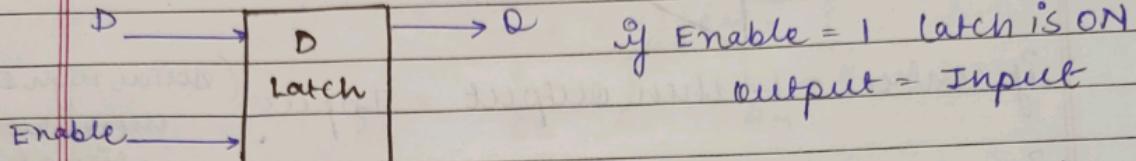


Memory

→ address lines on address line usually refers to a physical connection b/w CPU/clipset of memory. They.

OR

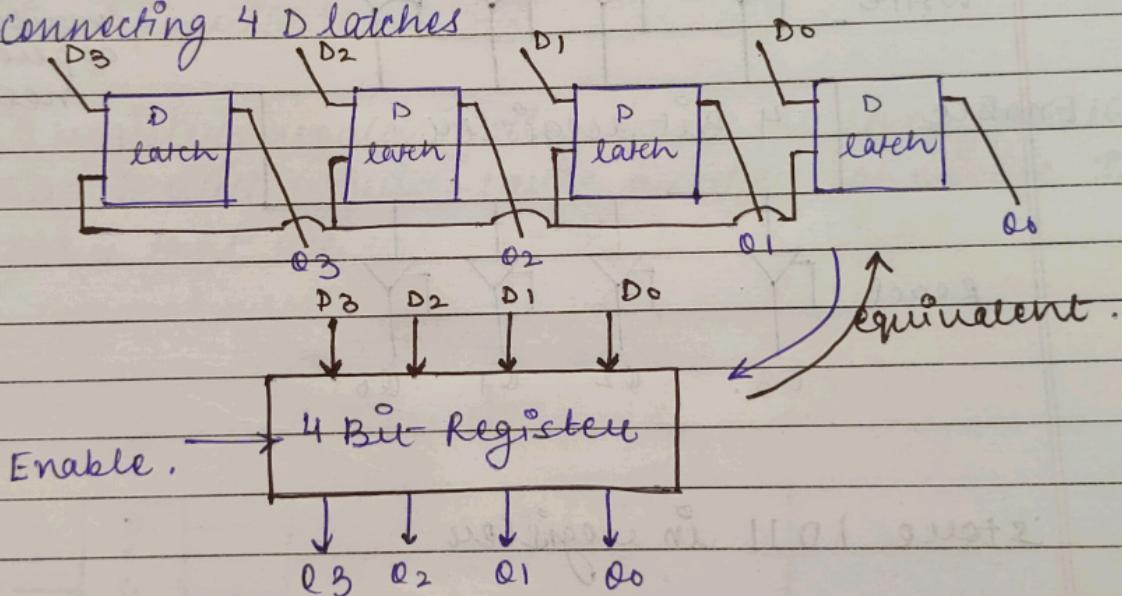
What is address in memory? specify which address to access in the memory.



If Enable = 1 latch is ON
output = Input

If Enable = 0 latch is OFF
Input has no effect on output
previous value is retained.

Connecting 4 D latches



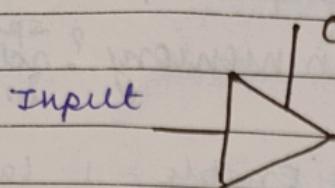
Buffer $\equiv \overline{Q}$ $Q = 1$.

\rightarrow high impedance state.

The connection b/w the two components (wire) is broken. There is no voltage.

Tri-state circuits ($0, 1$ and Z)

Tri-state Buffer

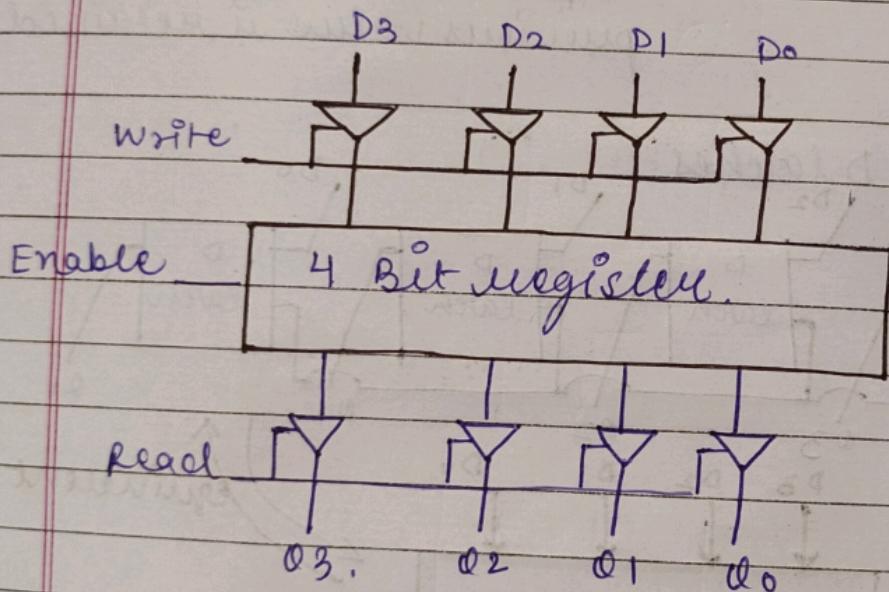


Need of tri-state Buffer?

We wanted a control signal so that read/write commands could work as per our will & not on its own.

If control = 1 then output = Input (active high signal works when signal is high or 1.)

else if control $\neq 1$ then output = Z
or control = 0



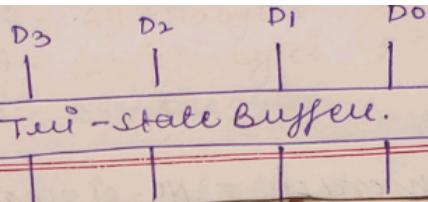
control here is renamed as write & read as per the need.

Store 1011 in register

1. Value is applied at $D_3 D_2 D_1 D_0$.
2. Activate Enable.
3. Activate Write.

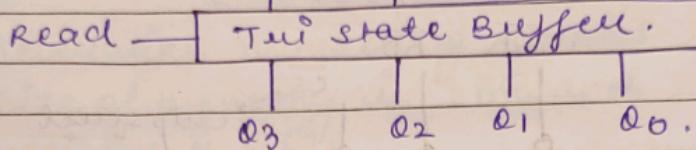
(4) means size of bit of the component.

write



PAGE NO.	
DATE	

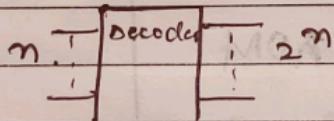
$$4 \times 4 = 16 \text{ bits.}$$



D₃ D₂ D₁ D₀.

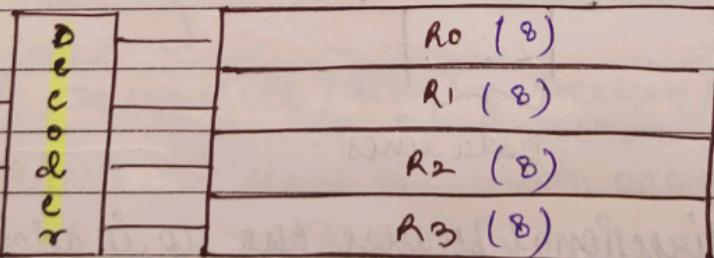
To read or write to a particular register then we need to enable specifically with activating its read / write command.

4 registers would need 4 enable to store just 16 bits but it will require large number of enable to select that we use decoder.



address lines.

A₁
A₀



A₁ A₀

0 0 → assigned to R₀

0 1 → R₁

1 0 → R₂

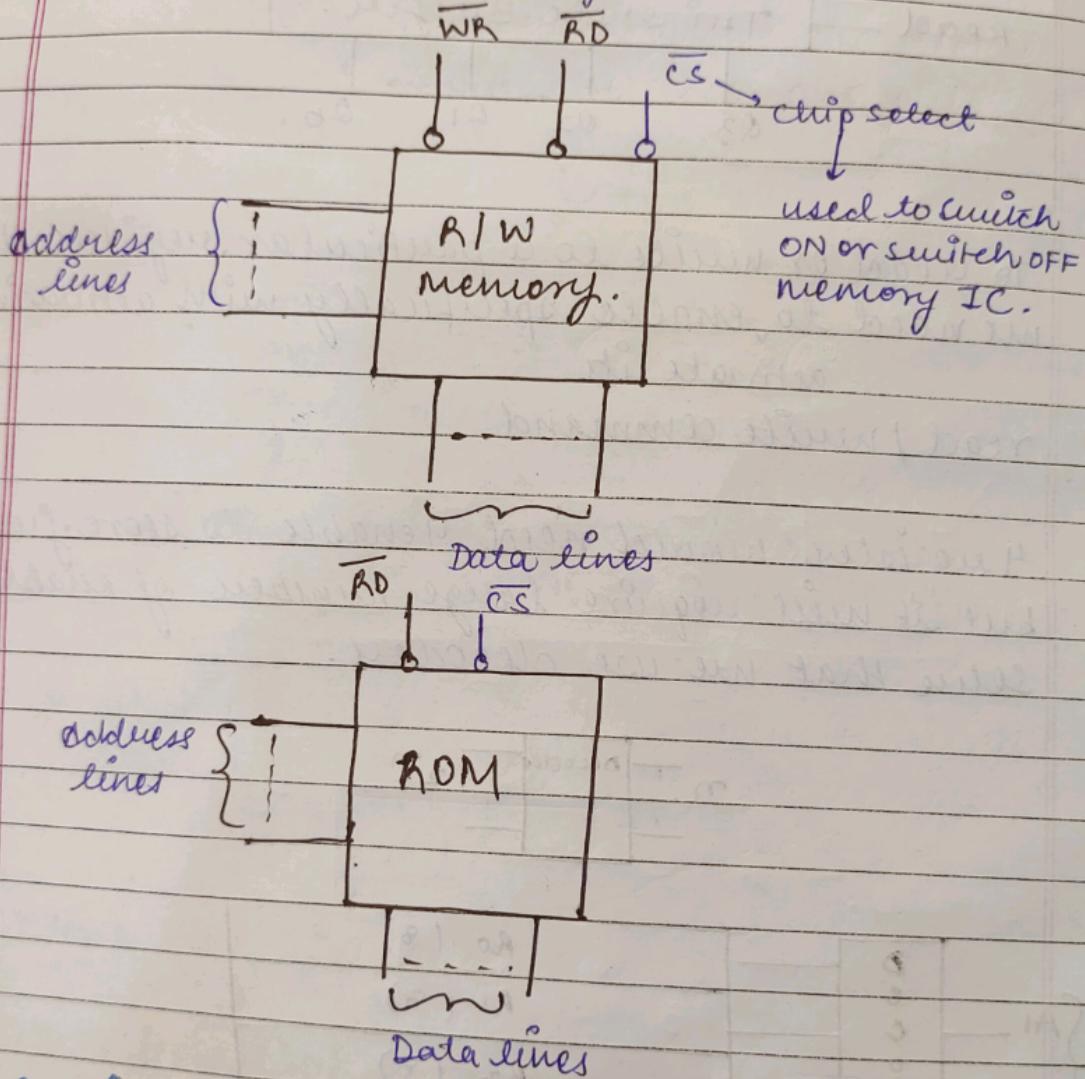
1 1 → R₃

Size of memory is = No. of registers \times size of each register.

$$= 4 \times 8 \text{ bits}$$

$$= 4 \times 1 \text{ bytes}$$

$$= 4 \text{ bytes.}$$



Address Bus is Unidirectional because the up is addressing a specific memory location.

