Instruction formats :-

- The format of an instruction is usually depicted in rectangular box symbolizing the bits of the instruction as they appear in memory word or control register.
- The bits of instruction are divided into groups called fields. The most common field found in instruction format are :

A) Operation Code - Specifies the operation to be performed

B) Address field - Designates a memory address or processor register.

C) Mode field - Specifies the operand or effective address is determined.

- Computers may have instructions of several different lengths containing varying number of addresses.
- The number of address fields in the instruction format of a computer depends on the internal organization of its registers.
- Most computers fall into one of the three types of CPU organization

1. Single Accumulator Organization
2. General Register Organization
3. Stack Organization

Dr Anuja Bhargava

Zero - Address Instructions

- A stack organized computer does not use an address field for the instruction ADD & MUL.
- The PUSH & POP instructions, needs an address field to specify the operand that communicates with the stack.

eg $X = (A + B) * (C + D)$

Write a program for stack organized computer.

PUSH	A	TOS ← A
PUSH	B	TOS ← B
ADD		TOS ← (A + B)
PUSH	C	TOS ← C
PUSH	D	TOS ← D
ADD		TOS ← C + D
MUL		TOS ← (C + D) * (A + B)
POP	X	M[X] ← TOS

Note :- The name "Zero - Address" is given to this type of computer because of the absence of an address field in the computational instructions.

* TOS - Top of Stack

Dr Anuja Bhargava

One Address Instruction

- It uses accumulator (AC) register for all data manipulation.
- Format

Opcode	Operand

Eg: $X = (A + B) * (C + D)$

Write a program to evaluate above equation.

LOAD	A	AC ← M[A]
ADD	B	AC ← AC + M[B]
STORE	T	M[T] ← AC
LOAD	C	AC ← M[C]
ADD	D	AC ← AC + M[D]
MUL	T	AC ← AC * M[T]
STORE	X	M[X] ← AC

Note :- 1. All the operations are done between the AC register and memory operand.

2. T is temporary memory location required for storing the intermediate result.

Dr Anuja Bhargava

Two Address Instructions

- It is most common in commercial computers.
- Here, each address field specify either a processor register or a memory word.
- format

Opcode	Destination Address	Source Address

eg $X = (A + B) * (C + D)$

MOV	R1, A	R1 ← M[A]
ADD	R1, B	R1 ← R1 + M[B]
MOV	R2, C	R2 ← M[C]
ADD	R2, D	R2 ← R2 + M[D]
MUL	R1, R2	R1 ← R1 * R2
MOV	X, R1	M[X] ← R1

Note :- 1) MOV instruction moves or transfers the operands to and from memory and processor registers.

2) The first symbol in instruction is assume to be both a source & destination where the result of the operation is transfered.

Dr Anuja Bhargava

Three Address Instructions

- It use each address field to specify either a processor register or a memory operand.
- format

Opcode	DA	SA	SA

eg

ADD	R1, A, B	R1 ← M[A] + M[B]
ADD	R2, C, D	R2 ← M[C] + M[D]
MUL	X, R1, R2	M[X] ← R1 * R2

Advantage - Results in short programs when evaluating arithmetic expressions.

Disadvantage - Binary coded instructions require too many bits to specify three addresses.

eg WAP to evaluate the arithment statement.

a) $X = \dfrac{A - B + C * (D * E - F)}{G + H * K}$

b) $X = A - B + (C+D)/E + F/G + (J*K)$

c) $X = (A*B+C) + D/E + F*G + (I*J)$

Using Three, Two, One & Zero Instruction.

Dr Anuja Bhargava

a) $X = \dfrac{A - B + C * (D * E - F)}{G + H * K}$

Soln:- RPN

$$X = \frac{(A - B + C) * (D * E - F)}{(G + H * K)}$$

$$= [(A - B + C) * (D * E - F)] (G + H * K) /$$

$$= [(A B - + C) * (D E * - F)] (G + H K *) /$$

$$= [(A B - C +) * (D E * F -)] (G H K * +) /$$

$$= A B - C + D E * F - * G H K * + /$$

$\therefore X =$

RPN → X A B - C + D E * F - * G H K * + / =

Dr Anuja Bhargava

Addressing Modes

- The way the operands are chosen during program execution is dependent on the addressing mode of the instruction.
- Computers uses addressing mode techniques for the purpose of accommodating one or both:
  - → To give programming versatility to the users by providing facilities like pointers to memory, counters for loop control.
  - → To reduce the number of bits in addressing field of the instruction.

