DATE / /
PAGE NO.

- Difference in Computer Organization and Computer Architecture

- Computer Architecture
  - computer architecture refers to those attributes of a system visible to a programmer or, put another way, those ~~the~~ attributes that have a direct impact on the logical execution of a program
  - designing of computer
  - Designer point of view
  - Ex of arch. attributes include the instruction set, the no. of bits used to represent various data types

- Computer Organization
  - C.O. refers to the operational units and their interconnections that realise the architectural specifications
  - Logic and Circuits
  - Programmer pt. of view
  - Ex O. attributes include those hardware details transparent to the programmer, such a signals, interfaces between the computer and peripherals.

- ~~XX~~ THE MAIN COMPONENTS OF COMPUTER

- What is Computer?

→ A computer is a complex system; contemporary computer contains millions of elementary electronic components

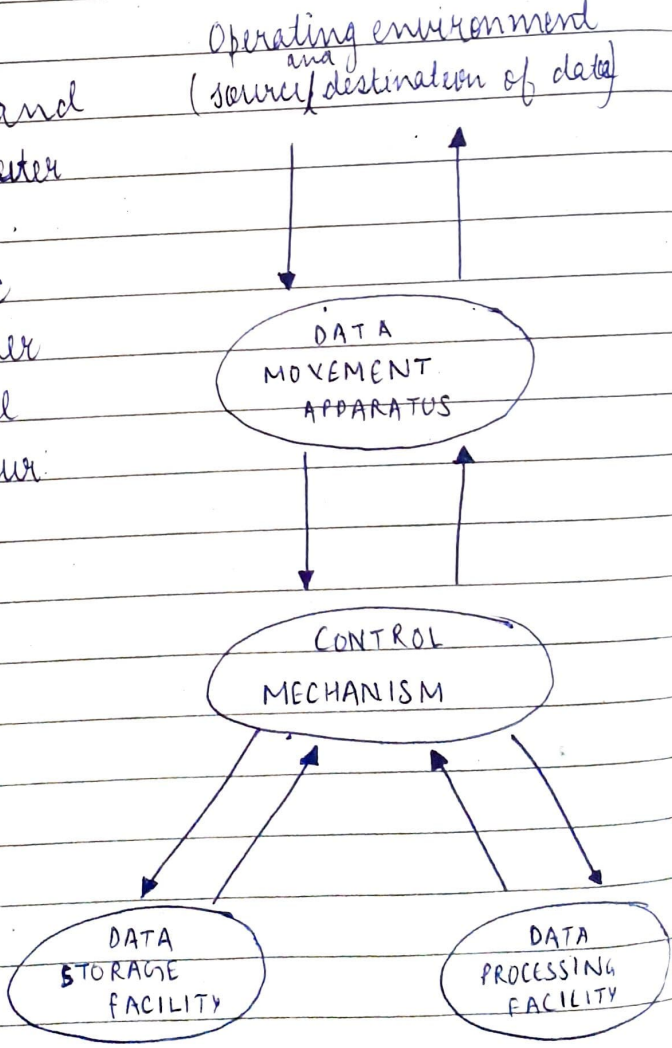
Due to complexity of computer contains millions of elementary electronic ~~cesst~~ components.

DATE / /
PAGE NO.

- Due to complexity of computer, the key is to recognize the hierachical nature of most complex systems including computer.
- The designer is concerned with structure and function.
  - STRUCTURE - The way in which the components are interrelated
  - FUNCTION - The operation of each individual component as part of the structure

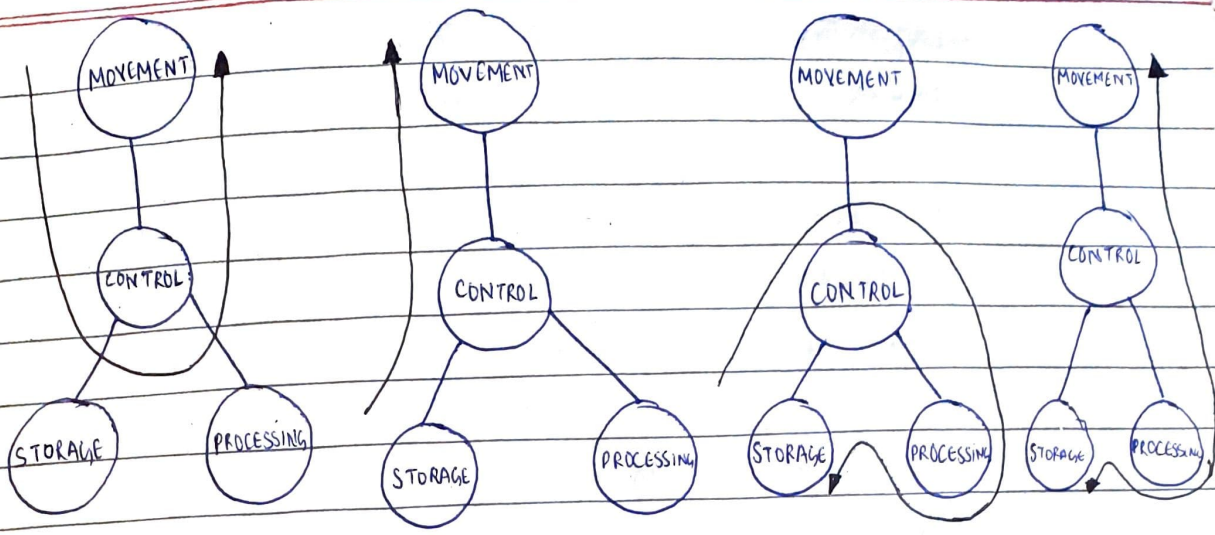
FUNCTIONAL DESCRIPTION

Functional
Both the structured and functional of a computer are, in essence, simple.
Figure depicts the basic function that a computer can perform. In general terms they are only four
- Data processing
- Data storage
- Data movement
- Control



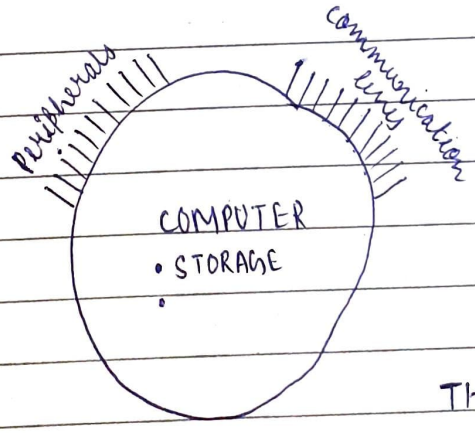
A FUNCTIONAL VIEW OF THE COMPUTER

DATE / /
PAGE NO.



POSSIBLE COMPUTER OPERATION

- STRUCTURE DESCRIPTION



THE COMPUTER.

