

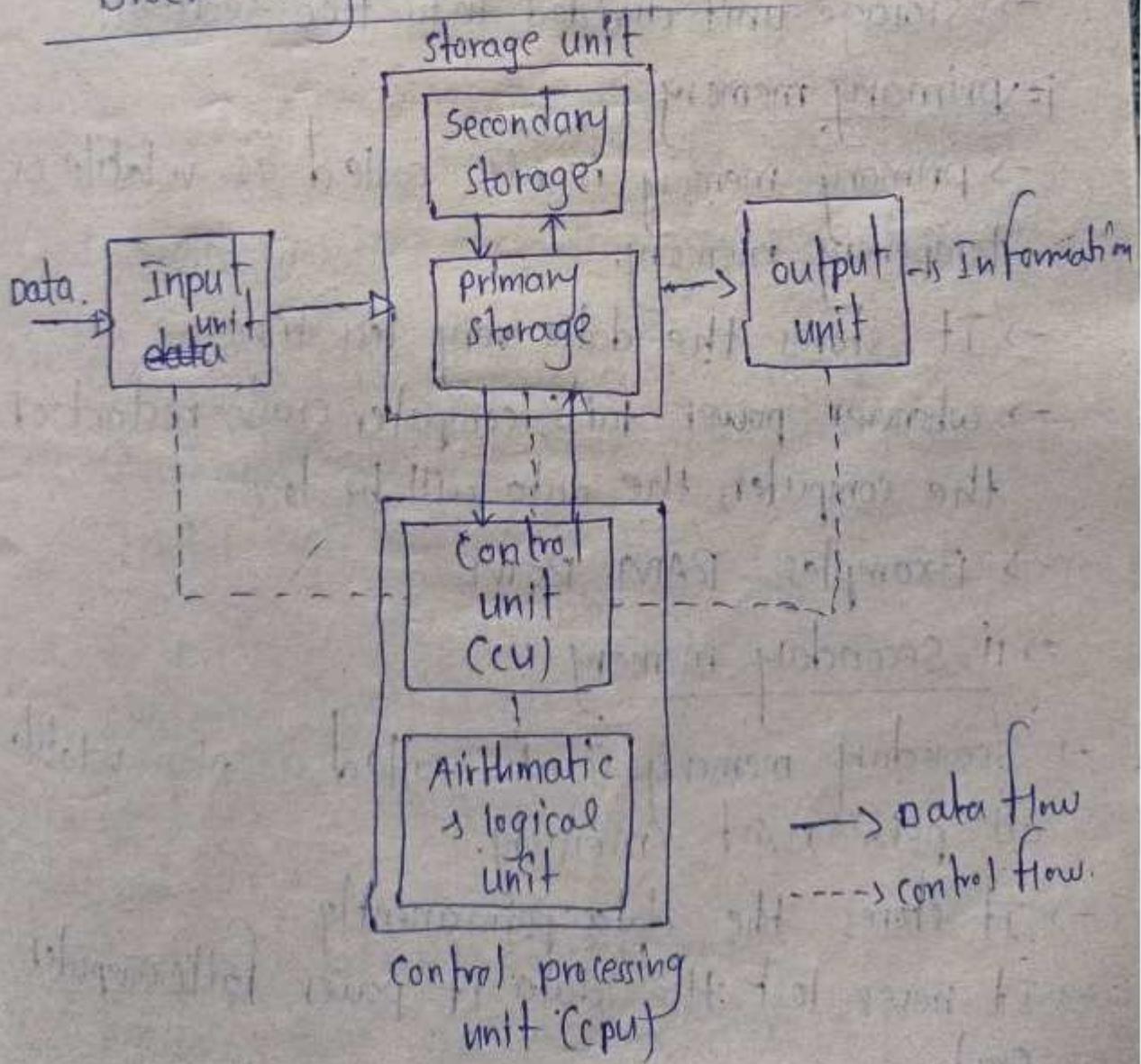
unit-1

Digital computer: A digital computer is considered to be a calculating device, that can perform arithmetic operations at enormous speed.

(or)

It is defined as device that operates upon information/data.

Block diagram of computer



Input unit

- > Instructions to the computer is called Input unit.
- > By using some devices we can give instructions to the computer.
- > Those devices are called input devices.
- > Input devices : Keyboard, Mouse, joystick, stylus etc.

Storage unit

- > Storage unit stores the information / data that we sent through input devices.
- > Storage unit divided into two sections.

i) Primary memory :

- > Primary memory is also called as volatile or temporary memory.
- > It stores the data very less time.
- > whenever power fails, computer crash, restart of the computer, the data will be lost.
- > Examples : RAM, ROM

ii) Secondary memory :

- > Secondary memory is also called as non-volatile (or) permanent memory.
- > It stores the data permanently.
- > It never lost the data, if power fails, computer crash.
- Ex. pen drives, compact disk (CD), digital video disk (DVD)

Central processing unit

- > CPU is also called as Heart or Brain of the computer.
- > CPU is divided into following types.
 - i) Control unit (CU)
 - ii) Arithmetic logic unit (ALU)

i) control unit

- > It controls and manages all the components of the computer system.
- > Control unit decides in which instruction will be executed and operations performed.
- > CU takes care of the step by step processing of all operations performed by the computer.

ii) Arithmetic logic unit

- > Arithmetic logic unit is also called as digital logic unit.
- > It performs all arithmetic operations like addition, subtraction, multiplication.
- > It takes the data from input and send to the CU.

- > Output unit : result of the input is called output.
- > By using some devices we get output from the computer.
- > Those devices are, monitor, printer.

Definition of computer organization

Computer organization is defined as the internal arrangement of computer which includes the design of the processor, memory and input/output circuit.

(Cor)
It is defined as the realization of what is specified by the computer architecture.

Differences b/w computer architecture and organization

- | | |
|----------------------------------|------------------------------|
| <u>Computer org architecture</u> | <u>computer organization</u> |
|----------------------------------|------------------------------|
- > Computer architecture describe what the computer does.
 - > It deals with functional behaviour of computer system.
 - > It deals with high level design issue.
 - > It comes before computer organization.
 - > Also called as instruction set architecture.
- > Computer organization describe how does it
 - > It deals with structural relationship.
 - > It deals with low level design issue.
 - > It comes after the architecture part.
 - > Also called as microarchitectures.

Register Transfer Language:

- > concerned with what to do.
- > It indicates its hardware - It indicates its performance.
- > Architecture involves
 - > organization involves physical components
 - * logic
 - * instruction sets
 - * data types
 - * addressing modes
 - * circuit designs
 - * address signals

- | | |
|-------------------|--|
| <u>Register</u> : | Register is a very fast computer memory used to store data/instruction, it is called register. |
|-------------------|--|
- > A register is a group of flip flops, each flip flop capable of storing one bit of information.
 - > Some registers, Accumulator, general purpose register, special purpose register.
- Register Transfer Language:
- > The term 'register transfer' can perform micro operations and transfer the result of operation to some other register.
 - (Cor) The information transformed from one register to another register.

form is called register transfer.

- Registers are designed by capital letters.
- Sometime followed by numbers.

(eg:- A, R1, R2, T1, T2)

Basic symbols of RTL

Symbol	Description	Example
Capital letter	denotes register	MAR, R2
()	denotes a part of register	R(0-7), R1(4)
\leftarrow	denotes transfer of info.	R ₂ \leftarrow R ₁
:	denotes termination of control function	R ₂
;	separates two micro operations	A \leftarrow B, B \leftarrow A

MR - memory address register

PC - program counter.

IR - Instruction register.

Various methods of RTL

(a)

control

clock

(b)

15
PC
PC(H)
PC(L)

Register Transfer: The term "register transfer" can perform micro operations and transfer the result of operation to some other register. It is called Register Transfer.

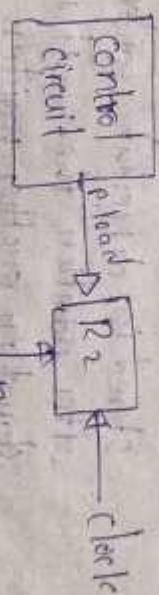
Standard notations for specifying operations

a) MAR - memory address register.

b) PC - program counter.

c) IR - Instruction register.

\rightarrow The following image shows block diagram that shows the transfer of data from R1 to R2.



R1

Micro operations

The operations executed on values stored in register are called micro-operations.

Types of micro operations

1. Arithmetic micro operation
2. logic
3. shift

1. Arithmetic micro operation

→ In CPU arithmetic logic unit [ALU] performs various operations like:

- i. Addition: In addition, two operands are stored in different registers, and after addition, the result is stored in another register.

$$R_1 \leftarrow R_2 + R_3$$

- ii. Subtraction: In subtraction, two operands are stored in different register and after subtraction, their result is stored in another register.

$$R_1 \leftarrow R_2 - R_3$$

2. Logic micro operations: logic micro operations are working on few logic gates

→ There are 4 types of logic micro operations:-

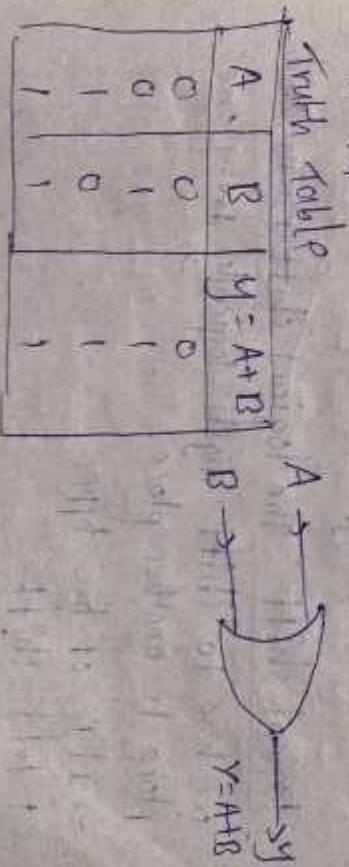
- i. AND: AND gate is known as series circuit.
→ R_2 and R_3 contain two inputs and the result is stored in R_1 .



Truth Table

		$y = A \cdot B$	
		A	B
		0	0
		0	0
		1	0
		0	1
		1	1

- * ii. OR gate: Two inputs are stored in R_2 and R_3 register, and the result is stored in R_1 .



Truth Table

		$y = A + B$	
		A	B
		0	0
		0	0
		1	0
		0	1
		1	1

iii) X-OR : It is like an OR-gate, but excluding both inputs are true.

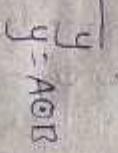
Truth Table

A	B	$y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0



iv) X-NOR : This is an X-OR gate but with an inverted output.

