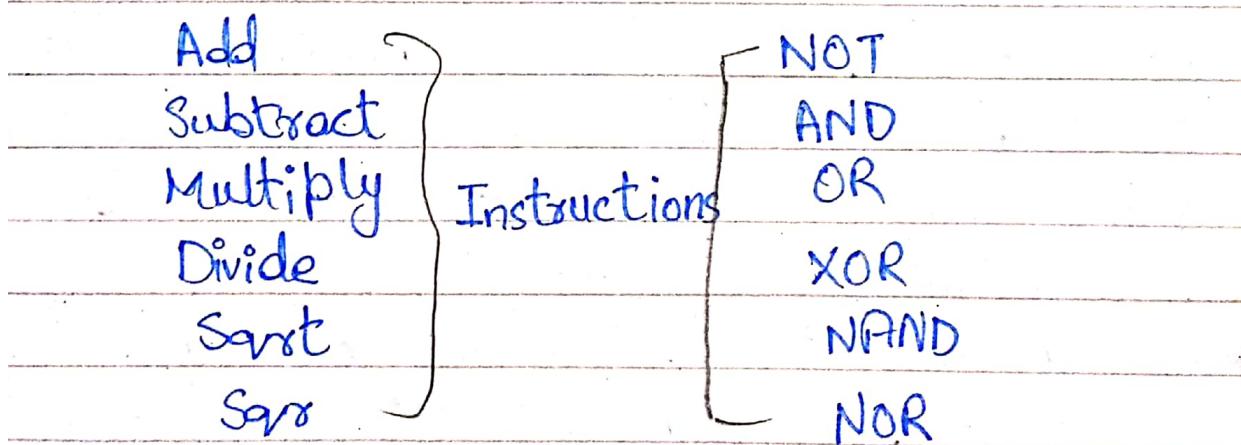
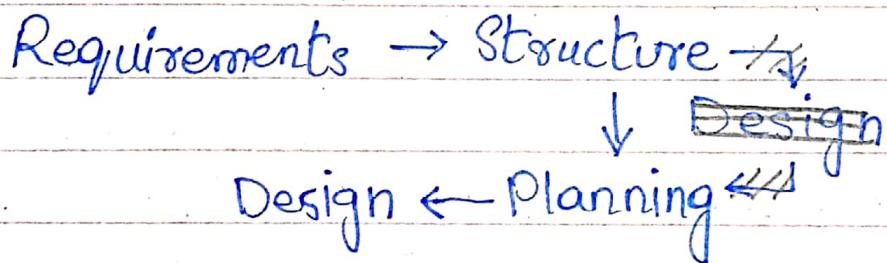


CS513- Computer Architecture

Computer: A device which performs Logical and arithmetic operations.

Architecture: Designing and planning of any model or structure by studying the requirements of a client.



In constructing any of the above structure instructions we have to ask about the inputs (numbers required for add operation) and its data type.

- * Computer Architecture affects the programmer
- * Computer organization affects the designer.

Architecture

- * Computer architecture refers to those attributes of a system which are variable for programmer
- * Those attributes that have a direct impact on the logical system or execution of the program, for example: **instruction set**, no. of bits used to represent various data types, I/O mechanism.
- * It is an architectural issue whether an instruction is available or not (for e.g. Add instruct)
- * An architecture may survive for long.

Organization

computer organization refers to their inter-connections that realizes the architectural specifications.

The organization attributes are related to hardware details or features that are **transparent** to the programmer. For example: **control signals, connection b/w peripherals and system memory technology used etc.**

It is an organization issue that how an instruction is implemented (adder instruction uses same special circuit)

An organization may change more frequently.

Evolution:

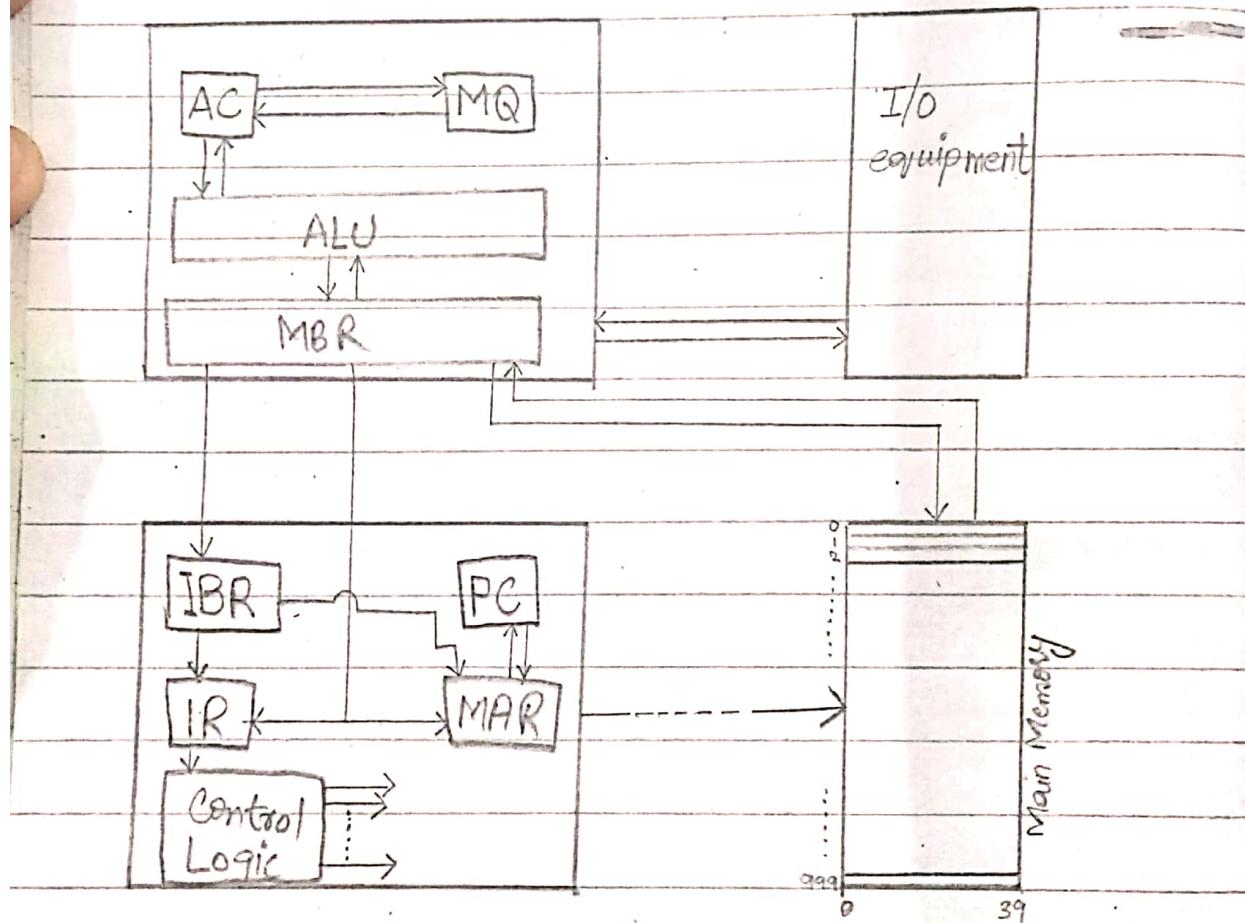
- * Decrease in component size
- * Memory capacity increases
- * Decrease in power consumption
- * Increase in processor performance
- * Increase in I/O capacity and performance.