- There are four main structured components:
  - central Processing unit (CPU): control the operation of the computer and perform its data processing functions; ~~its~~ d often simply referred to as processor.
  - Main memory - stores data
  - I/O - Moves data between the computer and its external environment
  - System interconnection: some mechanism that provides for communication among CPU, main memory, and I/O. A common example of system interconnection is by means of a system bus, consisting of a no of

DATE. / /
PAGE NO.

conducting wires to which all the other components attach.

- Its major structural components are as follows:
- Control unit - Controls the operation of the CPU and hence the computer
- Arithmetic and logic unit (ALU): Performs the computer's data processing function.
- Registers - Provides storage internal to the CPU.
- CPU interconnections - Some mechanism that provides for communication among the control unit, ALU and registers

• Historical Development: First through Fourth Generation computers.

→ Defination of computer.

• Computer is a programmable machine.
• Computer is machine that manipulates data according to a list of instructions
• computer is any device which aids humans in performing various kinds of computations or calculation

→ Three principles characteristic of computer.

• It responds to a specific set of instruction in a well defined manner
• It can execute a pre-recorded list of instruction
• It can quickly store and retrieve large amount of data

DATE / /
PAGE NO.

- Tally Sticks - A tally stick was an ancient memory aid device to record and document numbers, quantities, or even message.

- Abacus
- An abacus is a mechanical device used to aid an individual in performing mathematical calculations
- The abacus was invented in Babylonia in 2400 B.C
- The abacus in the form we are most familiar with was first used in China in around 500 B.C
- It used to perform basic arithmetic operations

- Napier's Bones
- Invented by John Napier in 1614.
- Allowed the operator to multiply, divide and calculate square and cube roots by moving the rods around and placing them in specially constructed boards

- Slide Rule
- Invented by William Oughtred in 1622
- Is based on Napier's ideas about ~~ab~~ logarithms
- Used primarily for
  - multiplication
  - division
- Not normally used for addition or subtractions

- Pascaline
- invented by Blaise Pascal in 1642
- It was its limitation to addition + and subtraction
- It is too expensive

- Stepped Reckoner
  - Inventage by Gottfried Wilhelm Leibniz in 1672
  - The machine that can add, subtract, multiply and divide automatically

- Jacquard Loom
  - The Jacquard loom is a mechanical loom, invented by Joseph-Marie Jacquard in 1881
  - It an automatic loom controlled by punched cords

- Arithmometer
  - A mechanical calculator invented by Thomas de Colmar in 1820,
  - The first reliable, useful and commericially successful calculating machine.
  - The machine could perform the 4 basic mathematic function.
  - The first mass-produced calculating machine

- Difference Engine and Analytical Engine
  - It an automatic, mechanical calculator designed to tabulate polynomial functions.
  - Invented by Charles Babbage in 1822 and 1834.
  - It is the first mechanical computer.

- First Computer Programmer
  - In 1840, Augusta Ada Byron suggests to Babbage that he use the binary system
  - She writes programs for the Analytical Engine.

DATE / /
PAGE NO.

- Scheutzian calculation Engine
  - Invented by per Georg Scheutz in 1843.
  - Based on on charles Babbage's different engine
  - The first printing calculator

- Tabulating Machine
  - Invented by Herman Hollerith in 1890.
  - To assist in summarizing information and accounting

- Havard Mark 1
  - Also known as IBM Automatic Sequence Controlled Calculator (ASCC)
  - Invented by Howard H. Aiken in 1943.
  - The first electro-mechanical computer

- Z1
  - The first programmable computer
  - ~~Great~~ created by Konrad Zuse in Germany from 1936 to 1938.
  - To program the Z1 required that user insert punch tape reader and all output was also generated through punch tape.

- Atanasoff-Berry computer (ABC)
  - It was the first electronic digital computing device
  - Invented by Professor John Atanasoff and graduate student Clifford Berry at Iowa State University between 1939 and 1942

- ENIAC
- ENIAC stands for Electronic Numerical Integrator and Computer.
- It was the first electronic general-purpose computer
- It completed in 1946
- Developed by John Presper Eckert and John W. Mauchl.

- UNIVAC 1
- The UNIVAC 1 (Universal Automatic computer 1) was the first comercial computer.
- Designed by J. Presper Eckert and John Mauchly

- EDVAC
- EDVAC stands for Electronic Discrete Variable Automatic Computer.
- The First stored program Computer.
- Designed by Von Neumann in 1952
- It has a memory to hold both a stored program as well as data

- The First Portable computer.
- Osborne 1 - the first portable computer
- Released in 1981 by the Osborne computer corporation

- The First computer company.
- The first computer company was the Electronic controls company
- Founded in 1949 by J. Presper Eckert and John Mauchly

DATE / /
PAGE NO.

Computer Generations

There are five generation of computer

- First Generation → 1946 - 1958
- Second Generation → 1959 - 1964
- Third Generation → 1965 - 1970
- Fourth generation → 1971 - today
- Fifth generation → today to future.

MOORE'S LAW

- Figure reflects the famous Moore's law, which was propounded by Gordon Moore, cofounder of Intel, in 1965. Moore observed that the number of transistor that could be put on a single chip was doubling every year and correctly predicted that this place would continue into the near future. To the surprise of many, including Moore, the pace continued year after year and decade after decade. The pace slowed to doubling every 18 months in the 1970s but has sustained that rate ever since.

- The consequences of Moore's law are profound:

1. The cost of a chip has remained virtually unchanged during this period of rapid growth in density. This means that the cost of computer logic and memory circuitry has fallen at a dramatic rate.
2. Because logic and memory element are placed closer together on more densely packed chips, the electrical path length is shortened increasing operating speed.
3. The computer becomes smaller, making it more convenient to place in a variety of environments.

DATE / /
PAGE NO.

4. There is a reduction in power and cooling requirements.
5. The interconnections on the integrated circuit are much more reliable than solder connection. With more circuitry on each chip, there are fewer interchip connections.

• THE VON NEUMANN AND NON-VON NEUMANN

→ John W. Mauchly and J. Presper Eckert conceived of an easier ~~to~~ way to change the behavior of their calculating machine. They reckoned that memory device, in the form of ~~me~~ mercury delay lines, could provide a way to store program instructions. This would forever end the tedium of rewiring the system each time it had a new problem to solve or an old one to debug. Mauchly and Eckert documented their idea, proposing it as the foundation for their next computer, the EDVAC.

→ One of these people was a famous Hungarian mathematician named John von Neumann (pronounced von noy-man). After reading Mauchly and Eckert's proposal for the EDVAC, von Neumann published and publicized the idea. So effective was he in the delivery of this concept that history has credited him with its invention. All stored-program computer have come to be known as ~~non~~ von Neumann system using the von Neumann system using the von Neumann architecture.

DATE / /
PAGE NO.

- Today's version of the stored-program machine architecture satisfies at least the following characteristics.