Data Bus is Bidirectional because up can read data from memory or write data to the memory.

Control Bus carries control signals from the up to other components. Normally it is unidirectional but sometimes it is bidirectional because up needs to know the status of other buses as well.

$$\begin{aligned}2^{16} &= 64 \text{ kilobytes} \\2^{17} &= 128 \text{ kilobytes} \\2^{18} &= 256 \text{ kilobytes}\end{aligned}$$

$$2^{20} = 1024 \times 1024 \text{ kilobytes}$$

PAGE NO.	
DATE	

ques

Determine the size of memory if number of address line is 16 and number of data line is 8.

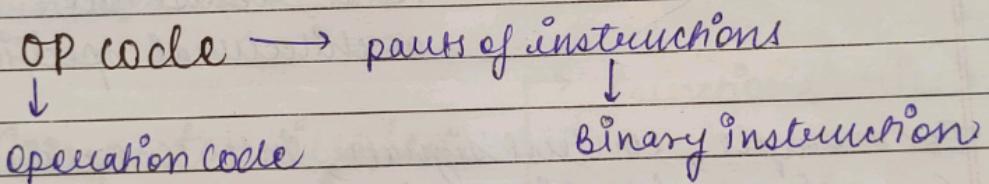
$$\begin{aligned}2^{16} \times 8 &= 2^{16} \text{ bytes} \\&= 64 \text{ kilobytes.}\end{aligned}$$

Determine number of address lines if the size of memory is 1 MB and size of each register is 8 bit.

$$\begin{aligned}8 \times 10^6 &= 8 \times 2^n \\10^6 &= 2^n \\8 \times 1024 \times 1024 &= 8 \times 2^n \\8 \times 1024 \times 128 &= 2^n.\end{aligned}$$

$$n = 20$$

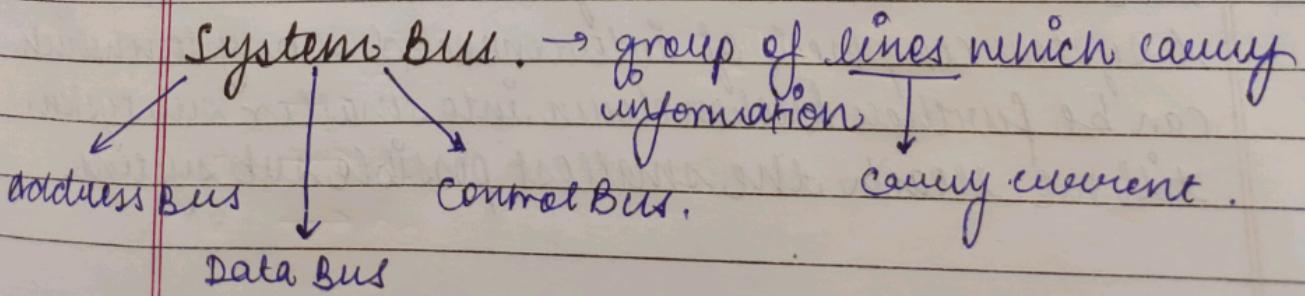
The Opcode is the instruction that is executed by the CPU.



- Fetch
 - Decode
 - Perform
- } processor Functions.

Is the data / memory location used to execute a instruction.

OPERAND → data on which operation is to be performed.



The Main Component Of a Computer System.

The main : architecture vs organization.

Architecture : are the attributes of a system visible to a programmer and decides the logical execution of a program.

For example, instruction set, number of bits used to represent data type, I/O mechanism and techniques for addressing memory.

Computer organisation : refers to the operational unit and their inter connections that realise the architectural specification

For example, control signals, interconnection b/w computer and peripherals and the memory technology used.
I/O devices

A computer is a complex system and to design and analyse such complex system we use Hierarchical system methodology.

Ques what is a hierarchical system?

A system composed of interconnected sub-system which can be further broken down into smaller sub-system till we reach the smallest possible sub-system.

Hierarchical system
Designed.

↓
Top - Down approach
(Prescribed)

↓
Bottom - Up approach.

Practically, combination of both.

Function

Function of a sub system tells us about of the operation that can be performed by the system.

Structure

Is the way to connect various sub system to achieve the desired function.

Computer System

→ Functional view
→ Structural view.

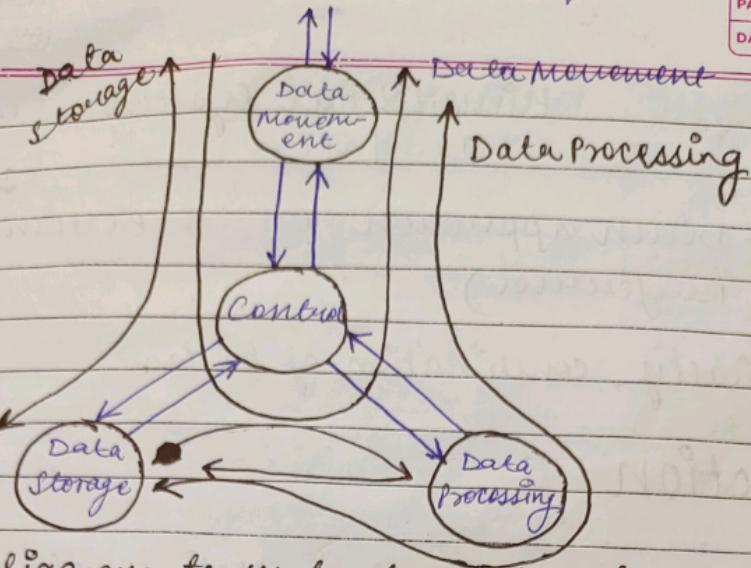
Functional view of computer:

There are in general 4 functions performed by a computer.

1. Data Processing.
2. Data Movement.
3. Data Storage.
4. Control.

Operating environment.
(source and destination of data)

PAGE No.	
DATE	



Block diagram to understand functional view of computer.

Data Movement : Data from one peripheral is transferred to some other peripheral.

Data storage : Data from peripheral gets stored and vice versa.

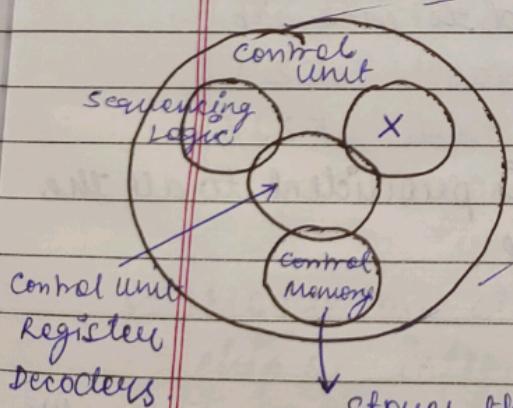
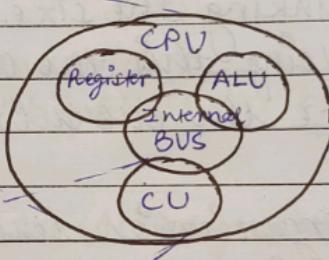
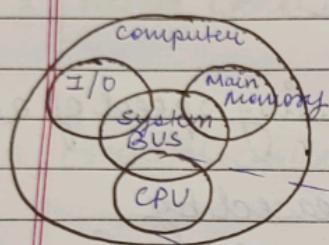
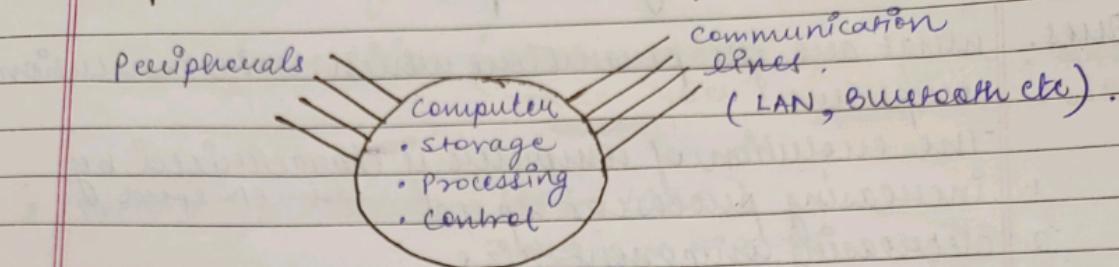
Data Processing : Various processing ways are possible.

Control : Control has nothing to do with the operational environment.

The instruction decoder is a CPU component that decodes and interprets the contents of the instruction register.

PAGE NO.	
DATE	

Top View (Structural) Of Computer.



stores the steps to implement a functionality (program).

A CU typically uses a binary decoder to convert coded instructions into timing and control signals that direct the operation of the other units.

use a binary decoder to convert coded instructions

into timing and control signals that direct the operation of the other units.

Time taken by the data to reach from one component to another reduces as the size of spacing b/w the components has reduced.

PAGE No.	
DATE	

Evolution Of Computer

ques. what are the parameters which decide evolution of computers?

The evolution of computer is characterized by faster R/W speed.

1. increasing processor speed.
2. decreasing component size.
3. increasing memory size.
4. " I/O capacity and speed.

ques how can be improve the processing speed of a computer?

The processing speed can be increased by

1. Shrinking the size of component such as microprocessor by doing this the distance b/w component reduce and hence the overall speed will improve.
2. At organization level use of parallel processing and pipelining techniques are employed to increase processing speed.
more core on the CPU allow multiple programs on different pipelines.