Advantage - It provide flexibility for writing programs that are more efficient with respect to number of instructions and execution time.

Program Counter - There is one register in the computer called Program counter or PC that keep the tracks of the instructions in the program stored in the memory. It holds the address of the instruction to be executed next and is incremented each time an instruction is fetched from memory.

Dr Anuja Bhargava

Example - Instruction format with a distinct addressing mode.

- Opcode specifies the operation to be performed.
- Mode is used to locate the operands needed for operation.
- Address field designates memory address.

Implied Mode -

- In this mode, the operands are specified implicitly in the definition of the instruction.
- eg Complement Accumulator
  (Operand is in the accumulator)

Immediate Mode -

- In this mode, the operand is specified in the instruction itself.
- These are useful for initializing registers to constant value.

Register Mode -

- In this mode, the operands are in registers that reside with in the CPU.
- The particular register is selected from a register field in the instruction.

Dr Anuja Bhargava

Register Indirect Mode -

- In this mode the instruction specifies a register in the CPU whose contents give the address of the operand in the memory.
- The selected register contains the address of the operand rather than the operand itself.
- Advantage - Address field of the instruction use fewer bits to select a register.

Autoincrement or Autodecrement Mode -

- It is similar to register indirect mode except that the register is incremented or decremented after its value is used to access memory.

Note :- Sometimes the value given in the address field is the address of the operand, but sometimes it is just an address from which the address of the operand is calculated. To differentiate among addressing modes, it is necessary to distinguish between address part of the instruction and effective address

Effective Address - It is defined to be memory address obtained from the computation dictated by the given addressing mode.

Dr Anuja Bhargava

Direct Address Mode-

- In this mode the effective address is equal to the address part of the instruction.
- The operand resides in memory and its address is given directly by the address field of the instruction.

Indirect address Mode-

- In this mode the address field of the instruction gives the address where the effective address is stored in memory.
- Control fetches the instruction from memory and uses its address part to access memory again to read the effective address.

Effective Address = Address part of Instruction + Content of CPU Register

Relative Addressing Mode.

- In this mode the content of program counter is added to the address part of the instruction in order to obtain the effective address.
- The address part is usually a signed number (+ve or -ve)
- When this number is added to the program counter, the result produces the effective address whose position in memory is relative to the address of next instruction.