A	B	$y = A \otimes B$
0	0	1
0	1	0
1	0	0
1	1	1



③ Shift micro operations

There are three types of shift micro operations.

i) logical shift : The logical shift means that we have to shift digits logically from one place to another place.

-> it is of two types
i) left shift
ii) right shift.

Left shift : Let us suppose a register has 4 bit value 1011, and if we shift this value to left then the left most digit is discarded and remaining three digits are shifted to the left and new digit '0' added to the number.

$$1011 \rightarrow 0110 \quad [1011 \rightarrow 0110]$$

Right shift : Let us suppose a register has 4 bit value 1011, and if we shift this value to right, then the right most value is discarded and the remaining three digits are shifted to the right and new digit '0' added to the number.

$$1011 \rightarrow 1010 \quad [1011 \rightarrow 1010]$$

ii) circular shift : As the name defined

circular means circle 'con rotation'. Here digits are not discarded, they are shifted in a circular motion.

i) left circular shift : Let us suppose a register has 4 bit value 1011 and if we shift this value in circular, then the left most digit is discarded, it is shifted to right most corner, remaining three shifted to left.

$$1011 \rightarrow 0111$$

ii Right circular shift

Let us suppose a register has 4 bit left and if we shift left value in circular, & right most digit is not discarded, it is shifted to the right most corner and remaining three are shifted to left-right
 Ex: 1011 \rightarrow 1101

3. Arithmetic shift:

Arithmetic operation is shift applied to signed values of numbers.

i Left arithmetic shift

It is same as logical shift, but sign is not going to change.

Ex: 1101 \rightarrow -1010

ii Right arithmetic shift

It is same as logical shift.

+ve on the left most digit is copied

Ex: 1011 \rightarrow 1101

Computer Registers

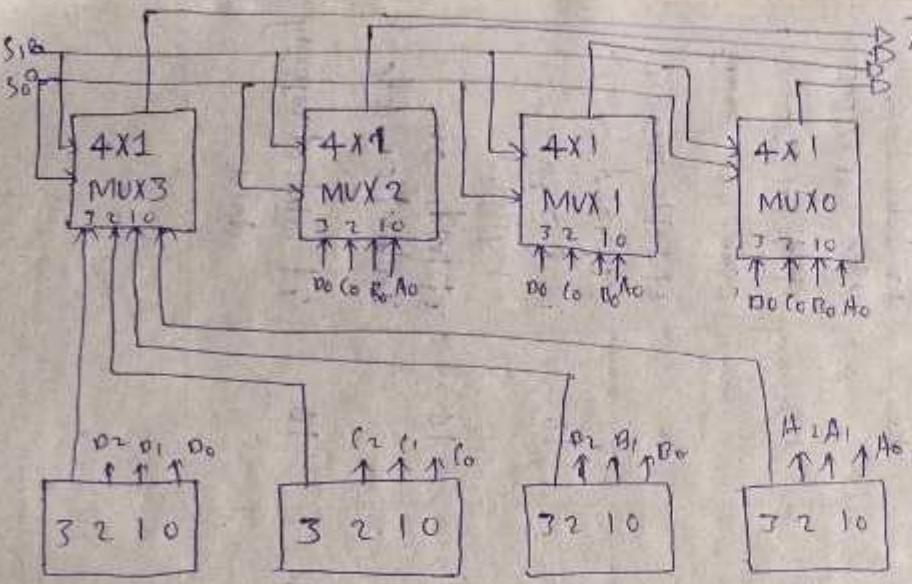
Register Name	Register Symbol	No. of bits	Function
Output Register	OUTR	8	It holds a/p character
Input Register	INTR	8	It holds the character
Program Counter	PC	12	It holds the address of instruction
Address Register	AR	12	It holds the address of memory
Data Register	DR	16	It holds memory operand
Accumulator	AC	16	It is a processor register
Temporary Register	TR	16	It holds on instruction code
			It holds temporary data.

BUS and memory Transfers

- A more efficient scheme for transferring information b/w registers in a multiple register configuration is a common bus system.
- one way of constructing a common bus system is with multiplexers.

4 line common Bus

Block diagram



Ex Bus system for four registers

- The two selected lines S_1 and S_0 are connected to the selection inputs of all four multiplexers.
- The following table shows the four possible binary values of the selection lines.
- Function table for bus of above figure

	S_1	S_0	Register selected
	0	0	A
	0	1	B
	1	0	C
	1	1	D

Computer Instruction

- Each computer has three instruction formats, as shown in below figure.

→ Computer instructions

1. Arithmetic, logical, shift instructions
2. Instructions for moving information from memory to register.
3. Program control instructions
4. Input and output instructions
5. Memory reference instructions

15 14 12 11
[I] [opcode] [Address] (opcode = 000 through 11)

15	12 11	0
0 1 1 1	Register operation	(opcode = 111, I=0)

b) Register reference Instruction.

15	12 11	0
1 1 1 1	i/o operation	(opcode = 111, I=1)

Basic Computer Instructions

Symbol	Microcode Code ^I	Description
AND	0XXX	AND memory word to AC
ADD	1XXX	Add memory word to AC
LDA	2XXX	Load memory word to AC
STA	3XXX	Store AC content in memory
BUN	4XXX	Branch unconditionally
BSA	5XXX	Branch and save return address
ISZ	6XXX	Increment and skip if 0

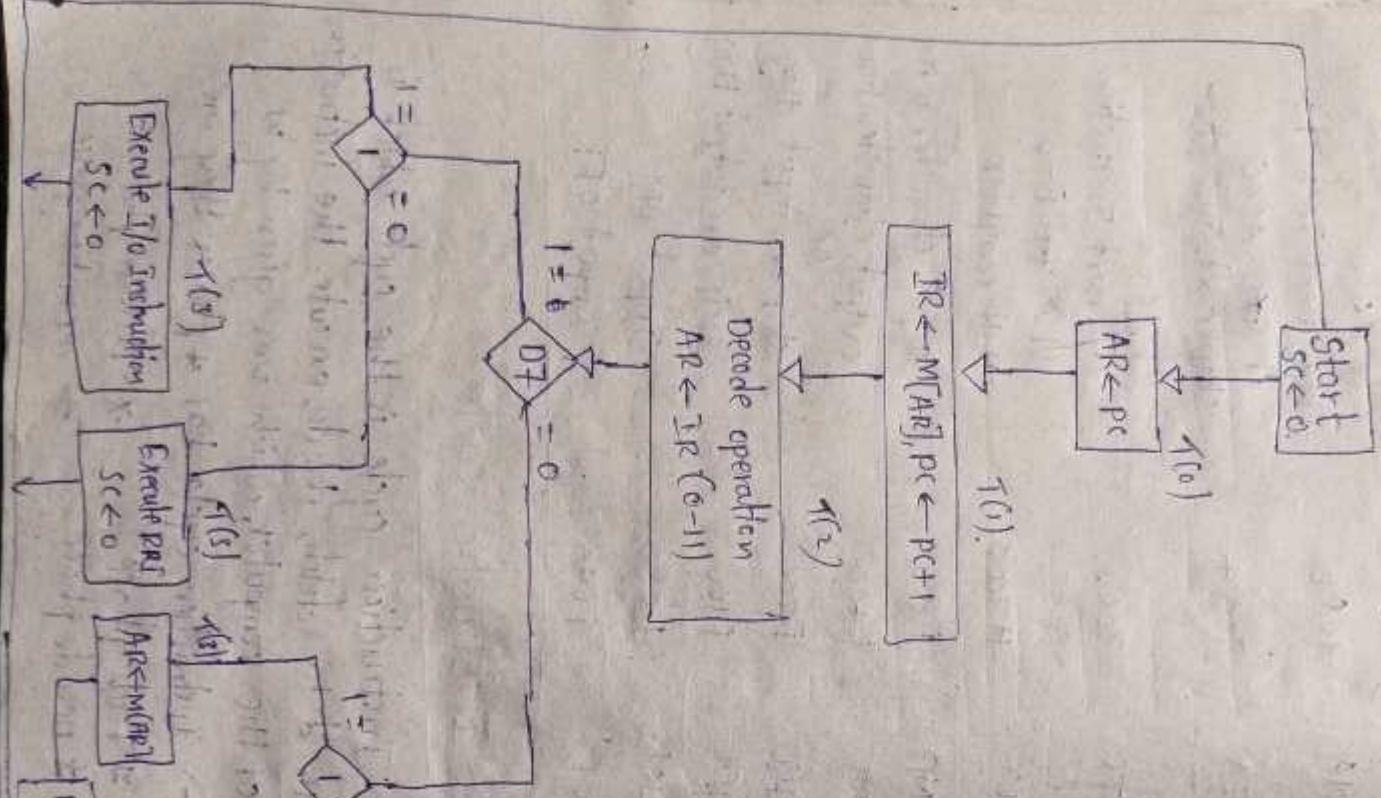
INP	HT	
F000	→	Input character to AC
F400		Output character from AC
SKT		Skip on input flag
SKO		Skip on output flag
ION		Input on
IOF		Input off
F040		Input off

Instruction cycle

Instruction cycle is the cycle which is used to fetch, decode, execute. the instruction of the computer, which was given by us.
 -> Instruction cycle has 4 phases they are:
 * fetch phase
 * decision phase
 * execute phase
 * decode phase

SPA F010 skip next instruction
 if AC > 0
 SNA F008 skip next instruction
 if AC = 0, AC<0
 STA F004 skip next instruction
 if AC = 0.
 SPA F002 skip next instruction
 if AC < 0, F = 0
 HLT F001 halt computer.

→ flow chart of instruction cycle



(wher

AR = Address Register

PC = program counter

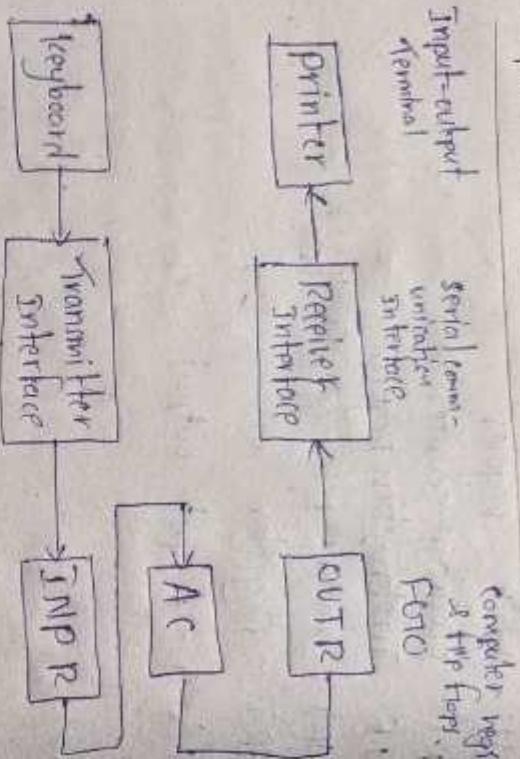
IR = Instruction Register

MRI = memory reference instruction

$$RR_1 = \eta_{\text{req}} e^{-\gamma_1}$$

No \Rightarrow Input-output

Input - output and interrupt



FOR

- > The above figure shows the flow chart of input - output and interrupt.
- > Here input information is given from keyboard to transmitter interface.
- > Transmitter receiver passes to input register.
- > In between input and output registers accumulator is used.
- > The information Accumulator projects the information to output register which projects to receiver interface, finally reaches to printer.
- > printer prints the data (i.e.) information which initially given by keyboard.