Processor Scheduling : Equal time of a processor is provided to all the processes scheduled in a queue.

Multi-processor system.

Multi-core system : Single microprocessor with various ALU.

Generation Of Computer

1. First Generation (1946 - 1959)

vacuum tubes
Based.

programming thru
switches and wires.

2. Second Generation (1959 - 1965)

Transistor based

3. Third Generation (1965 - 1971)

Integrated circuit Based

4. Fourth Generation (1971 - 1980)

VLSI processor based

5. Fifth Generation (1980 onwards)

DOS VLSI processor based

SSI 10 - 100 small scale Integration

MSI 100 - 1000 Medium scale "

LSI 1000 - 10,000 Large scale "

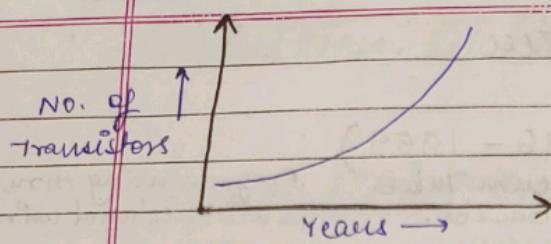
VLSI 10,000 - 2,000,000

ULSI 2,000,000 - No Limit

Moore Law states that number of transistors on a chip or integrated circuit will double every 2 years

How INTEL processors are fabricated?
fab lab.

PAGE No.	
DATE	



Von Neumann Architecture and Non Von Neumann Architectures.

Types of computer

Fixed Program
(Calculators)

Stone Program

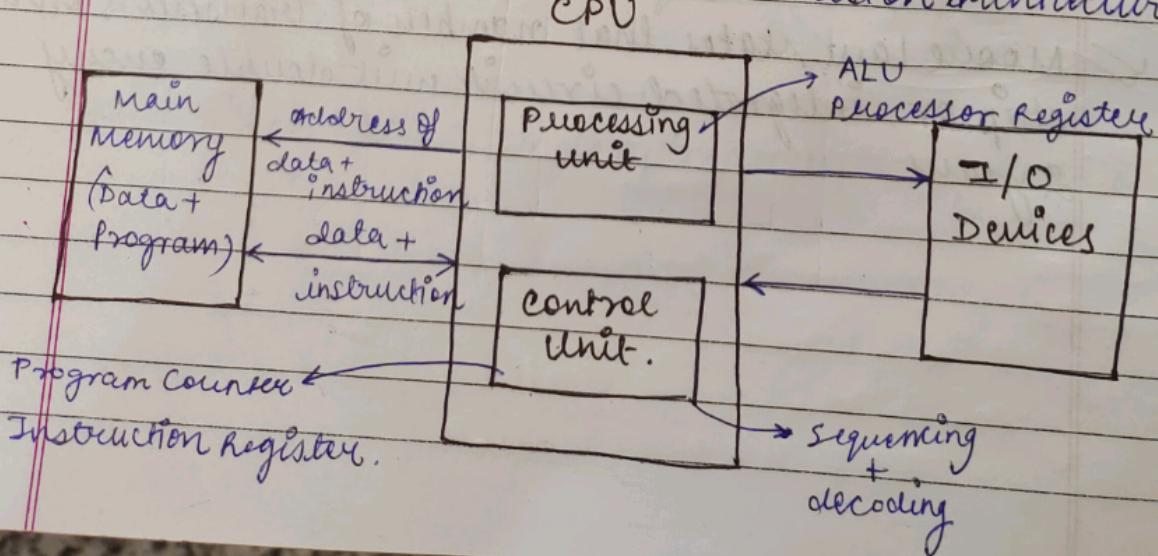
• concept by John von Neumann

1. They cannot be re-programmed.
- Re-programmable because of the presence of memory in which new instructions could be fed.

• IAS Architecture.

Institute of Advanced Studies
in Princeton.

• Princeton Architecture.



instruction : The action/process that is to be performed on the data is coded and fed.

Data : on which the action is to be performed.

PAGE NO.	
DATE	

By default : Microprocessor fetches instruction from the memory.

Ans. when everything is stored as 0s and is how the microprocessor tells whether it has fetched data or instruction?

Now suppose, microprocessor decodes the instruction as ADD so it will understand that next two memory addresses will be operands.

Instruction	fetch1
Data 1	0101 000
	fetch2
Data 2	0111 1011
	fetch3
instruction (new)	0101 0110
	1001 0110

PC is a CPU register in the computer processor which has the address of the next instruction to be executed from memory.

Program Counter : Register that stores address from where instruction is fetched.

Instruction register : where the instruction resides before its implementation.

Von Neumann Bottleneck

Hindrance due to which speed cannot be increased further.

The hindrance is instruction and data couldn't be fetched at the same time.

Bobo → Father of processors.

PAGE No.	
DATE	

AIW CPE

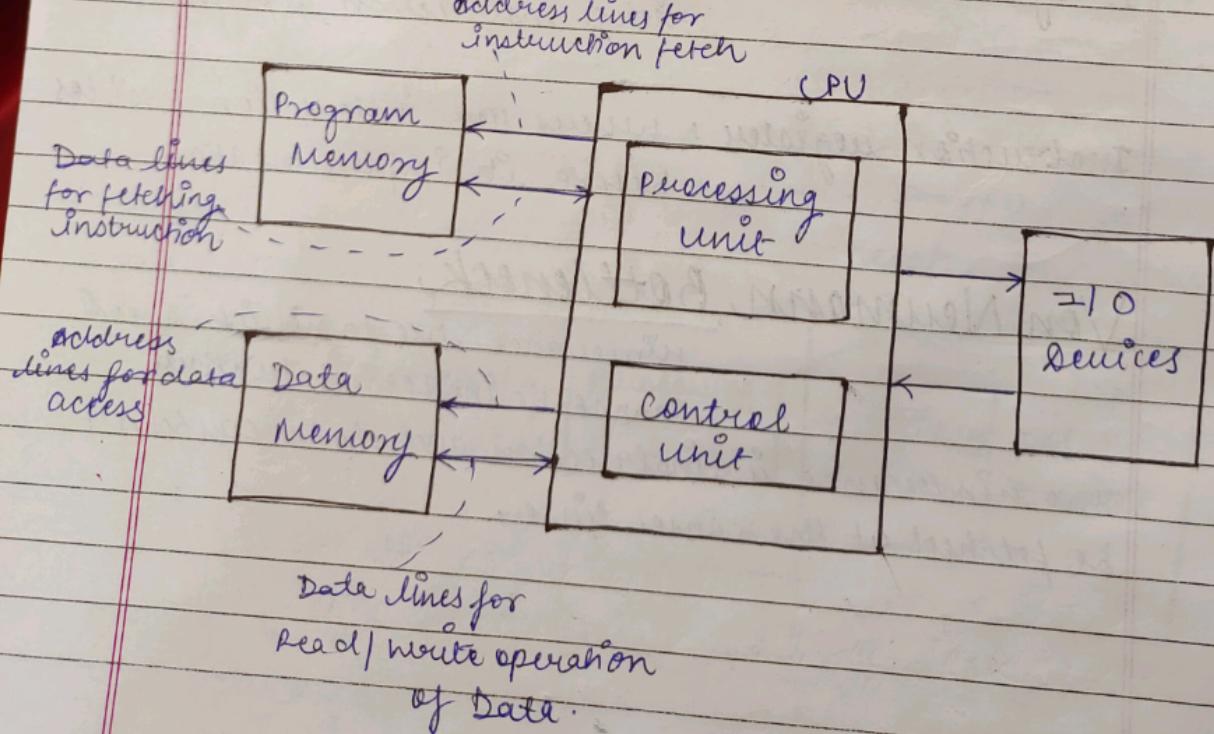
What is von Neumann Bottleneck?

The performance of the system is limited by the fact that an instruction fetch and reading/writing of data cannot occur at the same time because a common bus (address lines & data lines) is shared by instruction & data operation.

To overcome this problem another architecture was proposed.

Harnard Architecture. used in digital signal processing

To overcome Von Neumann Bottleneck Harvard architecture uses separate memory for data and separate memory for program hence dedicated address & data lines are used for accessing programs and a dedicated address of data lines are used for accessing data this will allow instruction fetch & data operation (R/W) at the same time.



modified von Neumann architecture, comb'n of Harvard and von Neumann.

A/W speed. Cache memory
high processing speed, lower storage space
memory b/w CPU and main memory.

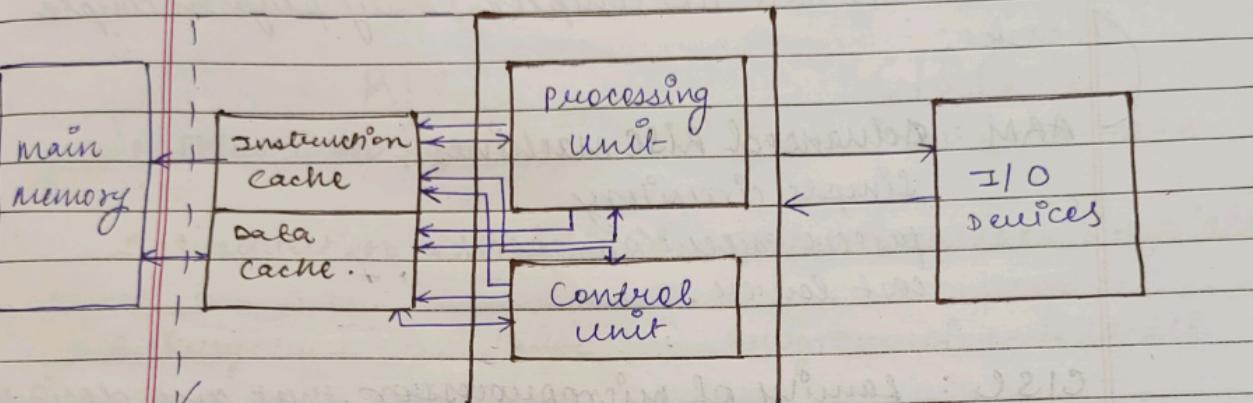
PAGE NO.	
DATE	

Harvard

Levels of cache memory

- L1 { split Cache memory
- L2
- L3 often
- L4 Rarely (supercomputers).

Instruction Cache
Data Cache.



may/may not reside on CPU.

Evolution of INTEL X86 Architecture

ones. What is X86 architecture?
8086.

x86 family (CISC)
architecture family
ARM family (RISC)

CISC → Complex Instruction set computer

Instruction Set.

All the instructions that are supported by a processor we group them as a set and call it instruction set.

RISC → Reduced Instruction set computer.

