

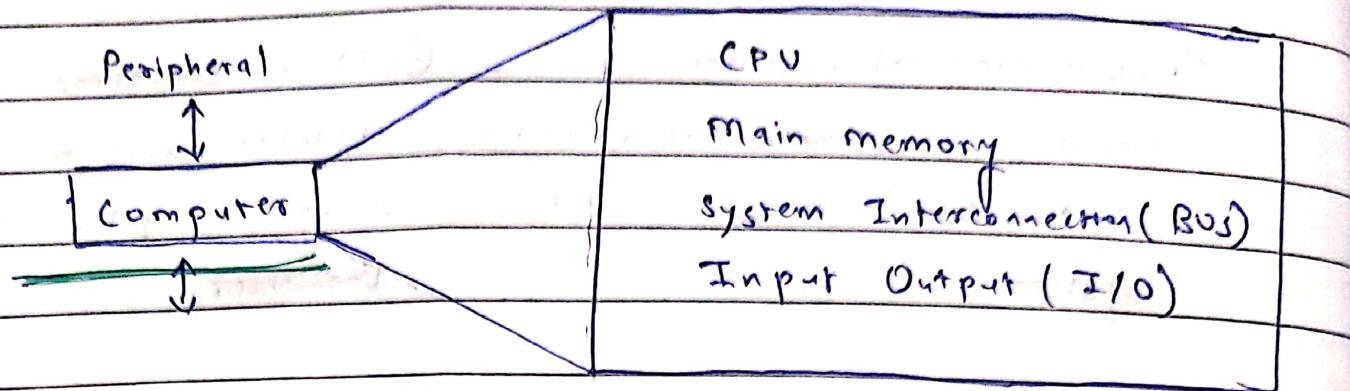
CH-1 Computer Architecture and Organization

Computer Architecture Computer Organization

- ① It is concerned with the way hardware components are connected together to form a computer system.
- ② It acts as an interface between hardware and software.
- ③ It helps us to understand functionalities of a system.
- ④ A programmer can view architecture in terms of instructions, addressing modes and registers.
- ⑤ While designing a computer system architecture is considered first.
- ⑥ It deals with high-level design issues.
- ⑦ It involves logic (Instruction sets, Addressing modes, Data types, cache optimization).
- ① It is concerned with the structure and behaviour of a computer system as seen by the user.
- ② It deals with the components of a connection in a system.
- ③ It tells us how exactly all the units in a system are arranged and interconnected.
- ④ Organization expresses the realization of architecture.
- ⑤ An organization is done on the basis of architecture.
- ⑥ It deals with low level design issues.
- ⑦ It involves physical components (Circuit design, Adder, Signals, Peripherals).

Basic organization of computer and block level description.

Structure of a computer:



4 components of computer

i] CPU:

- Main part of the computer
- Performs all the operations of the computer.
- Heart of the computer
- Known as processor.

ii] Main Memory:

- It is used to store the data

iii] I/O devices:

- Used for sending and receiving the data
- Known as channel (Between computer and external world)
- Other communication lines

iv] System Interconnections

- Lines that connect several components to enable them to perform their specific operations
- Mechanism used for communication between CPU, Main memory and I/O devices.

4 Components of CPU:

I] Control unit:

- Main part of the CPU.
- It does all the processing.

II] ALU:

- Arithmetic Logic Unit
- Used to perform all arithmetic functions.

III] Registers:

- Small unit of CPU
- Used to store small amount of data.

IV] CPU Interconnection:

- Mechanism used for the communication between registers, ALU and Control Unit.

3 Components of Control Unit:

I] Sequencing Logic:

II] Control memory (CU memory)

III] Control Unit Registers and decoders.

- The control memory stores the microinstructions loads it into the control unit register and the sequencing logic gives these signals in a proper sequence to execute a instruction.

Functions of control unit.

- It interprets instructions
- Controls data flow inside the processor
- Receive external commands, converts it to sequence of control signals.
- Handles multiple tasks like fetching, decoding, execution handling, storing result.

