04/02/25

COA

3 components are there → 1) Software
2) Hardware
3) Humans

* Hardware
a) Cpu :- what is cpu, parts of cpu (CU, ALU)
b) Inputs & outputs

* Software :-
a) OS, ...

* Computers :- → Analog (continuous signal
→ Digital

4 parts of computer →
1) CPU (Input)
2) Keyboard (")
3) Mouse (")
4) Monitor (output)

with all devivers there is a deviver then there is os then there is application software then humans

Computer slow → Context switching.
CPU can do one work at a time

* Operating Systems :-
1) Batch processing (before 90's)
2) Multiprogramming OS.
3) Time Sharing.
4) Multithreading.
5) Real time systems (predefined deadline fixed like airbags)
6) Distributive OS (90's) → One machine is available in world that can be accessed
→ Telecommunication (Fb.,

1) Vacuum tubes → they used to build computers (1940-50)
2) Transistor
3) Ic .... VLSI.

Poweron self test → sound where comp. on

* Memory :- that stores
RAM [primary]
ROM (store in ROM
Bootstrap nodal that is the reason of loading os.

CPU is connected to RAM but RAM is not connected to harddisk

07/02/25

1) Architecture :- Computer architecture is the branch of study that deals with the conceptual design & basic overview of computer system.

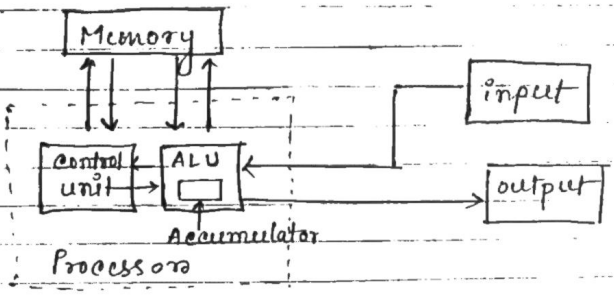
~~Organization~~ :- Organization :-

The computer organization is a lower level more concrete description of system that involves how the constituent parts of system are interconnected & how they interoperate in order to realise the architectural specifications.

2) → It refers to the parameter of computer system that are visible to the user.
→ The organizational issues are transparent to the user which is invisible to user.

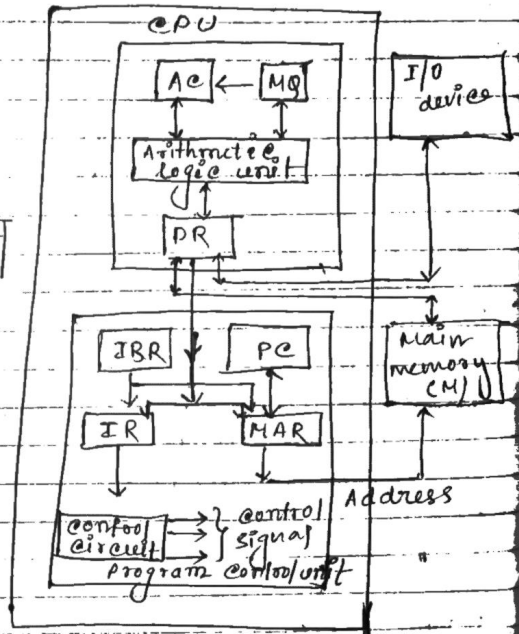
3) → It includes the instruction addressing mode, instruction formats, instruction sets, i/o mechanism etc.
→ The organisational aspects include memory technology, interfacing b/w CPU & peripherals, the system interconnect components like bus etc.

Von-Neumann Computer :-



Von-neumann Basic Architecture

IAS → Institute of advance study (1946 ~~Princton~~ University). Princeton



Structure of an IBS computer.

Stored program :- Program & data will store in same memory.

Hard ward architecture :- Program & data will store in different memory.

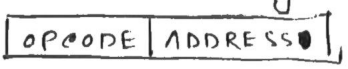
Von neumann concept :-

- The vonn neumann architecture proposed the stored-program concept which states that
  i) Program & data can be stored in the same memory.
  ii) The computer executes the program in sequence as directed by the instructions in the program.
  iii) A program can modify itself when the computer executes the program.

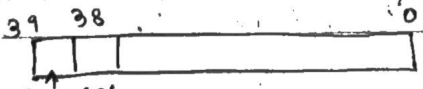
- Components of Von-Neumann Computer :-

1) Memory :- i) A random access cathod-ray tube (CRT) main memory is used which permitted an entired word to be accessed in one operation.

ii) Each instructions contain only one memory address & had the format like



iii) The central processing unit contain several high-speed vacuum tube ~~resistors~~ registers.
iv) The basic unit of infor" is 40-bit word.
v) The memory has a capacity of 2^{12} (4096 words). so the size is 4096 × 40 bits IAS
vi) Data :- The basic data item of an computer 11/02/25 is a 40 bit word with the left most bit represent the sign bit (0 - positive, 1 - negative) while the remaining 39 bits represent the magnitude of a number. The numbers are represented as fixed point numbers



vii) The IAS instructions are 20 bit long so two instructions can be stored in each memory location.

- Advantages of Von-newmann Architecture :-

1) Uses a single memory for data & instructions thus reducing hardware complexity.
2) flexible - can run various programs suitable for general purpose computing.
3) Ease of programming.

• Von-neumann bottleneck :-

One of the major factors for contributing to a computer performance is the time required to move instructions & data b/w CPU & the main memory.

The CPU has to wait longer to obtain a data word from the memory than from its registers because registers are very fast & are logically placed inside the CPU. This CPU-memory speed disparity is referred to as the von-neumann bottleneck.

This performance problem can be improved by →

1) Using a special memory called the cache memory b/w the CPU & the main memory.

2) Using a special type of computer known as Reduced Instruction Set Computers. (RISC).

• Registers :-

→ CPU Register :- CPU Registers are high speed memory units which are essential for efficient program execution & enable quick access to frequently used values.

Types of CPU Registers :-

i) General purpose Registers ii) Special Purpose Registers

i) General Purpose :- These registers work in coordination with the CPU's memory to optimize processing.
They are varied from 8 to 32 bits numbers.

ii) Special Purpose :- These registers have a specific purpose

a) Accumulator (ACC/A) :- This is the most frequently used registers used to store data taken from memory. It acts as the source of an operation & the destination of the operands.

b) Memory Address Register (MAR) :- It holds the address of the location to be accessed from memory.

c) Memory Data Register (MDR) :- It contains data to be retained into or to be read out from the addressed location.
MAR & MDR together facilitate the communication of the CPU & the main memory.

d) Program Counter (PC) :- It is used to keep the track of the execution of the program. It contains the address of the next instructions to be fetched from the main memory when the previous instructions has been successfully completed.

e) Instruction Register (IR) :- It holds the instructions which is just about to be executed. The instructions from the PC are fetched and stored in the instruction Registers.

f) Stack Pointer (SP) :- It points to the top of the stack which is a part of the memory used to store func calls & other operations.

g) Flag Registers / Status Registers / condition code Register :- It is a special purpose Registers used to indicate the status of the CPU or the outcomes of various operations.

h) Instruction buffer Registers (IBR) :- IBR is used to hold temporarily the right handed instructions from a word in memory.

i) Multiplier-Quotient (MQ) :- It is used in ALU to store the result of multiplication & division operations.