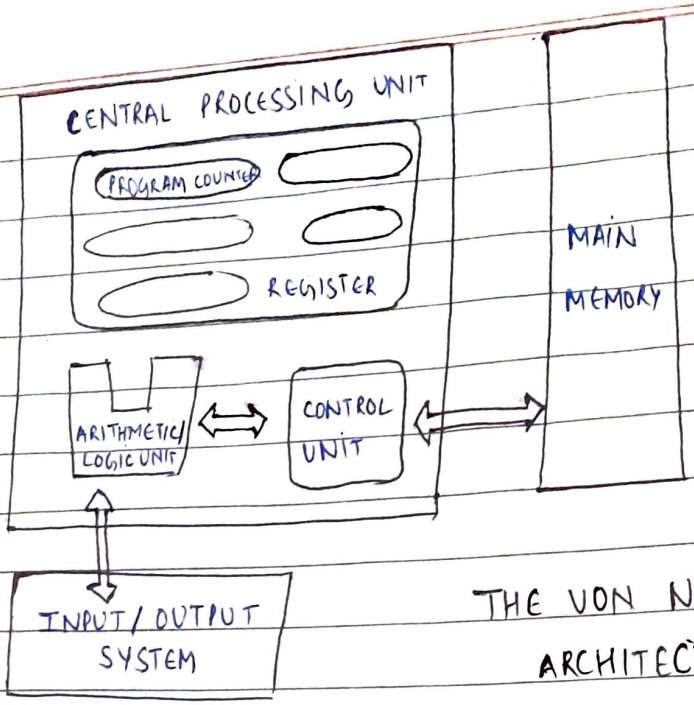
→ consists of three hardware systems: A central processing unit (CPU) with a control unit, an arithmetic logic unit (ALU), registers (small storage areas), and a program counter, a main-memory system, which holds program that control the computer's operation; and an I/O system.

→ capacity to carry out sequential instruction processing

→ contains a single path, either physically or logically, b/w the main memory system and the control unit of the CPU, forcing ~~alternating~~ alternation of instruction and execution cycles. This single path is often referred to as the von Neumann bottleneck.

- Von-Neumann architecture runs programs in what is known as the von Neumann execution cycle (also called the fetch-decode-execute cycle), which describe how the machine works. One iteration of the cycle is as follows:

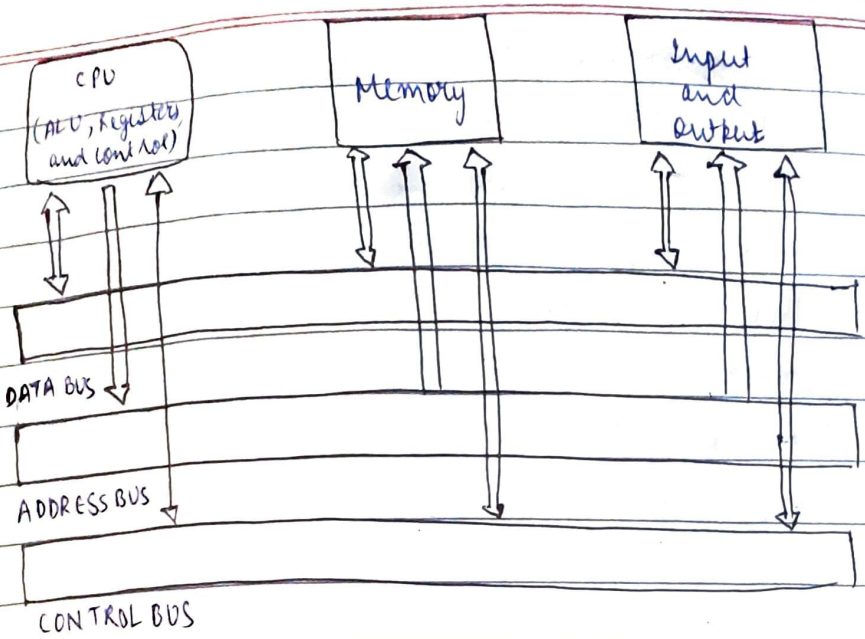
1. The control unit fetches the ~~#~~ next program instruction from the memory, using the program counter to determine where the instruction is located.
2. The instruction is decoded into a language the ALU can understand.
3. Any day operands required to execute the instruction ~~from the~~ are fetched from memory and placed into registers within the CPU.
4. The ALU executes the instruction and place the results in registers or memory



THE VON NEUMANN ARCHITECTURE

- MODIFIED VON NEUMANN ARCHITECTURE
  - The idea present in the von Neumann architecture have been extended so that program and data stored in a slow-to-access storage medium, such as a hard disk, can be copied to a fast access, volatile storage medium such as RAM prior to execution.
  - This architecture has also been streamlined into ~~the~~ what is currently called the system bus model, which is shown in figure.
    - → The data bus moves data from main memory to the CPU registers
    - → The address bus holds the address of the data that the data bus is currently accessing
    - → The control bus carries the necessary control signals that specify how the information transfer is to take place.

DATE / /
PAGE NO.



THE MODIFIED VON ARCHITECTURE, ADDING A SYSTEM BUS

- NON-VON NEUMANN MODEL
  - other then Von-Neumann architecture
  - Ex. including neutral networks, genetic algorithms, quantum computation, and parallel computers, of these, parallel computing is currently the most popular
  - Arise concept of parallel computing.
  - If a computer isn't fast enough, instead of trying to develop a faster, more powerful computer, why not simply use multiple computer?

- The Evolution of the Intel x86 Architecture
  It is worthwhile to list some of the highlight of the evolution of the Intel product line
  - 8080: The world's first general-purpose microprocessor. This was an 8-bit machine, with an 8-bit data path to memory. The 8080 was used in the first personal computer, the Altair.
  - 8086: A far more powerful, 16-bit machine. In addition to a wider data path and large registers, the 8086 sported an instruction cache, or queue, that prefetches a few instruction before they are executed. A variant of this processor, the 8088, was used in IBM's first personal

computer, securing the success of Intel. The 8086 is the first appearance of the x86 architecture.

- 80286: This extension of the 8086 enabled addressing a 16-MByte memory instead of just 1 MB.
- 80386: Intel's first 32-bit machine, and a major overhaul of the product. With a 32-bit architecture, the 80386 rivaled the complexity and power of minicomputer and mainframes introduced just a few years earlier. This was a first intel processor to support multitasking meaning it could run multiple programs at the same time.
- 80486 - The 80486 introduced the use of much more sophisticated and powerful cache technology and sophisticated instruction pipelining. The 80486 also offered a built-in math coprocessor, offloading complex math operations from the main CPU.
- Pentium - With the Pentium, Intel introduced the use of ~~much sophisticated~~ superscaler techniques, which allow multiple instruction to execute in parallel.
- Pentium PRO: The pentium Pro continued the move into superscaler organization begun with the Pentium with aggressive use of register renaming, branch prediction, data flow analysis and speculative execution
- Pentium II: The pentium II incorporated Intel MMX technology, which is designed specifically to process video, audio, and graphics data efficiently
- Pentium III: The Pentium III incorporates additional floating-pt instruction to support 3D graphics software
- Pentium 4: The Pentium 4 includes additional floating-pt and other enhancements for multimedia

Date / /
Page No.