Evolution of computers

- Mechanical Era

- ↳ Wilhelm Schickard (1623)
- ↳ Blaise Pascal (1642)
- ↳ Gottfried Liebniz (1673)
- ↳ Charles Babbage (1822)
- ↳ Herman Hollerith (1889)
- ↳ Konrad Zuse (1938)

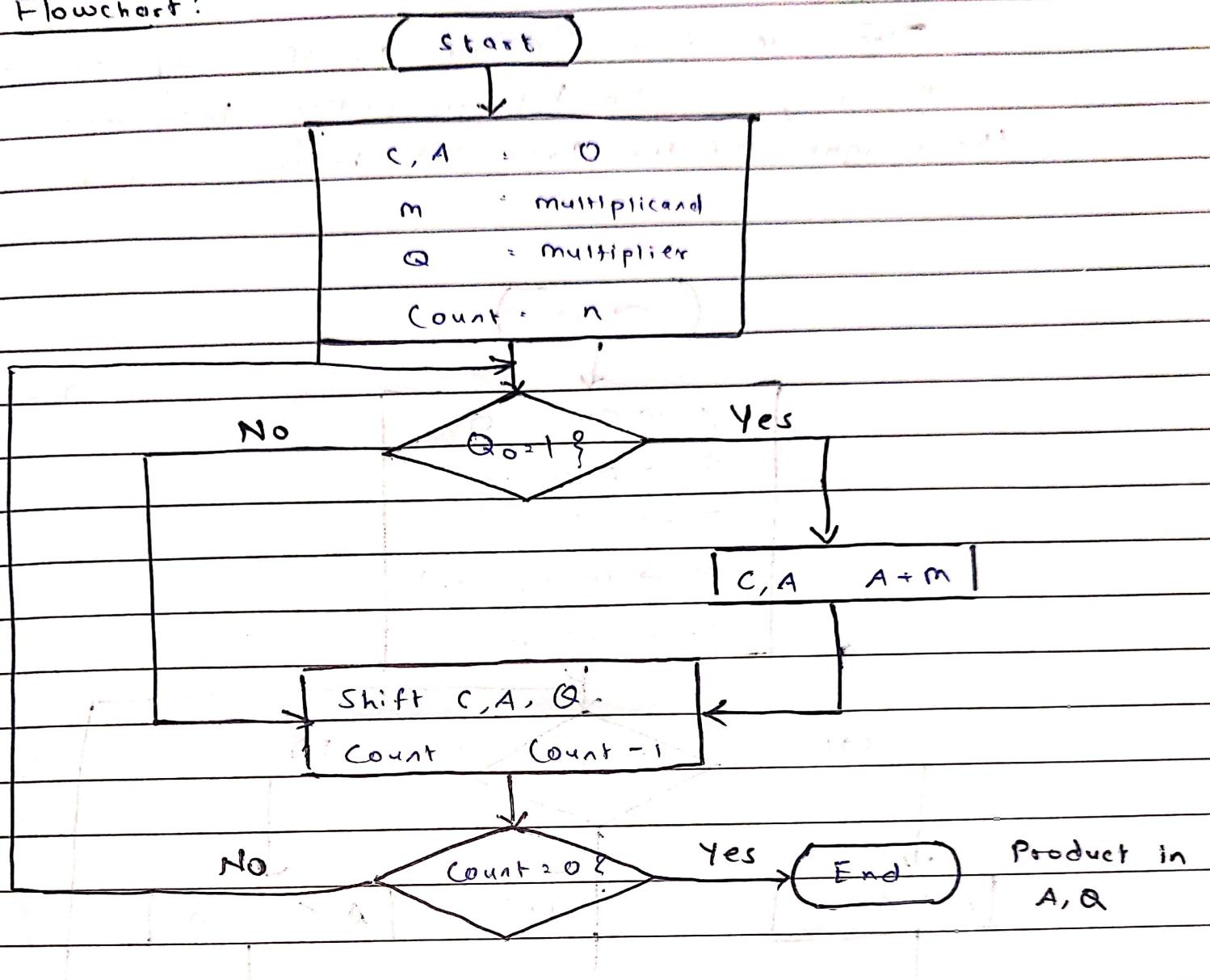
- Electronic Era

- ↳ Generation I : Vacuum tubes (1945 - 1958)
- ↳ Generation II : Transistors (1958 - 1964)
- ↳ Generation III : Integrated chip (IC) (1964 - 1974)
- ↳ Generation IV : Microprocessors

(INTEL HISTORY)

Multiplication : Unsigned Multiplication.

