What is Addressing Modes ?

The operation field of an instruction specifies the operation to be performed. And this operation must be performed on some data. So each instruction need to specify data on which the operation is to be performed. But the operand(data) may be in accumulator, general purpose register or at some specified memory location. So, appropriate location (address) of data is need to be specified, and in computer, there are various ways of specifying the address of data. These various ways of specifying the address of data are known as "Addressing Modes"

So Addressing Modes can be defined as "The technique for specifying the address of the operands " And in computer the address of operand i.e., the address where operand is actually found is known as **"Effective Address".** Now, in addition to this, the two most prominent reason of why addressing modes are so important are:

First, the way the operand data are chosen during program execution is dependent on the addressing mode of the instruction.

Second, the address field(or fields) in a typical instruction format are relatively small and sometimes we would like to be able to reference a large range of locations, so here to achieve this objective i.e., to fit this large range of location in address field, a variety of addressing techniques has been employed. As they reduce the number of field in the addressing field of the instruction.

Thus, Addressing Modes are very vital in Instruction Set Architecture(ISA).

some notations are

A= Contents of an address field in the instruction

R= Contents of an address field in the instruction that refers to a register

EA= Effective Address(Actual address) of location containing the referenced operand.

(X)= Contents of memory location x or register X.

Types Of Addressing Modes

Various types of addressing modes are:

1. Implied and Immediate Addressing Modes
2. Direct or Indirect Addressing Modes
3. Register Addressing Modes
4. Register Indirect Addressing Mode
5. Auto-Increment and Auto-Decrement Addressing Modes
6. Displacement Based Addressing Modes

1.Implied and Immediate Addressing Modes:

Implied Addressing Mode:

Implied Addressing Mode also known as "Implicit" or "Inherent“ addressing mode is the addressing mode in which, no operand(register or memory location or data) is specified in the instruction. As in this mode the operand are specified implicit in the definition of instruction.

“Complement Accumulator” is an Implied Mode instruction because the operand in the accumulator register is implied in the definition of instruction. In assembly language it is written as:

CMA: Take complement of content of AC Similarly, the instruction,

RLC: Rotate the content of Accumulator is an implied mode instruction.

All Register-Reference instruction that use an accumulator and Zero-Address instruction in a Stack Organised Computer are implied mode instructions, because in Register reference operand implied in accumulator and in Zero-Address instruction, the operand implied on the Top of Stack.

Immediate Addressing Mode:

In Immediate Addressing Mode operand is specified in the instruction itself. In other words, an immediate mode instruction has an operand field rather than an address field, which contain actual operand to be used in conjunction with the operand specified in the instruction. That is, in this mode, the format of instruction is:

As an example: The Instruction:

MVI 06 Move 06 to the accumulator

ADD 05 ADD 05 to the content of accumulator

- One of the operand is mentioned directly.
- Data is available as a part of instruction.
- Data is 8 0r 16 bit long.
- No memory reference is needed to fetch data

Immediate Mode :Eg.

Example 1 :

MOV CL, 03H

03 – 8 bit immediate source operand

CL – 8 bit register destination operand

Example 2:

ADD AX, 0525H

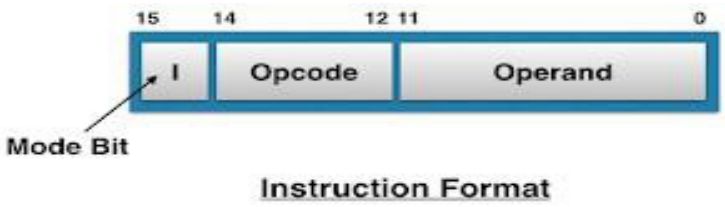
0525 – 16 bit immediate source operand

AX – 16 bit register destination operand.