ARM — Advance Risk Machine
Architecture

IAS (Von-Neuman Architecture)

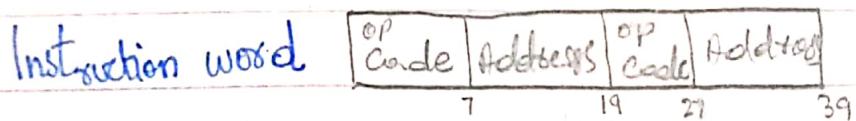
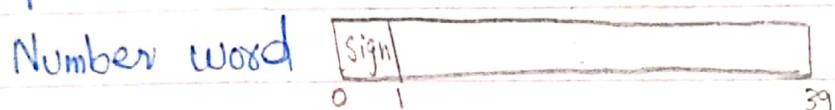
1945 - 1952

- * Concept of memory introduced
- * Instructions and data should be save in a single memory.
- * To access the memory we have to call the address of a cell in a memory.
- * Executions of instructions runs sequentially
- * Every instruction also holds the address of the next instruction to run.



- * This architecture has 21 instructions divided into 5 groups:
 - Conditional branching instructions
 - Unconditional branching instructions
 - Arithmetic instruction group
 - Data transfer instructions group
 - Address modify instructions group

- * Number word and instruction word is comprise of 40 bit (0 - 39)



- * All computers in today's world have Von-Neumann Architecture (IAS) which means the following concept :

- Data and instruction are stored in a single read and write memory.
- The contents of this memory are addressable by location without regard to the type of data contain there.
- Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next.

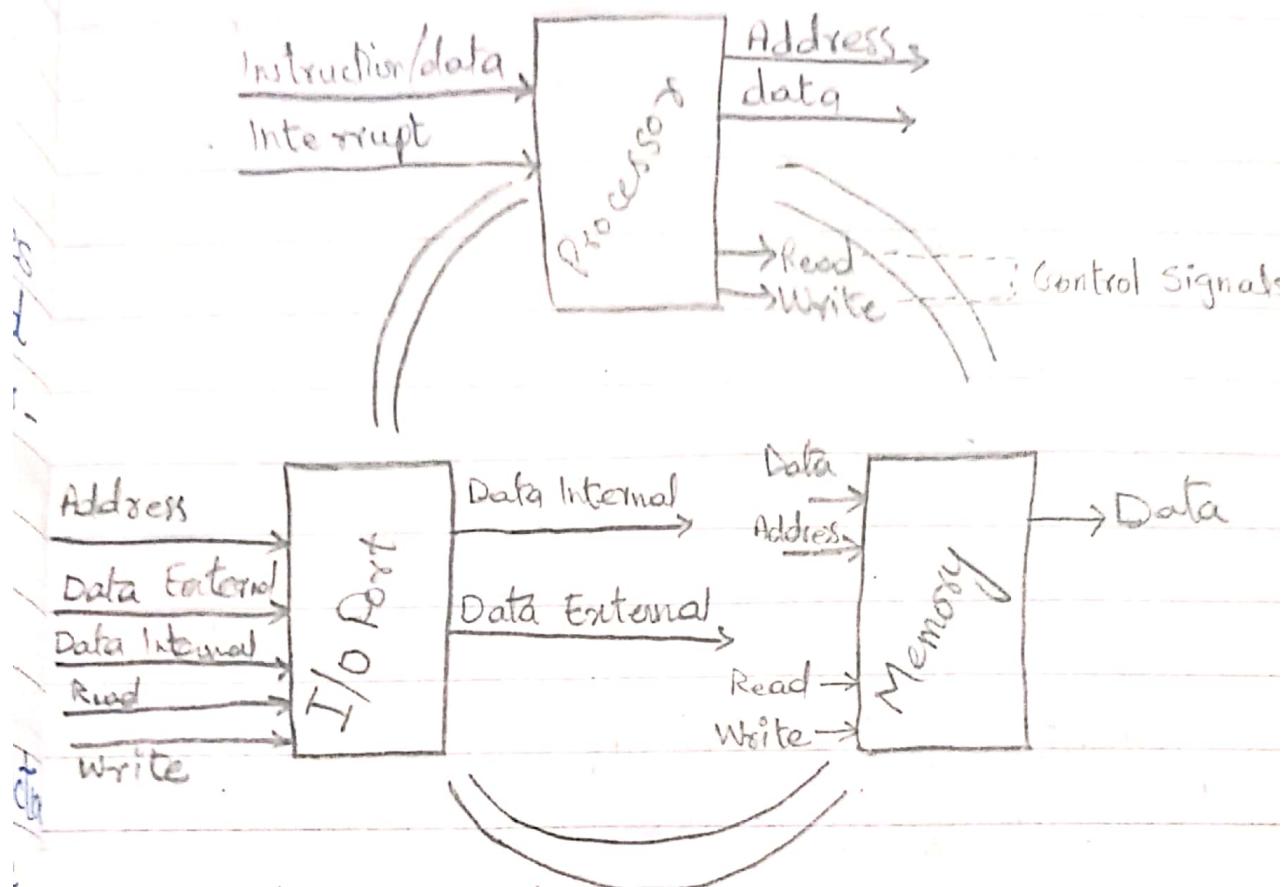
An IAS architecture includes :

- * A main memory which stores both the data and instructions ⁱⁿ same form. No separate area is reserved for the two. This resulted in many advantages one of them being that machine could treat instructions as data and accordingly could modify it just as it did for the data.
- * An arithmetic and logic unit capable of operating on binary data.
- * A control unit which interprets the instruction in memory and causes them to execute.
- * Input and output equipment operated by control unit.