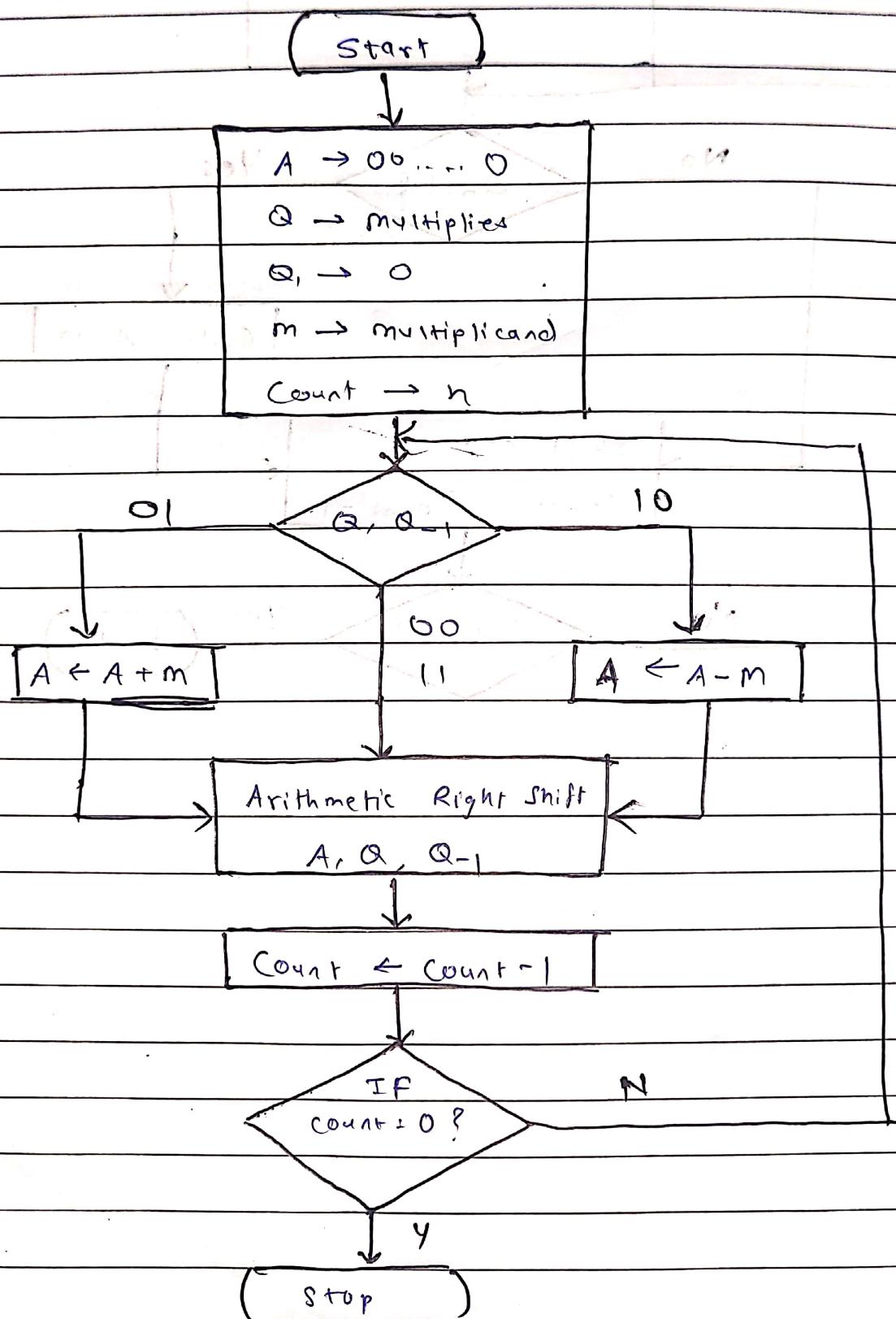
Flowchart:



