data \rightarrow Data Bus
- Transfer ACK : Indicates success (Data)
- Bus Request : Req. to again control of bus.
- Bus Grant : Control granted
- Interrupt Request : Pending Interrupt
- Interrupt ACK : Pending INTR recognised.
- Clock : Synchronization of operation.
- Reset : Initialization of modules.

A. Addressing Modes for Control Transfer Instructions :-

Addressing modes for control transfer instructions. They are:

- 1- Intra Segment Direct Mode :- If the location to which control is to be transferred is in the same segment, it is called intra segment mode. If address to which the control is to be transferred appears directly in the instruction as a displacement value, it is what

called as Intra Segment direct mode.

2. Intra Segment Indirect mode :-

If the location to which control is to be transferred is in the same segment, it is called Intra segment mode. If address to which the control is to be transferred appears indirectly in the instruction, it is what called as Intra segment direct mode.

3. Inter Segment Direct Mode :-

If the location to which control is to be transferred is not in the same segment, it is called Inter segment mode. If address of segment to which the control is to be transferred and location in the segment appears directly in the instruction, it is what called as Inter segment direct mode.

4. Inter Segment Indirect mode :-

If the location to which control is to be transferred is not in the same segment, it is called Inter segment mode.

If address of segment to which the control is to be transferred and location in the segment appears indirectly in the instruction, it is what called as inter segment indirect mode.

* Addressing modes :-

1.- Addressing modes are nothing but the different ways in which the location of an operand can be specified in an instruction. The number of addressing modes that a processor supports changes according to the instruction set.

It is based on, however there are a few generic ones that are present in almost all processors and are thus of utmost importance.

2.- They are as follows:-

1) Immediate mode

2) Register mode.

3) Indirect mode.

4) Index mode

5) Base with Index

6) Base with Index and offset

7) Relative

1. Immediate mode :- In this mode, the operand is specified on the instruction itself.
2. Register mode :- The operand is the contents of a register. We specify the operand in this case by specifying the name of the register in the instruction. Processor register often used for intermediate storage during arithmetic operations. This addressing mode is used at that time to access the registers.
3. Indirect mode :- The effective address (E.A.) of the operands is the contents of a register in the memory location where address appears in the instruction. The name of the register or the memory address is placed in parentheses to denote Indirection or in other words that the contents are addresses of the operands.
4. Index mode :- The effective address of the operand is calculated by adding a constant

value to the contents of a register, which is clearly shown. The address can be in a register used specially for this purpose or any of the general purpose registers. In either case it is called as an Index register.

5.- Base with Index mode:- The effective address is the sum of contents of two registers. The first register as before is called the Index and the second register is called the base register. This mode provides more flexibility since both the components are registers and can thus be changed.

6.- Base with Index and offset mode:- The effective address is the sum of contents of two registers and a constant. The constant value in this case is often called the offset or the displacement.

7.- Relative mode :- For relative addressing, also called PC-relative addressing, the

Implicitly referenced register is the program counter (PC). That is, the next instruction address is added to the address field to produce the EA, typically, the address field is treated as a two's complement number for this operation.

Example:-

- Auto increment mode :- The effective address of the operand is the contents of a register specified in the instruction. After accessing the operand, the contents of this register are automatically incremented to the next value. This increment is 1 for byte sized operands, 2 for 16 bit operands and so on.
- Auto decrement mode :- The effective address of the operand is the contents of a register specified in the instruction. Before accessing the operand, the contents of this register are automatically decremented and then the value is accessed.

* Data Transfer and Manipulation :-

Data Transfer Instructions cause transfer of data from one location to another without changing the binary information.

The most common transfer are between

the

- Memory and Processor registers.
- Processor registers and Input output devices.
- Processor register themselves.

Typical Data Transfer Instructions.

Name	Hemomic.
Load.	LD
Store	ST
Move	MOV
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP
RET	
MUL	

A. Data manipulation Instructions.

Data manipulation instructions perform operations on data and provide the computational capabilities for the computer. These instructions perform arithmetic, logic and shift operations.

1- Arithmetic Instructions :-

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Subtract with borrow	SUBB
Negate (2's complement)	NEG

2- Logical and Bit Manipulation Instructions :-

Name	Mnemonic
Clear	CLR
Complement	COM

AND

OR

Exclusive-OR

Clear carry

Enable Interrupt

Disable Interrupt

Set carry

Complement carry

AND

OR

XOR

CLRC

SETB CI

OR

SETC

CONC.