Complex instruction: requires several steps to execute the instruction.
Binary multiplication.

advantage : size of problem becomes smaller because up does the work itself.

RISC : complex circuitry
lower execution speed.
cost higher.

Instructions are not complex, they perform simple tasks.

ARM : Advanced RISC Machines.
Simple circuitry
faster execution speed
cost lower.

CISC : family of microprocessor that are designed with complex instructions to reduce the program size hence a single instruction can execute several low level operation.

But due to complex circuit speed of program execution is limited. Instructions generally take more than one clock cycle to execute. It has complex addressing modes and support large datatype. The size of instruction is also more than 1 word - 16 bits.

RISC : family of cpus that are designed to execute competing task with the simplest instruction in the shortest amount of time possible most of the instructions are executed in a single clock cycle.

$$\text{CPU time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instructions}} \times \frac{\text{seconds}}{\text{cycles}}$$

PAGE NO. _____
DATE _____

It has simple addressing mode but with fewer datatypes. Instructions are generally equal to less than one word in size. RISC architecture provide high pipelining possibility.

* RISC VS CISC

RISC

1. Focus is on software.

2. Instruction size is fixed Instruction size is variable

3. Program size is large. Program size is small.

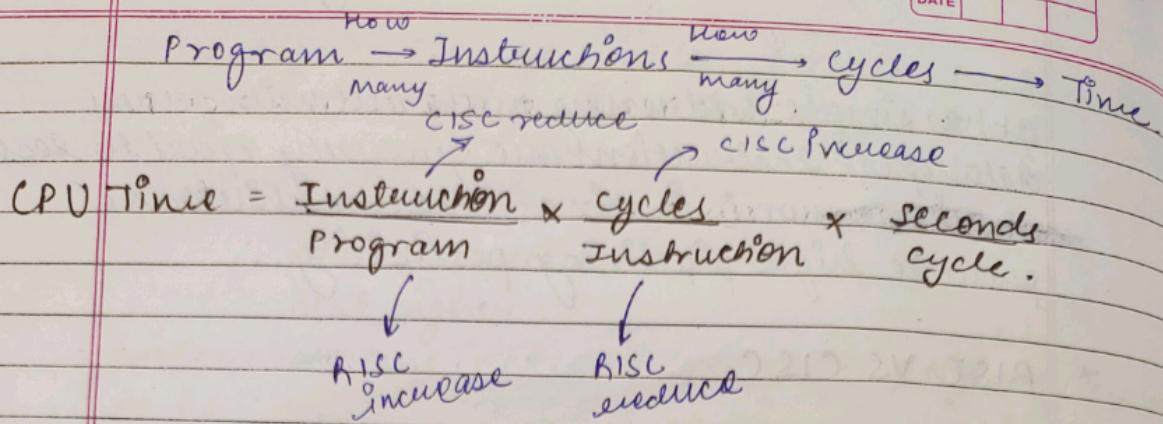
4. Uses Hard wired control unit uses hard wired as well as microprogrammed control unit.

5. Has more internal registers comparatively less internal registers.

6. Low power consumption. High power consumption

7. Higher cost

lower cost



Real Time Systems : For real time situations e.g., ATMs

1. Intel X86 family began with 8080 microprocessor which was first general purpose processor, it is an 8 bit machine with 8 bit data path to memory. It is the processor that was used in first personal computers.
2. 16 bit microprocessor with wider data path and larger register array. It was the first processor to introduce cache memory in the form of instruction queue. Pipelining was introduced in this processor which enables it to fetch the next instruction while the current instruction is being executed. This gave the family name X86.
3. 80286 : an extension of 8086 up where memory size was increasing to 1MB to 16MB
4. 80386 : was the first INTEL 32 bit machine which became a major computing source in

mainframe computers. It was the first processor to support multi-tasking : run multiple programs at the same time.

5. 80486 : A further enhancement with built-in math coprocessor for performing heavy computational operation. It also introduced complex instruction pipelining.
6. Pentium : Was the first processor by INTEL which employed super scalar techniques which allows multiple instructions to execute in parallel.
7. Pentium Pro : An enhancement to Pentium with introduction to branch prediction logic, data flow analysis for higher performance.
8. Pentium II : INTEL introduced MMX technology which allows efficient processing of video / audio and graphics data.
9. Pentium III : INTEL incorporated floating point instructions to support 3-D graphics software.
10. Pentium 4 : Included additional floating point support for multimedia operations.
11. Core : Was the first X86 family processor with Dual Core CPU which means implementing

two processors on a single chip.

12. Core 2 : It extends the architecture to 64 bits with dual core.

13. Core 2 Quad : Provides 4 processors on a single chip.

Performance Assessment Of Computer.

$$\text{Performance}_A = \frac{1}{\text{execution time}_A}$$

$$\text{Performance}_B = \frac{1}{\text{execution time}_B}$$

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{\text{execution time}_B}{\text{execution time}_A} = n.$$

If computer A runs a prog. in 10 sec and computer B runs the same prog. in 25 sec how much computer A is faster.

$$\frac{P_A}{P_B} = \frac{25}{10} = 2.5.$$

$$P_A = 2.5 P_B.$$

Response Time / Execution Time / Latency.

Amount of time that the program will take from start to end.

PAGE NO. _____
DATE _____

clock frequency is variable so that power consumption could be reduced because clock signal switch on and off frequently at max clock frequency.

Throughput.

How many amount of program/instruction/task can be performed in a specific amount of time.

Processor Clock.

CLOCK RATE / CLOCK FREQUENCY. (cycles per sec)

$$A \rightarrow R = 4 \text{ MHz}$$

$$P_1 \rightarrow 50 \text{ cycles}$$

$$B \rightarrow R = 2 \text{ MHz}$$

$$P_1 \rightarrow 50 \text{ cycles}$$

Time to execute

$P_1 = \text{No. of cycles required} \times \text{Time of execute one cycle (cycle time, CPU Time Period T-state)}$

$$= 50 \times \frac{1}{4} \times 10^{-6}$$

$$= 50 \times \frac{1}{2} \times 10^{-6}$$

$$= 50 \times 0.25 \times 10^{-6} = 50 \times 0.5 \times 10^{-6}.$$

$\text{CPU Time} = \frac{\text{No. of machine instructions required to execute a program}}{\text{instructions per instruction}}$

\downarrow

CPI

\downarrow

$\text{cycles per instruction}$

\leftarrow
execution time.

$$T = N \times S \times P$$

$$T = N \times CPI \times P$$

$$T = N \times CPI$$

R. Basic Performance Equation.

Let us assume that 2 computers use same instruction set architecture. Computer A has clock cycle time of 250 psec and a CPI of 2 for some program. Computer B has clock cycle time of 500 psec and a CPI of 1.2 for same program. Which computer is faster & by how much.

$$\frac{T_A}{T_B} = \frac{N \times 250 \times 2}{N \times 500 \times 1.2}$$

$$1.2 T_A = T_B.$$

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{T_B}{T_A} = 1.2$$

Computer B is faster.

MIPS : Million Instruction Per Second Rate

$$= \frac{1}{\frac{\text{avg execution}}{\text{time of instruction}} \times 10^6}$$

$$= \frac{1}{\frac{1}{\text{CPI} \times P} \times 10^6}$$

$$= \frac{R}{\text{CPI} \times 10^6}$$

$$= \frac{N \times R}{N \times \text{CPI} \times 10^6}$$

$$= \frac{N}{T \times 10^6}$$

Delay Programs : Programmed to generate a delay.

PAGE NO.	
DATE	

MFLOPS: Million Floating Point Operations per second.

Ques. The computer A has the operating frequency = 4 MHz. Determine the number of instructions required in a program to generate a delay of 1ms if CPI is 2.5.

$$T = \frac{N \times CPI}{R} \quad \text{or} \quad T = N \times CPI \times P.$$

$$R = 4 \text{ MHz}$$

$$P = \frac{1}{R} = \frac{1}{4 \times 10^6} = 0.25 \mu\text{s}.$$

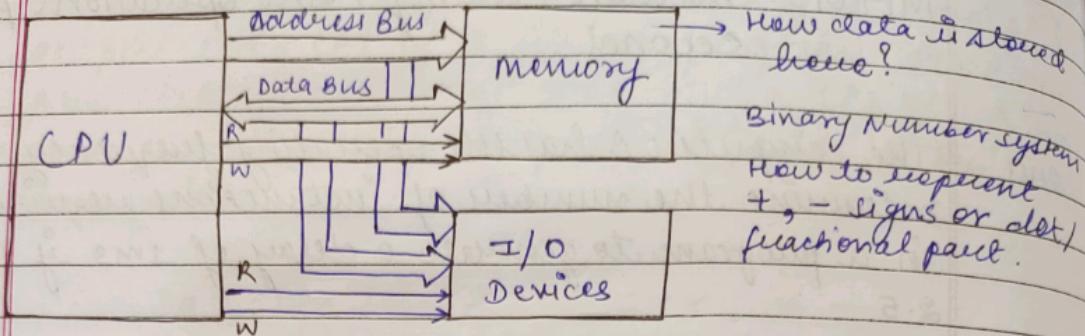
$$1 \times 10^{-3} = N \times \frac{2.5}{10^6} \times 0.25 \times 10^{-6}$$

$$\frac{1 \times 10^6}{2.5 \times 25} = N$$

$$\frac{1000 \times 1000}{25 \times 25}$$

$$1600 = N$$

Data Representation



Sign Magnitude representation of signed binary numbers.

$\underbrace{11}_{\text{sign bit}} \underbrace{010}_{\text{magnitude}}$ -10

This representation is not used in computers because:

- ① Zero has two representations
- ② Design of arithmetic circuit is difficult.

One's Complement representation of signed Numbers.

$b_{n-1} b_{n-2} b_{n-3} \dots b_0$.
if sign bit = 0, magnitude is determined by $b_{n-2} b_{n-3} \dots b_0$

if sign bit = 1, take one's complement or inversion of
 $b_{n-2} b_{n-3} \dots b_0$.

-18 Representation

$$\begin{array}{r} +18 \\ -18 \end{array} \quad \begin{array}{r} 010010 \\ \cancel{1000} 101101 \end{array}$$

This is also not used in computers because.

$$\begin{array}{l} 1111 \rightarrow 0000 \\ 0000 \rightarrow 0000 \end{array} \quad \begin{array}{l} \text{Two representations of zero.} \\ \hline \end{array}$$

- * positive numbers are represented as in case of sign-magnitude.

- * Negative Numbers.

Two's Complement Representation.