18/02/25

• Instruction formats :- | opcode | operand | mode |

The instruction format refers to the way the instructions are encoded and represented in machine language format. Instructions in a computer are divided into groups called fields. The commonly used fields are :- i) Opcode (Operation code) → This field states the operation to be performed. This field defines various processor operations such as add, subtract, multiply etc.

ii) Operand :- An operand can be a value or an address specifying a memory location or a processor register.

iii) Mode :- This field specifies the method to get the operand or the effective address of the operand.

• CPU organization :-
There are 3 ways in which CPU can be organized.

1. Single Accumulator Based CPU Organization :-
→ In this type of organization computers have accumulator based CPU.
→ It is the simple CPU in which the accumulator register is used implicitly for processing all instructions of a program and the intermediate results are stored in this register.
→ The instructions format uses in this computer uses one address field. That is why the CPU is known as one address machine. For e.g. → the instruction MUL X
AC ← AC * M[X]
X is an operand location in memory.
→ E.g PDP-8 processor
→ Advantages → i) One of the operand is always held by the accumulator register. This results in short instruction & less memory space.
ii) Instruction cycle takes less time because it saves time in instruction fetching from memory.
→ Disadvantages :-
i) When complex expression are computed program size is increased due to the uses of many short instructions.
ii) As the no. of instructions increases for a program, the execution time increases.

21/02/25

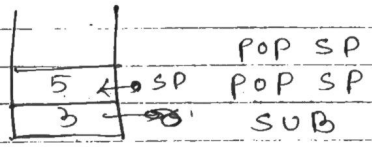
2. General Register based CPU organization :-
→ Instead of using a single accumulator register multiple CPU registers are used in this type of CPU organization.
→ This type of computer uses 2 or 3 address fields in their instruction format. either
→ Each address field can specify a general register or a memory word.
for e.g → MUL R1, R2, R3 | MUL R1, R2
R1 ← R2 * R3 | R1 ← R1 * R2
↓ | ↓
Three address | Two address

Examples → IBM-360, PDP-11
→ Advantages →
• since a large no. of registers are used in this organization

the efficiency of CPU increases.
• Less memory space is used to store the program since the instructions are used in more compact way.

→ Disadvantages →
• This organization is more costly since the large no. of registers are used
• The compilers need to be appropiately designed to avoid the unnecessary use of registers.

3. Stack based CPU organization :-
→ In stack based computers operated instructions based on a data structure called stack. A portion of the memory used to store operands in successive locations can be considered as a stack in computers.
→ The 2 operations performed on the operands are 'PUSH' & 'POP'.
→ The registers that holds the address for the topmost operand in the stack is called a stack pointer (SP).
→ The operation type instructions donot need an address field in stack based computers. for e.g.



→ Examples → Intel 8085
→ Advantages →
• Efficient computation of complex arithmetic expression
• Execution of instructions is fast because operand data are stored in consequel consequtive memory location.
• since instructions donot have an address field the length of the instructions is short.

→ Disadvantages →
• Program size increases.

• Instruction Length :-
The length of an instruction depends on the number of address fields used in it.

Ques 8) what are the advantages of using the different no. of addresses in the instructions?
→ i) The fewer the addresses the shorter is the length of the instructions. long instructions with multiple addresses usually required complex decoding circuits.

ii) Limiting the no. of addresses also limits the range of funs each instruction can perform.
iii) Long programs require longer execution time.

Q) What are the factors that affects the length of an instructions?

→ i) Memory size.
ii) Memory organization.
iii) Bus structure
iv) CPU complexity.
v) CPU speed.

LOAD → Memory to CPU
STORE → CPU to Memory

$X = (A+B) - (C+D)$

3-address instruction format

ADD R1, A, B // R1 ← M[A] + M[B]
ADD R2, C, D // R2 ← M[C] + M[D]
SUB X, R1, R2 ← M[X] ← R1 − R2

Advantages :-
i) It generates shorter programs

Disadvantages :-
i) It generates longer instructions

2- Address

~~ADD~~ LOAD R1, A // R1 ← M[A]
ADD R1, B // R1 ← R1 + M[B]
LOAD R2, C // R2 ← M[C]
ADD R2, D // R2 ← R2 + M[D]
SUB R1, R2 // R1 ← R1 − R2
STORE X, R // M[X] ← R1

Longer programs & shorter instructions.
Advantages :- less complex decoding circuits compare to three address instructions
Disadvantages :- less flexibility

1- address

LOAD C // AC ← M[C]
ADD D // AC ← AC + M[D]
STORE T // M[T] ← AC
LOAD A // AC ← M[A]
ADD B // AC ← AC + M[B]
SUB T // AC ← AC − M[T]
STORE X // M[X] ← AC.

Adv & disadv same as single accumulator

H.W $X = (A + B * C) / (D - E)$ using 0, 1, 2 & 3 address instructions

0 - Address

PUSH A // TOS ← A
PUSH B // TOS ← B
ADD // TOS ← B + A
PUSH C // TOS ← C
PUSH D // TOS ← D
ADD // TOS ← D + C
SUB // TOS ← (A+B) − (C+D)
POP T // M[T] ← TOS



Adv & disadv. same as stack based

• Addressing Modes :- The different ways in which the location of an operand is specified in an instruction is referred to as addressing modes.

Advantages of addressing modes :-
i) It gives programming versatility on flexibility to the programmers w.r.t the no. of instructions & execution time.
ii) To reduce the length of the instructions on the size of the program.