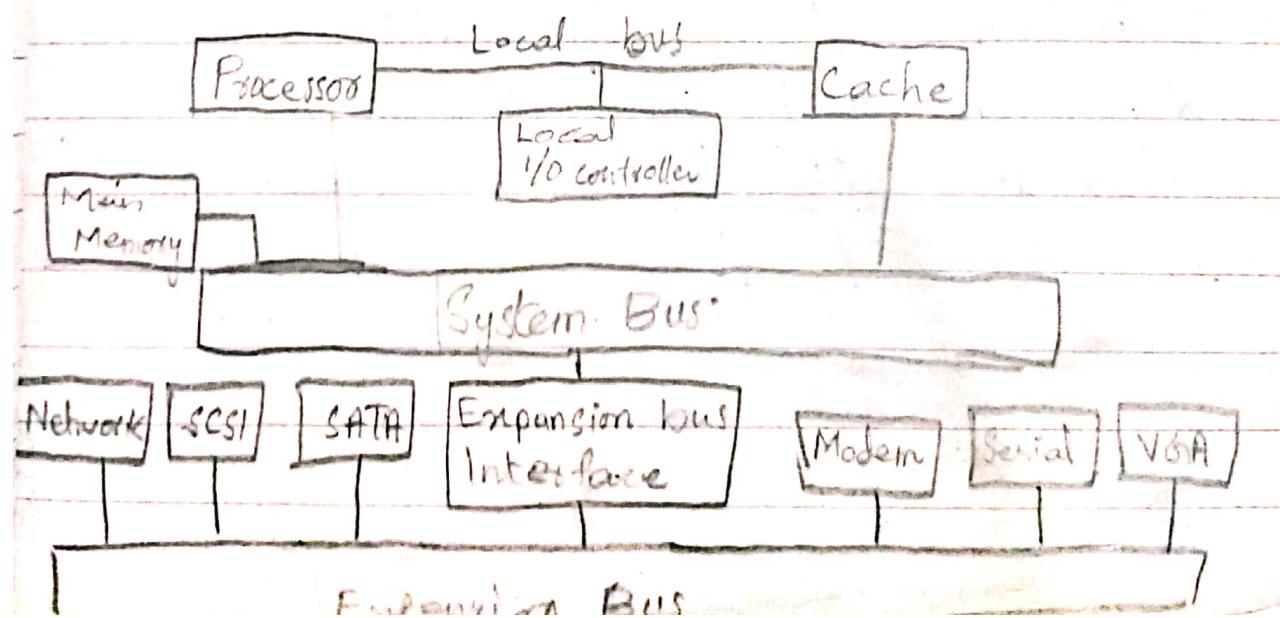
35
15- Sept- 2021

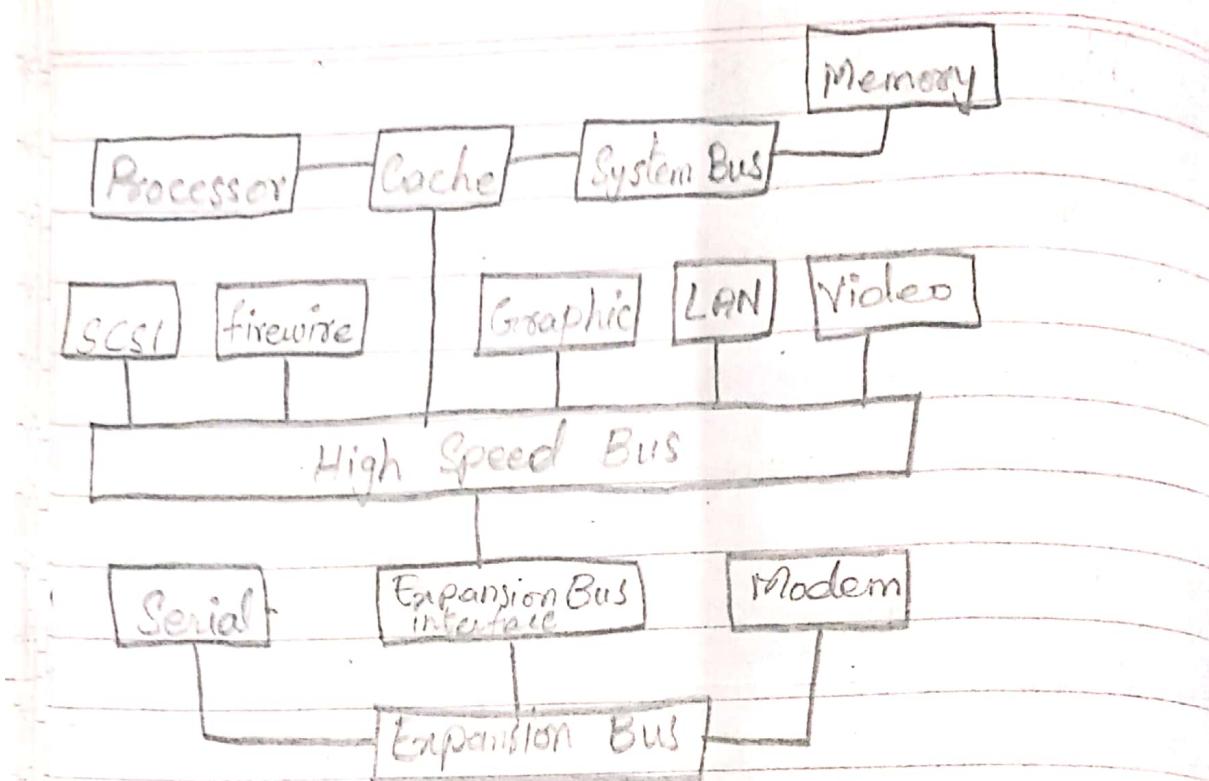
Interconnection Structure:



* Bus interconnection

* Point-to-point interconnection with packetized data transfer.





Elements of Bus Design:

- 1) Bus type : Dedicated / Multiplexed
- 2) Method of distribution: Arbitration
Centralized / Distributed
- 3) Timing: Synchronous / Asynchronous
- 4) Bus width: Data bus width or
Address bus width
- 5) Data transfer type supported:
Write, read, read modify write,
read after write, block transfer, etc.

switch: To connect two or more computers / devices

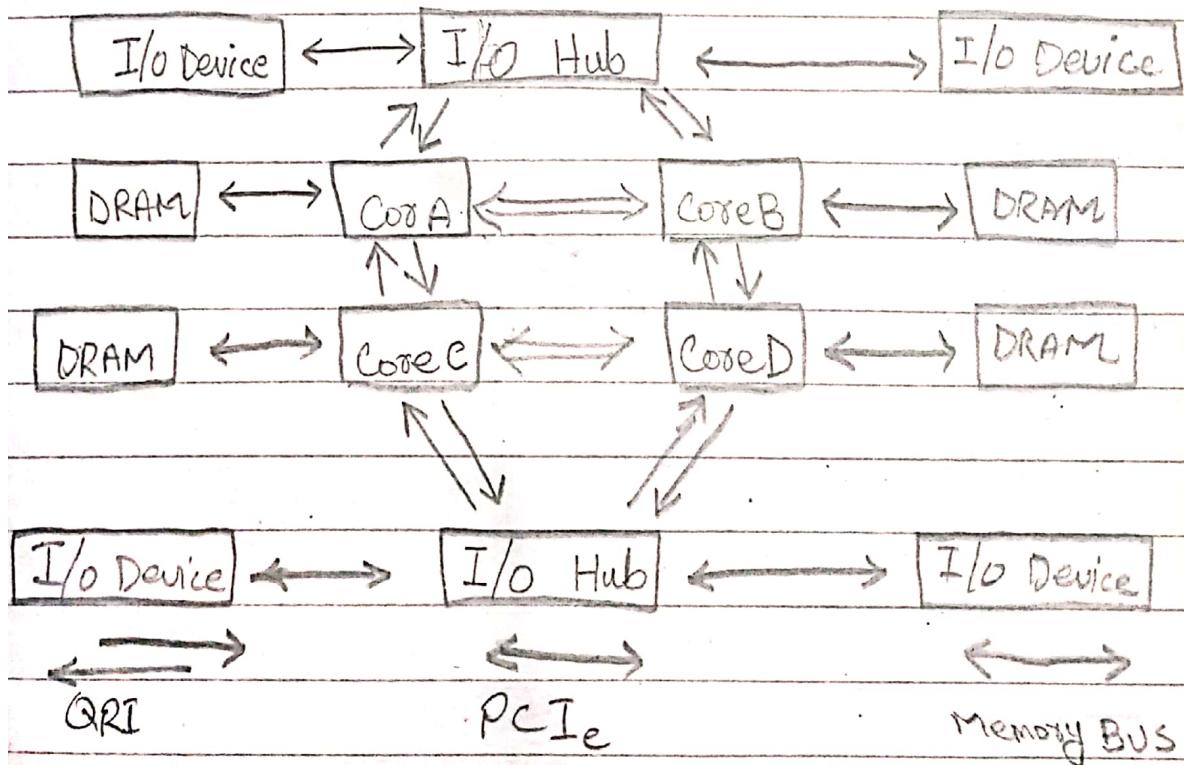
Router: To connect two or more networks

Point - to - Point \rightarrow Interconnection with
Packetized Data transfer:

A point-to-point connection between
two devices in a packetized form.