Multiplication : Signed Multiplication : Booth's Algorithm

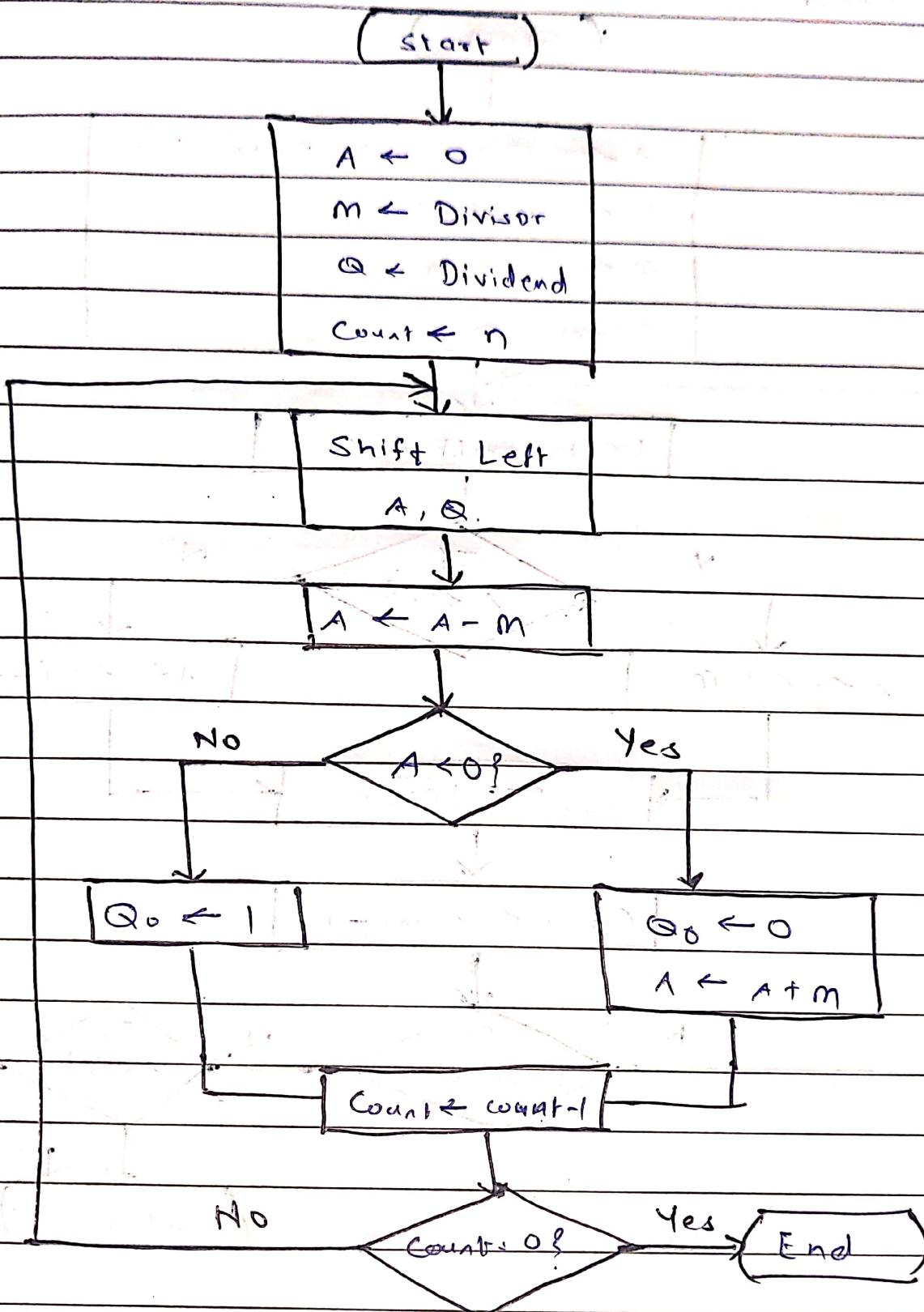
Booth's Principle :

The value of a series of 1's of binary can be given as the weight of the bit preceding the series minus the weight of the last bit in the series.