25/02/25

Case 1 :- No address field is specified as a part of instruction :-

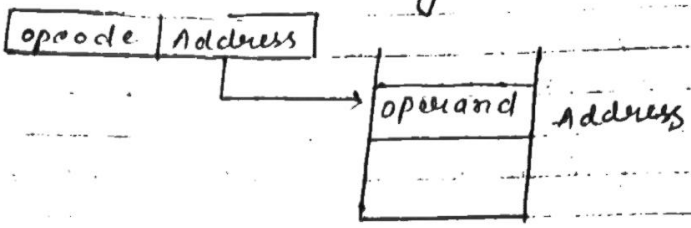
1. Implied (Inherent Addressing Mode) :
In this mode the operands are indicated implicitly by the instructions. The accumulator registers is generally used to hold the operand & storing the result after operation.
for eg. - The instruction CMA (complement accumulator)
" " RAL (Rotate Accumulator left)
↓ Rotates the contents of the accumulator to the left.
↓ complements the contents of the accumulator

2. Immediate Addressing Mode :-
In this mode the operande is mentioned explicitly in the instruction. An immediate mode instruction contains an operand rather than the address of the operand.
for eg. - MVI A, 06H → Load 06 into A register.
(Move immediate)
ADI 05H → AC ← AC + 05H
(Add immediate)
This mode is very useful for initialising a register to a constant value

3. Stack Addressing Mode :-
The stack organized computers used this addressing mode. In this mode all the operands for an instructions are taken from the top of the stack. The instruction doesnot have any operand field
for e.g. → SUB

Case 2: Address field is a memory address:-

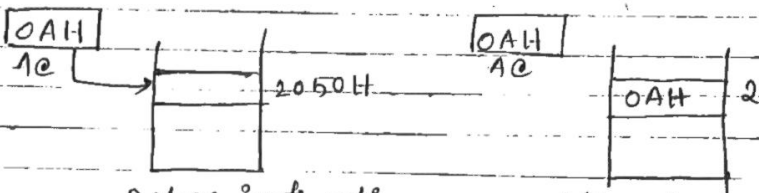
1. Direct (Absolute) Addressing Mode:-



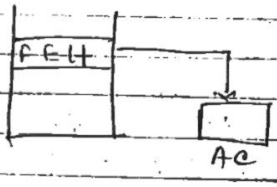
In this mode the instruction contains the memory address of the operand explicitly. Thus the address field of the instruction is the effective address.

↳ The address where the operand is located is known as effective address.

E.g → 1) STA 2050 H // Store accumulator in the (Store Accumulator Direct) address 2050



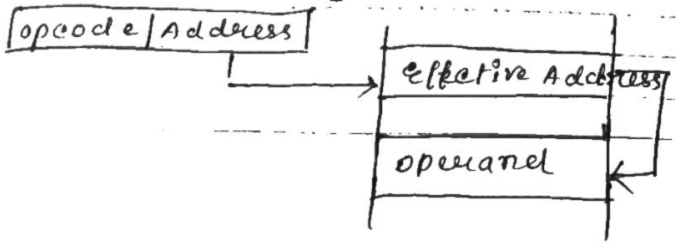
2) LDA 2050 H → 2050
(Load Accumulator Direct)



* Advantages:-
i) Since the operand address is directly available in the instruction there is no need for the effective address calculation step.
ii) The instruction cycle time is reduced.

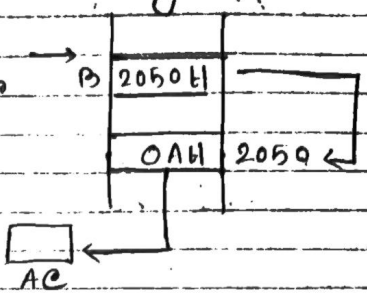
* All branch type instructions uses direct addressing mode.

2. Indirect Addressing Mode:-



In this mode the instructions gives a memory address in its address field which holds the address of the operand. Thus the address field of the instruction gives the address where the effective address is located in memory.

for eg → LDAX B
(Load Accumulator Indirect)



Advantages :-

i) There is a scope of changing the address during run time of the program without changing the instruction content. This type of addressing modes gives flexibility in programming.

ii) It is used for pointers in C language.

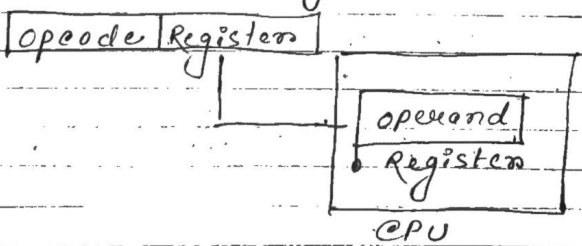
Disadvantages:-

The instruction cycle increases as there are two memory addresses

27/02/25

Case 3 :- Address field is a CPU Register :-

1. Register (Direct) Addressing mode :-



In this mode the processor registers holds the operands. The address field is a register field which is selected using a K-bit field where CPU contains 2^K registers.

E.g ADD R_1, R_2 // $R_1 \leftarrow R_1 + R_2$

• Advantages:-

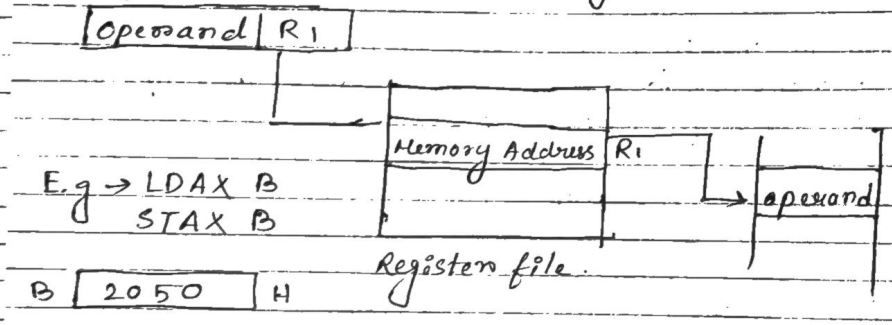
i) We are saving time & fast execution time since register accessing is much faster than memory accessing.

• Disadvantage:-

→ Limited no. of Registers available in the CPU.

ii) This mode is useful for storing intermediate results in the registers rather than in memory.

2. Register (Indirect) Addressing Mode :-



In this mode the instructions specify an address of CPU register that holds the address field of the operand in memory.

• Advantages :-

i) This mode is very useful for rapid access of main memory such as an array.

ii) We can use a small number of bits in the address field of the instruction for accessing a memory location.

• Disadvantages :- Same as the indirect mode (case 2)

Case 4 : Address field does not contain effective Address :- Effective address is calculated using →

Effective Address :- Address part of instruction + Contents of a special CPU register.

1> Relative Addressing Mode (PC - relative) :- PC → Program Counter

Instruction

opcode	offset

operand address : Contents of PC + offset .