-18
first we'll write +18.

$$\begin{array}{r} 010010 \\ 1's C \ 101101 \\ \hline + \ 1 \\ \hline 101110 \end{array}$$

arithmetic operations becomes much much ~~work~~ easier & hence used in computers.

$$\begin{array}{r} 1000 \\ * \end{array} \xrightarrow{1's C} \begin{array}{r} 0111 \\ +1 \end{array} \xrightarrow{*} \begin{array}{r} 1000 \\ -8 \end{array}$$

Min of Max value to represent in 2's C

$$-2^{n-1} \text{ to } +2^{n-1} - 1$$

$$\begin{array}{r} 1010 \\ -8421 \\ \hline = -6 \end{array}$$

PAGE No. _____
DATE _____

$$\begin{array}{r} 11010001 \\ 128 64 32 16 8 4 2 1 \\ +1 \end{array}$$

$$-128 + 64 + 16 + 1$$

$$-128 + 81$$

$$-47$$

$$\begin{array}{r} 00101110 \\ 32 16 8 4 2 1 \\ +1 \end{array}$$

$$32 + 8 + 4 + 2 + 1$$

$$32 + 8 + 4 + 2 + 1 = -47.$$

$b_{n-1} b_{n-2} \dots b_0$,
 negative weight positive weight.

Data Representation

Integer representation

(a) Unsigned Integer representation : using binary bits in bits binary representation

$$a_{n-1} a_{n-2} \dots a_1 a_0.$$

$$A = \sum_{i=0}^{n-1} 2^i a_i.$$

1010

$$A = \sum_{i=0}^3 2^i a_i = 2^0 a_0 + 2^1 a_1 + 2^2 a_2 + 2^3 a_3.$$

(b) Signed Integer representation :

* MSB is referred as sign bit.

sign bit = 1 -ve

sign bit = 0 +ve.

$$\begin{array}{r} -128 + 16 \\ -112 \end{array}$$

PAGE NO.	
DATE	

Sign Magnitude Representation.

$$A = \sum_{i=0}^{n-2} 2^i a_i \rightarrow [a_{n-1} = 0] \quad \text{sign bit.}$$

$$A = -\sum_{i=0}^{n-2} 2^i a_i \rightarrow [a_{n-1} = 1]$$

Two's Complement Representation.

$$-2^{n-1} \text{ to } 2^{n-1} - 1.$$

$$* A = -2^{n-1} a_{n-1} + \sum_{i=0}^{n-2} 2^i a_i.$$

$$a_{n-1} = \text{sign bit} = 0.$$

$$\begin{array}{r} 10010000 \\ (-2^7) \quad 2^4 \end{array}$$

$$-128 + 16 = -112.$$

Convert an n -bit 2's complement representation to m bit representation where $m > n$.

$$1011 \rightarrow 4 \text{ bit}$$

$$11111011 \rightarrow 8 \text{ bit. sign extension.}$$

In sign extension we can copy the sign bit towards the left hand side (towards \oplus ing $m8$) in order to represent an n bit number to m bit number where $m > n$.

Integer Arithmetic.

PAGE NO.
DATE

1. Negation

$$\begin{array}{rcl} +7 & \rightarrow & 0111 \\ -7 & \rightarrow & 1000 \\ & & \downarrow \\ & & +1 \\ & & 1001 \end{array}$$

Takes 2's complement

2. Adding signed integers in 2's complement form

$$\begin{array}{rcl} 1001 & \rightarrow -7 & 1001 \rightarrow -7 \\ +0101 & \rightarrow +5 & +1111 \rightarrow -1 \\ 1110 & \rightarrow -2. & \boxed{111000} \\ & & \text{Result.} \\ & & \text{discard the carry.} \end{array}$$

$$\begin{array}{rcl} 0111 & \rightarrow +7 \\ 0010 & \rightarrow +2 \\ 1001 & \end{array}$$

overflow condition in 2's complement

* Overflow rule : If the two numbers are added and both are either positive or negative but the result has opposite sign to the number, then it is a overflow condn.

If carry in the sign bit is not equal to the carry out of the sign bit, then overflow has occurred.

3. Subtraction of signed integer in two's complement representation.

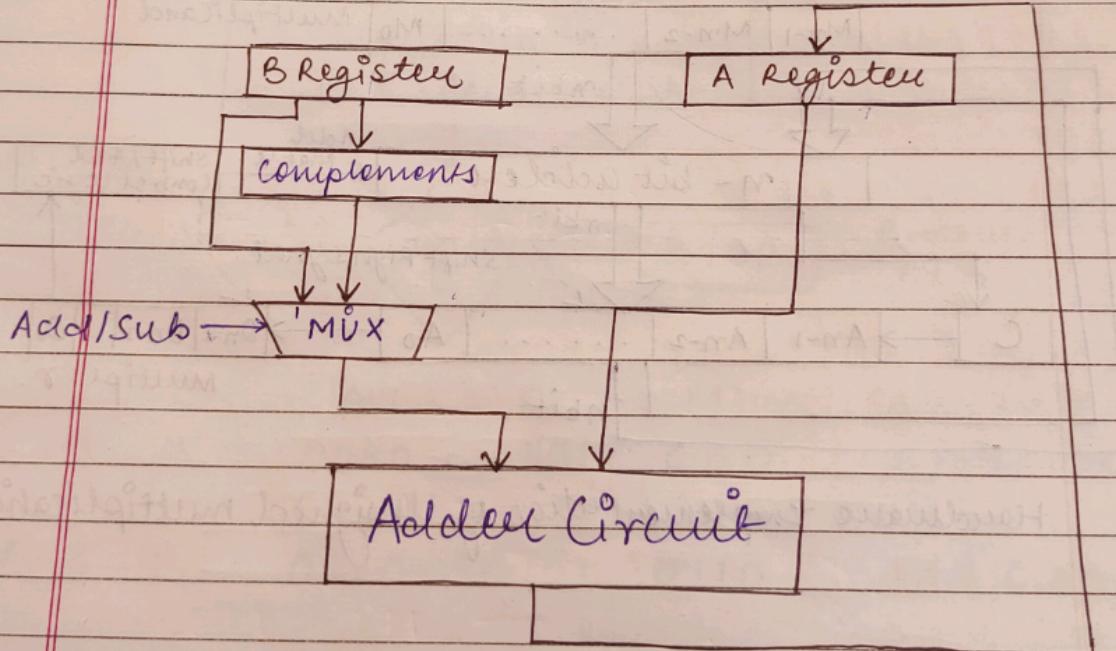
$$A - B = A + (-B)$$

Minuend Subtrahend.

$$A = +7 \quad 0\ 111$$

$$B = -1 \quad \text{Now } 1\ 111 \xrightarrow{\text{1's C}} 0000 \xrightarrow{\text{2's C}} 0001$$

$$\begin{array}{r}
 0\ 111 \rightarrow +7 \\
 + 0\ 001 \rightarrow +1 \\
 \hline
 1\ 000 \rightarrow -8 \rightarrow \text{overflow.}
 \end{array}$$



Block diagram of Addn & subn of signed integer in 2s complement representation.

if multiplier bit is 1 then the result of product is multiplicand only hence only add n else shifting is required.

PAGE No.	
DATE	

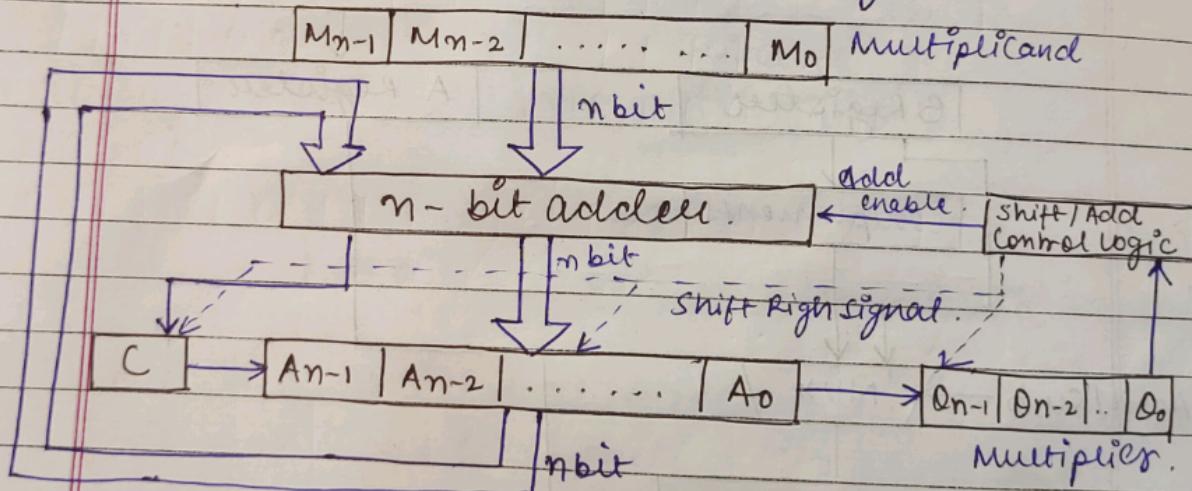
Multiplication Of Unsigned Integer

$$\begin{array}{r} 0110 \text{ (Multiplicand)} \\ \times 0101 \text{ (Multiplier)} \\ \hline 0110 \\ 0000 \\ 0110 \\ 0000 \\ \hline 0011110 \end{array} \rightarrow n \quad \rightarrow n$$

Partial Products.