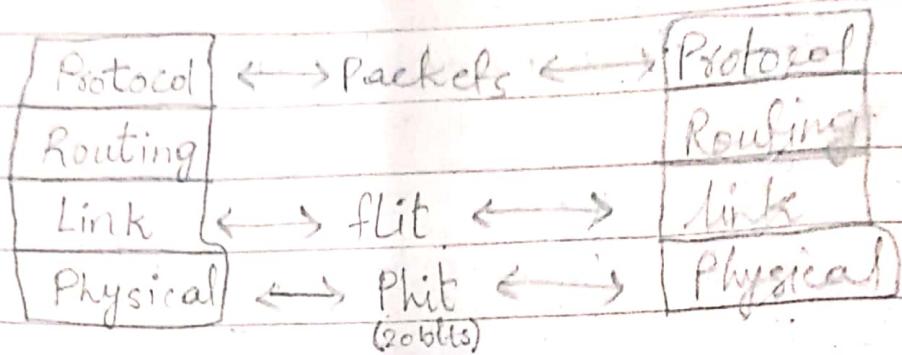
It has two types:

- ① QPI - Quick Path Interconnection
- ② PCIe - Peripheral Computer Interconnection Express



① QPI

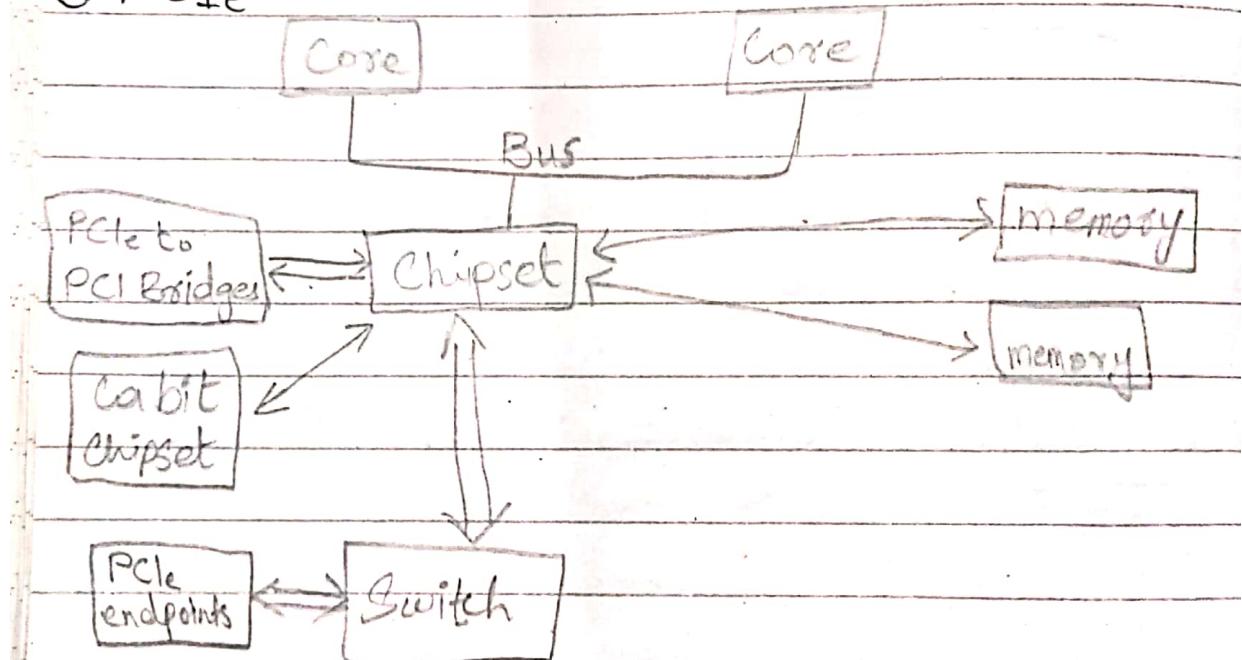
- i) Multiple Direct Connection
- ii) Layered Protocol Architecture

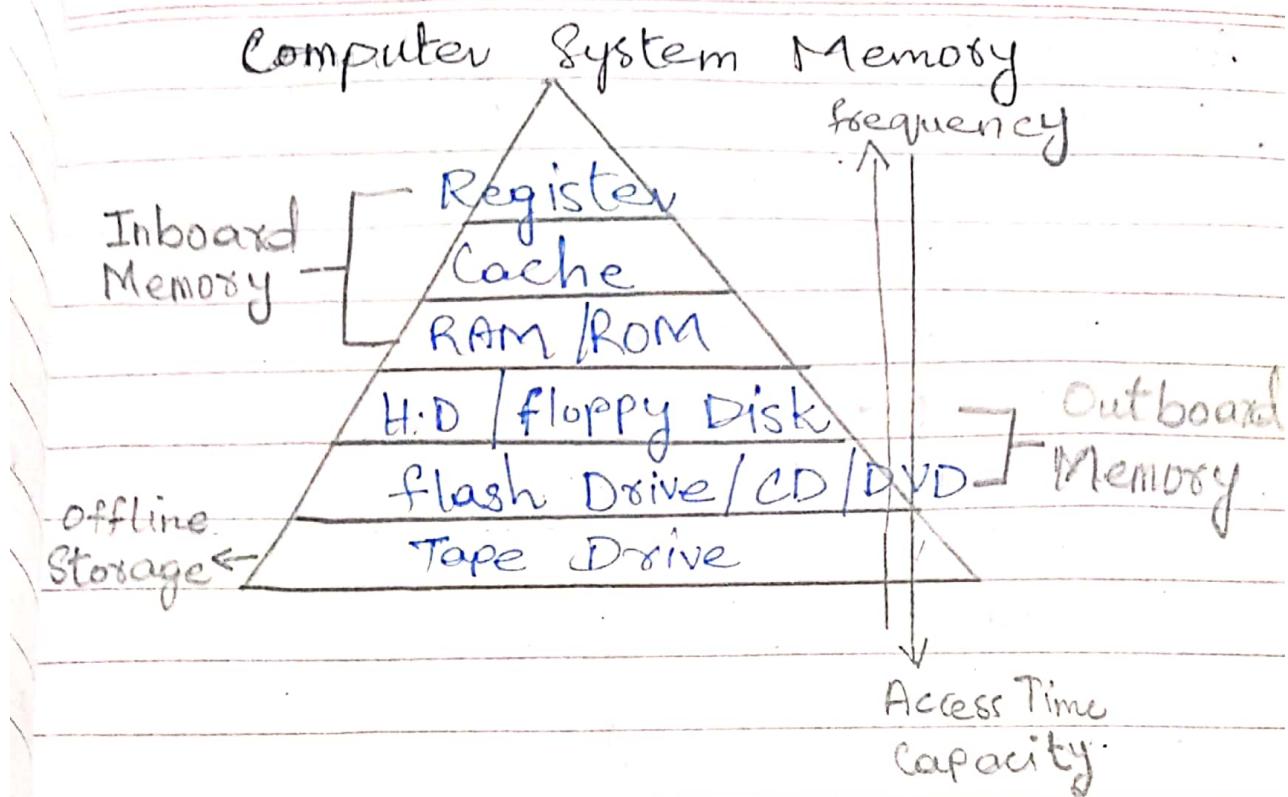


Phit - Physical Unit

flit - flow control unit

② PCIe





Characteristics:

1- Location

- (i) Internal (Register, Cache, MainM)
- (ii) External (Optical, Magnetic, flashD)

2- Capacity

Expressed in no. of bits (32 or 64 bit)

3- Access Method

- (i) Sequential - Linear Access
- (ii) Random - Randomly Access
- (iii) Direct - Seq. and random
- (iv) Associative - Access Memory using data

4- Unit of transfer instead of address.

(i) Word

(ii) Block

Access Methods:

5 - Performance

- (i) Access time
- (ii) Cycle time
- (iii) Transfer rate

RAM (Internal, Associative)

Non-RAM (Sequential & Direct, External)

if a and b are two address, a takes 1.2ms and b takes 1.15ms, then b has lower cycle time so, b is useable It is only Random Access memories.