JUMP	50H

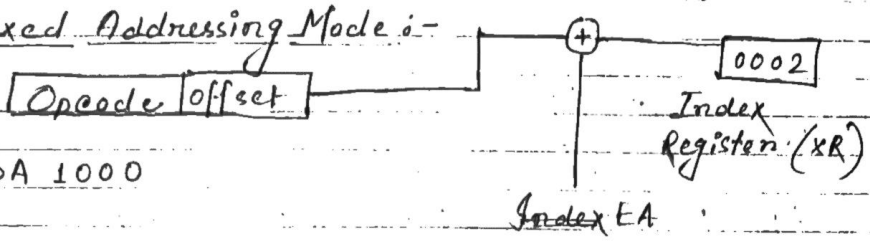
00011
PC

EA : 00011 + 50H : 0065H

In this mode the EA is obtained by adding the contents of the program counter register with the address part of the instruction.

• Advantage :-
This mode is used to specify the branch address of the branch instruction.

2> Indexed Addressing Mode :-

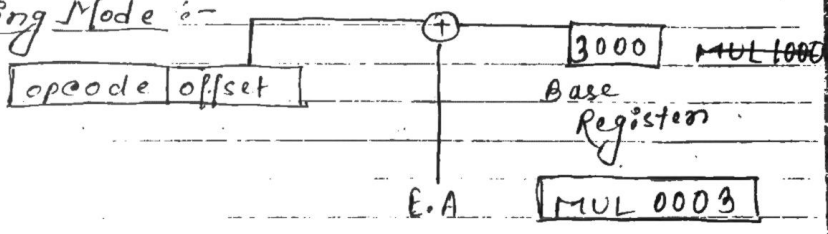


In this mode the EA is determined by adding the content of the Index Register with the address part of the instruction.

• Advantage :-
This mode is useful for accessing operand array.

* EA = Address part of instruction + Contents of a XR

3> Based Addressing Mode :-



EA ◆ = Address part of instruction + contents of a Base Register

This mode is used for allocation of the programs in memory. The content of the base register is added to the address part of the instruction. This mode is useful for skipping through various memorys.

28/02/25

- Instructions :- An instruction is a group of bits that instructs the computer to perform a specific operation.

- Instruction set :- An instruction set of a processor is a collection that defines all the instruction. A complete instruction set is expressed as Instruction set Architecture (ISA) of the processor. An instruction set of a processor must have the following characteristics:

1) Completeness :- we should be able to construct a machine language program to evaluate any function i.e. computable using a reasonable amount of memory space.

2) Efficiency :- Frequently required function can be performed rapidly using relatively few instructions.

3) Regularity :- The instruction set should contain expected opcode & addressing mode.

4) Compatibility :- to reduce hardware & software design cost the instructions may be required to be compatible with those of existing machine.

- Instruction types :- The instructions in an instruction set are classified into 5 types :-

1) Data transfer Instructions :- This instructions copy information from one location to another location either in the processor's internal registers or in the external main memory.
E.g :- (MOVE, LOAD, STORE, PUSH, POP) etc

2) Arithmetic instructions :- These instructions perform operation on numerical data.
E.g :- ADD, SUB, MULT, INCR, DIV, DECR.

3) Logical Instructions :- These instructions include boolean & other

non numeric operation.
E.g → AND, OR, XOR, NOT.

4) Program Control Instructions → These instructions change the sequence in which programs are executed.
E.g → JUMP, WAIT, JNC (Jump NOT Carry), JC (Jump On Carry).

5) Input Output operations :- These instructions cause information to be transferred b/w the processor or its main memory & external io device.
E.g → IN, OUT,
(Read) (Write)

RISC Vs CISC

RISC → (Reduced Instructions Set Computers)
CISC → (Complex Instructions Set Computers)

RISC	CISC
i) Lesser no. of instructions (less than 100).	i) Greater no. of instructions (Greater than 200)
ii) Single fixed length instructions	ii) Variable length instructions.
iii) Large Relatively few addressing modes (≤05)	iii) A large no. of addressing modes (>10)
iv) Mostly hardware control units.	iv) Mostly microprogrammed control units.
v) Memory access limited to LOAD & STORE operations	v) Most instructions manipulate operands in memory.
vi) All operations are executed within the registers of the CPU.	vi) A small no. of registers are used.

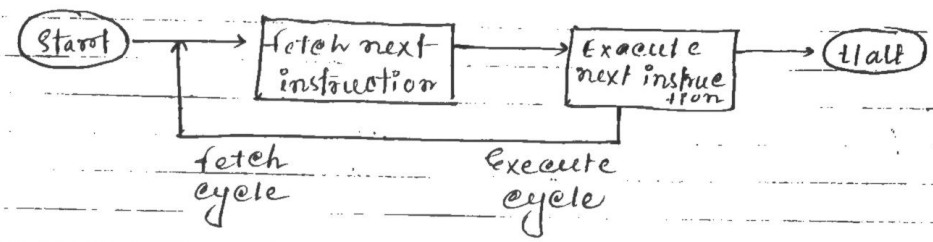
LDA 2500
Contents of PC = 2000
R_1 = 2300
XR = 20
AC = 2

* EA contents of PC + offset
= 2002 + 2500
= 4502 → for Relative.

Address	Memory
2000	LDA MODE
2001	Address = 2500
2002	Next instruction
2300	2450
2500	2700
2520	2800
2700	2250
4502	3400

Addressing Mode	Effective Address	Contents of AC
Immediate	2001	2500
Register	—	2300
Register-indirect	2300	2450
Direct	2500	2100
Indirect	2100	2250
Relative	4502	3400
Indexed	2620	2800

- Instruction Cycle :-

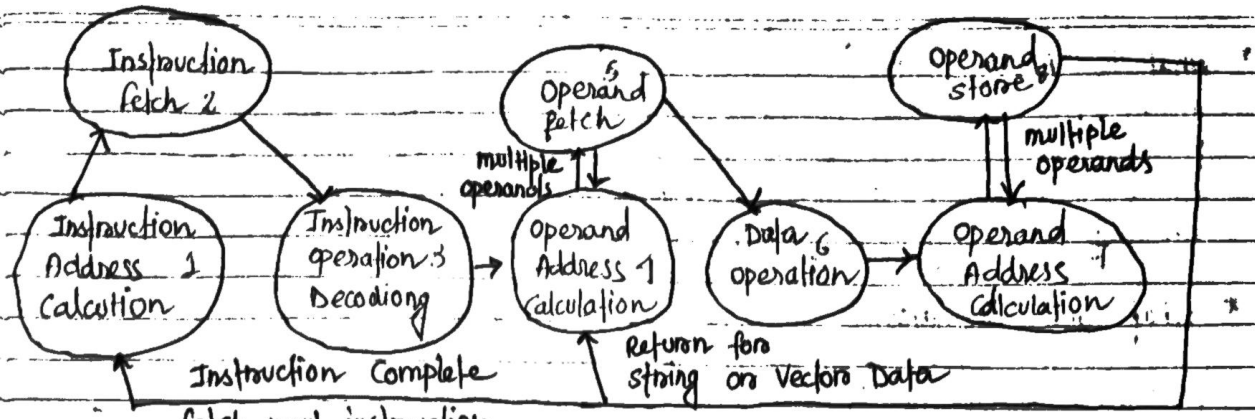


Basic Instruction Set

4/03/25

i) The instruction processing consists of 2 steps
The processor reads (fetches) instructions from memory one at a time and excutes each instructions from memory one at a. The program execution consists of repeating the process of instructions fetch and instruction execution in a typical processor a register called the program Counter holds the address of the instruction to be fetched next. The processor always implements the program Counter after each instruction phase so that it will fetch the next instruction in sequence

ii) The fetched instruction is loaded in a register in the processor known as the instruction register (IR). The instruction contains bits that specify the action the processor is to take. The processor interprets the instruction and performed the required action.