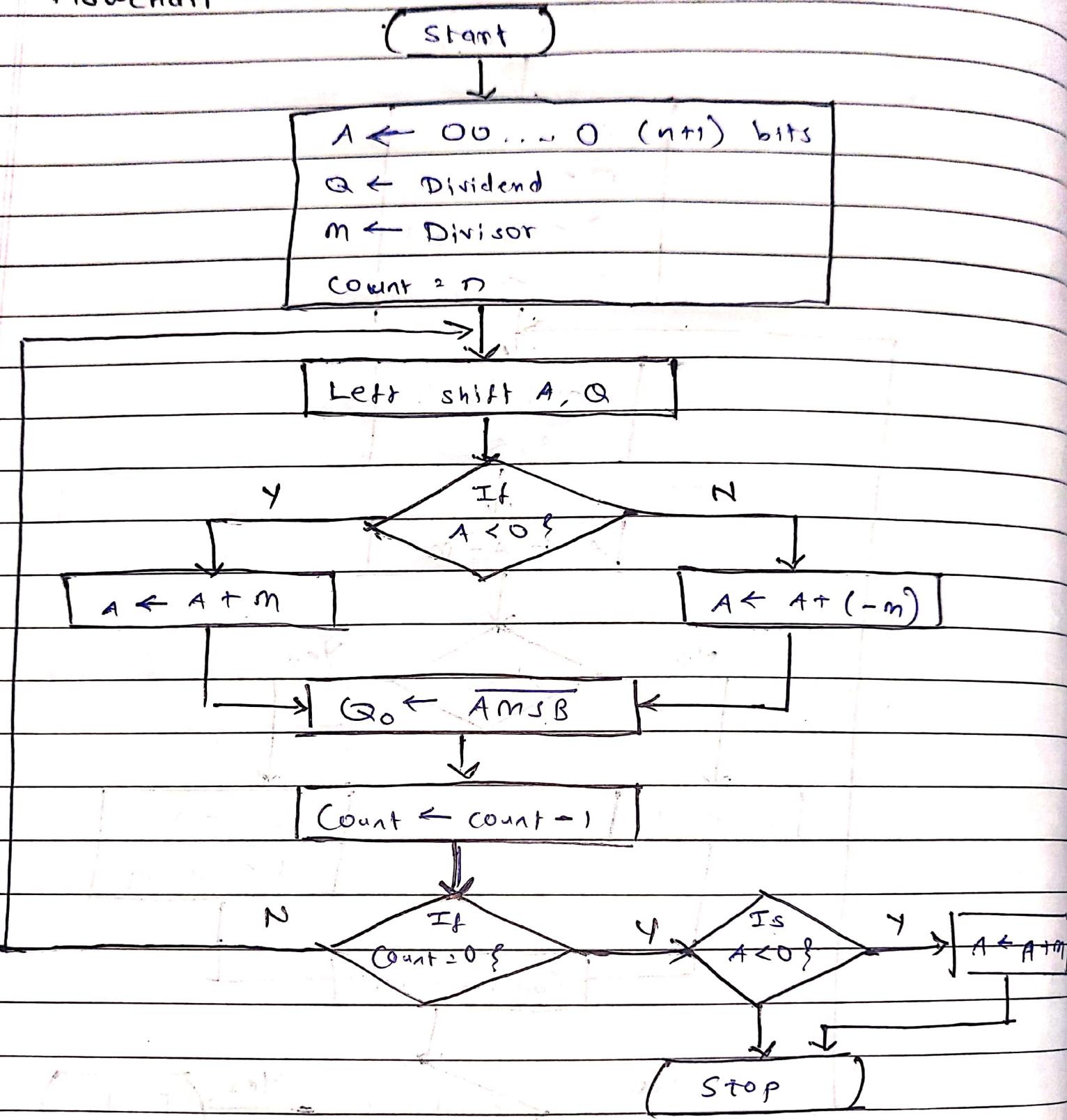
Division of Integers : Restoring method

Flowchart



Division of Integers : Non - restoring method

Flowchart



Floating point representation (IEEE 754 standard)

i) Single precision floating point representation

Representation:

S	Exponent	Mantissa
1 bit	8 bits	23 bits

(i) S:

- This bit represents the sign of the number. 0 represents +ve and -1 represents -ve number.

(ii) Exponent:

- 8 bit signed exponent in excess 127 representation

(iii) Mantissa: (m):

- 23 bit Mantissa (m) fraction

(iv) Value represented: = $(-1)^S \times 1.m \times 2^{\text{Exponent} - 127}$

ii) Double precision floating point representation.

Representation:

S	Exponent	Mantissa
1 bit	11 bits	52 bits

(i) S:

- This bit represents the sign of the number. 0 represents +ve and 1 represents -ve number.

(ii) Exponent:

- 11 bit signed exponent in excess 1023 representation

(iii) Mantissa: (m):