2. Direct and Indirect Addressing Modes

The instruction format for direct and indirect addressing mode is shown below:

It consists of 3-bit opcode, 12-bit address and a mode bit designated as(I).**The mode bit (I) is zero for Direct Address and 1 for Indirect Address**. Now we will discuss about each in detail one by one.



Instruction Format

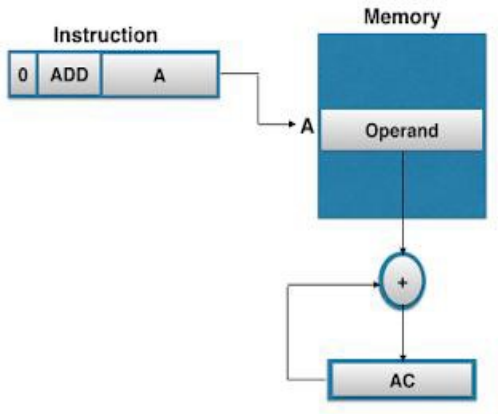
Direct Addressing Mode

Direct Addressing Mode is also known as "Absolute Addressing Mode". In this mode the address of data(operand) is specified in the instruction itself. That is, in this type of mode, the operand resides in memory and its address is given directly by the address field of the instruction. Means, in other words, in this mode, the address field contain the Effective Address of operand i.e., EA=A

As an example: Consider the instruction:

ADD A Means add contents of cell A to accumulator .

It Would look like as shown below:



Here, we see that in it Memory Address=Operand.

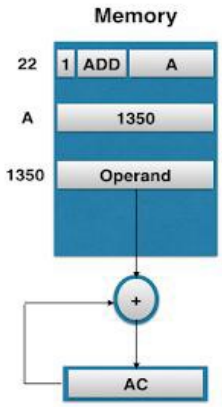
Indirect Addressing Mode:

In this mode, the address field of instruction gives the memory address where on, the operand is stored in memory. That is, in this mode, the address field of the instruction gives the address where the "Effective Address" is stored in memory. i.e., EA=(A)

Means, here, Control fetches the instruction from memory and then uses its address part to access memory again to read Effective Address.

As an example: Consider the instruction:

ADD (A) Means adds the content of cell pointed to contents of A to Accumulator. It look like as shown in figure below:



Thus in it, AC <-- M[M[A]]

[M=Memory]

i.e., (A)=1350=EA

3.Register Addressing Mode:

In Register Addressing Mode, the operands are in registers that reside within the CPU. That is, in this mode, instruction specifies a register in CPU, which contain the operand. It is like Direct Addressing Mode, the only difference is that the address field refers to a register instead of memory location.

i.e., EA=R

It look like as:

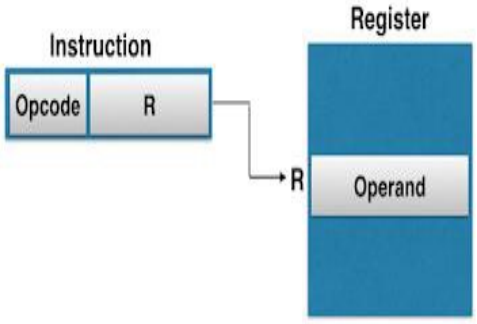
Example of such instructions are:

MOV AX, BX Move contents of Register BX to AX

ADD AX, BX Add the contents of register BX to AX

Here, AX, BX are used as register names which is of 16-bit register.

Thus, for a Register Addressing Mode, there is no need to compute the actual address as the operand is in a register and to get operand there is no memory access involved



4. Register Indirect Addressing Mode:

In Register Indirect Addressing Mode, the instruction specifies a register in CPU whose contents give the operand in memory. In other words, the selected register contain the address of operand rather than the operand itself. That is,

i.e., EA=(R)

Means, control fetches instruction from memory and then uses its address to access Register and looks in Register(R) for effective address of operand in memory.

It look like as:

Here, the parentheses are to be interpreted as meaning contents of.