3- Shift Instructions:-

Name Mnemonic.

logical shift right SHR

logical shift left SHL

arithmetic shift right SHRA

arithmetic shift left SHLA

rotate right ROR

rotate left ROL

rotate right through carry RORCT

rotate left through carry ROLC

A. What is Pipelining?

Pipelining is the process of accumulating instruction from the processor through a pipeline. It allows storing and executing instructions in an orderly process. It is also known as pipeline processing. Pipeline is a technique where multiple instructions are overlapped during execution. Pipeline is divided into stages and connected with one another to form a pipe like structure. Instruction enters from one end and exit from another end.

Pipeline increases the overall instruction throughput. In pipeline system, each segment consists of an input register followed by a combinational circuit. The register is used to hold data and combinational circuit performs operations on it. The output of combinational circuit is applied to the input register of the next segment.

Types of pipeline — It is divided into 2

Categories —

1.- Arithmetic pipeline.

2.- Instruction pipeline

1.- Arithmetic pipeline — Arithmetic pipelines are usually found in most of the computers. They are used for floating point operations, multiplication of fixed point numbers etc.

2.- Instruction pipeline — In this a stream can be executed by overlapping fetch, decode and execute phases of an instruction cycle. This type of technique is used to increase the throughput of the computer system.

Advantages of pipelining —

- 1.- The cycle time of the processor is reduced.
- 2.- It increases the throughput of the system.
- 3.- It makes the system reliable.

Disadvantages of pipelining :-

- 1.- The design of pipelined processor is complex and costly to manufacture.
- 2.- The Instruction Latency is more.

A. RISC Processor :-

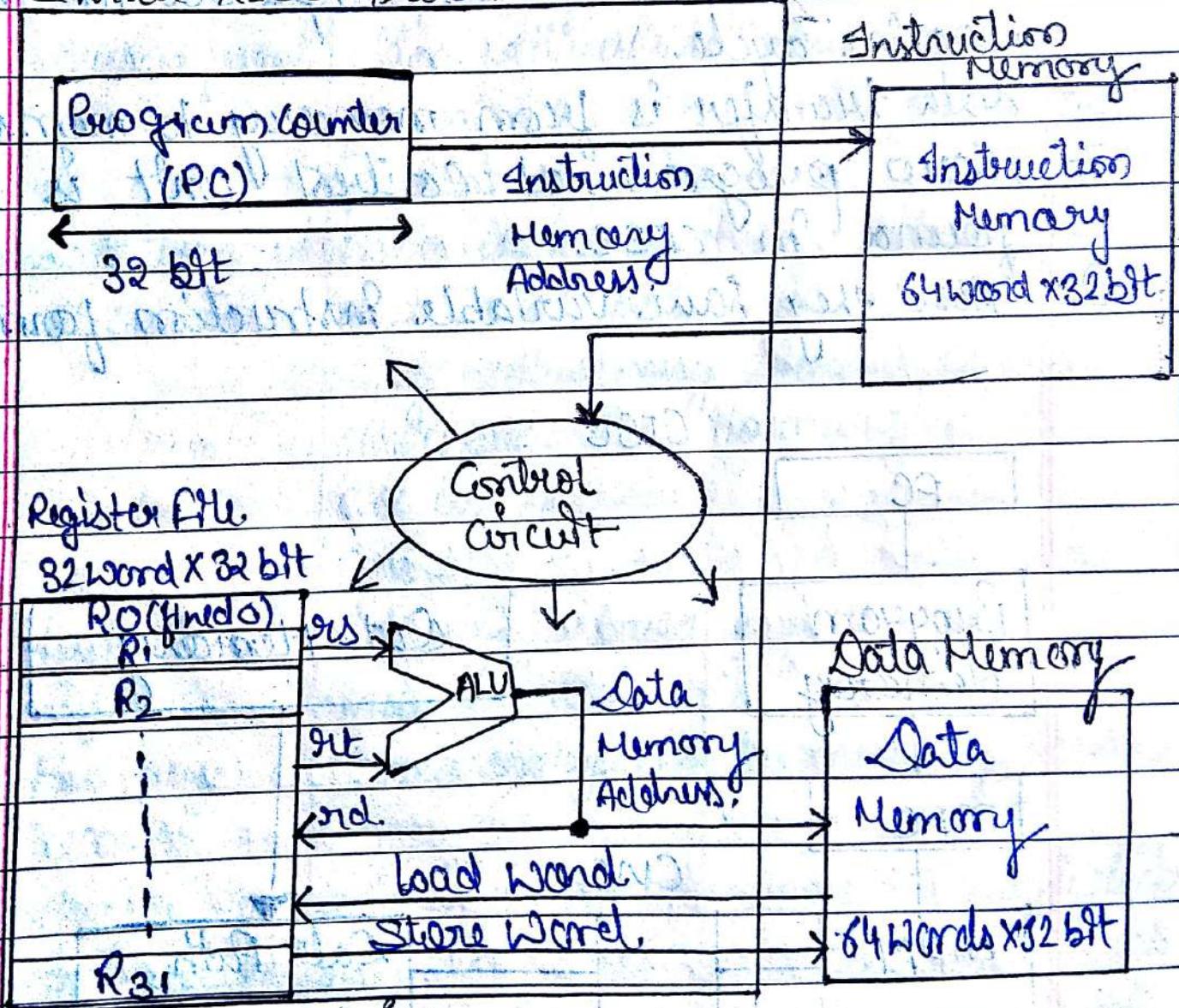
It is known as Reduced Instruction set Computer. It is a type of microprocessor that has a limited number of instructions. They can execute their instructions very fast because instructions are very small and simple.