↑ Instruction cycle State Diagram

The states can be described as follows :-

Instruction Address Calculation :-
Determined the address of the next instruction to be executed Usually this involves adding a fixed number to the address of previous instruction.

Instruction fetch :-
Read instruction from its memory location into the processor.

Instruction operation Decoding :-
Analyse the instruction to determine type of operation to be performed and operands to be used.

operand address Calculation
If the operation involves reference to at operand in memory or available by I/O then determine the address of the operand.

operand fetch
fetch the operand from memory or read it from I/O.

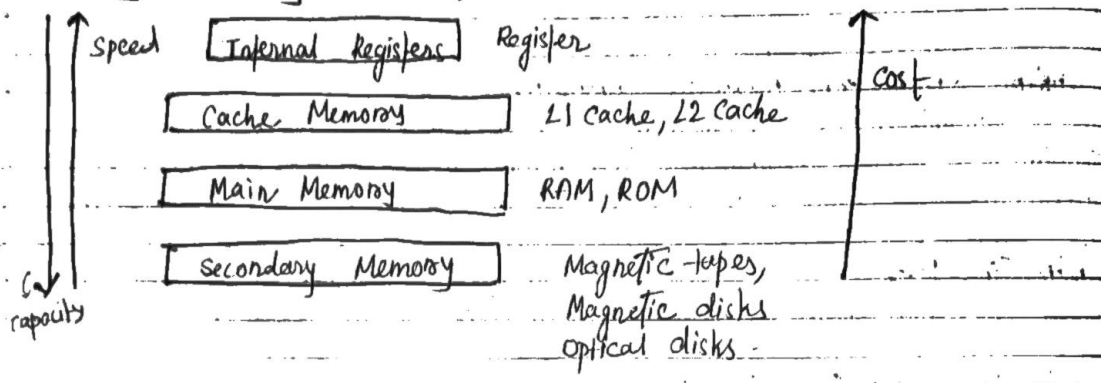
Data Operation :-
perform the operation at indicated in the instruction.

operand store :-
Write the result into memory or out to I/O.

* What is Machine cycle?
A machine cycle of consisting of necessary step carried out to perform memory access operation. each of the basic operations such as fetch or read or write, consits of a machine cycle.
An instruction cycle consists of several machine cycles.

* What are T-states?
One clock cycle of the system clock is defferred to as T-state. A machine cycle consists of several T-states.

Memory Hierachy



Secondary Memory
The slow speed low cost devices that provide back up storage are called Secondary Memory. Programs that are not currently needed by the main memory are transferred into secondary memory to provide space for currently used program and data.
ex:- Magnetic tapes, optical devices.

Main Memory:-
This is the memory that communicates directly with the CPU. Only programs and data currently needed by the CPU for execution reside in the main memory.

Cache Memory:-
This is a special high speed main memory which is used for increasing the speed of processing by making the current programs and data available to the CPU at a faster rate.

Internal Memory :-

This memory refers to high speed registers used inside the CPU. This register hold temprorary results when computation is in progress.

Access Method :-

i) Sequential Access :- In this method the memory is access in a specific linear sequential manner.
ex :- Magnetic Disk, Magnetic tapes, CD ROM

ii) Random Access :- In this method mode any location of the memory can be accessed randomly.
ex :- RAM, ROM

iii) Associative Access :- It is a special type of random access method. that performs comparison of desired word for a specific match for all the words in the memory simultaneously.

Memory Parameters :- There are three basic parameters for choosing a memory.

i) capacity

$2^m \times n$ (memory size × word size) m : address length ex : $2^{16} \times 8$

ii) Speed :-

Access time :- The access time is the time taken by the memory to complete a read operation from the moment of reciving the read control signals.

iii) Data Transfer Rate / Band Width :- The maximum amount of information that can be transferred to and from the memory for unit time is called the band width and is expressed as numbes of bytes or words per second.

11/03/25

Main Memory: - The main memory of a computer system is a central storage unit which is directly accessible by the CPU.

→ It is a large & fast external memory used to store programs & data during computer operations

→ RAM one of 2 types :- i) RAM :- The term as RAM memory consists of integrated circuit chips which are volatile i.e., (if the power goes off the stored information is lost). RAM is used to store the program & data that are being used by CPU in real time.

→ There are 2 types of RAM :- SRAM & DRAM.

→ There is another type of main memory called ROM (Read only Memory). Data stored in the ROM is written even after the computer is turned off i.e., non volatile.

→ The types of ROM are :- i) Programmable ROM
ii) Erasable Programmable ROM
iii) Electrically Erasable Programmable ROM.

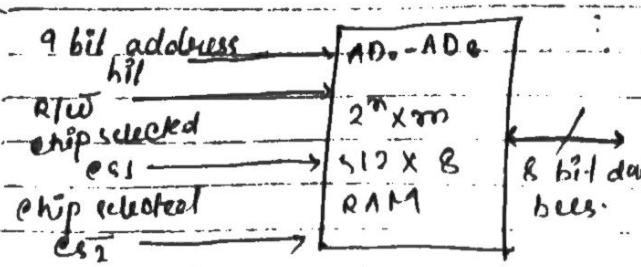
→ The ROM portion of the main memory is needed for storing an initial program called the Boot Strap Loader. The Boot strap loader is a program whose func" is to start the computer software apparently when the power is turned on. Since ROM is volatile its contents are destroyed when power is turned off. The contain of ROM remain unchanged after power is turned off & on again. Thus when the power is turned on the hardware of the computer sets the program counter to the first address of the Bool strap Loader. The Boot strap Loader loads a portion of the operating system from disk to main memory & control is then transferred to the operating system.