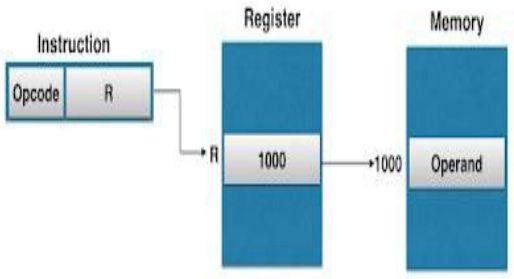
Example of such instructions are:

MOV AL, [BX]

Code example in Register:

MOV BX, 1000H

MOV 1000H, operand



From above example, it is clear that, the instruction(MOV AL, [BX]) specifies a register[BX], and in coding of register, we see that, when we move register [BX], the register contain the address of operand(1000H) rather than address itself.

5. Auto-increment and Auto-decrement Addressing Modes

These are similar to Register indirect Addressing Mode except that the register is incremented or decremented after(or before) its value is used to access memory. These modes are required because when the address stored in register refers to a table of data in memory, then it is necessary to increment or decrement the register after every access to table so that next value is accessed from memory.

Thus, these addressing modes are common requirements in computer.

Auto-increment Addressing Mode:

Auto-increment Addressing Mode are similar to Register Indirect Addressing Mode except that the register is incremented after its value is loaded (or accessed) at another location like accumulator(AC).

That is, in this case also, the Effective Address is equal to

EA=(R)

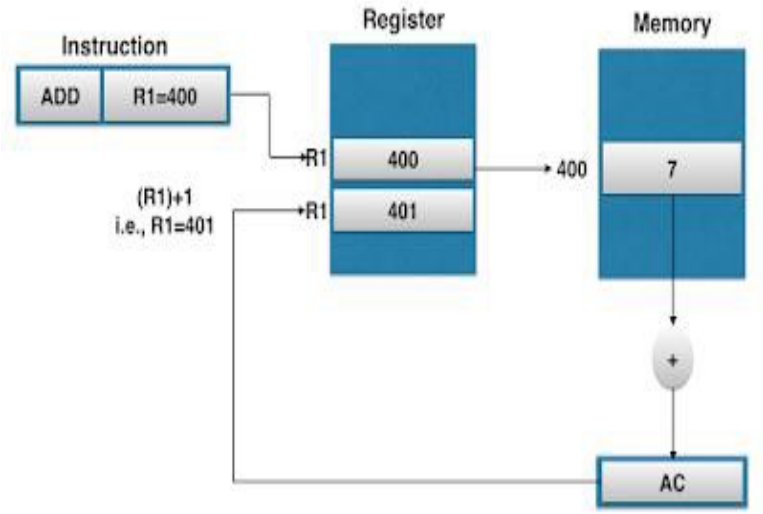
But, after accessing operand, register is incremented by 1.

As an example:

It look like as shown below:

Here, we see that effective address is (R)=400 and operand in AC is 7. And after loading R1 is incremented by 1.It becomes 401.

Means, here we see that, in the Auto-increment mode, the R1 register is increment to 401 after execution of instruction.



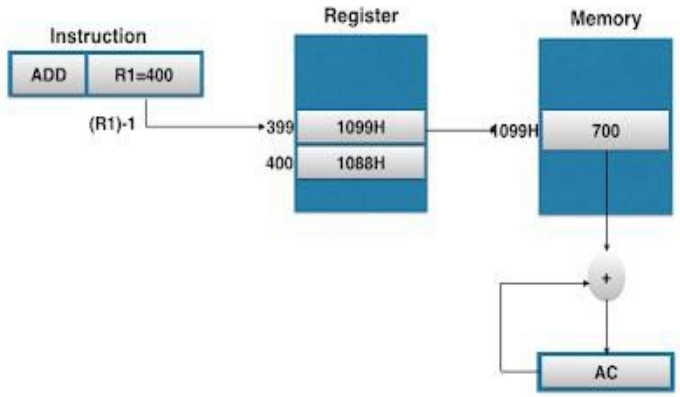
Auto-decrement Addressing Mode:

Auto-decrement Addressing Mode is reverse of auto-increment , as in it the register is decrement before the execution of the instruction. That is, in this case, effective address is equal to

EA=(R) - 1

As an example:

It look like as shown below:



Here, we see that, in the Auto-decrement mode, the register R1 is decremented to 399 prior to execution of the instruction, means the operand is loaded to accumulator, is of address 1099H in memory, instead of 1088H.Thus, in this case effective address is 1099H and contents loaded into accumulator is 700.

6. Displacement Based Addressing Modes:

Displacement Based Addressing Modes is a powerful addressing mode as it is a combination of direct addressing or register indirect addressing mode. i.e., EA=A+(R)

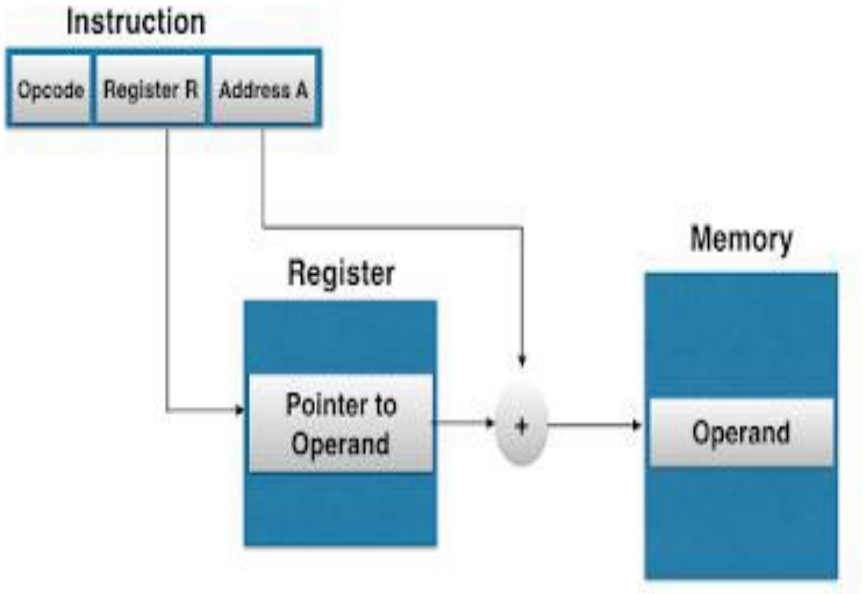
Means, Displacement Addressing Modes requires that the instruction have two address fields, at least one of which is explicit means, one is address field indicate direct address and other indicate indirect address.

That is, value contained in one addressing field is A, which is used directly and the value in other address field is R, which refers to a register whose contents are to be added to produce effective address.

There are three areas where Displacement Addressing modes are used. In other words, Displacement Based Addressing Modes are of three types. These are:

1. Relative Addressing Mode
2. Base Register Addressing Mode
3. Indexing Addressing Mode

Now we will explore to each one by one.



1. Relative Addressing Mode:

In Relative Addressing Mode , the contents of program counter is added to the address part of instruction to obtain the Effective Address.

That is, in Relative Addressing Mode, the address field of the instruction is added to implicitly reference register Program Counter to obtain effective address.

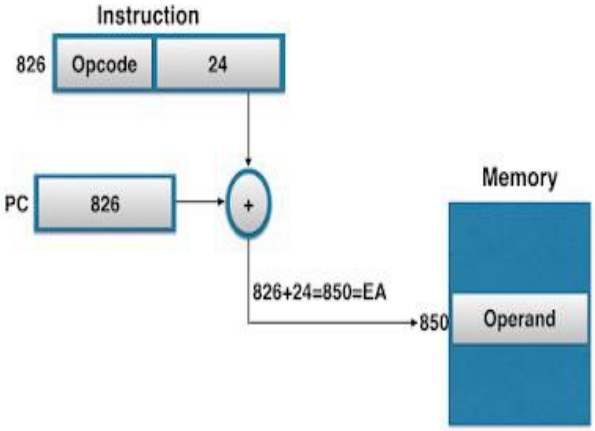
i.e., EA=A+PC

It becomes clear with an example:

Assume that PC contains the no.- 825 and the address part of instruction contain the no.- 24, then the instruction at location 825 is read from memory during fetch phase and the Program Counter is then incremented by one to 826.

The effective address computation for relative address mode is 26+24=850

Thus, Effective Address is displacement relative to the address of instruction. Relative Addressing is often used with branch type instruction



2. Index Register Addressing Mode

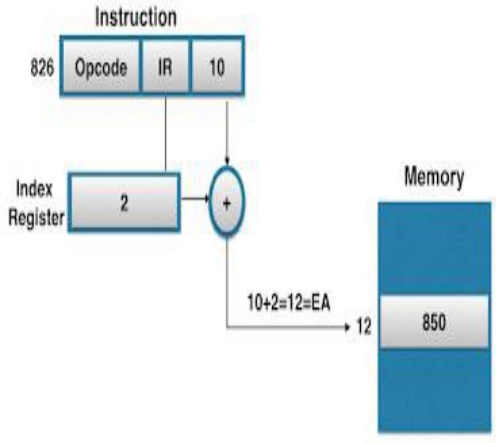
In indexed addressing mode, the content of Index Register is added to direct address part(or field) of instruction to obtain the effective address. Means, in it, the register indirect addressing field of instruction point to Index Register, which is a special CPU register that contain an Indexed value, and direct addressing field contain base address.

As, indexed type instruction make sense that data array is in memory and each operand in the array is stored in memory relative to base address. And the distance between the beginning address and the address of operand is the indexed value stored in indexed register.

Any operand in the array can be accessed with the same instruction, which provided that the index register contains the correct index value i.e., the index register can be incremented to facilitate access to consecutive operands.

Thus, in index addressing mode

EA=A+Index



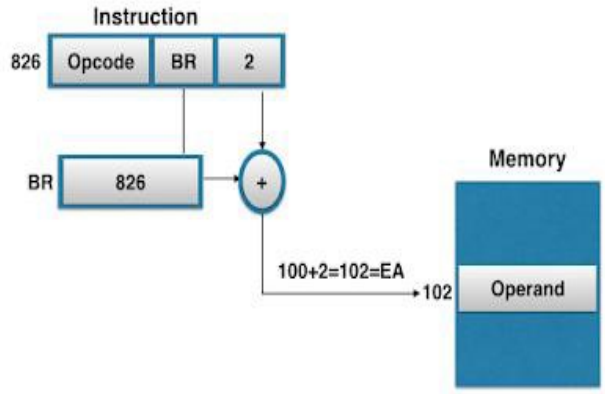
3. Base Register Addressing Mode:

In this mode, the content of the Base Register is added to the direct address part of the instruction to obtain the effective address.

Means, in it the register indirect address field point to the Base Register and to obtain EA, the contents of Instruction Register, is added to direct address part of the instruction.

This is similar to indexed addressing mode except that the register is now called as Base Register instead of Index Register.

That is, the EA=A+Base



Thus, the difference between Base and Index mode is in the way they are used rather than the way they are computed. An Index Register is assumed to hold an index number that is relative to the address part of the instruction. And a Base Register is assumed to hold a base address and the direct address field of instruction gives a displacement relative to this base address.

Thus, the Base register addressing mode is used in computer to facilitate the relocation of programs in memory. Means, when programs and data are moved from one segment of memory to another, then Base address is changed, the displacement value of instruction do not change.

So, only the value of Base Register requires updation to reflect the beginning of new memory segment.