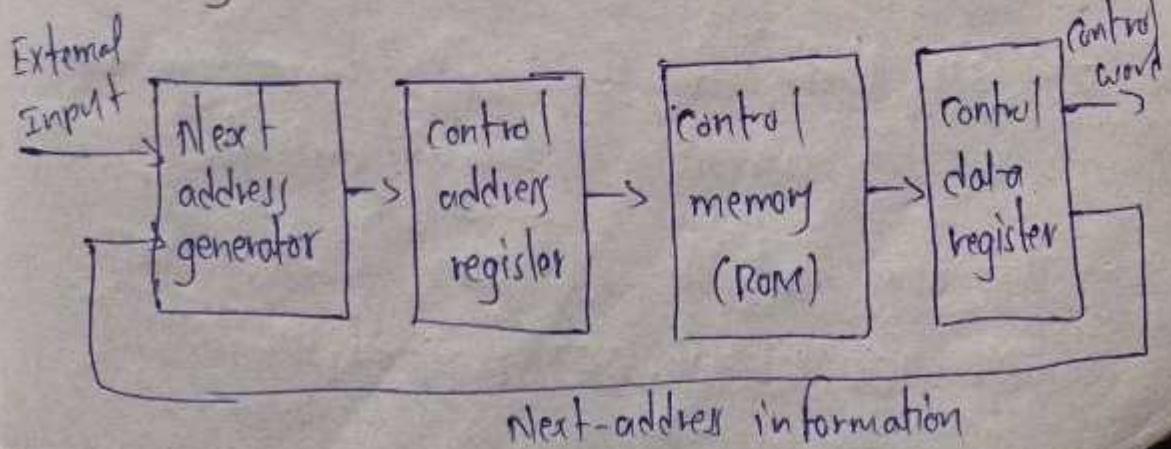
Input - output Instructions

P: SET - C.	INP	PR 11	Input character.
OUT	PR 10	Output character.	
SFT	PR 9	skip on input flag.	
SKP	PR 8	skip on output flag.	
TON	PR 7	Input enable on	
TOF	PR 6	Input enable off.	

unit - 2

control memory

- control memory is the storage in the microprogrammed control unit to store the microprogram.
- control memory is a part of the control unit.
- Any computer that involves microprogrammed control consists of two memories. They are main memory and control memory.
- programs are usually stored in main memory by user.
- Microprograms are usually stored in control memory.
- The general configuration of a microprogrammed control unit shows in below block diagram.



→ The above block diagram shows micro-programmed control organization.

→ The control memory assumed to be ROM in which all control information is permanently stored.

→ The control memory address register specifies address of micro-instruction.

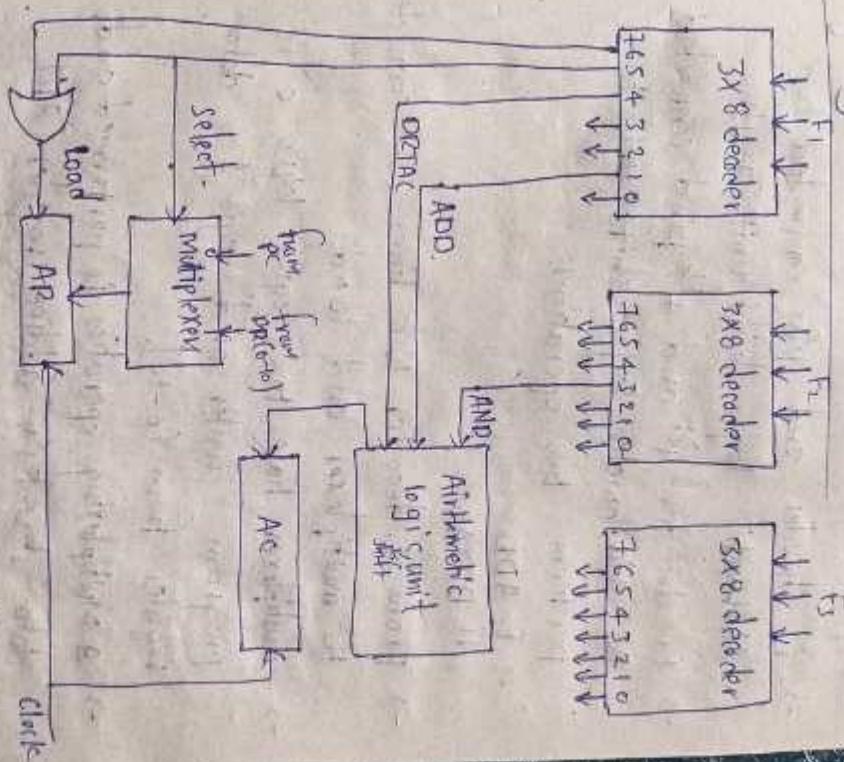
→ The control data register holds the microinstruction.

→ The next address generator is sometimes called as 'microprogram sequencer'.

→ The data register sometimes called as pipeline register.

→ Control data register is again connected to next address generator.

Design of Control unit



→ The block diagram shows design of control unit.

→ Design of control unit is of two types.

i) Hardwired control unit

2. micro programmed control unit

→ Here 3 three 3x8 decoders are used namely F₁, F₂, F₃.

- F₃ decoder not connected to anything.
- F₂ decoder and pin connected to Arithmetic logic shift unit (ALU).
- F₁ decoder 1st and 4th pins connected to Arithmetic logic shift unit to perform bus operation.

- i ADD
- ii DRTAC
- iii:
- From F₁ decoder two lines are connected to multiplexer and load.
- Multiplexer has two logic. Logic '0' is program counter and logic '1' is data register from (0-10).
- A multiplexing operation is performed and data stored in address register (AR).
- Arithmetic logic shift unit takes the result in accumulator (AC).
- In three decoders only some pins are used connected to get output.
- The few outputs of three decoders are AND, ADD, DRTAC, DRTAP, PCTA, PCTP.

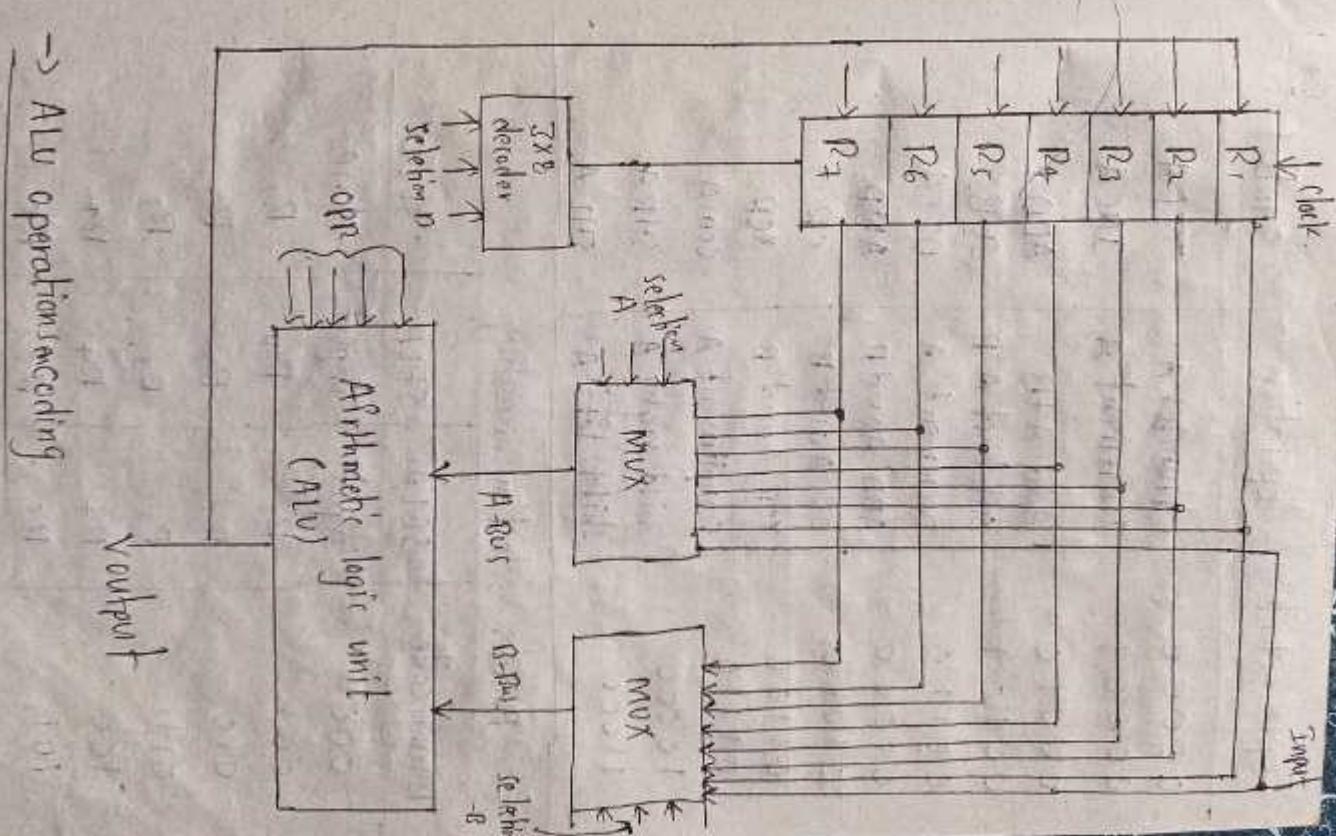
- They also designed intell. instruction using gates to generate control signals.
- The op's stored in Accumulator.
- The multiplexer takes many I/O and store result in AR.

Hardwired control unit	Microprogrammed control unit
→ The hardwired control unit induces the control signals required for processor.	→ The microprogrammed control unit induces the control signals through micro-instructions.
→ It is faster than microprogrammed control unit.	→ It is slower than Hardwired control unit.
→ It is difficult to design.	→ It is easy to design.
→ It is more expensive.	→ It is less expensive.
→ It is difficult to manage complex instructions.	→ It is easy to manage complex instructions.
→ It uses limited instructions.	→ It uses many instructions.