• Difference b/w RAM & ROM :

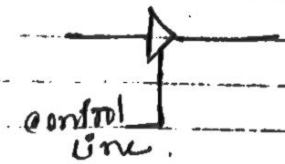
RAM	ROM
i) RAM is a volatile memory that could store a data as long as power is supplied	i) ROM is a non-volatile memory that could retain data even when the power is switched off.
ii) Read & write operations are supported.	ii) Only Read operation is supported.
iii) Used to store data that has to be currently processed by the CPU.	iii) It is used to store firmware which is used for initializing the components of the computer
iv) It is a high speed memory	iv) It is much slower than RAM.
v) CPU can easily access data stored in RAM.	v) CPU can't easily access data stored in ROM.

RAM chips

Z = logically disconnected from circuit

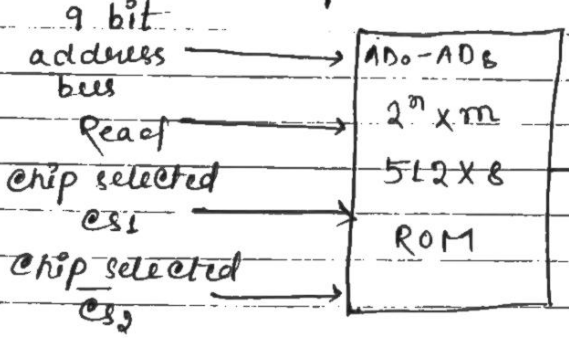


Tristel buffer → 0, 1, Z



CS1	$\overline{CS_2}$	R/W	Memory func"	State of the bus
0	0	X	Inhibit	High impedence
0	1	X	"	"
1	0	0	write	Input data to RAM
1	0	1	Read	output data from RAM
1	1	X	Inhibit	High impedence state.

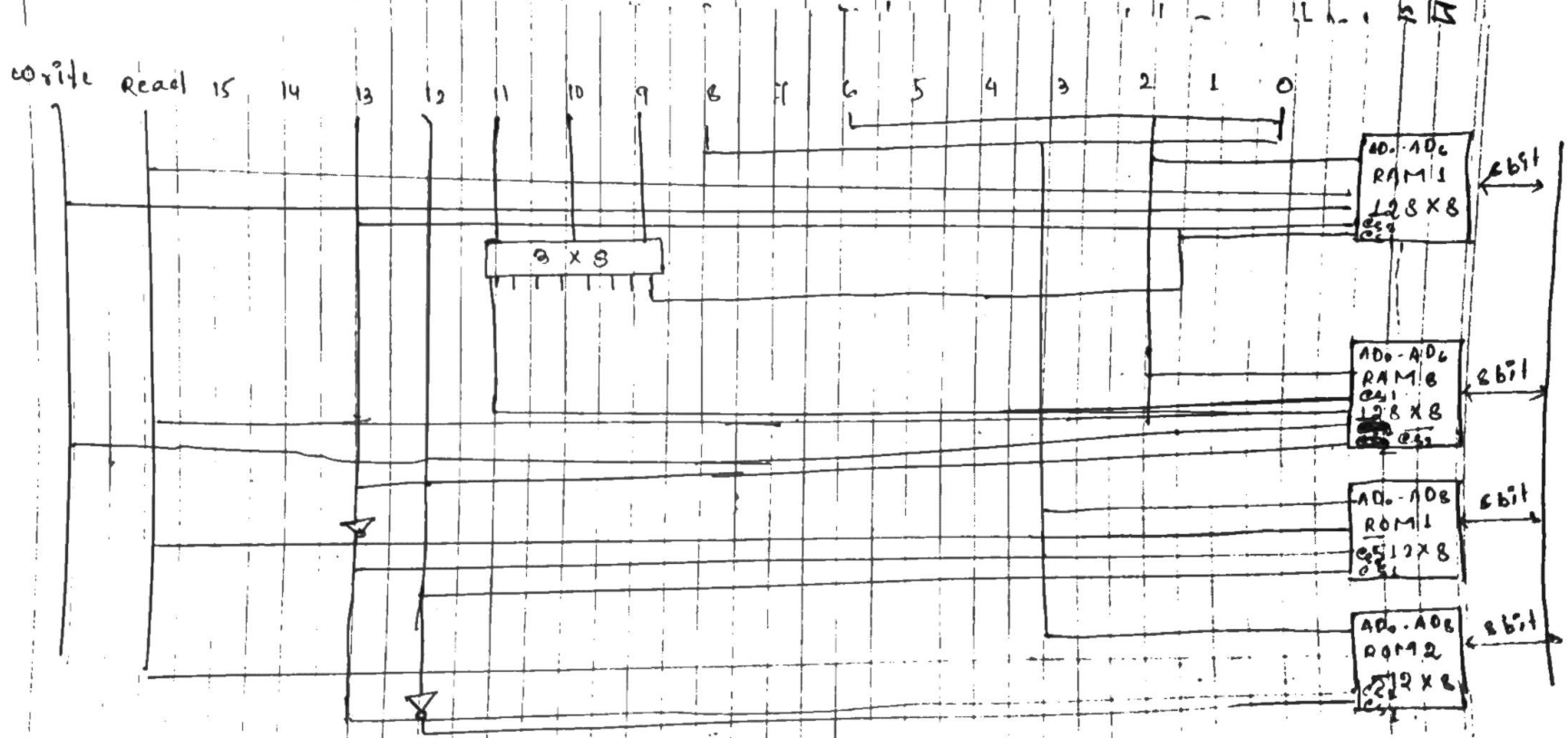
ROM chips



Q) Suppose we have 2 sets of memories with RAM of sizes 128 x 8 & ROM of sizes 512 x 8. we have to design a memory of capacity 1024 x 8, find the no. of chips & draw the diagram.

→ 1. No. of RAM chips = $\frac{1024 \times 8}{128 \times 8} = 8$

2. No. of ROM chips = $\frac{1024 \times 8}{512 \times 8} = 2$


Write
Read
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
3 x 8
AD0 - AD6
RAM 1
128 X 8
8 bit
AD0 - AD6
RAM 8
128 X 8
8 bit
AD0 - AD8
ROM 1
512 X 8
8 bit
AD0 - AD8
ROM 2
512 X 8
8 bit