RISC chip requires fewer transistors which make them cheaper to design and produce.

In RISC, the instruction set contains simple and basic instructions from which more complex instructions can be produced. Most instructions complete in one cycle, which allows the processor to handle many instructions at same time.

In this, Instructions are register based and transfer takes place from register to register.

Small RISC Processor (SRP).

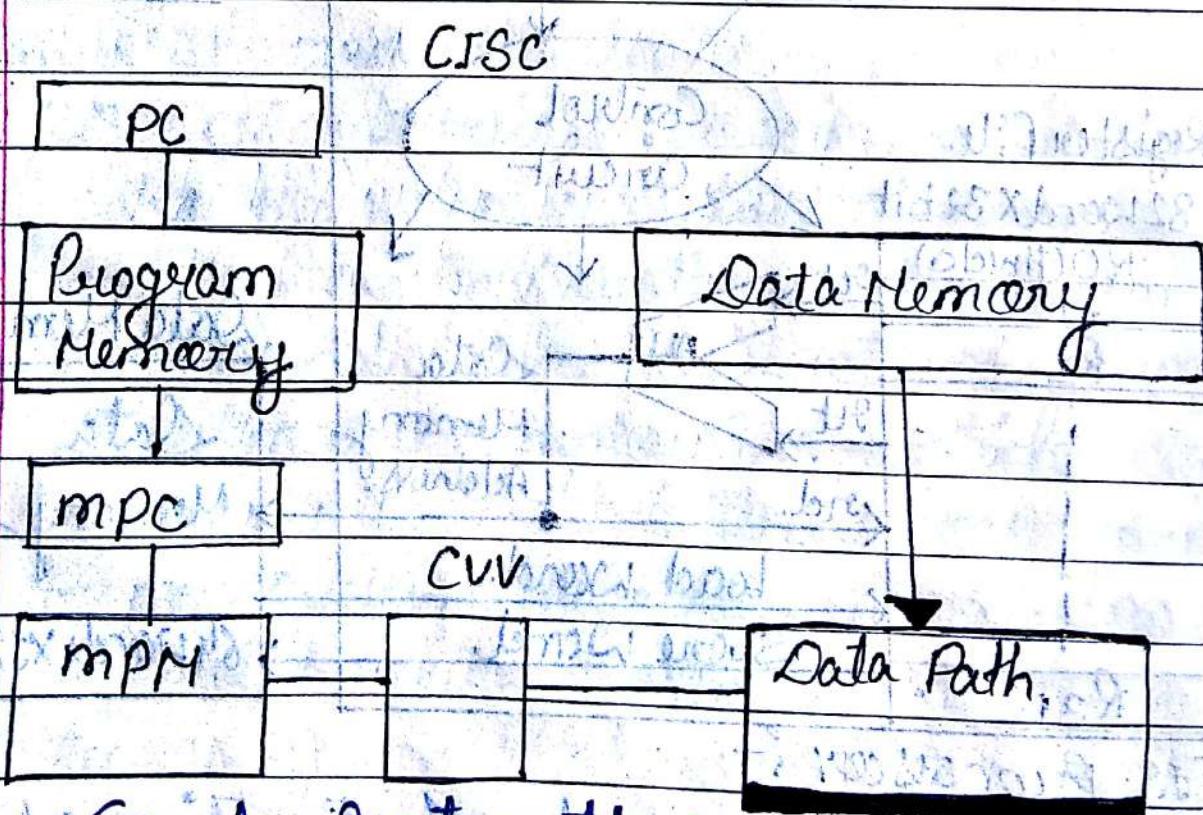


A. CIS Processor :-

1. It is known as Complex Instruction set Computer.

2. It consists of large number of instructions and

- 2.- It was first developed by Intel.
- 3.- It contains large number of complex instructions.
- 4.- In this Instructions are not register based.
- 5.- Instructions cannot be completed in one machine cycle.
- 6.- Data transfer is from memory to memory.
- 7.- More programmed control unit is found in CISC.
- 8.- Also they have variable instruction formats.



Complex Instructions
One instruction in several CVV PM standard.

A. Difference between CISC and RISC