6 - Physical Type

- (i) Semiconductor
- (ii) Magnetic
- (iii) Optical
- (iv) Magneto-optical

7 - Physical Characteristics

- (i) Volatile / non-volatile $\frac{\text{Transfer rate}}{\text{Cycle Time}} = 1 \text{ ms}$
- (ii) Erasable / non-erasable

for RAM

for Non-RAM

$$T_n = T_a + \frac{N}{R}$$

$T_a = \text{Access time}$

$N = \text{No. of bits}$

$R = \text{Transfer rate}$
in (bps)

$T_n = \text{Avg. Access Time}$

(i) Access Time:-

(a) RAM [Internal, Associative]

(b) Non-RAM [External, Sequential & Direct]

(ii) Cycle Time :-

$$\text{Cycle Time} = T_a + T_s$$

$\therefore T_a$ = Access Time

$\therefore T_s$ = Transient Time

(iii) Transfer Rate :-

(a) For RAM

$$\text{Transfer Rate} = \frac{1}{\text{Cycle Time}}$$

(b) For Non-RAM

$$\therefore T_n = T_a + \frac{N}{R}$$

$$\therefore R = \frac{N}{T_n - T_a}$$

$\therefore T_n$ = Avg. Access Time

* to w/f N bits

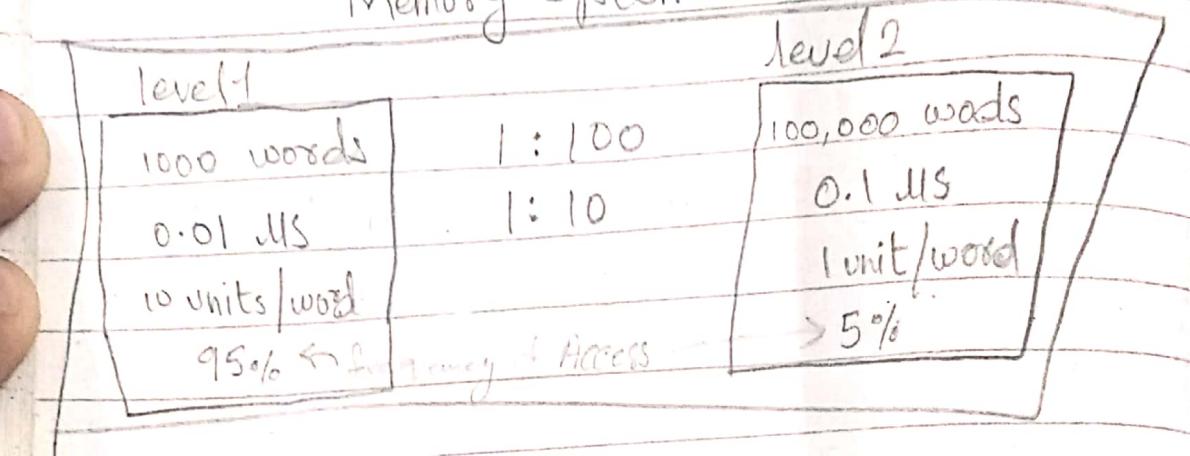
$\therefore T_a$ = Avg. Access Time

$\therefore N$ = No. of bits

$\therefore R$ = Transfer Rate

in bps

Memory System



$$\text{Total Capacity} = 101000 \text{ words}$$

$$\begin{aligned}\text{Total Cost} &= (1000 \times 10) + (100,000 \times 1) \\ &= 110,000 \text{ units}\end{aligned}$$

$$\text{Avg. Cost} = \frac{\text{Total Cost}}{\text{Total Capacity}}$$

$$= \frac{110000}{101000}$$

$$= 1.089 \text{ units/word}$$

$$\text{Avg. Access Time} = \frac{(95 \times (0.01 \mu\text{s})) + (5[0.1 + 0.01])}{100}$$

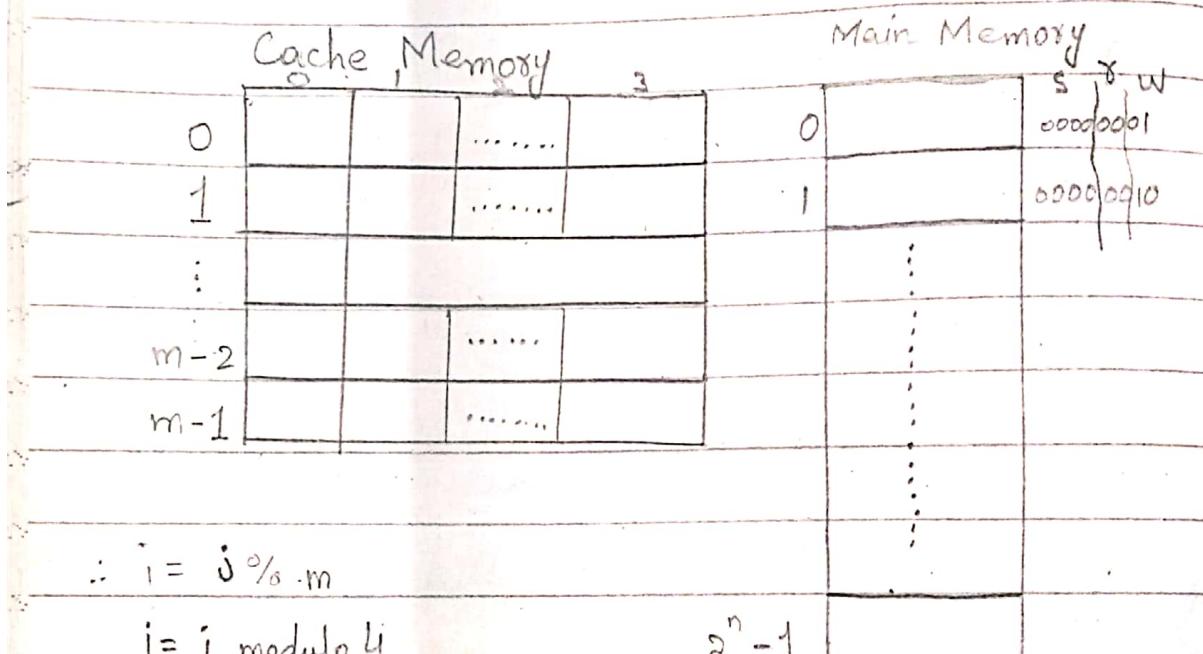
$$= 0.015 \mu\text{sec}$$

Cache Memory :-

A special buffer storage, smaller and faster than main memory, that is used to copy of instructions and data in main memory that are likely to be needed next by the processor and that have been obtained automatically from main memory /storage.

Elements of Cache Design:-

- i) Cache Size
- ii) Write Policy
- iii) Replacement Algorithm
- iv) Mapping functions.
- v) Line size / Block size.
- vi) Number of caches.



(i) Cache Size:

Smaller the cache size faster the access time.

L1 is smallest size in computers

(ii) Write Policy:

(a) Write Through

(b) Write Back

(a) Write Through: Works on non-shareable cache

All write operations are made to main memory as well as to cache.

The main disadvantage of this technique is that it generates substantial memory traffic and may create a bottleneck. Bus watching technique

(b) Write Back:

It minimizes memory writes, It only updates only cache. When an update occurs, a dirty bit or use bit associated with the line is set.

(iii) Replacement Algorithm:-

- (a) Least Recently Used (LRU)
- (b) First in First out (FIFO)
- (c) Least Frequently Used (LFU)
- (d) Random.

(iv) Mapping Functions:

- (a) Direct → advantage: Don't need to use Replacement Algo.
- (b) Associative
- (c) Set Associative

(v) Line Size / Block size

tag	$S-8$	γ	W
Block Size			Line Size
2^S	.	.	2^W

↓
no. of lines = 2

ML

Thrashing :-

If a program happens to reference words repeatedly from two different blocks that map into the same line, then blocks will be continually swapped in the cache, and the hit ratio will be low (a phenomenon known as Thrashing)

Victim Cache :-

Use to overcome thrashing problem.