$\overline{CS2}$ ↑ CS1 for ROM ↑ CS1 for RAM

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex Range
RAM1	0	0	0	X	0	0	0	X	X	0	0	0	0	0	0	0	0000H
RAM2	0	0	0	X	0	0	1	X	X	0	0	0	0	0	0	0	007FH
RAM3	0	0	0	X	0	1	0	X	X	0	0	0	0	0	0	0	0200H
RAM4	0	0	0	X	0	1	1	X	X	0	0	0	0	0	0	0	027FH
RAM5	0	0	0	X	1	0	0	X	X	0	0	0	0	0	0	0	
RAM6	0	0	0	X	1	0	1	X	X	0	0	0	0	0	0	0	
RAM7	0	0	0	X	1	1	0	X	X	0	0	0	0	0	0	0	
RAM8	0	0	0	X	1	1	1	X	X	0	0	0	0	0	0	0	
ROM1	0	0	1	1	X	X	X	0	0	0	0	0	0	0	0	0	
ROM2	0	0	1	0	X	X	X	0	0	0	0	0	0	0	0	0	

* ⇔ X → 0

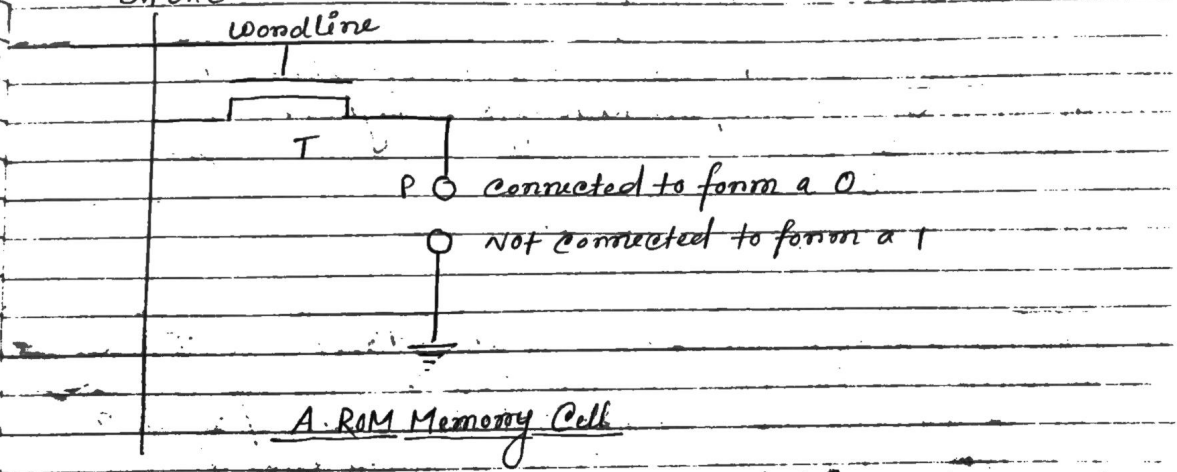
$\overline{CS2}$ CS1 for ROM CS1 for RAM

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Hex Range
RAM 1	0	0	0	x	0	0	0	x	x	0	0	0	0	0	0	0	0000-07FF
RAM 2	0	0	0	x	0	0	1	x	x	0	0	0	0	0	0	1	0800-0FFF
RAM 3	0	0	0	x	0	1	0	x	x	0	0	0	0	0	1	0	1000-17FF
RAM 4	0	0	0	x	0	1	1	x	x	0	0	0	0	0	1	1	1800-1FFF
RAM 5	0	0	0	x	1	0	0	x	x	0	0	0	0	1	0	0	2000-27FF
RAM 6	0	0	0	x	1	0	1	x	x	0	0	0	0	1	0	1	2800-2FFF
RAM 7	0	0	0	x	1	1	0	x	x	0	0	0	0	1	1	0	3000-37FF
RAM 8	0	0	0	x	1	1	1	x	x	0	0	0	0	1	1	1	3800-3FFF
ROM 1	0	0	1	1	x	x	x	0	0	0	0	0	1	0	0	0	4000-47FF
ROM 2	0	0	1	0	x	x	x	0	0	0	0	0	1	0	0	1	4800-4FFF

x ⟷ X → 0

22/04/25

ROM chips :-



A ROM Memory Cell

A rom is a main memory which is used to store permanent system programs and system data. A logic value 1 is stored in the cell if the transistor is not connected to the ground at point p, otherwise a binary 0 is stored.

In order to read the state of the cell the wordline is activated to close the transistor which acts as a switch. The voltage on the bit line drops to 0 if the point p is connected otherwise it retains the state of 1.

• Types of ROM :- 1) PROM (Programmable ROM)
2) EPROM (Erasable Programmable PROM)
3) EEPROM (Electrically Erasable PROM)
4) Flash Memory.

1) PROM :- Some ROM designs allow data to be loaded into the cell by the user. This type of ROM is known as PROM. A fuse is inserted at point P. Before it is programmed the memory contains all zeros. The user can insert 1's at the required location by burning the fuses with high voltage current.

2) EPROM :- An erasable programmable ROM uses a transistor in each cell that acts as a programmable switch. The contains of a EPROM can be erased by burning the device under an ultra violet light.

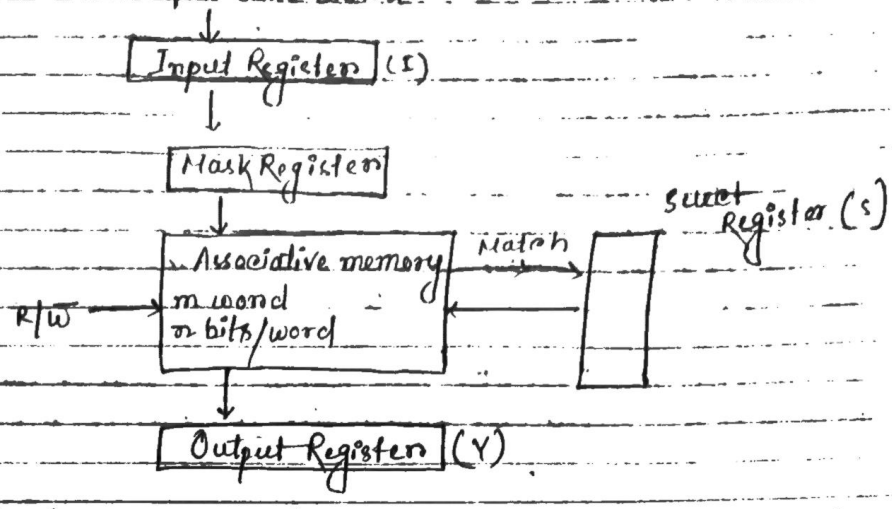
3) EEPROM :- EEPROM has the advantage that the contains can be selectively erased by writing 1's & each bit of the information can be stored again.

4) Flash Memory :- A flash memory is a type of EEPROM in which the density of data storage is higher. USB are flash memory.

• Difference between SRAM & DRAM.