Architectural characteristics	Complex Instruction set Computer (CISC)	Reduced Instruction Set Computer (RISC)
Instruction size and format	Large set of instructions with variable formats (16-64 bits per instruction).	Small set of instructions with fixed format (32 bit).
Data transfer	Memory to memory.	Register to register.
CPU control	Most micro coded using control memory (ROM) but modern CISC use hardwired control.	Mostly hardwired without control memory.
Instruction type	Not register based instructions.	Register based instructions.
Memory access	More memory access.	less memory access.
Clocks	Includes multi-clocks.	Includes single clock.
Instruction of nature	Instructions are complex.	Instructions are reduced and simple.

A Vector (Array) Processor and Its Types.
Array processors are also known as multiprocessors or vector processors. They perform computations on large arrays of data. Thus, they are used to improve the performance of the computer.

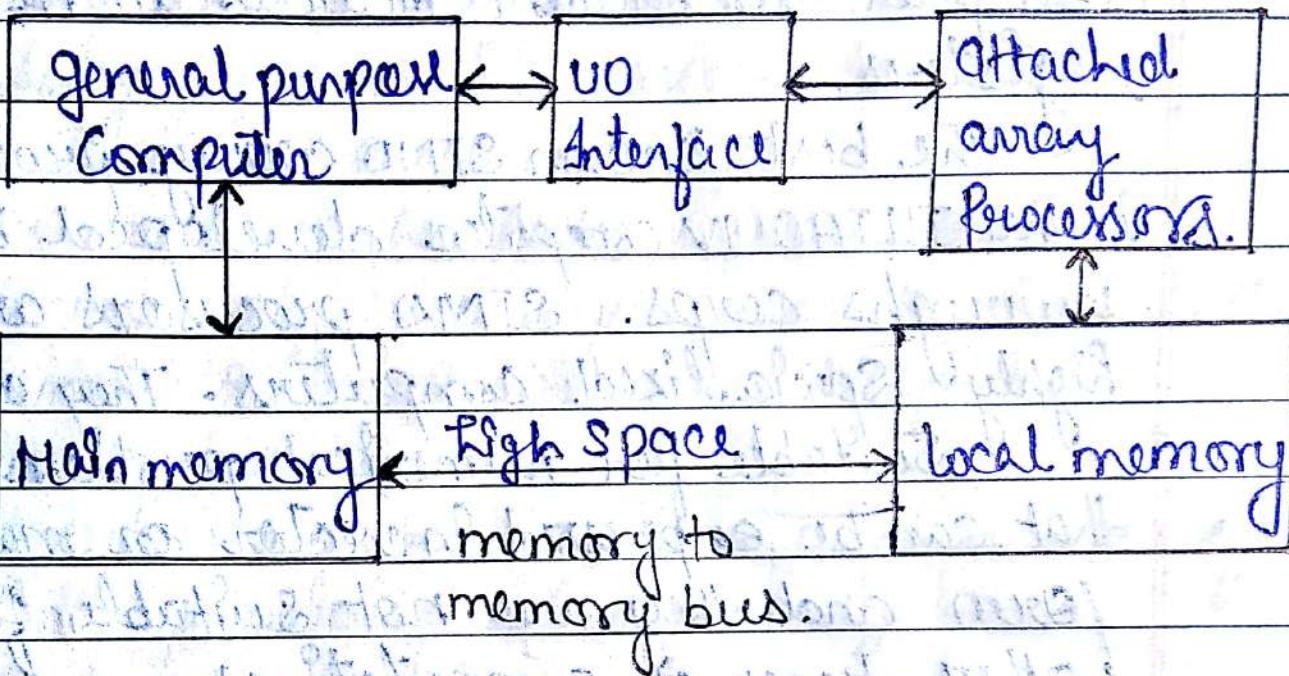
Types of array processors -

There are basically two types of array processors:

- 1.- Attached array Processors.
- 2.- SIMD Array Processors.