- Core : This is the first Intel x86 microprocessor with a dual core, referring to the implementation of 2 processors on a single chip
- Core2 : The core 2 extends the architecture to 64 bits. The core 2 Quad provide 4 processors on a single chip

UNIT - 1 PART 2

Data Representation in Computer Systems :

→ Signed Integer Representation
→ Complement Systems : 1's complement & 2's complement
→ Addition and subtraction using signed nos
→ Multiplication of +ve nos
→ Signed Operand Multiplication
→ Integer Division ;
→ Floating Point Representation
→ The IEEE-754 Floating pt standard, Floating Pt Arithmetic, Floating Pt Errors

SIGNED INTEGER REPRESENTATION

- consider an n-bit vector
  → $B = b_{n-1} \cdots b_1 b_0$ ex - 01 0010 ....
- where $b_i = 0$ or 1 for $0 \le i \le n-1$. This vector can represent an unsigned integer value
- V(B) in the range 0 to $2^n - 1$ where
  → $V(B) = b_{n-1} \times 2^{n-1} + \cdots b_1 \times 2^1 + b_0 2^0$

Eg - if there are 4-bit no: 1010 then it can be represented as $1 \times 2^3 + 0 \times 2^2 \times 1 \times 2^1 + 0 \times 2^0 = (10)_{10}$

- We need to represent both +ve and -ve no. Three system are used for representing such nos
  1. Sign and magnitude 2. 1's complement
  3. 2's complement

DATE / /
PAGE NO.

B	VALUES REPRESENTED		
$b_3\ b_2\ b_1\ b_0$	Sign and magnitude	1's complement	2's complement
0 1 1 1	+7	+7	+7
0 1 1 0	+6	+6	+6
0 1 0 1	+5	+5	+5
0 1 0 0	+4	+4	+4
0 0 1 1	+3	+3	+3
0 0 1 0	+2	+2	+2
0 0 0 1	+1	+1	+1
0 0 0 0	+0	+0	+0
1 0 0 0	-0	-7	-8
1 0 0 1	-1	-6	-7
1 0 1 0	-2	-5	-6
1 0 1 1	-3	-4	-5
1 1 0 0	-4	-3	-4
1 1 0 1	-5	-2	-3
1 1 1 0	-6	-1	-2
1 1 1 1	-7	-0	-1

- Addition and subtraction.

1- (+65) 01000001
+ (+5) 00000101
+70 01000110 → 70

2- +65 01000001
+(-5) 11111011
60 (1)00111100 ⇒ 60

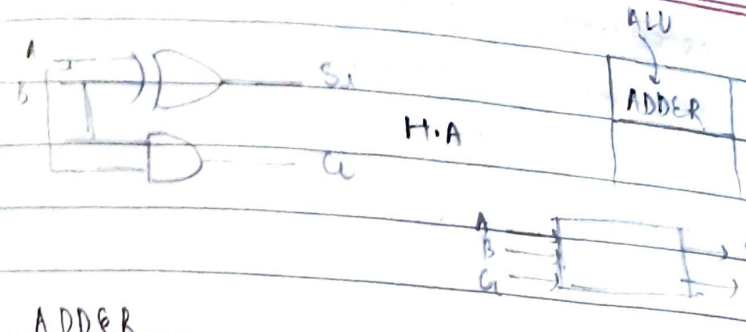
3 (-65) 10111111
+ (+5) 00000101
-60 11000100 -2's complement of 60

4. (-65) 10111111
+(-5) 11111011
-70 (1)10111010

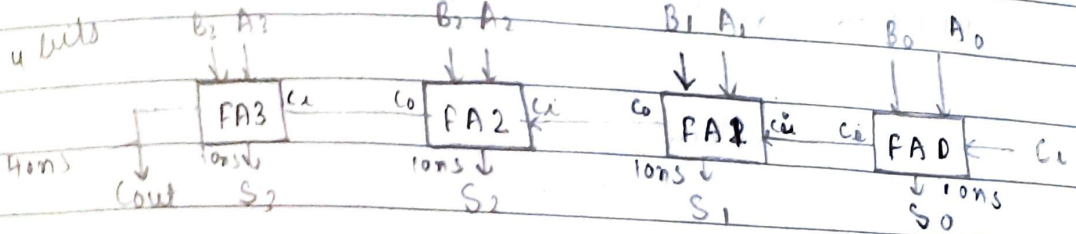
→ -ve → complement + 1 → -ve

DATE / /
PAGE NO.

- ADDER



1. RIPPLE CARRY ADDER



8 - bits - 80ns

16 - bits - 160ns

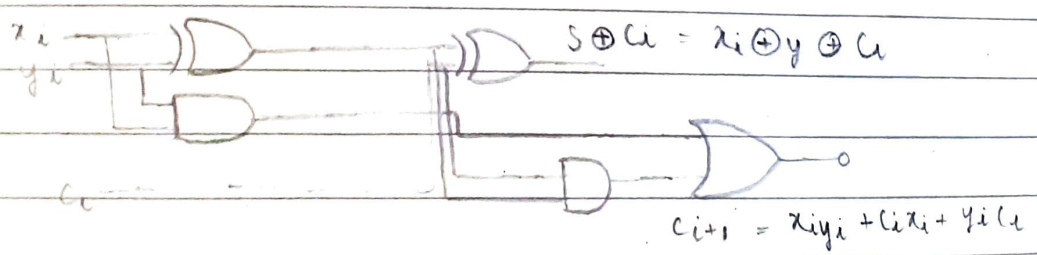
32 - bits - 320ns

64 - bits - 640ns

For 1 output we have to wait for 640ns.

Because of this the speed go slower so thats why we don't use ripple carry adder.

2. CARRY LOOK AHEAD ADDER / CARRY PROPAGATION ADDER



$C_{i+1} = x_i y_i + x_i C_i + y_i C_i$

$= x_i y_i + C_i (x_i + y_i)$

$C_{i+1} = G_i + P_i C_i$ $\quad G_i = x_i y_i, \; P_i = x_i + y_i$

$i = 0$

$C_1 = G_0 + P_0 C_0$

$C_2 = C_1 + P_1 C_1$

$= C_1 + P_1 (C_0 + P_0 C_0)$

$= C_1 + P_0 C_0 + P_1 P_0 C_0$ - By adding this the speed increases this adder is better than CLA

DATE / /
PAGE NO.