(iv) [c] set Associative Function

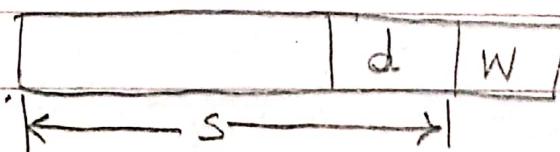
$$j = \frac{m}{k} \quad \leftarrow m = \text{no. of lines} = 2^8$$

$\leftarrow k\text{-way set associative Function}$

$$\cancel{i} = j \% m$$

$$\text{set no. } i = j \% N$$

$$j = \frac{m}{N}$$



Ques.

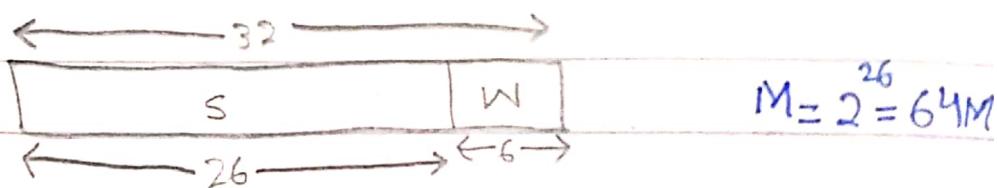
A system with 32 address lines and a cache of 64 KB i.e. organized as 8-way set associative with Line size 64 byte is used. How would memory system interpret memory address.

$$n = 32 \quad (\text{MM size} = 2^{32} = 4 \text{ GB})$$

$$\text{Cache size} = 64 \text{ KB}$$

$$K = 8 \quad (\text{set-associative})$$

$$k = 64 \text{ byte} = 2^6 \Rightarrow w = 6$$



$$M = 2^{26} = 64 \text{ M}$$

$$V = \frac{m}{K} = \frac{1024}{8}, \quad m = \frac{\text{cache size}}{K}$$

$$V = 128 = 2^7$$

$$= \frac{64 \text{ KB}}{1 \text{ KB}}$$

$$m = 1 \text{ K}$$

$$S = 26 = 19 \quad d = 26 - 19 = 7 \quad W = 6$$

Replacement Algorithm:-

- (1) FIFO First in first out Worst Algo
- (2) LFU Least Frequently Used
- (3) LRU Least Recently Used
- (4) Random (Best Algo)

"Locality of Reference"

Multilevel Cache:

- (1) L1
- (2) L2
- (3) L3

Unified vs Split Cache

4M

Lecture #09

Arithmetic & Logic Unit

$$\begin{array}{r}
 1001 \quad -1 \\
 0101 \quad +5 \\
 \hline
 X - 6 = 1110 \quad +4 \checkmark
 \end{array}$$

This is correct
in 2's complement

sign always be greater no.

Number Representation:

sign	data		2's complement
M	0000	→ +0	
	0001	→ +1	→ 1111 → -1
	(0010)	→ +2	→ 1101 → -2

Sign-Magnitude Representation:-

$$\begin{array}{r}
 -1 \quad 1001 \quad -7 \\
 +5 \quad \underline{0101} \quad +5 \\
 \hline
 +4 \quad \quad \quad -2
 \end{array}$$

$$\begin{array}{cccc}
 -2^3 & 2^2 & 2^1 & 2^0 \\
 \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} \\
 -8 & +4 & +2 & +1 = -1
 \end{array}$$

Adder, Subtract, multiplication



$$2 + (-3)$$



Adder
circuit

↙ 2's complement circuit

Sign [Range]
Formula

Bit extension

No. of zeros: 2

$$\text{Range: } -(2^{n-1} - 1) \leftrightarrow + (2^{n-1} - 1)$$

1001
↓

1000 0 001

Negation Rule: Magnitude Diff.

we move the sign bit
to MSB to convert to
8-bit

Overflow rule:

Two's complement [Range]
Formula

$$\text{Range: } -2^{n-1} \leftrightarrow + 2^{n-1} - 1$$

$$\begin{array}{r} 1001 \\ - 1111 \\ \hline 11111001 \end{array} = -7$$

Bit extension

$$-128 + 64 + 32 + 16 + 8 + 0 + 0 + 1 = -7$$

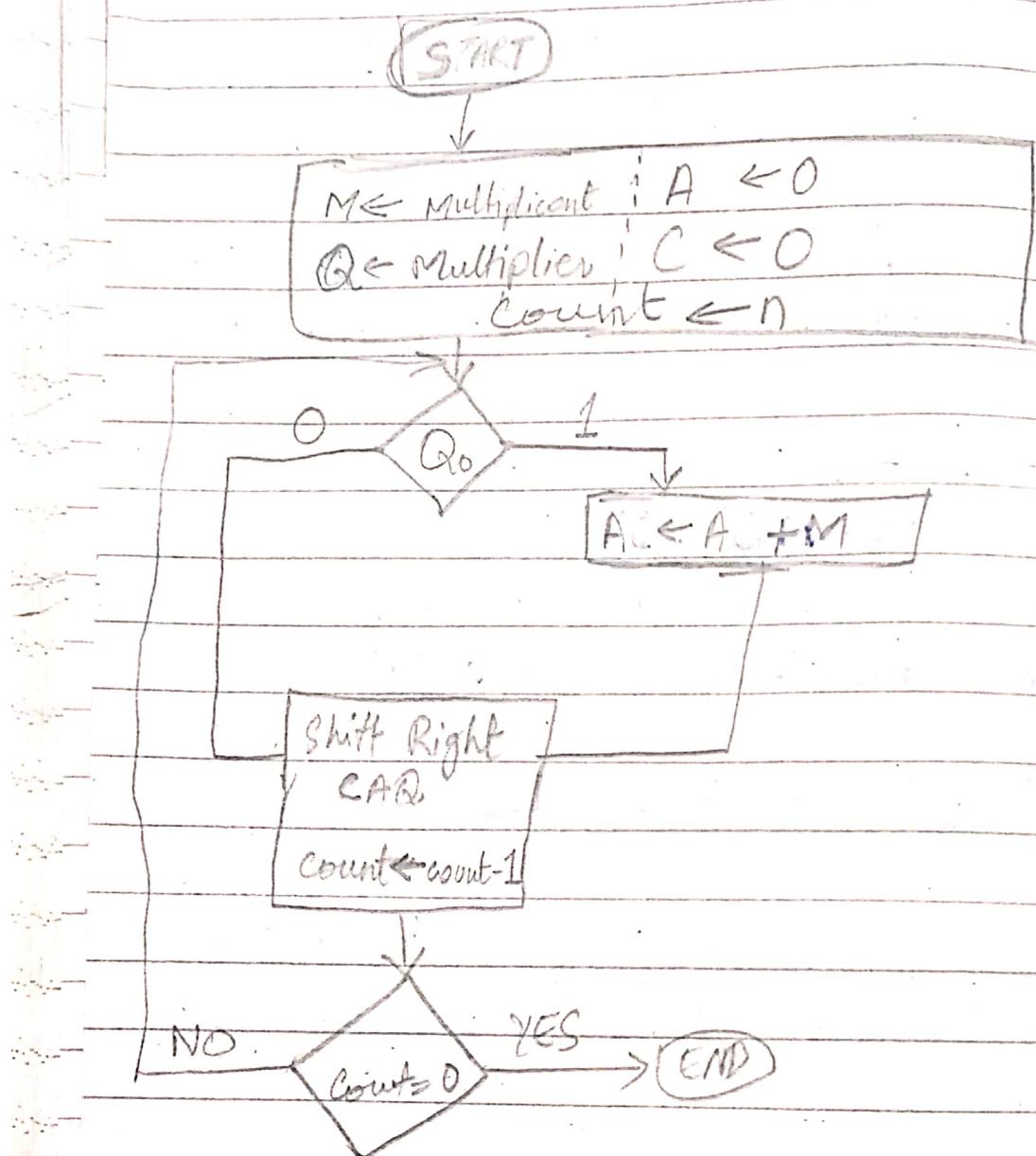
Negation Rule: $A - B = A + (-B)$

Number of zeros = 1

Overflow Rule:

