Addressing Mode - The different way to specifying the location of an operand in an instruction are called Addressing Mode.

→ Types -

- Implict/Implied → It is a type of address mode where no operand is specified in the instruction.
- Immediate → operant specifies in the instruction itself.
- Direct → specifies the instruction in the effective or actual address of the operand.
- Indirect → instruction specifies the address where the effective address of operand is stored.
- register direct → operand is stored in the register in the CPU (that contain operand)
- register indirect → address feild of instruction specifies the address that contain effictive address of operand.
- Auto Increment → The register is incremented after its value is used to access memory.

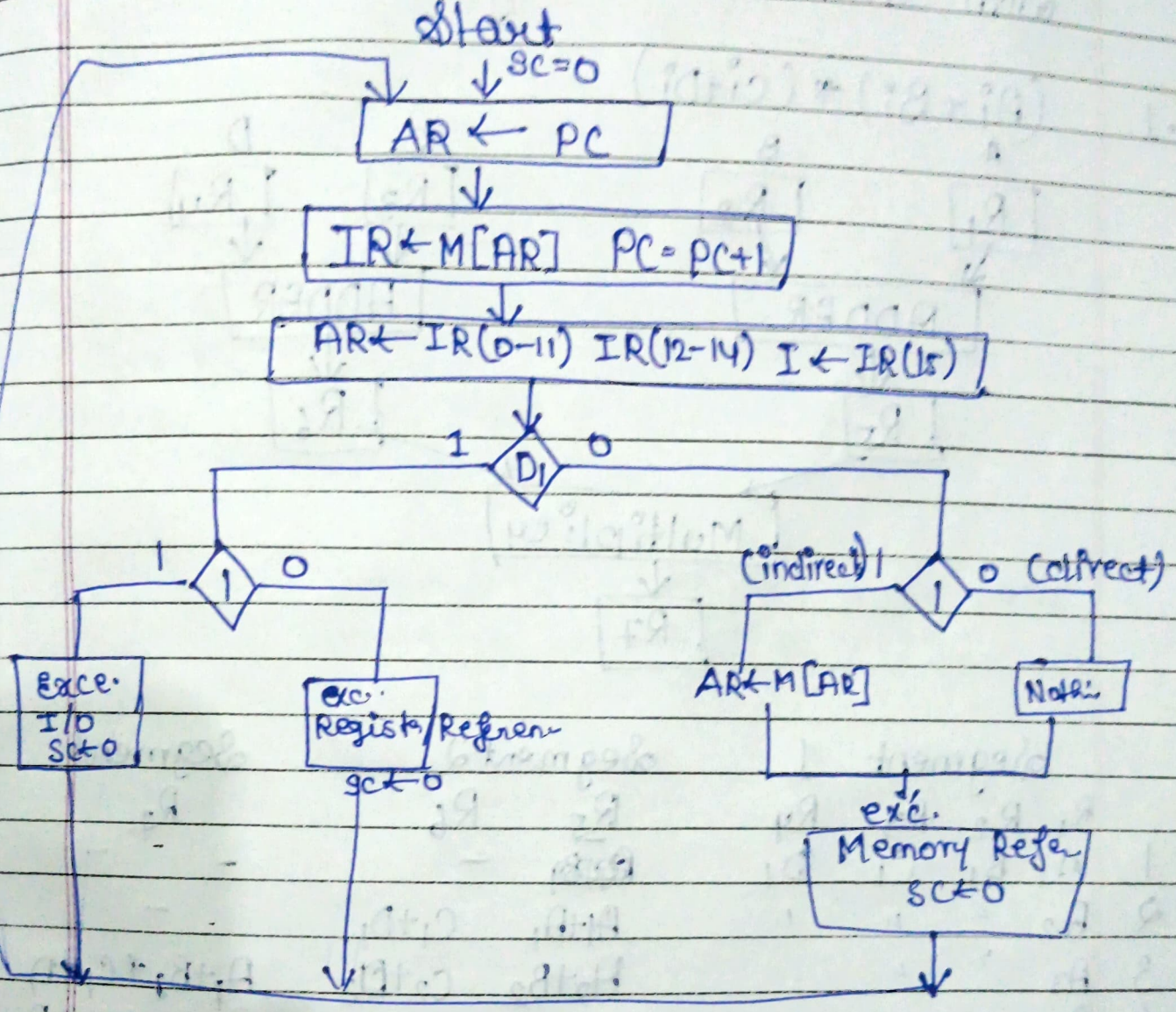
- Auto decrement mode - the register is decremented before its value is used to to access memory.
- Relative address → address - Program counter + Address field of Instruction.
- Index Address → address = Index register + Address field of instruction
- Base register addres. address - Base register + address field of Instruction