- ARITHMETIC OPERATION
- Addition / subtraction
- Multiplication
- Division

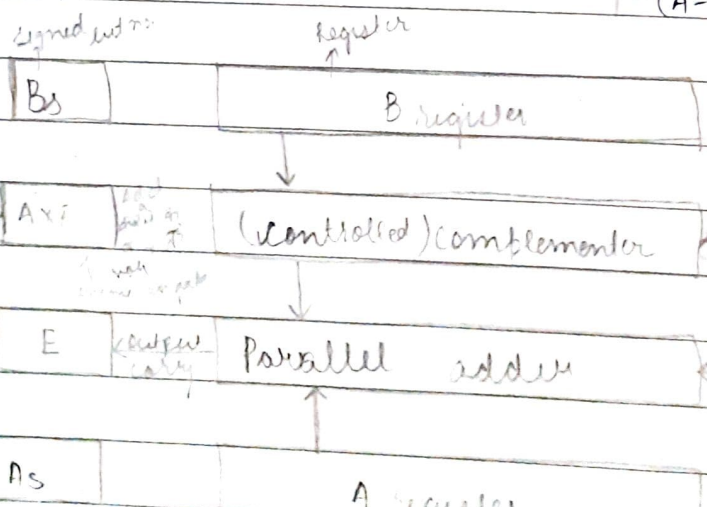
INT
unsigned signed

In this we will see 2 things - Hardware - Algorithm

(1) Addition and subtraction / for subtraction we use 2's C

	Magnitude Addition	Magnitude Subtraction		
		A > B	A < B	A = B
(+A) + (+B)	+ (A + B)			
(+A) + (-B)		+(A-B)	-(B-A)	±(A-B)
(-A) + (+B)		-(A-B)	+(B-A)	±(A-B)
(-A) + (-B)	- (A+B)			
(+A) - (+B)		+(A-B)	-(B-A)	+(A-B)
(+A) - (-B)	+ (A+B)			
(-A) - (+B)	- (A+B)			
(-A) - (-B)		-(A-B)	+(B-A)	+(A-B)

case for addition



* In addition case, AVF = 0 if both symbol is same the overflow condition generate

In subtraction case, AVF = 0

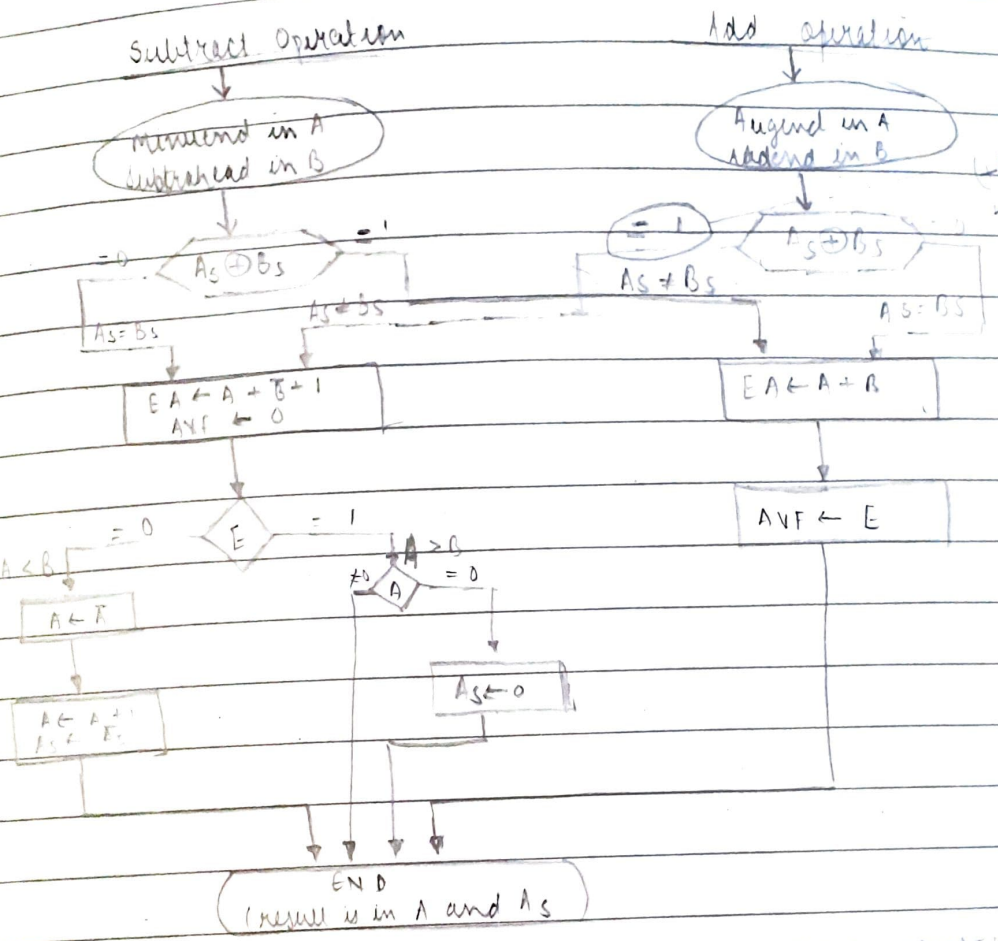
XOR

0 0	0
0 1	1
1 0	1
1 1	0

DATE / /
PAGE NO.

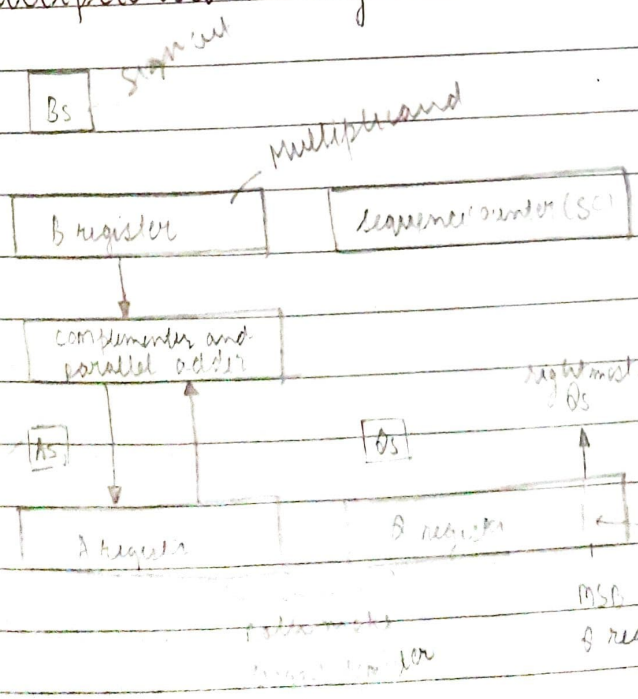
M = 0 → No complement & Cin = 0 (A ← A + B)

M = 1 → complement B & Cin = 1 (A ← A + $\bar{B}$ + 1)



FLOWCHART OF ADDITION AND SUBTRACTION OPERATION

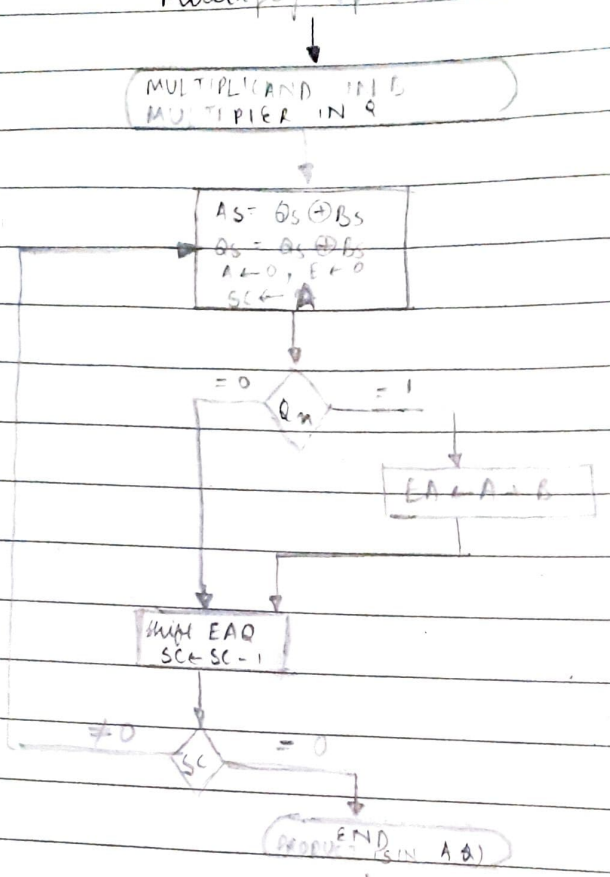
- **Multiplication Algorithm.**



23 → 10111 multiplicand
19 → 10011 multiplier
10111
10111
00000
00000
10111
1 0 1 1 0 1 0 1 = 437 PRODUCT

DATE / /
PAGE NO.

FLOWCHART FOR MULTIPLY OPERATION



Multiplicand B: 10111 (multiplicand)

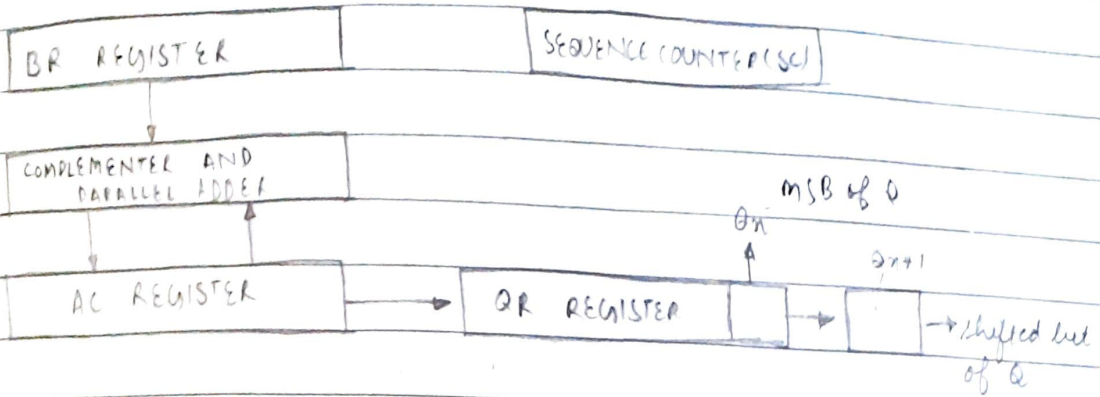
	E	A	Q (multiplier)	SC (sequence counter)
Multiplier in Q	0	00000	10011	101 5
$Q_n = 1$, add B		10111		
First partial product	0	10111		
shift right EAQ	0	01011	11001	100 4
Q = 1; add B		10111		
Second partial product	1	00010		
Shift right EAQ	0	10001	01100	011 3
$Q_n = 0$; shift right EAQ	0	01000	10110	010 2
$Q_n = 0$; shift Right EAQ	0	00100	01011	001 1
$Q_n = 1$; add B		10111		
Fifth partial product	0	11011		
Shift right EAQ	0	01101	10101	000

Final product in AQ = 01101 10101

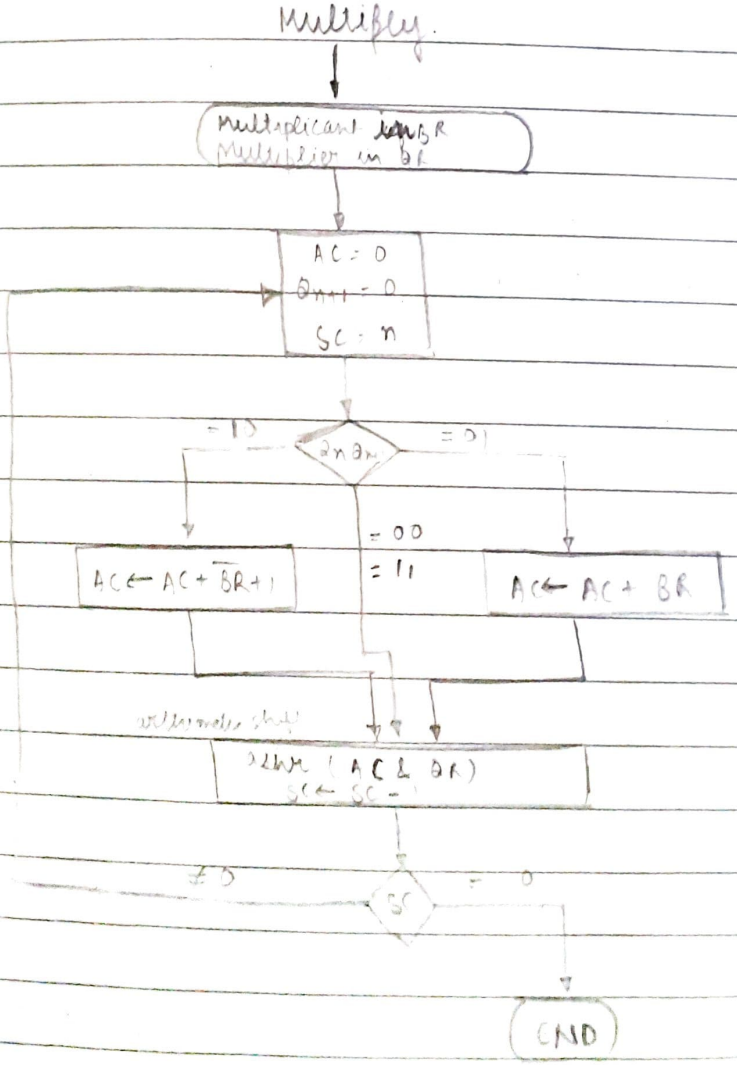
DATE / /
PAGE NO.