Multiplication

→ we need double size resultant bits



$\begin{array}{r} 1101 \\ \times 2^3 \\ \hline 2220 \end{array}$

2^3 complement \rightarrow $(-2^3) 2^2 2^1 2^0$

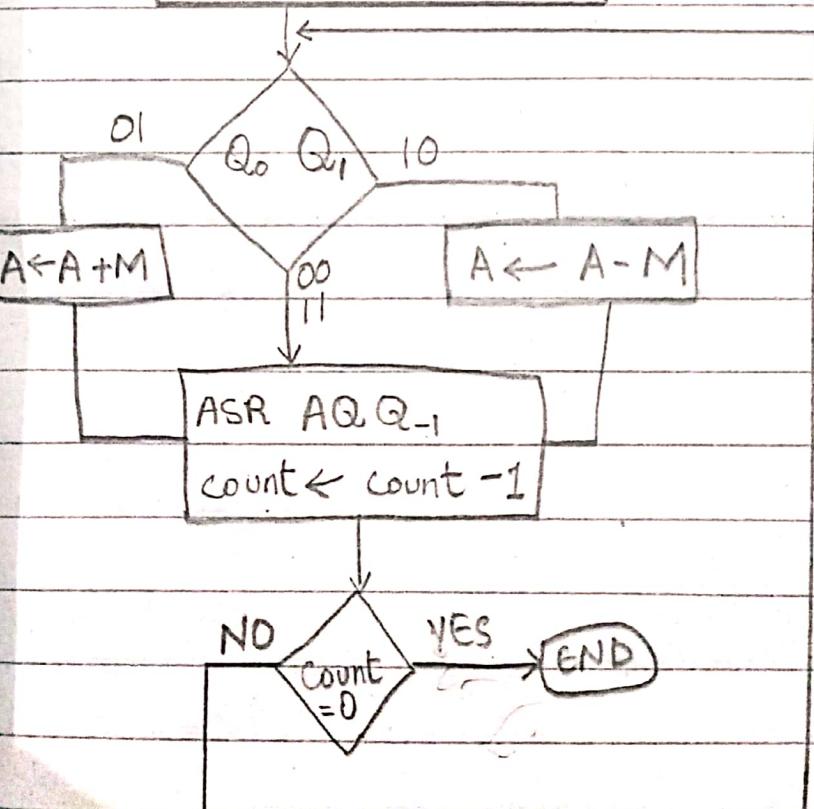
Sign
bit

Multiplication Signed Numbers

Booth's Algorithm:

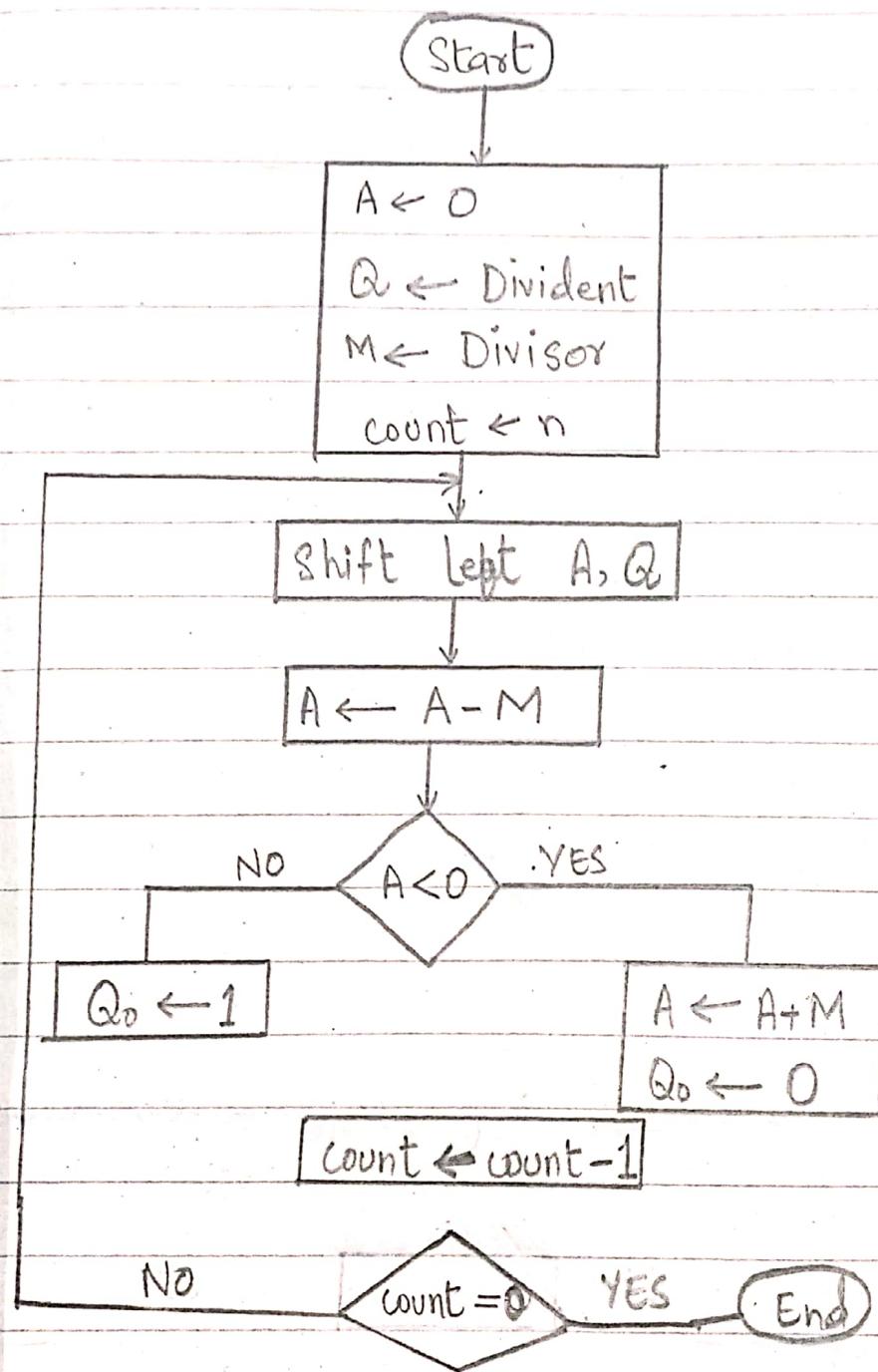
Start

$A \leftarrow 0, Q_{-1} \leftarrow 0$
 $Q \leftarrow \text{Multiplier}$
 $M \leftarrow \text{Multiplicand}$
 $\text{count} \leftarrow n$



Steps	A	Q	Q_{-1}	Count
Initialization	0000	0101	0	4
$Q_0 Q_{-1} = 10 \Rightarrow A \leftarrow A - M$	0111	0101	0	
ASR count-1	0011	1010	1	3
$Q_0 Q_{-1} = 10 \Rightarrow A \leftarrow A + M$	1100	1010	1	
ASR count-1	1110	0101	0	2
$Q_0 Q_{-1} = 10 \Rightarrow A \leftarrow A + M$				