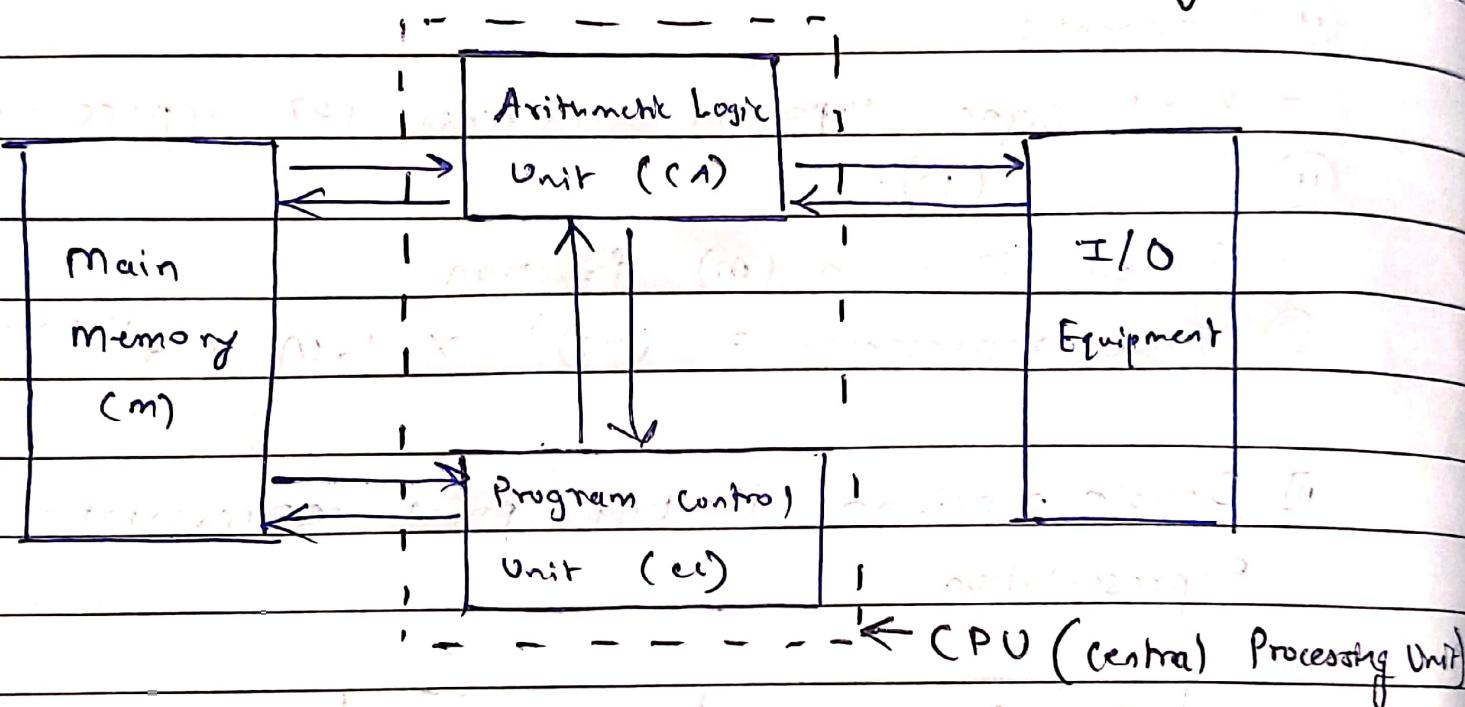
- 52 bit Mantissa Fraction

(iv) Value represented: = $(-1)^S \times 1.m \times 2^{\text{Exponent} - 1023}$

CH-2 Processor Organization and Architecture

Von Neumann Architecture

- First proposed by computer scientist John von Neumann.
- In this architecture, one data path or bus exists for both instruction and data. As a result, the CPU does one operation at a time.
- It either fetches an instruction from memory or performs read/write operation on data. Instruction fetch and a data operation cannot happen simultaneously.

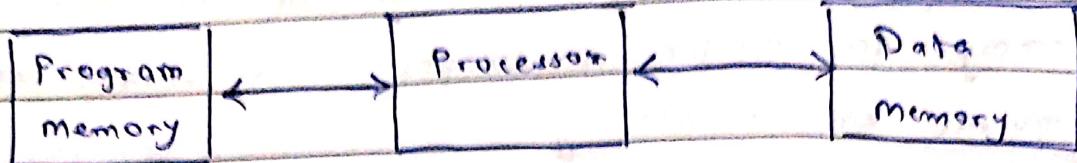


- Supports simple hardware
- Allows use of a single, sequential memory
- The computer has a common memory for data as well as code to be executed.
- This system has 3 units - ① CPU, ② memory, ③ I/O
- CPU has 2 units - ① Arithmetic Unit, ② Control Unit

Stored program concept:

- Stored program concept, storage of instruction in computer memory to enable it to perform a variety of tasks in sequence or intermittently.
- Designed by Hungarian mathematician John Von Neumann.
- Von Neumann architecture is a design model for a stored program digital computer that uses a processing unit and a single separate storage structure to hold both instructions and data.
- Stored program digital computer is one that keeps its programmed instructions as well as its data in read-write, random access memory (RAM).
- Before any data processed, instructions are read into memory. The processing starts with the first instruction in the program which is copied into a control unit circuit. The control unit executes the instructions sequentially until it finds one that causes it to break a sequence and go elsewhere in the program.
- I/O and processing are performed simultaneously.

Harvard Architecture



- Two separate memories for storing data and program
- The processor can simultaneously access instruction as well as the data and hence can complete an instruction execution in one cycle.

Von Neumann Architecture Harvard Architecture

- | | |
|---|---|
| ① Single memory to be shared by both code and data | ① Separate memories for code and data |
| ② Processor needs to fetch code in a separate clock cycle & data in another clock cycle. So it requires two clock cycles. | ② Single clock cycle is sufficient as separate buses are used to access code and data |
| ③ Higher speed, thus less time consuming | ③ Slower in speed, thus more time consuming |
| ④ Simple in design | ④ Complex in design |

IR (Instruction Register):

- It is used to hold current instruction
- Contents of IR are available to the control unit, which generate the timing signals that control the various processing elements involved in executing the instruction

PC (Program Counter):

- It always contains address of the next instruction to be executed

SP (Stack Pointer):

- It is used to point to the top activation record on the run time stack.
- Run time stack contains one activation record for each function that is currently unfinished in the program.