BOOTH'S MULTIPLICATION ALGORITHM

- Hardware for Booth algo



- Flow chart of booth algo for multiplication of a no.



- Example of Booth's algo

$Q_n Q_{n+1}$	BR = 10111, $\overline{BR}$ + 1 = 01001	AC	QR	Q_{n+1}	SC
	Initial	00000	10011	0	101
1 0	subtract BR	01001			
		01001			
	ashr	00100	11001	1	100
1 1	ashr	00010	01100	1	011
0 1	Add BR	10111			
		11001			
0 0	ashr	11100	10110	0	010
1 0	ashr	11110	01011	0	001
	subtract BR	01001			
		00101			
	ashr	00011	10101	1	000

- Division Algorithm

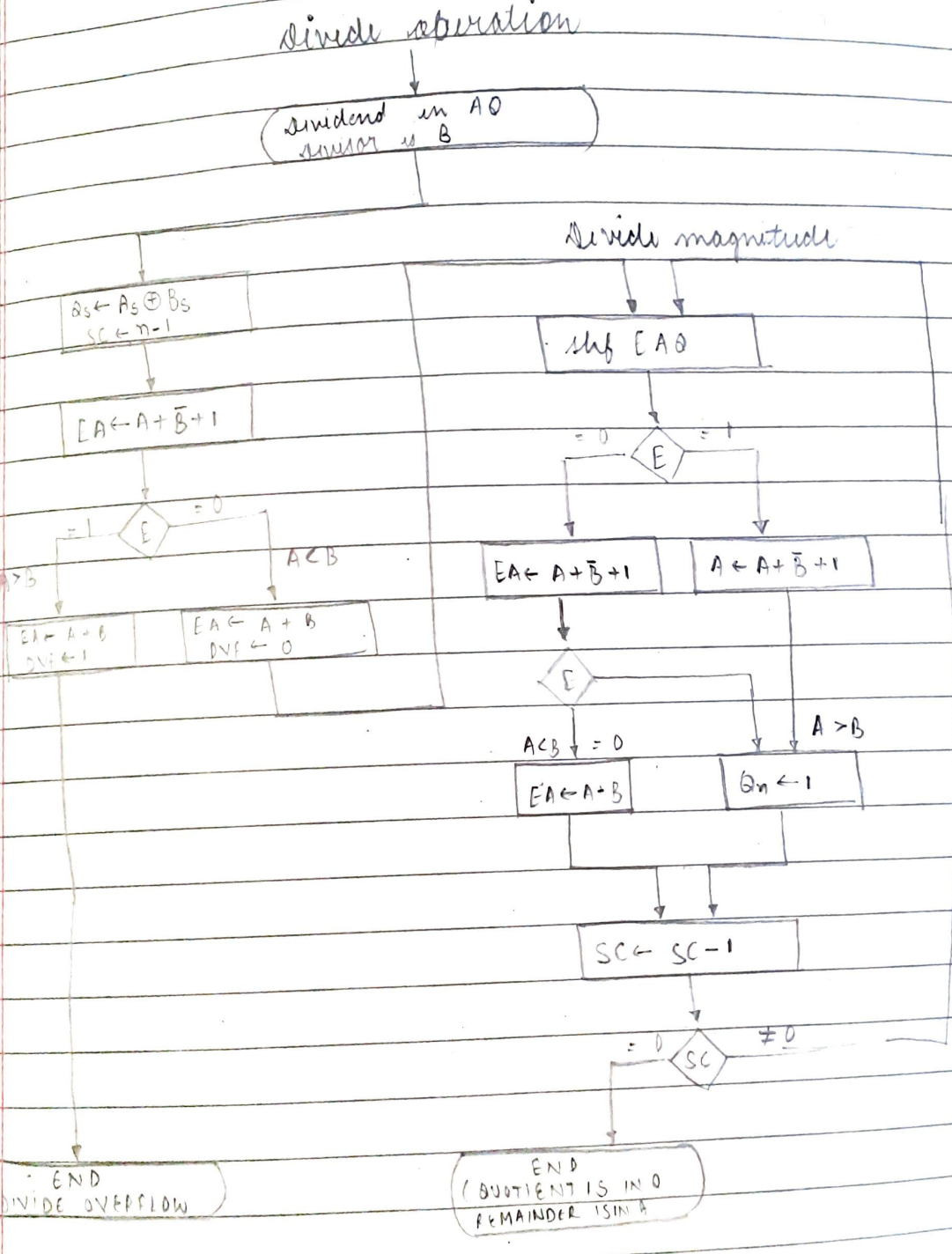
Example of binary division

Divisor:
B = 10001

	11010	Quotient = Q
	0111000000	Dividend = A
	01110	5 bit of A < B, quotient has 5 bit
	011100	6 bit of A > B
	10001	Shift right B and subtract; enter 1
	010110	7 bit of remainder > B
	10001	Shift right B and subtract; enter 1
	001010	Remainder < B; enter 0 in Q shift right
	010100	Remainder > B
	10001	Shift right B and subtract enter 1
	000110	Remainder < B; enter 0 in Q
	00110	Final remainder

DATE / /
PAGE NO.

Flow chart for divide operation



DATE / /
PAGE NO.

→ Example of binary division with digital hardware

Divisor B = 10001 $\quad\quad$ $\bar{B}+1 = 01111$

	E	A	Q	SC
Dividend		01110	00000	
shl EAQ	0	11100	00000	5
add $\bar{B}+1$		01111		
E = 1		01011		
Set $Q_n = 1$	1	01011	00001	
shl EAQ	1	10110	00010	4
Add $\bar{B}+1$	0	01111		
E = 1		00101		
Set $Q_n = 1$	1	00101	00011	
shl EAQ	1	01010	00110	3
Add $\bar{B}+1$	0	01111		
E = 0; leave $Q_n = 1$	0	11001	00110	
Add B		10001		
Restore remainder	1	01010		
shl EAQ add $\bar{B}+1$ E = 1	0 1	10100 01111	01100	2
		00011		
Set $Q_n = 1$	1	00011	01101	1
shl EAQ	0	00110	11010	
Add $\bar{B}+1$		01111		
E = 0, leave $Q_n = 0$	0	10101	11010	
Add B		10001		
Restore remainder	1	00110	11010	0
Neglect E		00110		