1.- Attached Array Processors -

An attached array processor is a processor which is attached to a general purpose Computer and its purpose is to enhance and improve the performance of that Computer in numerical computational tasks. It achieves high performance by means of parallel processing with multiple functional units.



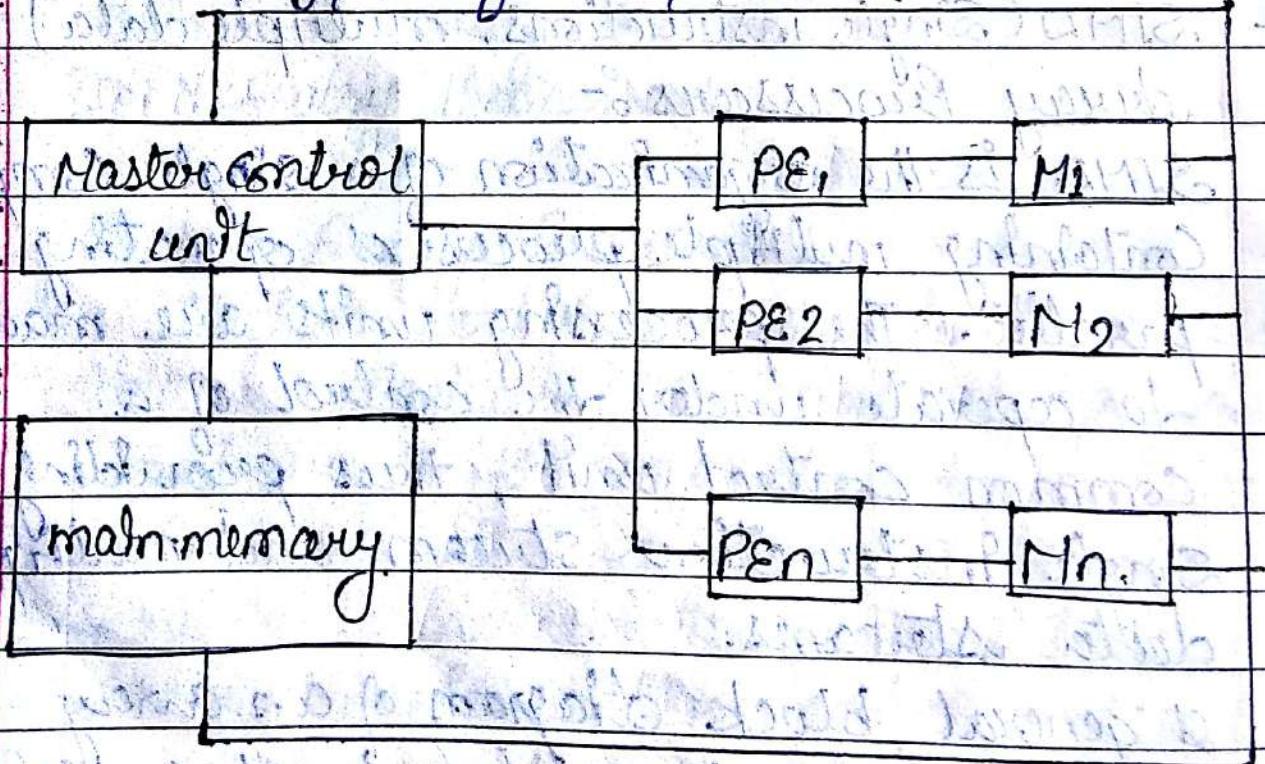
2- SIMD (Single Instructions, multiple data)
array Processors :-

SIMD is the organization of a single computer containing multiple processors operating in parallel. The processing units are made to operate under the control of a common control unit, thus providing a Single Instruction Stream and multiple data streams.

A general block diagram of an array processor is shown below. It contains a set of identical processing elements (PPE's), each of which is having a local memory M.

Each processor element includes an ALU and registers.

The best known SIMD array processor is the ILLIAC IV computer developed by the Burroughs corps. SIMD processors are highly specialized computers. They are only suitable for numerical problems that can be expressed in vector or matrix form and they are not suitable for other types of computations.



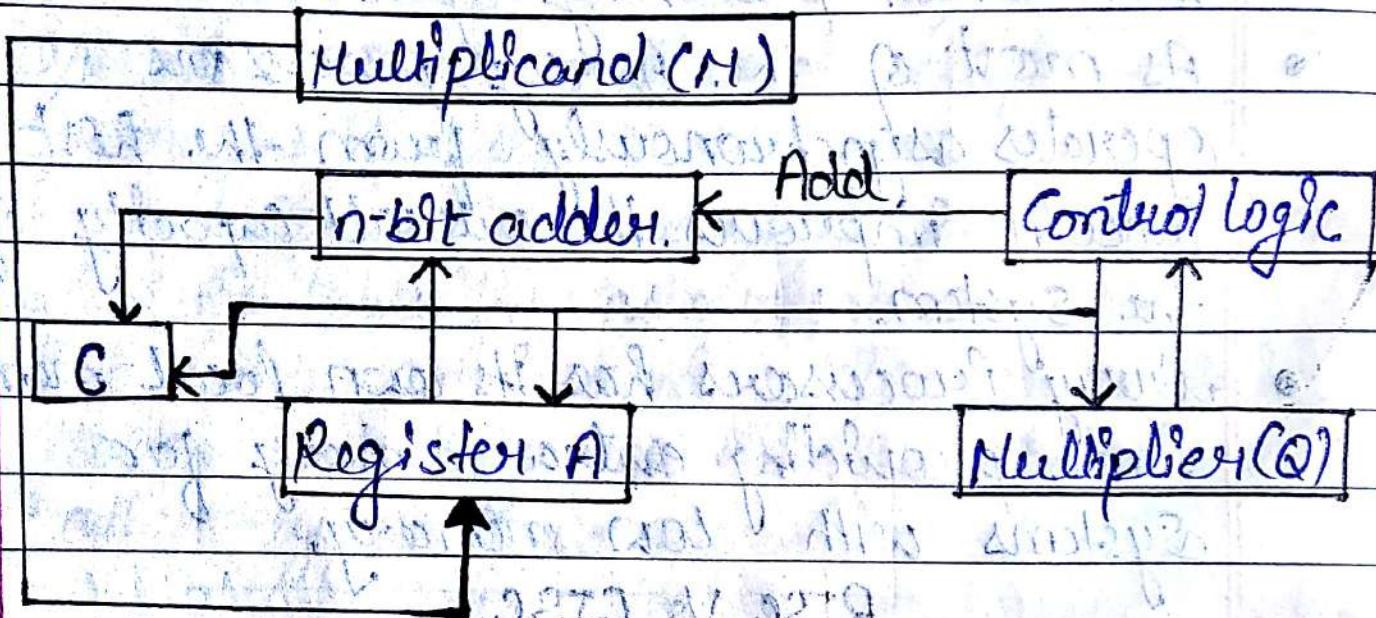
Q. Why use the Array Processor?

- Array processor increases the overall instruction processing speed.
- As most of the Array processors operates asynchronously from the host CPU, hence it improves the overall capacity of the system.
- Array Processors has its own local memory, hence providing extra memory for systems with low memory.

RISC Vs CISC

Parameter	RISC	CISC
Instruction types	Simple	Complex.
Number of Instructions	Reduced (30-40)	Extended (100-200)
Duration of an Instruction	One cycle	More cycles (4-120)
Instruction format	Fixed	Variable.
Instruction execution	In parallel (pipeline)	Sequential
Addressing modes	Simple	Complex
Instructions accessing the memory	Two: Load and Store	almost all from the set unique.
Register set	multiple	In CPU (micro-program).
Complexity	In computer	

Hardware Structure of Multiplication, using Shift and Add.



Booth's algorithm :-

Flow chart -

(Start)

$M \rightarrow$ multiplicand.

$Q \rightarrow$ Multiplier

$q_0 \rightarrow 0$

$A \rightarrow 0$

$n \rightarrow$ no. of bits

$A = A + M$

$A = A + M$

Arithmetic Shift Right AQn0

$n = n + 1$

NO

is
 $n=0$

?

Result In AQ

Stop.