AC (Accumulator):

- It is one of the general purpose registers
- It is specifically used to accumulate the result of the currently running instruction.

MAR (Memory Address Register):

- It is used to handle data transfer between the main memory and the processor.
- MAR holds the address of the main memory to or from which data is to be transferred.

MDR (Memory Data Register):

- Handles data transfer between the main memory and processor
- MDR contains the data to be written into or read from the addressed word of the main memory

CPU Architecture and Organization

CPU Archit is divided into 4 different groups

i) Registers

ii) Arithmetic and Logic Unit

iii) Interrupt Control

iv) Timing and control circuitry

- It consists of PIPD (Parallel In Parallel Out)
- Section is known as Scratch Pad Memory. It stores data and address of memory
- The archit of microcomputer depends upon the number and the type of registers used in microprocessor
- The registers are classified as
 - i) Temporary Registers
 - ii) General purpose registers
 - iii) Special purpose registers

Instruction Formats

- The control unit and the ALU along with some registers constitute CPU.
- Input devices are required to give instructions and data to the system. Output \rightarrow Output system
- The instructions and the data given by the input devices are to be stored, For storage we require memory
 - The memory module receives address and control signals.
 - The memory is assumed to be of 'n' locations with the addresses from 0 to n-1.
- CPU has the control unit that provides control signals to all the resources inside and outside the CPU.
- An instruction is normally made up of a combination of an operation code and some way of specifying an operand, like its location or address in memory

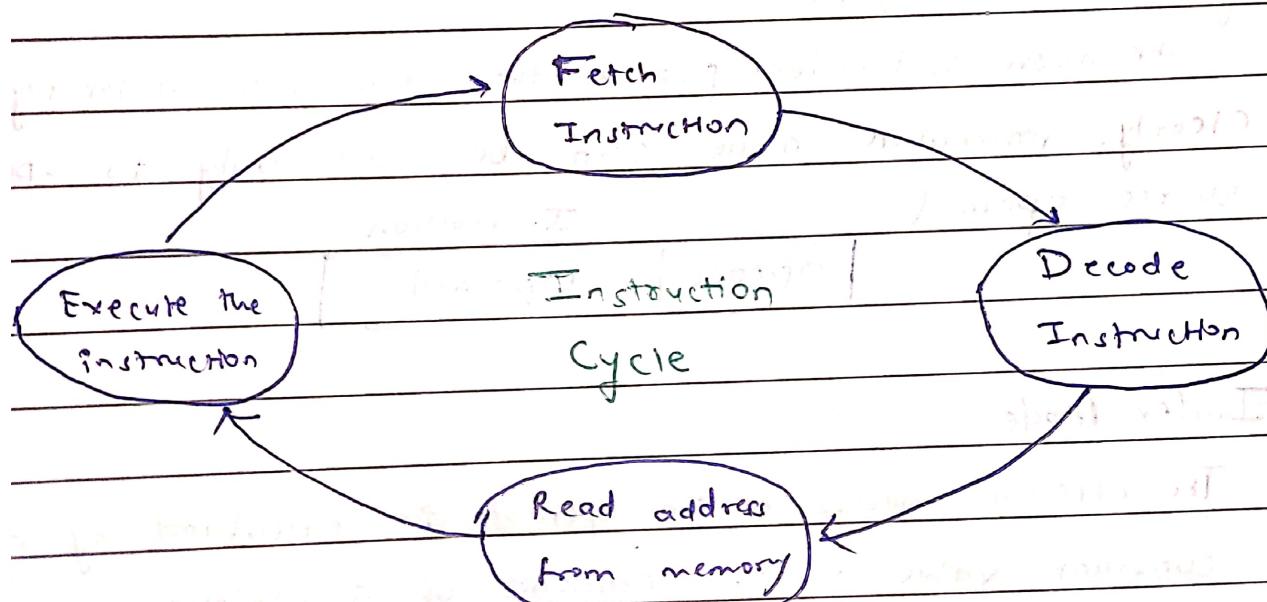
Opcode	Mode	Address or Operand
--------	------	--------------------

Reverse Polish Notation:

- RPN used to be a basic of computer programmer's world but today it is not as well known
- RPN is a method for representing expressions in which the operator symbol is placed after the arguments being operated on.
- In polish notation, operator comes before the operand.
- For ex, RPN expression will produce the sum of 2 and 3 namely, 5: 2 3 +.
- RPN is also known as postfix notation

Instruction cycle

- A program residing in the memory unit of a computer consists of a sequence of instructions.
- These instructions are executed by the processor by going through a cycle for each instruction.
- In basic computer, each instruction cycle consists of phases:
 - I Fetch instruction from memory
 - II Decode the instruction
 - III Read the effective address from memory
 - IV Execute the instruction



Interrupt Cycle

- In a instruction cycle, the interrupt is the last part
- Interrupts occur at random times during the execution of a program, in response to signals from hardware.