SRAM	DRAM
1) It has lower access time i.e., it is faster than DRAM.	1) The access time of DRAM is less than SRAM.
2) SRAM is made up of transistors.	2) DRAM is made up of capacitor.
3) It is very costly.	3) It is much cheaper.
4) SRAM requires constant power supply.	4) DRAM offers reduced power consumption because info" is stored in capacitors.
5) SRM retains the stored info" as long as the power is supplied.	5) DRAM tends to loose info" over a period of time thus the capacitor needs to be periodically recharged.

• Associative Theory :-



The searching time for desired data stored in the memory can be reduced largely if stored data can be searched only by the data value itself (rather than by an address. The memory access by data content is known as associative memory or content Addressable Memory (CAM).

• When a data is stored in the memory no address is given. At any first location the data is stored. When the data is read from the memory only the data word or part of the data is specified.

• Advantages :-

1. Searching in this memory is very fast due to its organisation.

• Disadvantage:

→ It is more expensive than a RAM memory because each chip should seperate logic circuit.

• Organication of Associative Memory :-

→ The associative memory consists of a memory array and a logic for 'm' words with 'n' bits/word.

1. Input Register (I) :- The Input register is used to hold the data to be written into or searched from the associative Memory.

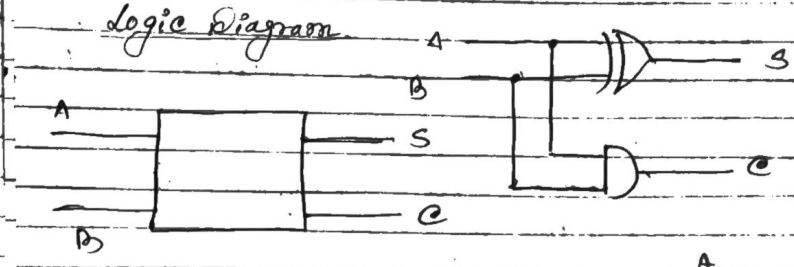
2) Mask Register :- The Mask Register provides a mask for choosing a particular field or key in the input register word. Only those bits in the input register that have one in their corresponding position of the mask reg. are compared.

3) Select Register (S) :- If match is found after comparing input data in I register with the key field with in the M register the corresponding bits in the select register is set.

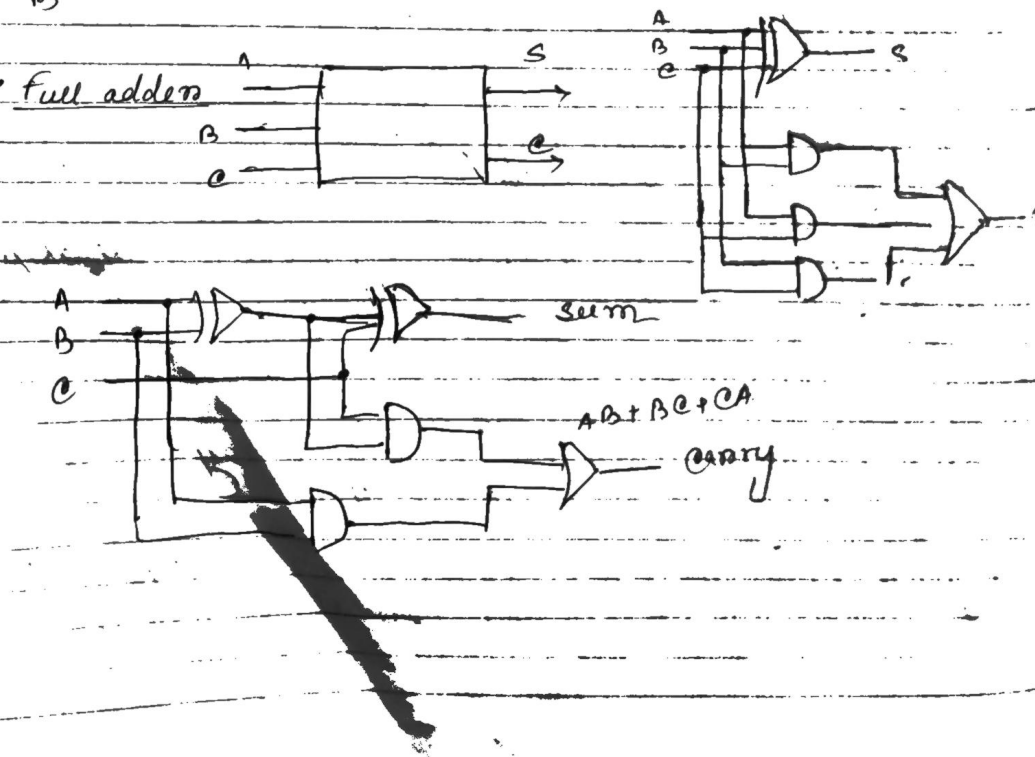
4) Output Register (Y) :- This register contains the match data word retrieved from the associative memory.

25/04/25

• Circuit Diagram of Half Adder :-



• Full adder



23/04/25

• Cache Memory :-

Cache Memory works on the locality of Reference.
Locality of Reference is of 2 types → 1) Spatial 2) Temporal.

1) Spatial :- There is a high chance that whole block will be executed.

2) Temporal :- the tendency of a program to access the same memory location repeatedly.

* Cache Memory is placed in b/w CPU & main Memory.

location :

• Cache hit :- Data is in the cache Memory.

• Cache Miss :- Data is not in cache memory, found in main memory.

Hit ratio = $\frac{\text{Number of hits}}{\text{total CPU refrences to memory}}$

= $\frac{\text{Number of hits}}{\text{no. of hits + number of misses}}$

$0 \leq h \leq 1$

t_c = Cache access time
h = hit ratio of cache
t_m = main memory access time

$t_{av} = h * t_c + (1-h)(t_c + t_m)$
$= t_c + (1-h) t_m$

Q) A hierarchical cache - main memory sub-system has the following specification →
1) Cache access time is 50 Nanosec.
2) Main memory accesstime is 500 Nanosec.
3) 80% of the memory request are for read
4) Hit ratio of 0.9 for read access & write-through scheme is used.
Calculate the following :-
i) Average access time of the memory system considering only memory read cycle.
ii) Average access time of the system for both read & write request.

Ans → Given, t_c = 50 nsec
t_m = 500 nsec
Probability of read (P_R) = 0.8
Hit ratio for read access = 0.9
write through scheme is used.

i) Considering memory read cycle,

Avg access time = $t_{av} = h * t_c + (1-h)(t_c + t_m)$

$= 1,00$ nsec.

ii) Considering both read & write

$P * t_{av} + (1-P) * t_m$ [since it is write through it will access main memory]

$= 0.8 \times 100 + (1-0.8) * 500$ nsec

$= 80 + 0.2 \times 500$

$= 80 + 100 = 180$ nsec.