Restoring Division Algorithm of Unsigned Numbers:



$$\text{Sign}(R) = \text{Sign}(D)$$

$$\text{Sign}(Q) = \text{Sign}(D) \times \text{Sign}(V)$$

$$M = 00001010$$

$$-M = 11110110$$

Steps	A	Q	Count
Initialization	0000 0000	0101 0101	8
Shift Left (SL)	0000 0000	1010 1010?	
$A \leftarrow A - M$	1111 0110	1010 101?	
$A < 0; Q_0 = 0; A+M; \text{count}-1$	0000 0000	1010 1010	7
SL	0000 0001	0101 0100	
$A \leftarrow A - M$	1111 0111	0101 010?	
$A < 0; Q_0 = 0, A+M; \text{count}-1$	0000 0001	0101 0100	6
SL	0000 0010	1010 100?	
$A - M$	1111 1000	1010 100?	
$A < 0; Q_0 = 0; A+M; \text{count}-1$	0000 0010	1010 1000	5
SL	0000 0101	0101 000?	
$A \leftarrow A - M$	1111 1011	0101 000?	
$A < 0; Q_0 = 0; A+M; \text{count}-1$	0000 0101	0101 0000	4

Floating Point Number Representation:

$$\pm S \times B^{\pm E}$$

S → Significant bit

B → Base

E → Exponential power

32-bit floating point

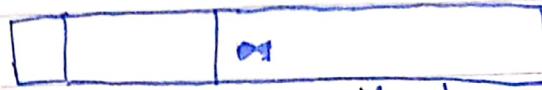
Sign	Exponent	Significant
1bit	8-bit	23-bit

Bias formula: $2^{K-1} - 1$ $K=8$

$$2^{8-1} - 1 = 2^7 - 1 = 128 - 1 = 127$$

Q. Why is bias is different from 2's complement.

Biased Representation of floating pointer



Sign Exponent
(biased) Significant

-10100
 1.0101×2

$$\text{Exponent (Biased)} = 2^{-10100 + 2^{k-1}}$$

$$= 2^{-10100 + 2^{b-1}}$$

$$= 2^{-10100 + 2^7 - 1}$$

$$= 2^{-10100 + 128 - 1}$$

$$= 2^{-10100 + 127}$$

$$-10100 = -20$$

$$= 2^{-20 + 127}$$

$$= 2^{-107}$$

0	11101011	01010000000000000000
---	----------	----------------------

$$2 - 3 \\ 2 + (-3)$$

IEEE 754 Specifications

- 1) $\text{Exp} = 0, \text{Frac} = 0 \Rightarrow \text{zero} \pm$
- 2) $\text{Exp} = 0, \text{Frac} \neq 0 \Rightarrow \text{subnormal number}$
- 3) $\text{Exp} = \text{all } 1, \text{Frac} = 0, \pm \infty$
- 4) $\text{Exp} = \text{all } 1, \text{Frac} \neq 0, \text{NaN}$

$$x = x_s \times B^{x_e}$$
$$y = y_s \times B^{y_e}$$

$$x * y = x_s * y_s \times B^{(x_e + y_e) - 127}$$

$$x \div y = x_s \div y_s \times B^{(x_e - y_e) + 127}$$

Addition/Subtraction

- 1) Check for zero
- 2) Align the significants
- 3) Add
- 4) Normalize the answer.

Comparison is very ^{easy} in bias representation

Instruction Set

Add 2+2

Opcode : 8 bit

Source operand Reference (address + AC)

Result operand Reference (AC)

Next instruction Reference (PC)

OP code	Source Operand Reference	Result Operand	Next Instruction Reference
---------	--------------------------------	-------------------	----------------------------------

* There are many instruction formats and it varies system to system.

$$Y = (A - B) / (C + D * E)$$

SUB Y, A, B	$Y \leftarrow A - B$	Three Instruction Format
MUL T, D; E	$T \leftarrow D * E$	
ADD T, C, T	$T \leftarrow C + T$	
DIV Y, Y, T	$Y \leftarrow Y / T$	

LOAD Y, A	$Y \leftarrow A$	Two Instruction Format
SUB Y, B	$Y \leftarrow Y - B$	
LOAD T, D	$T \leftarrow D$	
MUL T, E	$T \leftarrow T * E$	
ADD T, C	$T \leftarrow T + C$	
DIV Y, T	$Y \leftarrow Y / T$	

LOAD A	$AC \leftarrow A$	One Instruction Format
SUB B	$AC \leftarrow AC - B$	
STORE Y	$Y \leftarrow AC$	
LOAD D	$AC \leftarrow D$	
MUL E	$AC \leftarrow AC * E$	
ADD C	$AC \leftarrow AC + C$	
STORE Y	$Y \leftarrow AC$	
LOAD A	$AC \leftarrow A$	
SUB B	$AC \leftarrow AC - B$	
DIV Y	$AC \leftarrow AC / Y$	

Pronunciation

reh.puh.twaa

- Operation Repertoire
- Data types
- Instruction Set
- Registers
- Addressing Mode

- Intermediate
- Direct

Data Types

- Numbers
 - Integers.
 - Floating point
- BCD
- characters
- Data

etc

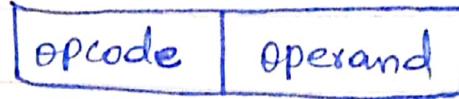
Operation ~~Rep~~ Repertoire:

- Types of Operations:
 - Arithmetic
 - Logical
 - Data Transfer
 - I/O

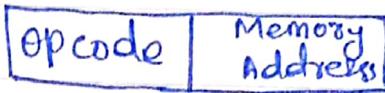
- Transfer of Control
- System control

Addressing Modes

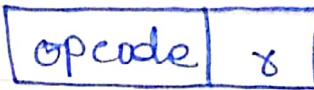
Intermediate



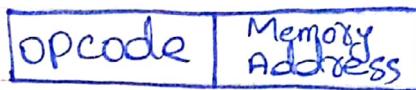
Direct



Registers



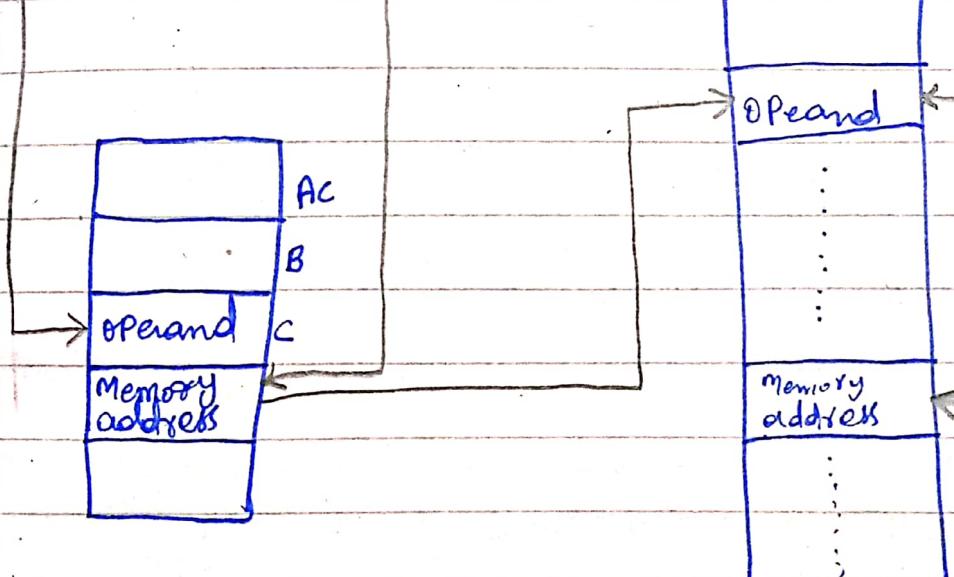
Indirect



Register Indirect



Displacement
Addressing Mode



of