- | | |
|--------------------------------------|--------------------------------------|
| -> control memory is absent. | -> control memory is present. |
| -> chip area is less | -> chip area is more. |
| -> Instruction set of size is small. | -> Instruction set of size is large. |

General register organization

- Genera register organization, refer to the structure and usage of general registers in CPU.
 - These registers are used for various computational and manipulation tasks.
 - Block diagram of general register organization



OPR Select	operation	Symbol
00000	Transfer A	TSFA
00001	Increment A	INCA
00010	Add A+B	ADD
00011	Subtract A-B	SUB
00110	Decrement A	DECA
01000	ADD A and B	ANP
01010	OR A and B	OR
01100	XOR A and B	XOR
01110	Complement A	COMA
10000	Shift right A	SHRA
11000	Shift left A	SHLA

→ Register selection encoding

Binary code.	SEL A	SEL B	SEL D
000	Input	Input	None
001	R ₁	R ₁	R ₁
010	R ₂	R ₂	R ₂
011	R ₃	R ₃	R ₃
100	R ₄	R ₄	R ₄
101	R ₅	R ₅	R ₅
110	R ₆	R ₆	R ₆
111	R ₇	R ₇	R ₇

Addressing Modes

- The addressing modes help us specify refers to the way in which operand of an instruction is specified.
- It is useful to determine the operand of an effective address.

→ A Bus organization & CPU registers shown in above table.

→ The OPR of each register is connected to two buses.

→ The A and B-buses are given. Inputs for Arithmetic logic unit [ALU].

→ Arithmetic logic unit performs Arithmetic and logical operations.

→ The output of ALU again connected to input of Registers.

→ It is a loop operation.

→ Types of addressing mode

1. Immediate addressing mode

→ In immediate addressing mode, in place of address we mention operand.

→ Address = operand.

2 Register addressing mode

→ In register addressing mode, register contains the operand.

→ Register = operand.

3. Register indirect addressing mode

→ Register having the effective address, so we have to go that effective address to determine operand.

→ Register = Effective address = operand

4. Auto-increment addressing mode

→ It is similar to register indirect addressing mode, but it follow the post increment approach.

→ operand = 700_16 .

5. Auto-decrement addressing mode

→ It is similar to register indirect addressing mode but it follows pre-decrement approach.

→ operand = 450_16 .

6. Direct addressing mode

→ In this mode instruction contains address that address is effective address.

Address → Effective address → operand.

7. Indirect addressing mode

→ In this mode, the instruction contains address that address is effective address that contains operand.

Instruction → address → effective address → operand.

8. Relative address mode

→ The content of the program counter is added to the address field value, we get effective address.

→ program counter + address field value = effective address

9. Base register addressing mode

The address content of base register value is added to the address field value, we get effective address

Base register value + address field value = effective address

Data transfer and manipulation Instructions

i. Data transfer instructions

→ Data transfer instructions are used to transfer of data from one location to another location without changing the binary information.

- The most common data transfer is between memory to processor register
- processor register to I/O devices
- processor register itself.

ii. Data manipulation Instructions

→ Data manipulation Instructions perform operations on data.

→ These instructions perform arithmetic, shift logic and shift operations.

1. Arithmetic Instructions

→ Typical data transfer instructions

Name	Mnemonic
Load	LD
Store	ST.
Move	Mov
Exchange	XCH
Input	IN
Output	OUT
Push	PUSH
Pop	POP

Name	Mnemonic
Increment	INC
Decrement	DEC
Add	ADD
Subtract	SUB
Multiply	MUL
Divide	DIV
Add with carry	ADDC
Add with borrow	SUBB
Register complement	NEG

2. Logical Instructions

Name	Mnemonic
clear	CLR.
complement	COM.
AND	AND
OR	OR.
Exclusive-OR	X-OR
Clear carry	CRC.
Set carry	SET.
Complement carry	COMC.
Enable Interrupt	EI.
Disable Interrupt	DI

3. Shift Instructions

Name	Mnemonic
Shift right	SHR.
Shift left	SHL.
Arithmetic shift right	SHRA.
Arithmetic shift left	SHLA.
rotate right	ROR.
rotate left	ROL.
rotate right with carry	RCR.
rotate left with carry	RCR.

program control instructions

- program control instructions provide a decision-making capability, and change the path taken by the program.
- some typical program control instructions

Name	Mnemonic
Branch	BR.
Jump	JMP.
Skip	SKP.
Call	CALL.
Compare	CMP.
Test	TEST.

Instruction format

- the instruction format are a sequence of bits (0 and 1).
- These bit groups known as fields.
- Instruction format also defines the layout of the bits.

- depending upon on the multiple address fields, the instruction is categorized as follows.

1. zero address instruction

- The instruction format in which there is no address field is called zero address instruction.
- In zero address instruction, stacker are used.
- There is no operand.

MODE	OPCODE
------	--------

Example:

- Assembly language instruction - ~~push~~ push-A, push-B, etc.

- Stack transfer operation - TOS ← A, TOS ← B

2. one Address instruction format

- The instruction format in which the instruction uses only one address field is called one address instruction format.
- It has only one operand.

MODE	OPCODE	OPERAND
------	--------	---------

→ Example

- Assembly language instruction - Load, ADD, store.
- operation register organisation - R1 ← M[A] + M[B] etc.

3. Two(r) address instruction format

- The instruction format in which the instructions uses only two address fields is called two-address instruction format.
- Mostly used instruction format.
- It has two operand.

→ Example

- Assembly language instruction - M[OP1,A], ADD R1,B etc.

4. Three (3) address instruction format

- The instruction format in which instruction uses three address fields is called...
- It has three operand.

unit - 3

DATA Representation

Data types

→ Computer programs (or) applications may use different types of data based on the problem.

→ different types of data.

- Numeric data - Integer and Real Numbers
- Non-numeric data - character data, address data, logical data.

i Numeric data

• Integer data :

→ It is used for whole numbers.

• whole Real numbers

→ It can be expressed as,

1. Fixed-point representation
2. Floating-point representation.

i) Non-numeric data

i character data: A sequence of character is called character data.

→ A character may be:

- alphabetic (A-Z) & (a-z)
- numeric (0-9)
- special character (+, #, *, @)

or combination of all these

-Example of character data

- Manoj#

- 561 prasanna123@

ii logical data: A logical data type

is used by computer to take logical decisions

→ It is different from numeric and alphanumeric.

→ logical data type is denoted by

True(T) or False(F).

→ logical data consisting of relational symbol (> < = etc)

→ It is also called as boolean data type

Address data type

→ Address and data are mostly used in computers.

→ Data is the term used to describe information.

→ Address is the term used to describe the location of the information.

Complements

→ Complements are used in digital computers for simplifying the subtraction operation and for logical manipulation.

→ There are two types of complements

i) i complement ii) (r-1)'s compliment

→ When base 'r' is substituted in name, r's ord (r-1)'s complements return to 0 if 1 and r's complements for binary number.

→ Similarly 10's and 9's complement for decimal number.

i) r^1 complement

\Rightarrow For decimal numbers, $r=10$.

($r-1$)

r^1 complement = 10^1 's complement.

\rightarrow The 10^1 complement of decimal number is obtained by adding 1 to the 9^1 complement.

\rightarrow For ex. 10^1 complement of 2389 is 7611

9 9 9 9

—
9 9 9 9
2 3 8 9
—————
7 6 1 0

(+) —
7 6 1 1
—————

\Rightarrow For binary numbers, $r=2$

\Rightarrow For binary numbers, $r=2$
 $(r-1)^1$ complement = $(2-1)^1$'s complement = 1^1 complement.
 \rightarrow The 1^1 complement of binary number is obtained by changing 1's into zeros and zeros into ones.

\Rightarrow For binary numbers, $r=2$

r^1 complement = 2^1 's complement.
 \rightarrow The 2^1 complement of binary number is obtained by adding 1 to 1^1 's complement.

for ex. 2^1 complement of 10100 is 01010
its complement: 101100

ii) $(r-1)^1$'s complement

\Rightarrow For decimal numbers, $r=10$.

$(r-1)^1$'s complement = $(10-1)^1$ = 9^1 complement.

\rightarrow 9^1 complement of decimal numbers is obtained by subtracting each number from 9.

\rightarrow 9^1 complement of 546700 is 453299.

9 9 9 9 9 9

—
9 9 9 9 9 9
5 4 6 7 0 0
—————
4 5 3 2 9 9

\rightarrow For ex. 9^1 complement of 00000 is 111010
 9^1 complement of 101100 is 0100101

Data representation

→ Data representation refers to the form in which data is stored, processed and transmitted is called data representation.

→ It is of two types they are:

i) Fixed point representation

ii) Floating point representation

Fixed point Integer representation

Signed mag

① Fixed-point representation : In fixed

point notation, there are a fixed number of digits after the decimal point.

is called fixed point representation.

Integer representation

1. signed-magnitude representation
2. signed 1's complement
3. 2's complement

For example,

→ Consider a signed number +14 stored in an 8-bit register.

→ +14 is represented by a sign bit of 0 in the left most position followed by the binary

14: 00001110

→ There is only one way to represent +14, but

Here are three different ways to represent

-14 with 8-bits, they are:

In signed magnitude representation:

10001110

In 1's complement : 11110001

In 2's complement : 11110010

n Arithmetic Addition :

$$\begin{array}{r}
 +6: 000000110 \\
 +13: 00000110 \\
 \hline
 +19: 000001001
 \end{array}$$

$$\begin{array}{r}
 +6: 000000110 \\
 -13: 11110011 \\
 \hline
 -7: 11111001
 \end{array}$$

Air Chemistry subtraction

→ The subtraction operation can be changed to an addition operation, if the sign is changed.

$$(\pm A) - (+B) = (\pm A) + (-B)$$

$$(\pm A) - (+B) = (\pm A) + (-B)$$

for example, $(-6) - (-13) = +7$, or $(-6) + (+13) = +7$.

-06-11-11010

+17.000 1101

111,000,000

पुस्तकालय

Concurrent sessions in Eiffel

Decimal fixed point representation

4385 : 0100 0011 1000 0101

$$\text{for ex: } (+375) + (-240) = +135$$

101. compliment of -240.

qqq qqq

~~1399~~

160

Floating-point representation

In floating point representation there is no fixed number of digits after the decimal point it called floating point representation.

→ The floating point representation have

two part:
A. first part is signed fixed point

number called montiss.

In the second part is position of defining point is called exponent.

The fixed point monads may be a fraction or integer.

Ex The decimal number +6132.789.

negative Exponent

6.6132789, +04,

→ The equivalent is $+0.6152789 \times 10^4$

→ The floating point represented as
 $\underbrace{ }_{\text{mantissa}} \times \underbrace{ }_{\text{exponent}}$

$M \times r^e$

where $m = \text{mantissa}$.