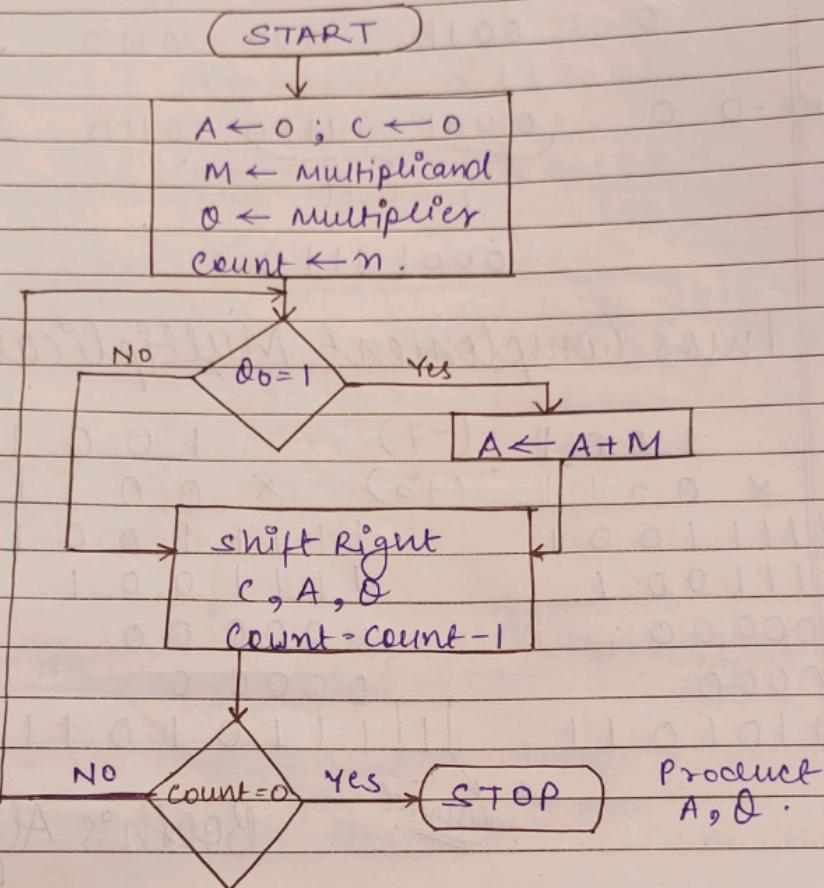
$\rightarrow 2n \rightarrow$ longest possible
(include carry).

Running Addition of Partial Product : addition of Partial Product while calculating it.



Handmade Implementation of Unsigned multiplication.

Flowchart to understand mechanism.



C A Q M comment.
 count = 4 0 0000 0101 0110 Initial value

count = 3 0 0110 0101 0110 Add C,A ← A+M

0 0011 0010 0110 shift right.

count = 2 0 0001 1001 0110 shift right

count = 1 0 0111 1001 0110 Add, C,A ← A+M

C A O M Commence

0 0011 1100 0110 sign right

Count = 0 0 0001 1110 0110 sign right

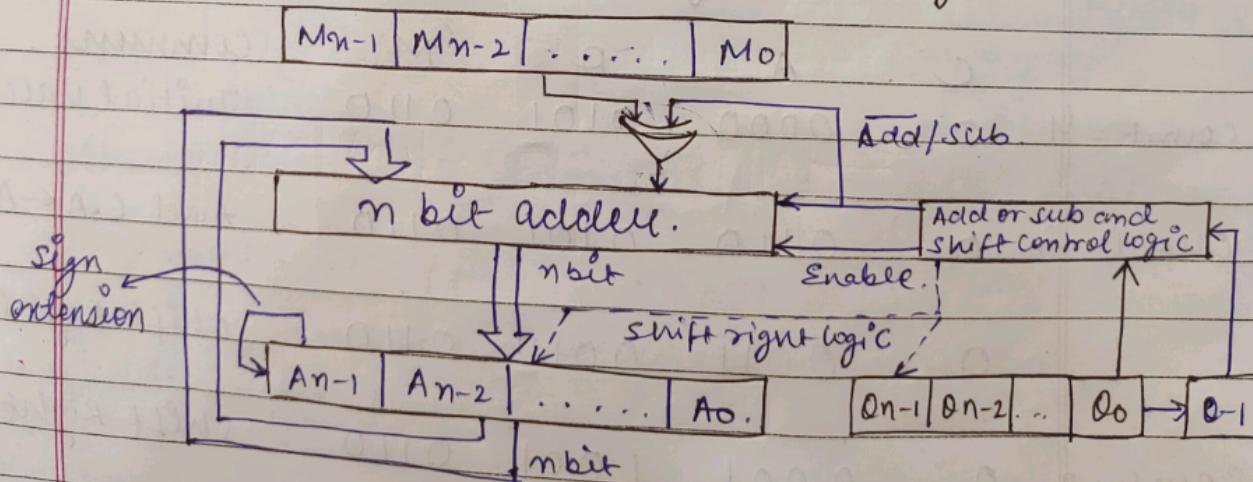
Product :
0001 1110

Two's Complement Multiplication

$$\begin{array}{r}
 1001 \quad (-7) \\
 \times 0011 \quad (+3) \\
 \hline
 11111001 \\
 11111001 \\
 \hline
 00000000 \\
 00000000 \\
 01011101011
 \end{array}
 \quad
 \begin{array}{r}
 1001 \\
 \times 0011 \\
 \hline
 11111001 \\
 11111001 \\
 \hline
 00000000 \\
 00000000 \\
 00000
 \end{array}
 \rightarrow \text{sign extension}$$

discard

Booth's Algorithm



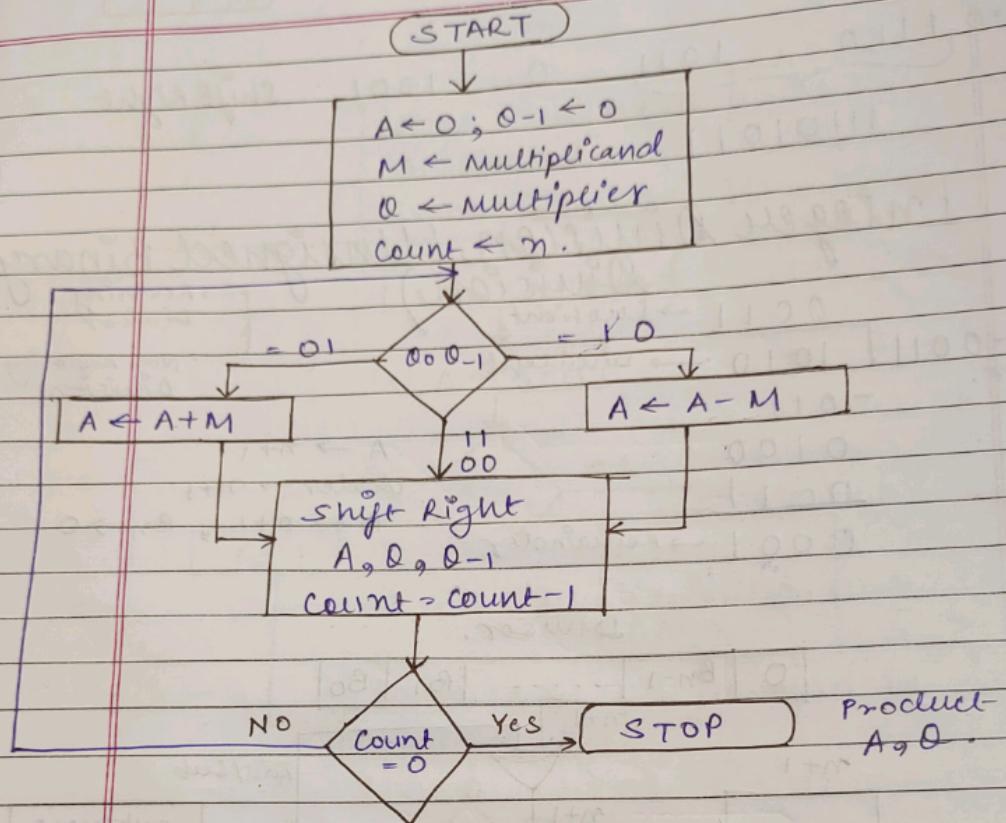
Comments

sign right
shift right

n

sign extension

m



A O O-1 M Comments.

Count=4 0000 0011 0 1001 Initial values

Count=3 0111 0011 0 1001 $A \leftarrow A - M$

0011 1001 1 1001 Shift Right

Count=2 0001 1100 1 1001 Shift Right

Count=1 1010 1100 1 1001 $A \leftarrow A + M$

1101 0110 0 1001 Shift Right

Count = 0 1110 1011 0 1001 shift right
 11101011

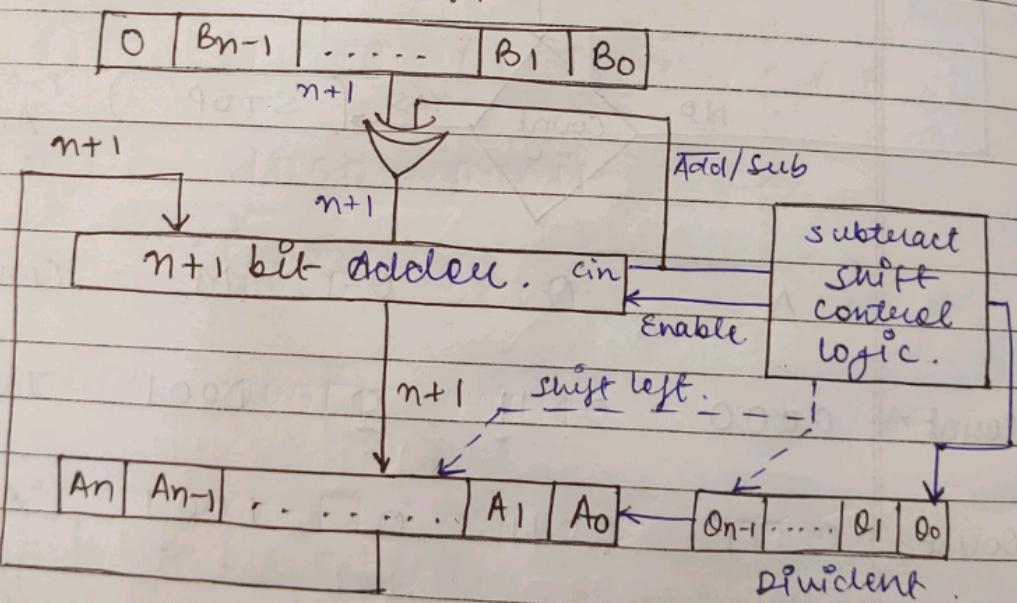
Integer Division (Unsigned Binary Division)

$$\begin{array}{r}
 \text{Quotient} \\
 \text{Divisor.} \quad 0011 \quad | \quad 1010 \quad \text{Dividend} \\
 -011 \\
 \hline
 0100 \\
 0011 \\
 \hline
 0001 \quad \text{Remainder}
 \end{array}$$

→ Restoring Division
 → Non Restoring Division

$A \rightarrow n+1$
 $\text{divider} \rightarrow n+1$
 $B \rightarrow n+1, B_n \rightarrow 0$