Dr Anuja Bhargava

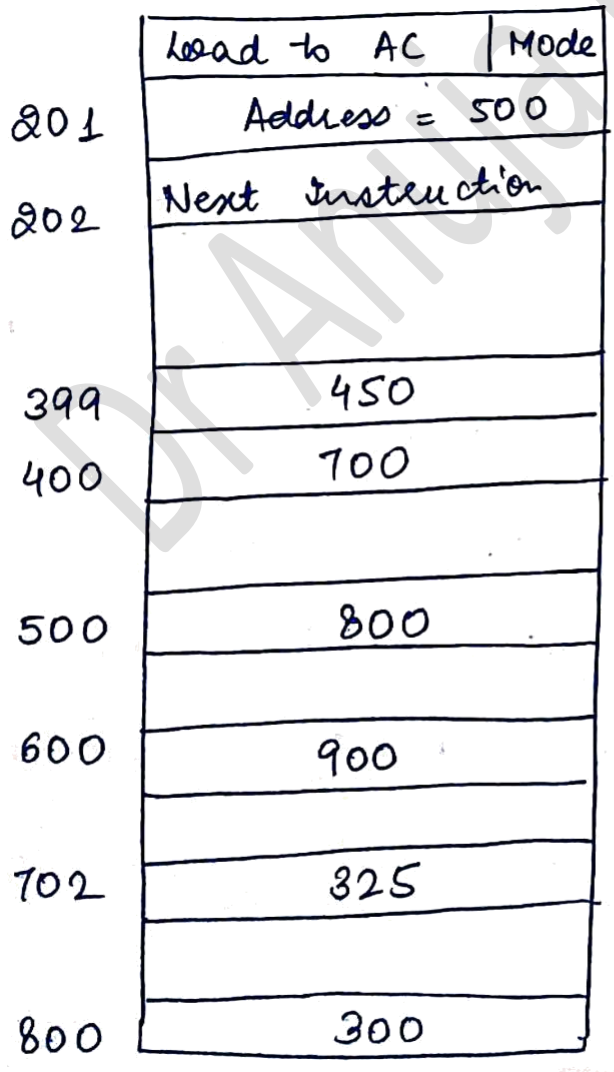
Indexed Addressing Mode-

- In this mode the content of index register is added to the address part of the instruction to obtain the effective address.

Base Register Addressing Mode

- In this mode the content of base register is added to the address part of the instruction to obtain the effective address

Example :-



Program Counter PC = 200

Processor Register R1 = 400

Index Register XR = 100

Dr Anuja Bhargava

Soln:-

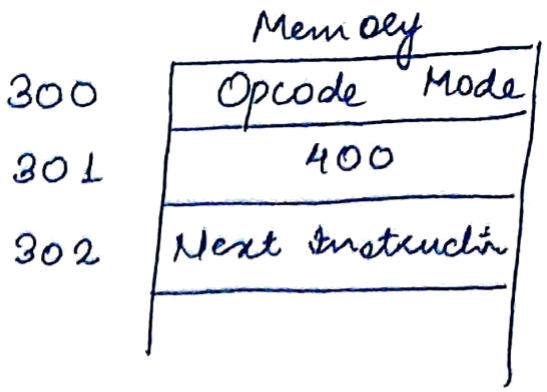
Addressing Mode	Effective Address	Content of AC
1. Direct Address	500	800
2. Immediate Address	201	500
3. Indirect Address	800	300
4. Relative Address	702 (500 + 202)	325
5. Indexed Address	600	900
6. Register Address	-	400
7. Register Indirect Address	400	700
8. Auto increement	400	700
9. Auto decrement	399	450

Example:- An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register R1 contains the number 200. Evaluate the effective address if the addressing mode of the instruction is : direct, immediate, relative, register indirect, index with R1 as index register.

Dr Anuja Bhargava

Soln:-

1. Direct : 400
2. Immediate: 301
3. Relative : 302 + 400 = 702
4. Register Indirect: 200
5. Indexed : 200 + 400 = 600

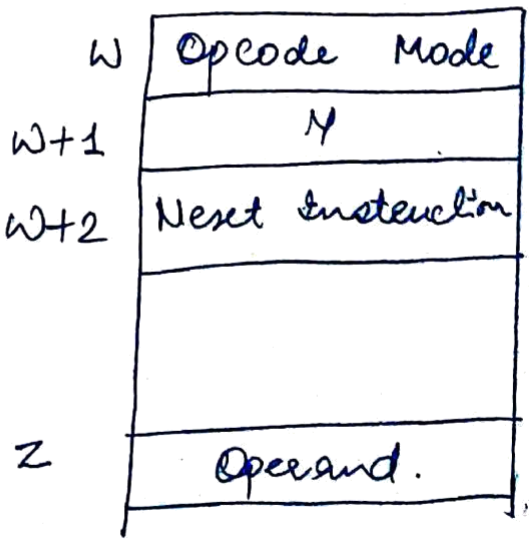


R1 = 200

Example: A two - word instruction is stored in memory at an address designated by symbol W. The address field of the instruction stored at (W+1) is designated by symbol Y. The operand used during the execution of the instruction is stored at an address symbolized by Z. An index register contains the value X. State how Z is calculated from the other addresses if the addressing mode of the instruction is : Direct, Indirect, Relative & Indexed.

Soln:-

1. Direct : Z = Y
2. Indirect : Z = M[Y]
3. Relative : Z = Y + W + Z
4. Indexed : Z = Y + X



Dr Anuja Bhargava