$e = \text{exponent}$.

$r = 10$ for decimal.

$r = 2$ for binary.

Floating point arithmetic operations

→ The floating-point number in computer register consists of two parts

i) Mantissa 'm'

ii) Exponent, ' e' '

→ This two parts represents a number obtained from multiplying 'm' times and radix ' r ' to the value of ' e '.

" $M \times r^e$ ".

i) Addition and subtraction

- During addition (or) subtraction, the two floating point operands are in AR and BR.
- The algorithm can be divided into four consecutive part.

1. check for zeros.
2. Align the mantissa.

→ Add (or) subtract the mantissa.

→ Normalize the result.

Multiplication

multiply

Multiplicand in BR
multiplier in CR

$= e$

$\neq 0$

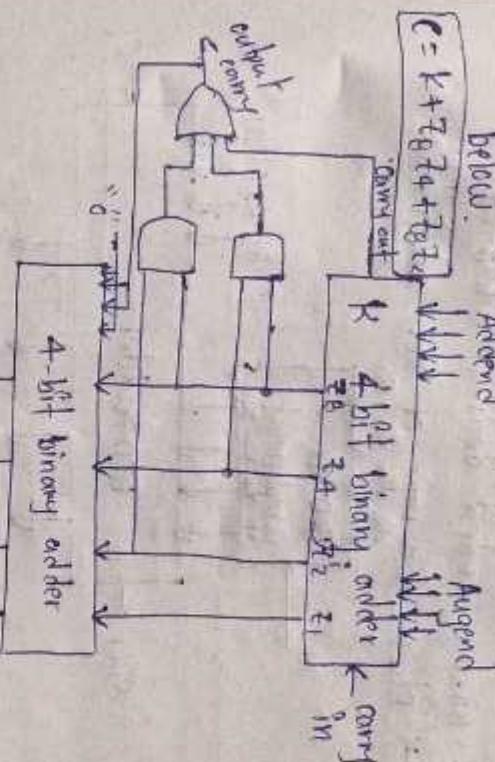
Division: The division of two floating-point numbers will always result in a normalized quotient.

- The division algorithm can be divided into five parts
 - check the term
 - Align the dividend
 - divide the mantissa.
 - subtract the exponent.
 - Initialize the register and

Decima | Arithmetik und

- To perform arithmetic operations with decimal data, it is necessary to convert the input decimal numbers to binary.
 - It is necessary to perform ~~to per~~ all calculations with binary and convert into 'decimal'.
 - It is very efficient method.
 - i BCD Adder
 - The binary coded decimal adder.

Derivation of BCB adder



ii BCD subtractor

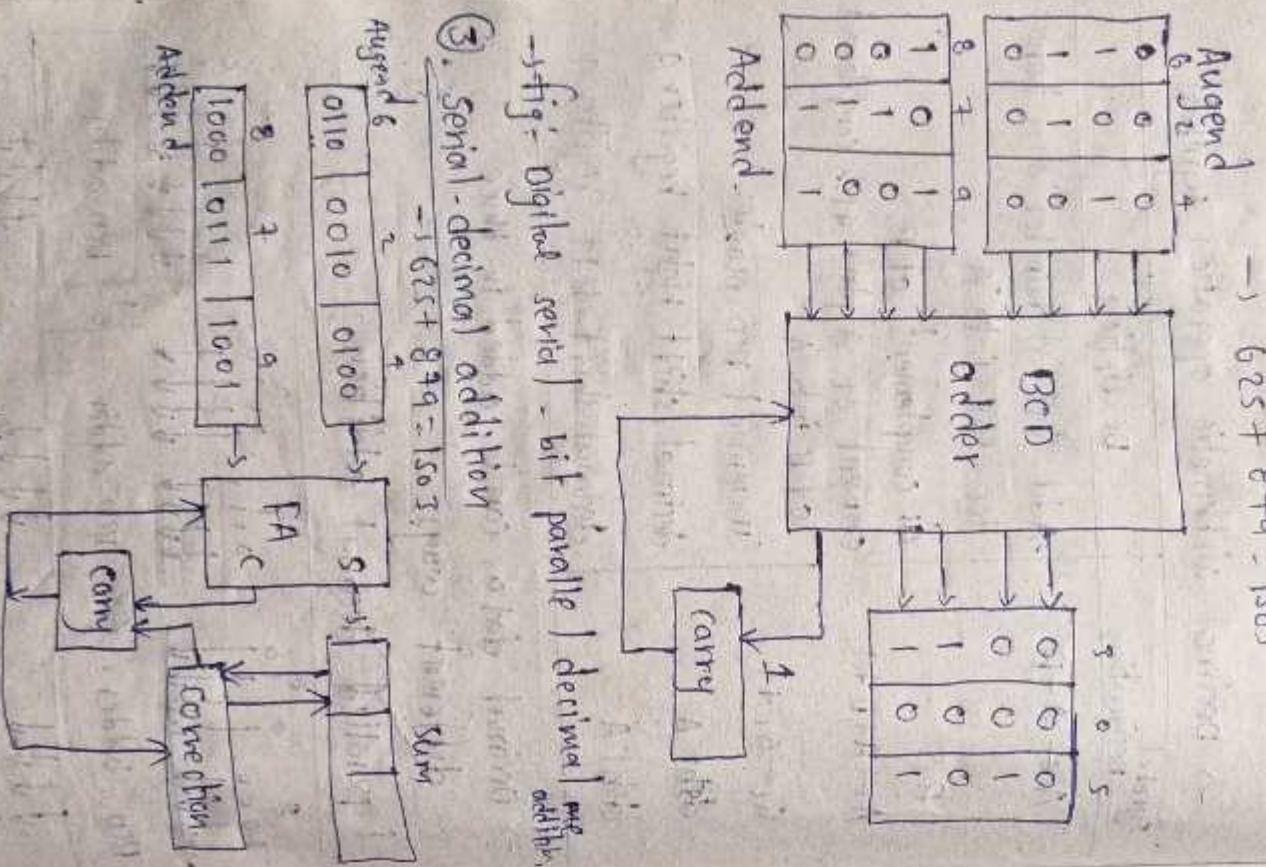
- > Binary coded decimal subtractor.
 - > Block diagram of BCD subtractor.
-
- $A \leftarrow A + B$
- $B \leftarrow \bar{B}$
- $A \leftarrow A + \bar{B} + 1$
- $Q_1 \leftarrow Q_1 + 1$
- Shift A
- $C \leftarrow A_1 A_2 A_3 A_4$

Decimal Arithmetic operations

- > Decimal arithmetic operation symbols
- | Symbol | Designation | Description |
|--------------------------------|---|-------------|
| $A \leftarrow A + B$ | Add decimal number and transfer sum into A. | |
| \bar{B} | q1 complement of B. | |
| $A \leftarrow A + \bar{B} + 1$ | content of A plus, q1's complement of B into A. | |
| $Q_1 \leftarrow Q_1 + 1$ | increment BCD number in Q1. | |
| Shift A | decimal shift right register A. | |
| $C \leftarrow A_1 A_2 A_3 A_4$ | decimal shift-left register A. | |
- > Block diagram shows one stage of decimal arithmetic unit.
 - > It consists of BCD adder and q1's complement.
 - > Mode ($M=0$) controls the operation
 - > S is the output of A plus q1's complement
 - + BCD complement - B.
 - > Decimal data can be added in three different ways.
 - i parallel decimal addition:
-
- $624 + 879 = 1503$
- fig: parallel decimal addition: $624 + 879 = 1503$

1. Digital serial - bit parallel decimal addition

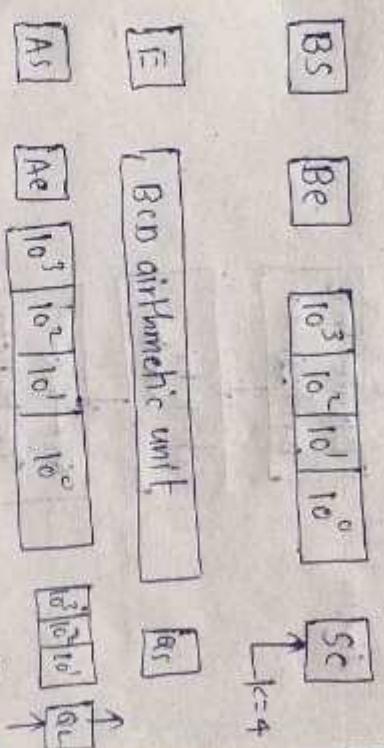
$$625 + 879 = 1503$$



→ fig: digital serial 1 - bit parallel decimal addition

2. Multiplication

→ The register configuration for decimal multiplication are shown in fig.



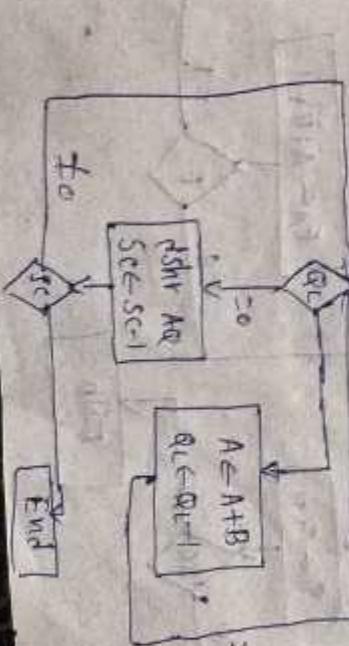
→ Flow chart for decimal multiplication

↓ multiply
multiplicand in 'B'
multiplier in 'A'