Type

Mean of Communication between

Interrupts

the CPU and the OS Kernel

Signals

the OS kernel and OS processes

Addressing Modes

- Addressing modes are nothing but the different ways in which the location of an operand can be specified in an instruction. The number of addressing modes that a processor supports changes according to the instruction set it is based on.

- I] Immediate mode

vii] Base with Index

ii] Register mode

viii] Base with index and offset

iii] Absolute mode

ix] Relative

iv] Indirect Mode

x] Auto increment

v] Index mode

xii] Auto decrement

- Immediate mode:

The operand is specified in the instruction itself.

Eg. move #200, R0

The above instruction places the value 200 in the register R0. Clearly, immediate mode can be used only to specify the source operand.

Instruction

Opcode	Operand
--------	---------

Index Mode:

- The effective address of the operand is calculated by adding a constant value to the contents of a register.

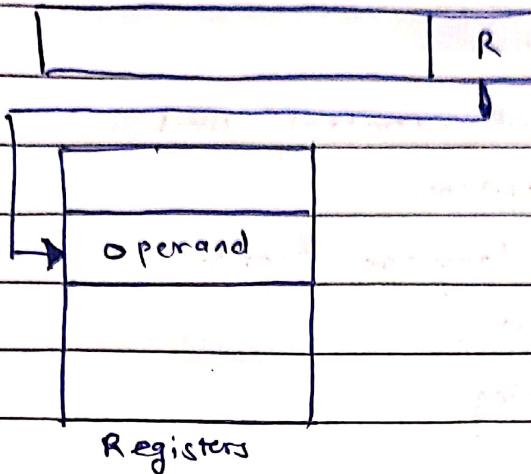
Registers Mode:

- Operand is the content of the register.

Eg. Move R0, R1

Contents of the R0 register are moved to R1 register.

Instruction



Absolute Mode:

- The operand is in a memory location; The address of the operand is passed explicitly in the instruction

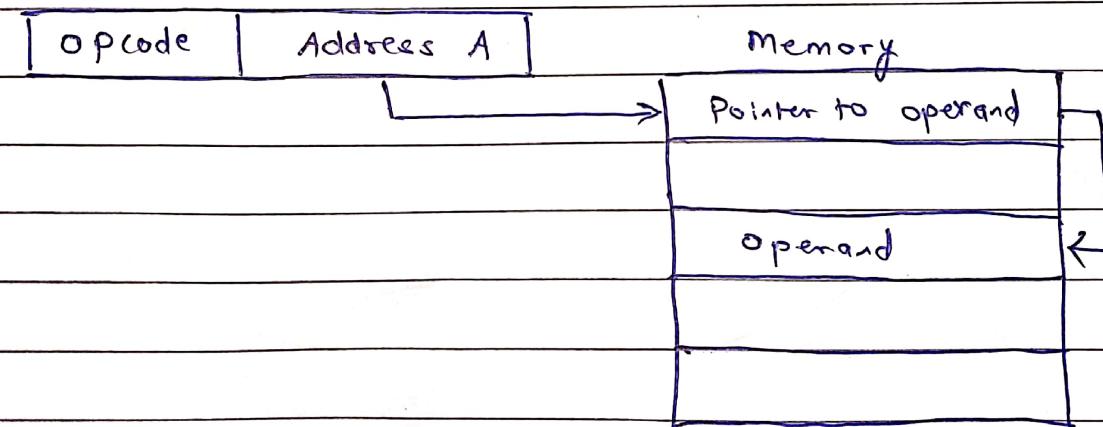
Eg. Move LOC, R0

Here LOC corresponds to the address from where the contents will be accessed by the processor and placed in R0.

Indirect Mode:

- The effective address of the operand is the content of a register or the memory location whose address appears in the instruction.

Instruction



Fetch cycle

- CPU fetch some data or instruction from main memory then store it into its internal memory called Register, this stage is called Fetch cycle.

Applications of Microprogramming

- In realization of control unit
- In Operating System
- In High level language support
- In microdiagnostics
- In User tailoring
- In Emulation.