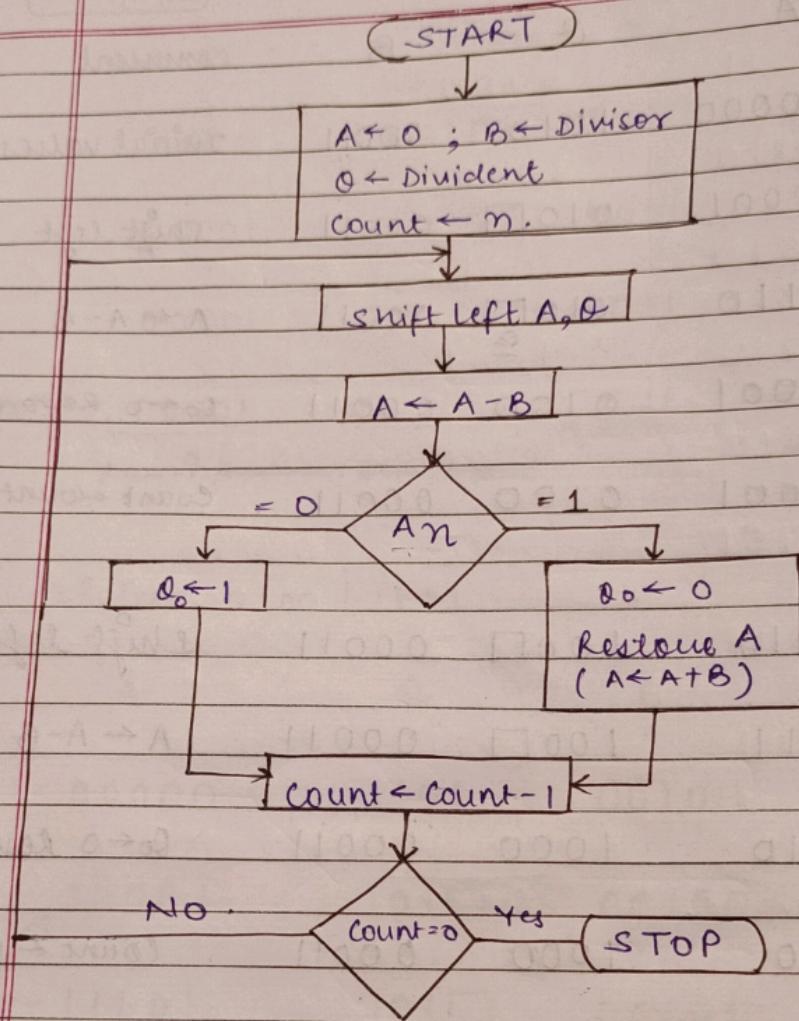
Divisor.



Hardware Implementation of Unsigned Binary Division.

right

Binary
Division
Restoring
Non-restoring



$$\begin{array}{r}
 B = 0011 \\
 -B = 1110 \\
 \hline
 -B = 11101
 \end{array}$$

Discard carry
on addition
in 2's C

PAGE NO.	
DATE	

$$\begin{array}{l}
 n=4 \\
 \text{Divisor} = 0011 \\
 \text{Dividend} = 1010
 \end{array}$$

A O -B

comment

00000 1010 00011

Initial values

00001 010□ 00011

shift left

11110 010□ 00011

$A \leftarrow A - B$

00001 0100 00011

$Oo \leftarrow 0$ Restore A.

count = B

Break

Count \leftarrow Count - 1

00010 100□ 00011

shift left

11111 100□ 00011

$A \leftarrow A - B$

00010 1000 00011

$Oo \leftarrow 0$ Restore A.

Count = 2

Break.

Count \leftarrow Count - 1.

00101 000□ 00011

shift left

00010 000□ 00011

$A \leftarrow A - B$

00010 0001 00011

$Oo \leftarrow 1$

Count = 1

00010

Count \leftarrow Count - 1.

Break.

comment .

A	O	B	
00100	001 <input type="checkbox"/>	00011	shift left .
<u>00001</u>	001 <input type="checkbox"/>	00011	$A \leftarrow A - B$
00001	0011	0011	$Q_0 \leftarrow 1$
count = 0	<u>00001</u>	<u>0011</u>	count \leftarrow count - 1
remainder quotient .		$B = 00100$	
		$-B = 11011$	
		11100	

$4 \overline{) 13}$

3	Divisor 0100	1101	
4	A	O	comment .
12	00000	1101	00100 initial values
1	00001	101 <input type="checkbox"/>	00100 shift left

A	O	B	comment .	
11101	101 <input type="checkbox"/>	00100	$A \leftarrow A - B$	
00001	101 <input checked="" type="checkbox"/>	00100	$Q_0 \leftarrow 0$ restore A	
Count = 3	00001	1010	00100	Count \leftarrow count - 1

$\cancel{0} 00011$ 010 00100 Shift left

$\cancel{11111}$ 010 00100 $A \leftarrow A - B$

00011 0100 00100 $Q_0 \leftarrow 0$ restore A

$$-B = 11100$$

~~11100
00100
00010~~

11100
00100
00000

PAGE No.
DATE

Count=2 00011

0100

00100

Count=Count-1

00110

100□

00100

shift left

An=0 00010

100□

00100

A ← A-B

00010

1001

00100

Q0 ← 1

Count=1 00010

1001

00100

Count=Count-1

00101

001□

00100

shift left.

An=0 00001

001□

00100

A ← A-B

00001

0011

00100

Q0 ← 1

Count=0 00001
Remainder,

0011
Quotient

00100

Count=Count-1

Flowchart for Non-Restoring

PAGE No. _____
DATE _____

count = count - 1
shift left

$\leftarrow A - B$

$B < 1$

$1 \leftarrow \text{count} - 1$

left.

B

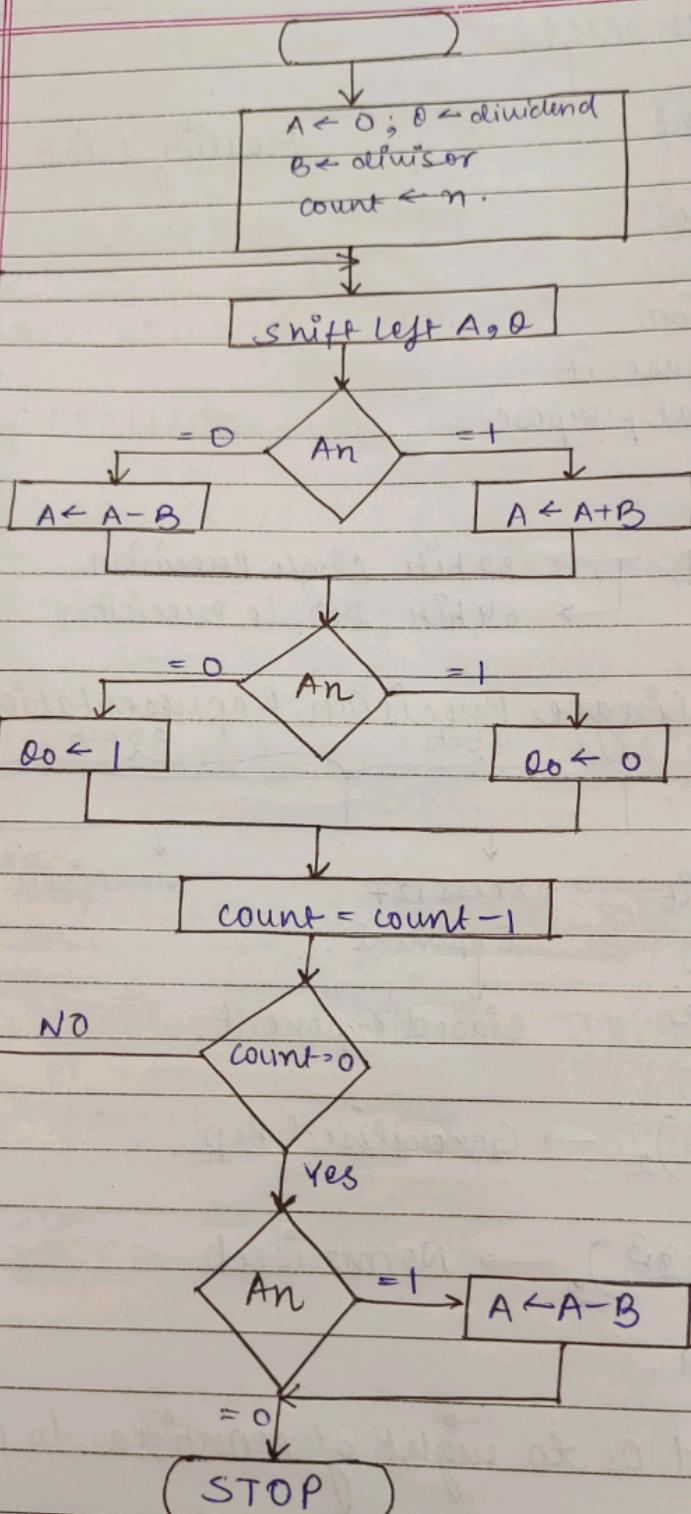
1

count - 1

NO

Yes

STOP



Real Numbers.

Fixed Point

→ Drawbacks

→ Range

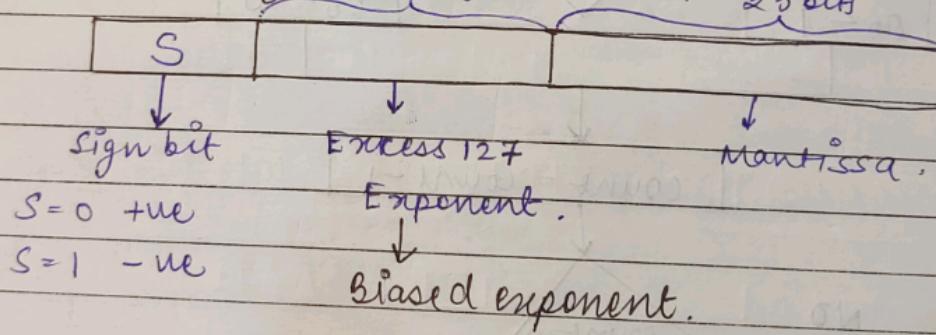
→ Precision

we do not use it
for general purpose
computer.

Floating Point.

IEEE 754. → 32 bits Single Precision
→ 64 bit Double precision

32 Bits Single Precision Representation.



$(1101.001)_2 \rightarrow$ Generalised Rep

$(1.\underbrace{101001}_\text{Mantissa} \times 2^3)_2 \rightarrow$ Normalised

Mantissa.

We append 0s to right of mantissa to make it
23 bits.

To avoid the use of signed representation of exponents we use excess 127 concept to represent positive & negative values.

Represent 625.125 in 32 bit single precision.

1100000000000000000000000000

18	625
	312

$$(1001110001.001)_2.$$

$$(1.001110001001 \times 2^9)_2$$

$s = 0$

$$\begin{array}{r} 127 \\ 9 \\ \hline 136 \end{array}$$

$$(10001000)_2, \quad \text{Excess 127 bits exponent.}$$

Mantissa.