↓
Are Q₁ & Q₂ = 0?
Ans No
See sc-1

↓
increment
decked

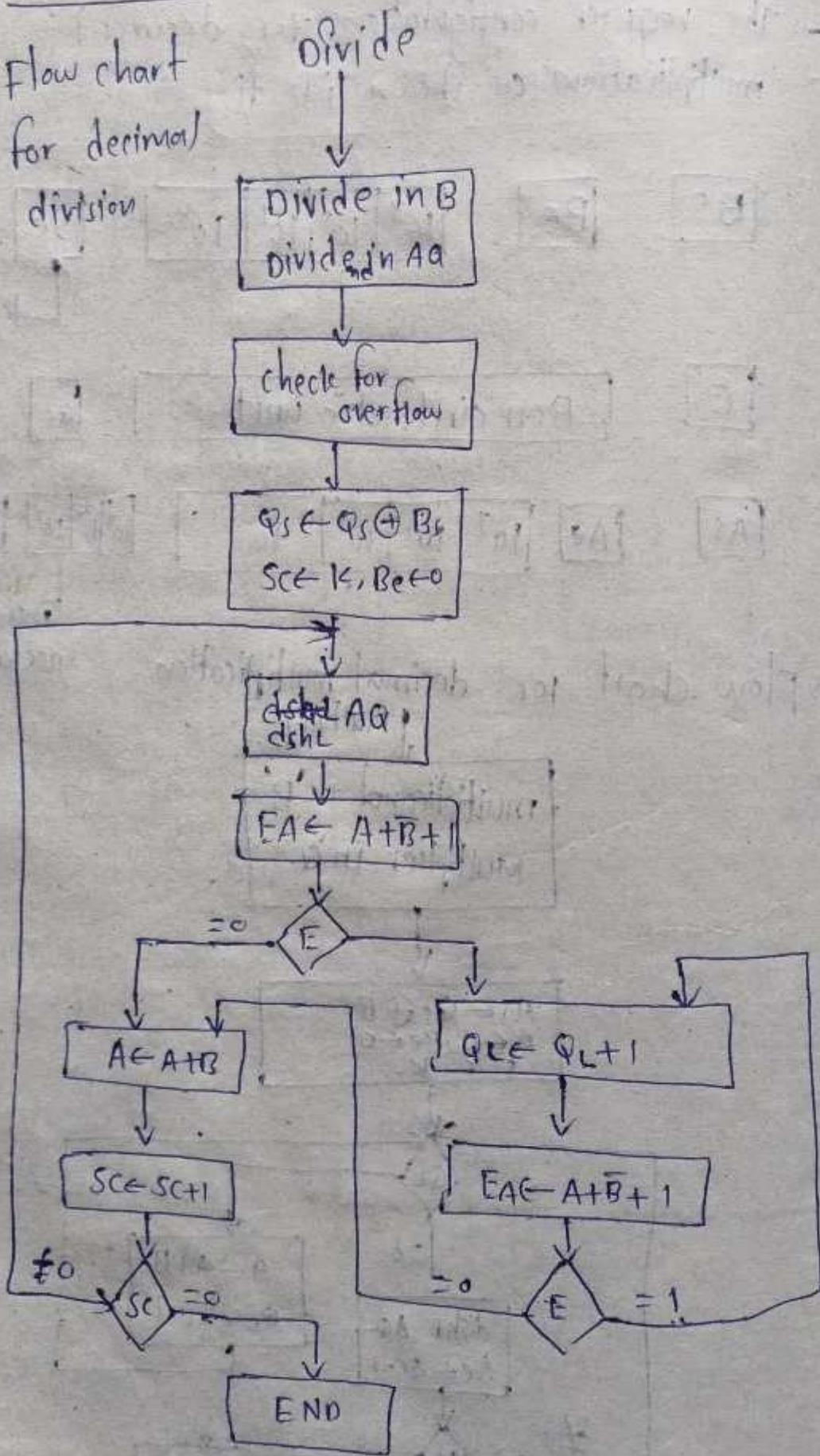
↓
 $A \leftarrow A + B$
 $Q_{12} \leftarrow Q_{12} + 1$



→ serial decimal addition

Division

→ Flow chart
for decimal
division

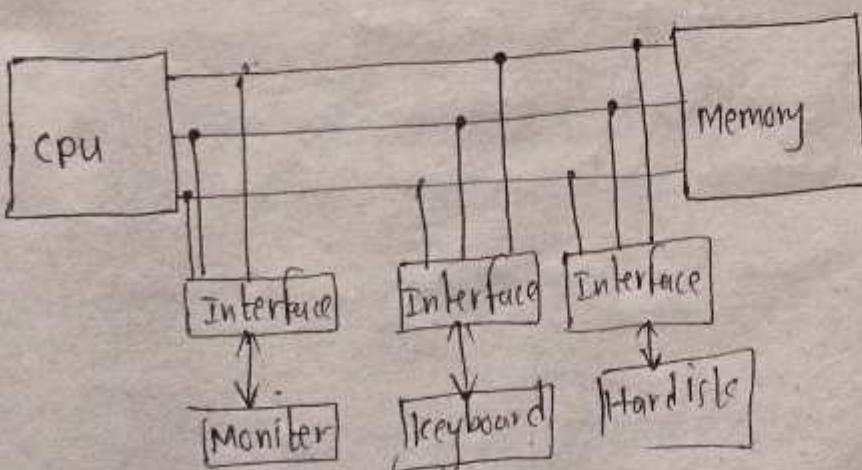


unit - 4

Input - output organization

Input - output interface :

- The input - output interface provides a method for transforming information between internal storage and external I/O devices.
- A peripheral device is that provides input and output for the computer.
- peripheral devices is also called as I/O devices
- for example keyboard and mouse provide input to the computer called input devices while monitor and printer provides output to the computer is called output devices



→ The above block diagram shows input-output interface.

→ The major differences are follow:

1. The nature of peripheral device is electro-mechanical and electronic.
2. The nature of CPU is electronic.
3. The data transfer rate of peripheral devices are slow than CPU.
4. The format of peripheral devices and CPU, memory are differ.

→ I/O commands

- i control command
- ii status command
- iii data output command
- iv data input command

A synchronous data transfer

→ Asynchronous data transfer is a method of data transmission, where data is sent in a non-continuous (or) non-synchronous manner. It is called Asynchronous data transfer.

→ Asynchronous data transfer between two independent units requires control signals.

→ Thus, two methods can achieve the asynchronous data transfer.

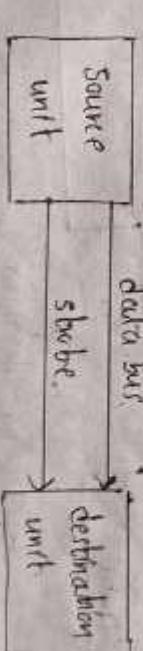
→ Two methods they are:

i) strobe control method

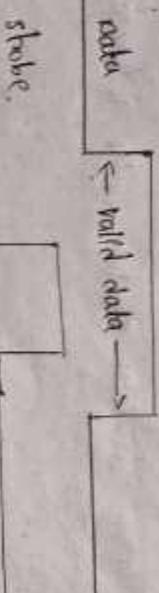
ii) handshaking method

i) Strobe control method:

i) source-initiated strobe



→ The above block diagram shows strobe control method.



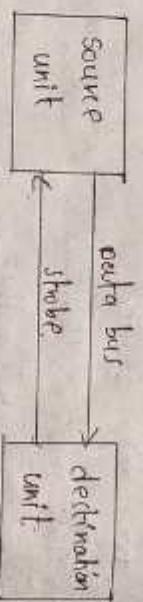
→ Above figure shows the timing diagram of

shake control method

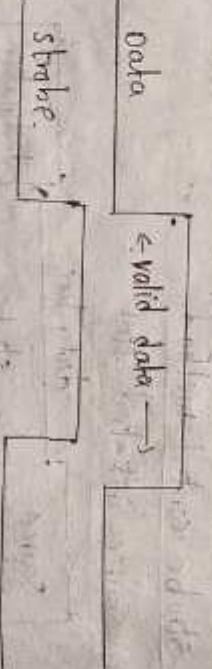
→ The block diagram of shake control method consist of source unit and destination unit, data bus etc.

→ Block diagram indicates source initiated shake for data transfer.

ii) Destination initiated shake

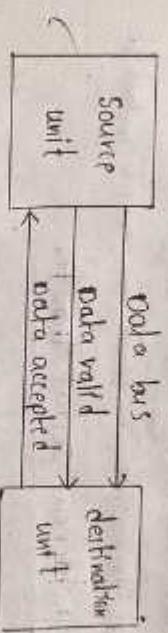


→ Block diagram



→ The two handshaking lines are data bus and data valid generated by source unit.

→ The data accepted is generated by destination unit.



→ Timing diagram

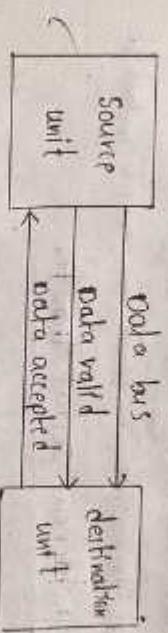
→ above block diagram is for destination initiated shake for data transfer.

→ Timing diagram, shake width increase,

ii) Handshaking Method

→ The handshake method solves the drawback. It goes along in shake control method by introducing seconding control signal.

i) Source initiated using handshaking

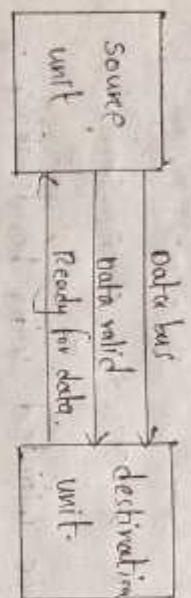


→ The two handshaking lines are data bus and data valid generated by source unit.

→ So data transfer is initiated by source unit, by enabling data valid.

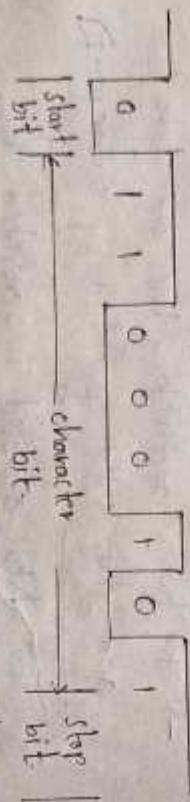
→ compare with control shake, extra signal data accepted is exist.

H Destination initiated data Transfer



- > Block diagram shows destination-initiated data transfer by handshaking
- > It procedure is same as source initiated

Asynchronous serial Transfer

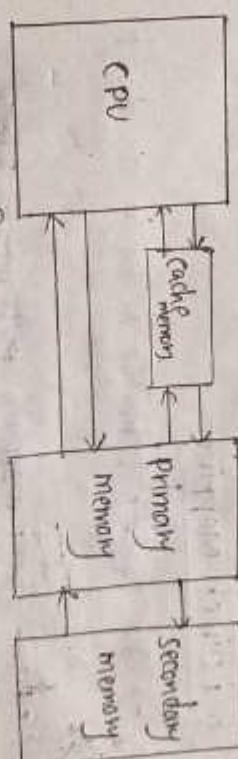


- > Block diagram shows asynchronous serial communication.
- > Serial communication may be synchronous or asynchronous.

- > Asynchronous:
 - > Initiation of the transfer starts from start bit which is always '0'.
 - > character bit follows start bit.

Cache memory:

- > Cache is a very high speed memory
- > Cache is a smaller and faster memory, that stores the data in main location
- > Cache is a CPU memory
- > There are different caches in CPU, which store instruction and data.
- > Cache memory block diagram.



- > levels of memory
- > Level 1 (L1) Register : It is a type of memory in which data is stored and accessed.
- > Most commonly used : register, Accumulator, address, register, etc.
- > Level 2 (L2) Cache memory :
 - > It is a faster memory, having faster access.
 - > Level 3 (L3) Main memory : It is the memory on which computer works currently.

→ Level 4 (64) Secondary memory

- It is the external memory, that is not fast.
- cache performance
- The performance of the cache measured in Hit ratio.

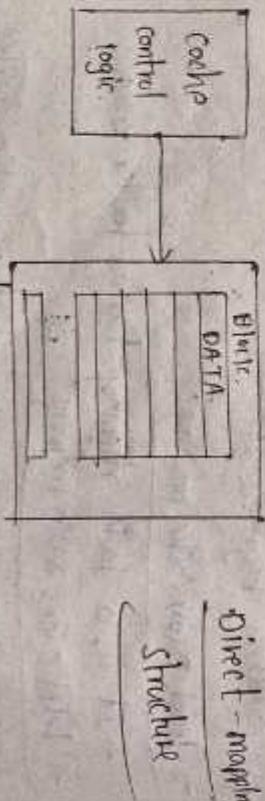
Cache mapping

- There are three types of mappings used for cache mapping.

i Direct mapping

- The simplest technique is known as direct mapping.
- In direct mapping each memory block assignments to specific line in cache.
- Direct mapping performance directly proportional to Hit ration.

Main memory

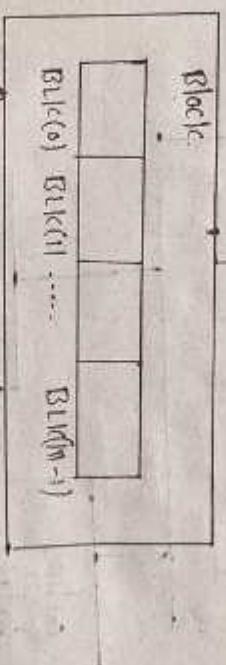


Direct-mapping structure

2. Associative Mapping

- In this type of memo mapping associative memory is used to store content.
- It is faster and flexible mapping.

Main memory



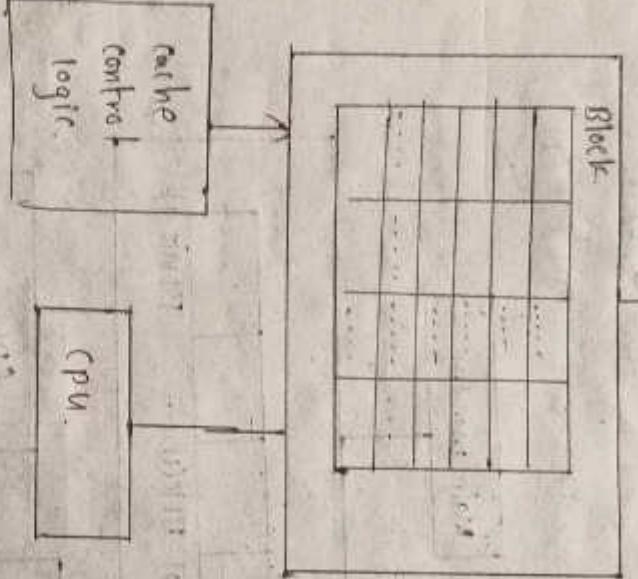
Associative mapping structure

- 3 set-associative mapping
- This is enhanced form of direct mapping.
- Drawbacks of direct mapping is removed.
- Structure of set-associative mapping

Structure of set-associative mapping

Hierarchy Memory

Main memory

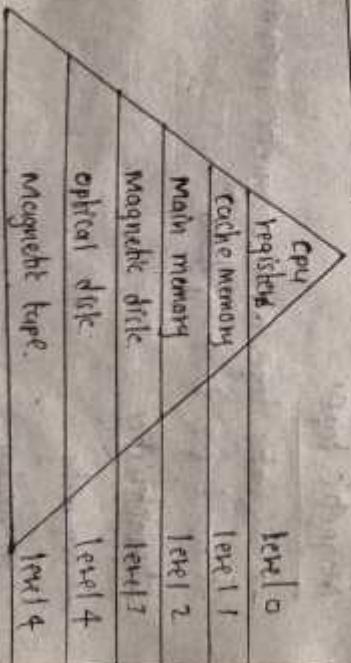


Types

i) External memory : consisting of magnetic disk, optical disk, and magnetic tape.

ii) Internal memory : consisting of main memory and cache memory and CPU registers.

Design of Hierarchy :



i) Registers

- Registers are small, high speed memory units located in CPU.
- peoples have fast access time

Cache memory

- > cache is a small and fast memory unit located in CPU.
- > It is a CPU memory

Main memory

- > Main memory is also known as RAM.
- > It is a primary memory
- > It have large storage capacity
- 1 static RAM
- 2 dynamic RAM.

Magnetic disk

- > magnetic disk are circular plates, fabricated by metal or magnified material.

Magnetic tape

- > magnetic tape is a magnetic recording device
- > used for back-up data.

Characteristics

- > capacity
- > hierarchy
- > performance
- > Access time

Main memory

- > The main memory is the central storage unit.
- > It is the essential component of the digital computer.

- > It stores the data and programs

- > Main memory is of two types
 1. RAM (Random access memory)
 2. ROM (Read only memory).

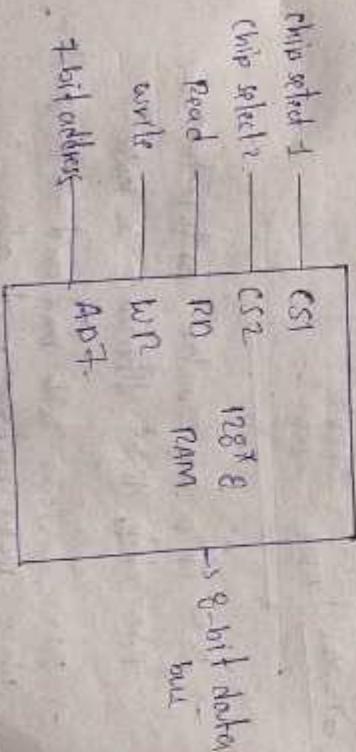
RAM : Random access memory is a volatile memory, volatile memory stores the data.

It is dependent on the power.
-> If power supply stops entire data erased.

RAM integrated circuit chips

- > The RAM main components are - flip-flop.

- > RAM chip block diagram



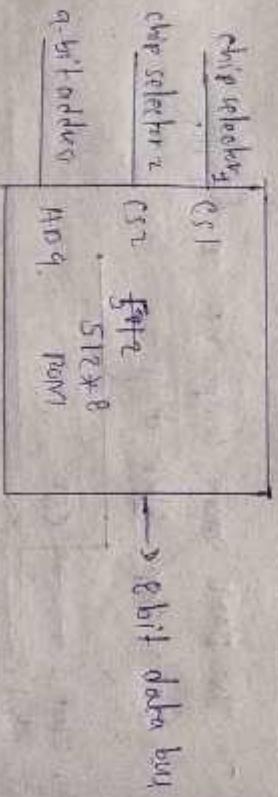
ROM Features

- A ROM chip is required for communication with CPU.
- A bidirectional data bus.
- The logic '0' and '1' signals standard digital signals.

② ROM : Read only memory is non-volatile memory, it keeps the data even if the power source is turned off.

ROM Integrated circuit chip

- ROM memory is used to store data and programs permanently.
- Block diagram of ROM chip



→ CS1 and CS2 pins must be '0' or '1'.

- ROM chip has similar architecture as RAM chip, slight difference is Here, RAM performs both read and write operations, while ROM performs read only.

Unit-5

Comparision of RISC and CISC

RISC	CISC
→ RISC stands for reduced instruction set computer.	→ CISC stands for complex instruction set computer.
→ It uses more registers.	→ It uses less registers.
→ Code size is more.	→ Code size is less.
→ RAM usage is more.	→ RAM usage is less.
→ less addressing modes.	→ More addressing modes.
→ pipelining is easy.	→ pipelining is difficult.
→ Simple and standard instructions.	→ complex and variable length instructions.
→ Emphasis on software.	→ Emphasis on hardware.
→ It consumes low power.	→ It consumes high power.
→ They are expensive.	→ They are relatively expensive.
→ It has fixed instruction format.	→ It has variable instruction format.
→ It doesn't external memory.	→ It requires external memory.

Interconnection structures

- There are different physical forms available for establishing an interconnection network.
- These physical forms are known as interconnection structures.
- There are five types of Interconnection structures

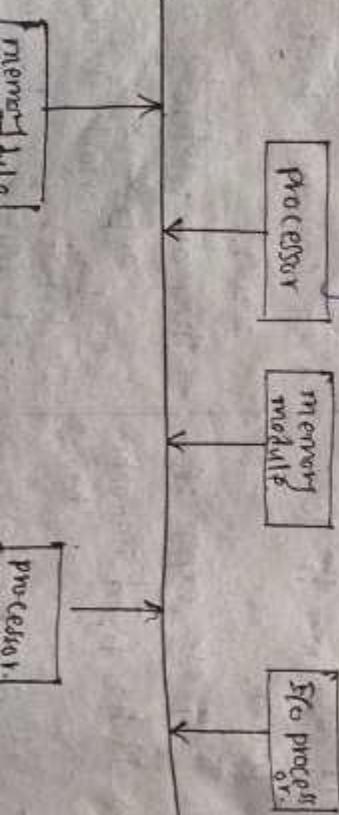
1. Time shared common bus.

2. Multipart memory.
3. Crossbar switch.
4. Multistage switching network.
5. Hypercube system.

1. Time shared common bus:

→ In any microprocessor system, the time shared common bus provides a common communication path by connecting all I/O processor, processor, memory units.

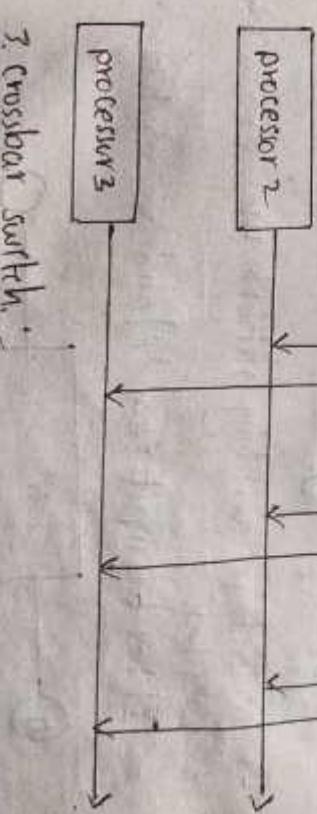
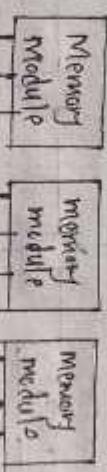
→ Below figure shows a single bus microprocessor system.