Remainder in A

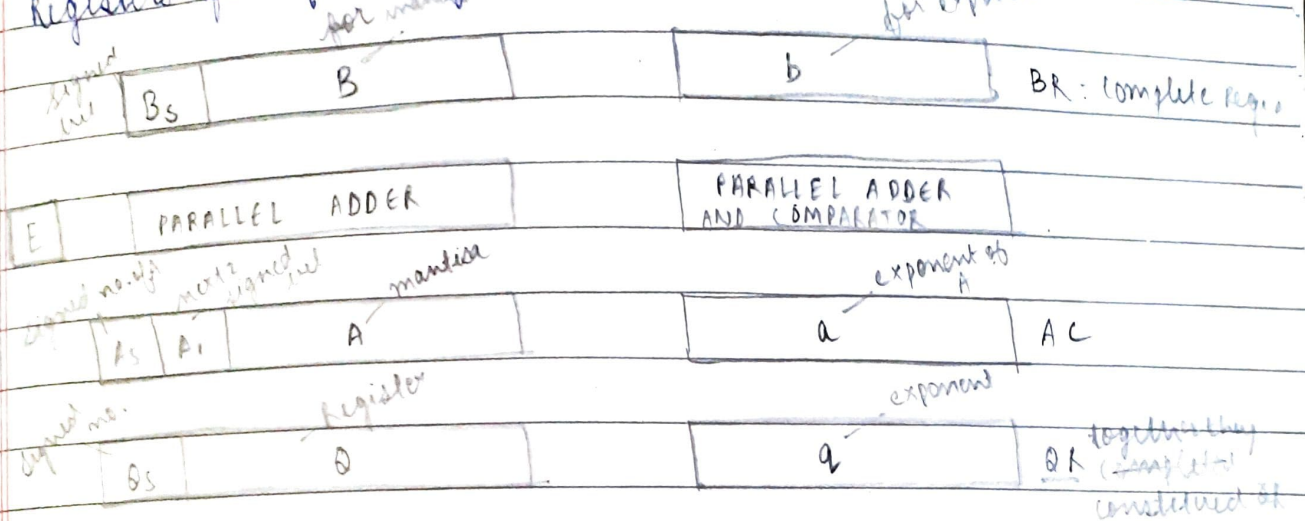
Quotient in Q

11010

DATE / /
PAGE NO.

FLOATING POINT ARITHMETIC OPERATION

- Registers for floating-point arithmetic operations



Addition and subtraction

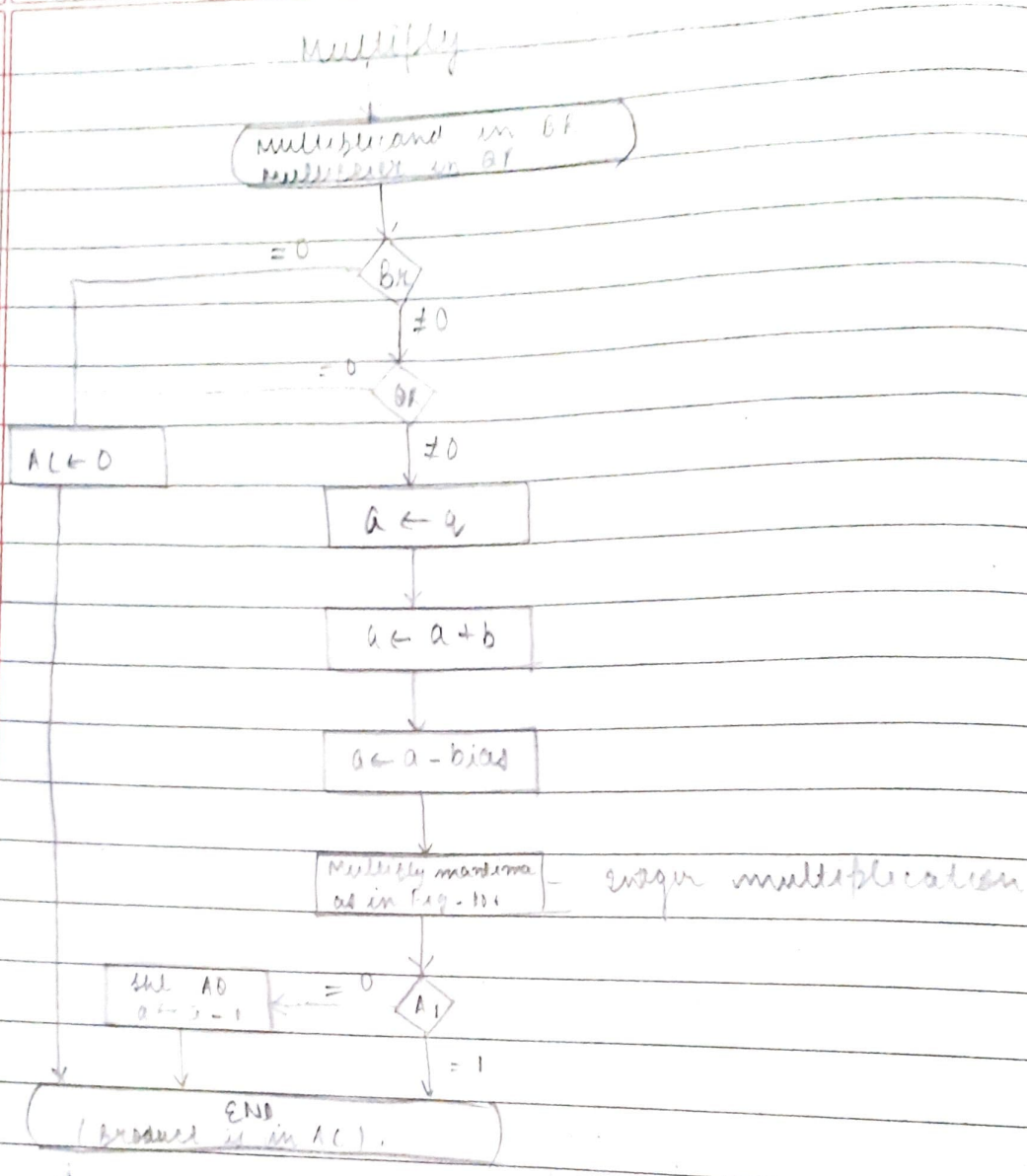
1. Check for zeroes.
2. Align the mantissas
3. Add or subtract the mantissas
4. Normalize the result

- FLOATING POINT MULTIPLICATION OPERATION

- The multiplication algorithm can be subdivided into 4 parts

1- Check for zeroes
2- Initialize registers and evaluate the sign.
3- Align the dividend.
4- Subtract the exponents.
5- Divide the mantissas.

2. Add the exponents.
3- Multiply the mantissas
4- Normalise the product.



- FLOATING POINT DIVISION OPERATION

The division algorithm can be subdivided into five parts:

1- check for zeroes
2- Initialize registers and evaluate the sign
3- Align the dividend.
4- subtract the exponents
5- divide the mantissas

- flow chart for division of floating-pt numbers.