00111000100100... 0
 ↓
 23 bits.

31	30	23 Q2	0
0	10001000	00111000100100... 0	0

Sign bit

Excess 127 exponent

Represent -3.625 .

$$2 \mid 3 \\ 1 \mid 1$$

$$(11.101)_2 \quad (1.1101 \times 2^1)_2$$

$$S = 1$$

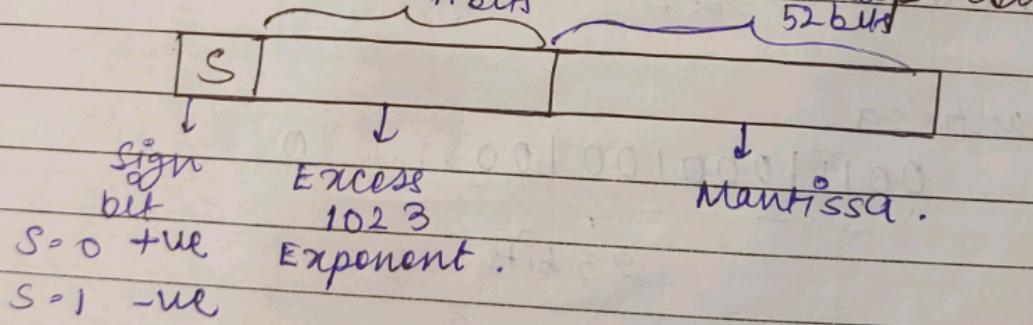
$$127 + 1 = 128$$

$$100000000$$

$$1101000\dots 0$$

$$[1 | 10000000 | 1101000\dots 0]$$

64 Bits Double Precision Representation.



$$0.125 + 0.0625 \quad .0011$$

$(-307.1875)_{10}$ in single or double precision.

$(0.0625)_{10}$ " " " "

$$\begin{array}{r} 307 \\ 153 \\ 76 \\ 38 \\ 19 \\ 9 \\ 4 \\ 2 \\ 1 \end{array} \Big| \begin{array}{r} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array}$$

significand \rightarrow Mantissa.

$$(0.0001)_2$$

$$127 - 4 = 123$$

$$(1.0 \times 2^4)_2.$$

$$\boxed{10|0111111|0000\ldots0}$$

$$1023 - 4 = 1019.$$

(Add)

Addition/Subtraction Of Floating Point Number.

	S	Biased Exp	Mantissa
X =	0	01101101	1101011010...0
Y =	0	01101100	010101000...0

$Z = X + Y \rightarrow$ Step 1: We will check whether any of them is zero.

$Z \leftarrow$ Non zero number (if $Y \neq 0$; $Z \leftarrow X$)

Step 2: Compare exponent.

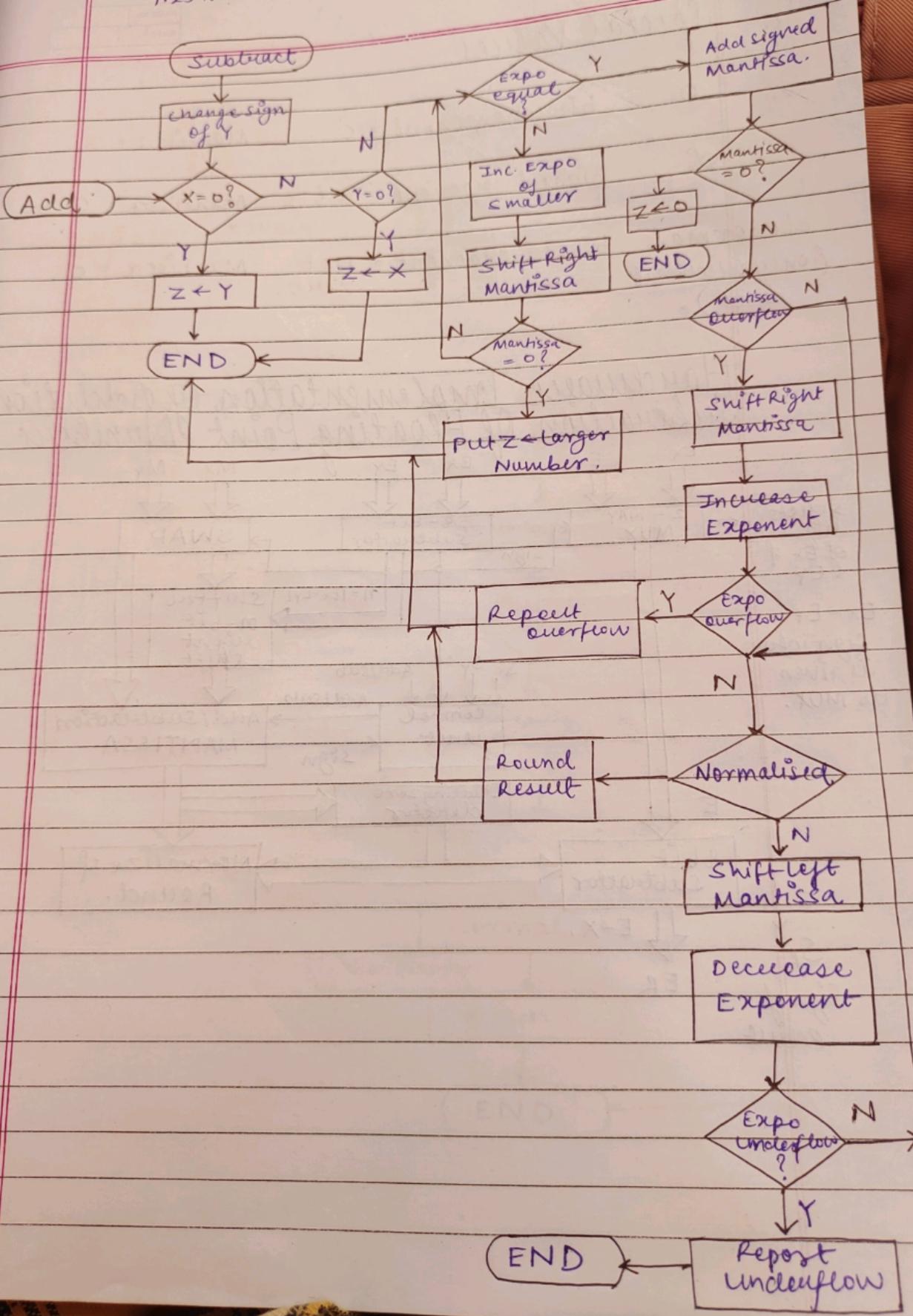
Increase exponent of smaller no.
shift eight significand / Mantissa.

Step 3: add the significand.

Step 4: Normalise.

12.5×10^5) slight right.
 1.25×10^6

Mantissa overflow
 after clipping mantissa
 23 bits
 PAGE NO. _____ DATE _____ of 24 or
 more bits.



Special Values.

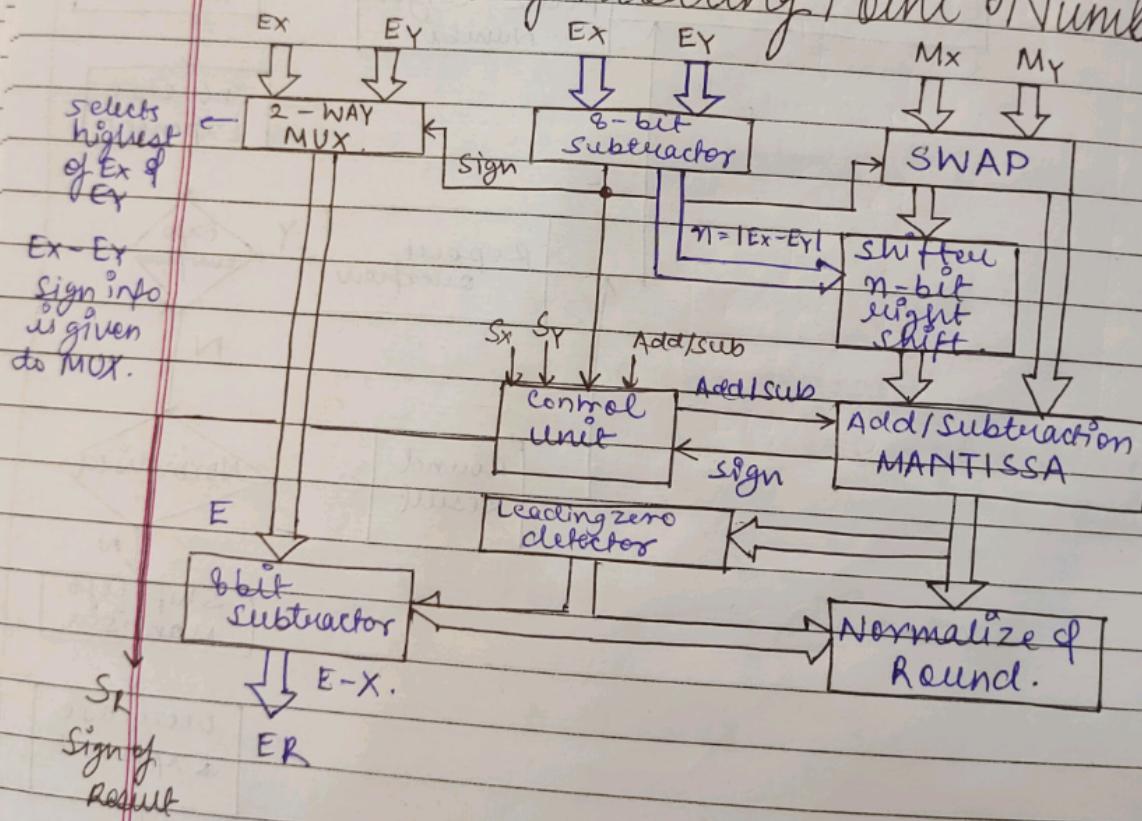
0 biased exponent = 0

Mantissa = 0

∞ biased exponent = all 1's Mantissa = 0

denormal biased exponent = Not zero. Mantissa = 0
(only fractional part).

Hardware Implementation Of Addition Subtraction Of Floating Point Numbers



$S_x \text{ } E_x \text{ } M_x$
 $X \text{ } S_y \text{ } E_y \text{ } M_y$
 \downarrow
 $E_x + E_y$
 $M_x \times M_y \rightarrow 48 \text{ bits}$
 $24 \text{ bits} \quad 24 \text{ bits}$
 \downarrow only 23 bits can be stored.
Rounding
Normalisation.