2. Multipart memory

- A multipart memory provides separate buses for every memory module.
- Every processor connected to each memory module.

→ The below figure shows multipart memory interconnection structure!

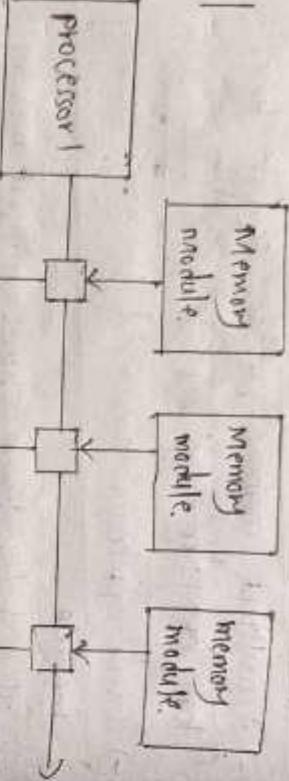


→ The crossbar system consist of interconnection cross point b/w processor and memory modules.

→ The below figure shows crossbar switch interconnection structure.

Characteristics of microprocessor

→ Microprocessor performance depends on following characteristics.



- clock speed
- instruction set
- work site

clock speed The speed at which microprocessor executes instruction is called the clock speed. in mega Hertz.

Instruction set The command which is given to the computer to perform an operation on data is called instruction set.

word size The no. of bits that can be processed by a processor in a single instruction is called word size.

Multi-stage switching Network:

- Multistage switching network uses 2x2 switch Crossbar.
- It has two inputs (A and B) and two outputs (C and D).



Hypercube system

- Hypercube system is a binary n-cube architecture.
- we can connect 2ⁿ processor

Interprocessor arbitration

→ The processor, main memory and I/O devices can be interconnected by means of a common bus.

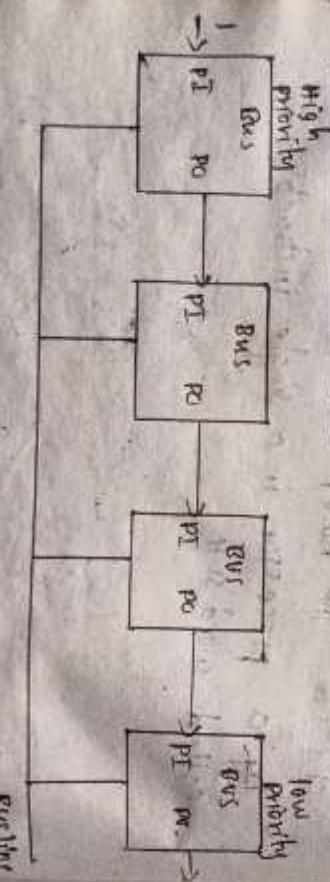
→ There are two bus arbitration method.

1. Static arbitration Technique

→ static arbitration technique.

→ static arbitration technique is of two type) they are

i. Serial arbitration



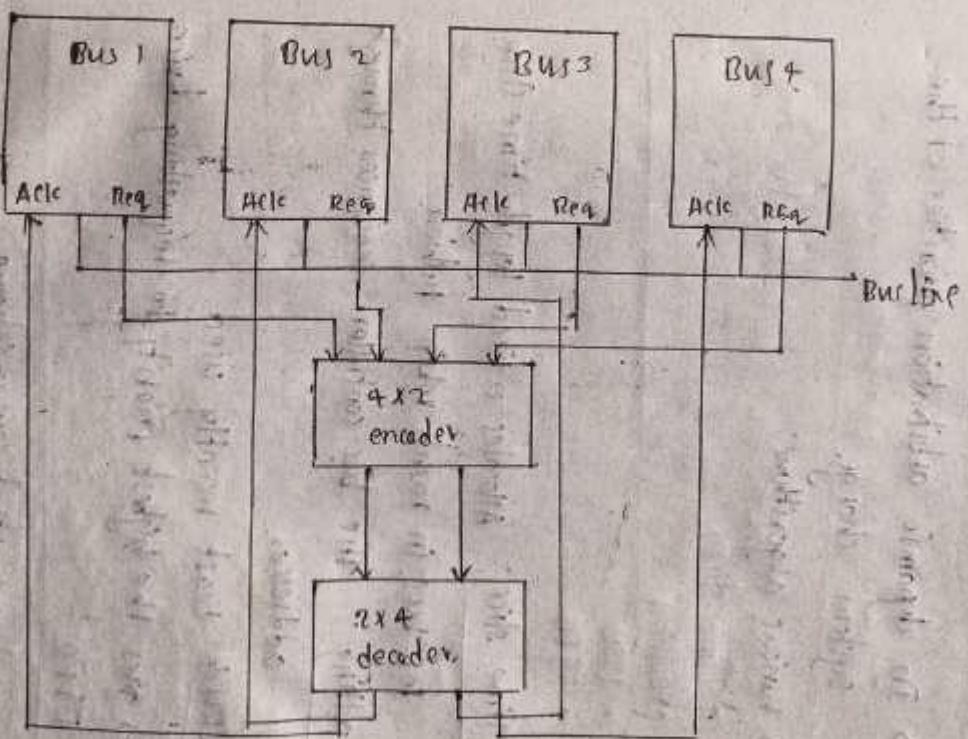
→ serial arbitration is also called as

serial chain arbitration.

→ It has 4 buses and they are connected to common bus line

→ first bus-high priority, last bus-low priority.

(2) Parallel Arbitration



→ parallel arbitration uses 4x1 encoder and 2x4 decoder.

→ each bus have acknowledge input and Request output

ii) Dynamic arbitration Technique :-

→ In dynamic arbitration priorities of the system change.

-s Various algorithms:

- Time slice

- polling

- LRU

- FIFO

1. Time slice: Allocates a fixed length time slice for bus, in round robin fashion.

2. polling: The bus controller sequences through addresses.

3. LRU: least recently used.

→ gives the highest priority to requesting device.

4. FIFO:

→ first come, first serve scheme.

Array processor

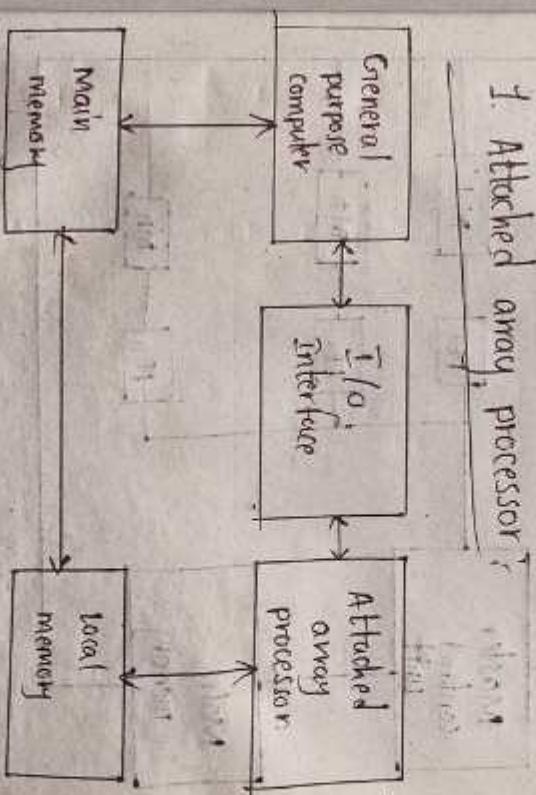
→ A processor that performs computations on array of data in called array processor.

→ Classification of array processors

1. Attached array processor

2. SIMD array processor.

1. Attached array processor



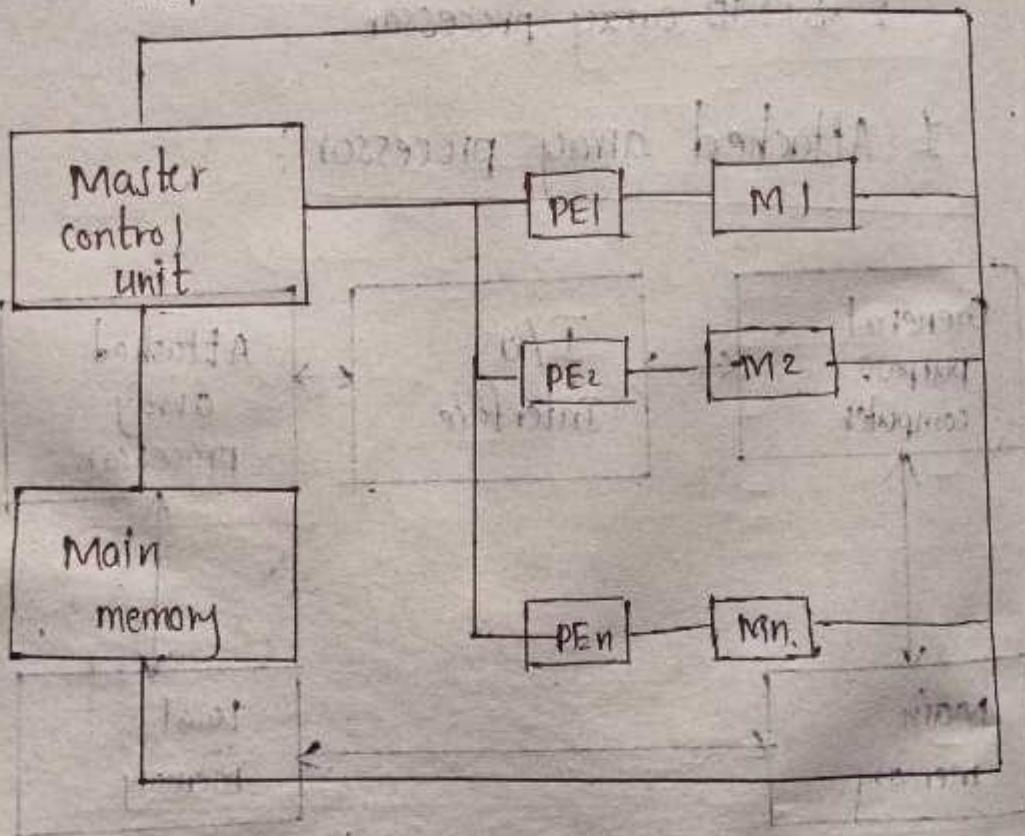
→ The attached array processor is the auxiliary processor.

→ The attached array processor consist of general purpose computer, I/O interface, attached array processor, main memory, local memory.

→ The main memory and local memory are linked.

2. SIMD Array processor

→ SIMD stands for single instruction and multiple data streams.



→ The block diagram shows array processor organisation in SIMD.

→ Applications

1. Radar systems
2. Sonar systems
3. wireless communication
4. Medical applications.