→ Why do we use -

- To give the programmig versability to the user.
- To reduce the number of bits in addressing field of instruction

→ Instruction cycle: It fetch the instruction from memory and then decode and then execute.

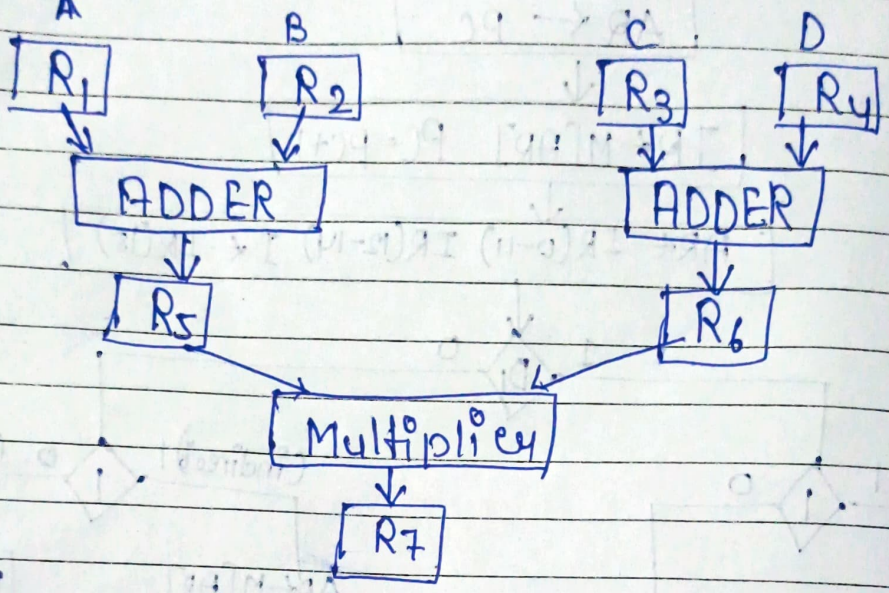
flow chart of Instruction cycle



Pipelinning →

Pipelinning is a technique of decomposing a sequential process int sub-operations with each sub-process being executed in a segment that operates concurrently with with other segment

Q.1 $(A_i + B_i) * (C_i + D_i)$




| | Segment 1 | | | | Segment 2 | | Segment |
|---|---|---|---|---|---|---|---|
| | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | $R_7$ |
| 1 | $A_1$ | $B_1$ | $C_1$ | $D_1$ | | - | - |
| 2 | $A_2$ | 〃 | 〃 | 〃 | $A_1+B_1$ | $C_1+D_1$ | - |
| 3 | $A_3$ | 〃 | 〃 | 〃 | $A_2+B_2$ | $C_2+D_2$ | $A_1+B_1 * C_1+D_1$ |
| 4 | $A_4$ | 〃 | 〃 | 〃 | 〃 | 〃 | $A_2+B_2 * C_2+D_2$ |
| 5 | $A_5$ | 〃 | 〃 | 〃 | 〃 | 〃 | 〃 |
| 6 | $A_6$ | 〃 | 〃 | 〃 | 〃 | 〃 | 〃 |
| 7 | | | | | 〃 | 〃 | 〃 |
| 8 | | | | | | | 〃 |

Locality of reference refers to a phenomenon in which a computer program tends to access same set of memory locations for a particular time period. In other words, **Locality of Reference** refers to the tendency of the computer program to access instructions whose addresses are near one another. The property of locality of reference is mainly shown by loops and subroutine calls in a program.

An **instruction** of a computer is a command given to the computer to perform a specified operation on given data. In microprocessor, the **instruction** set is the collection of the instructions that the microprocessor is designed to execute.

The programmer writes a program in assembly language using these instructions. These instructions have been classified into the following groups: