6.1 Addressing Modes: introduction

Addressing modes are an aspect of the instruction set architecture in most CPU designs. The various addressing mode helps to identify the type of operands in the instruction. An addressing mode specifies how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction or elsewhere. It is one of the important threads(المواضيع) that attract full attention of compiler writers and those who programmed using assembly language directly. Be think with the following instructions:

add R1,x

add R1,500

add R1,[x]

What is the similarity of these instructions?

The answer is: all of these instructions implement adding operation on R1 and all of them have two addresses.

Now ,What is the difference among the three instructions?

The answer is: the value that added to R1.

In the first instruction the value stored in a memory address called (X) , here the CPU has one visit to the memory .

The second instruction has the value directly which means that there is no need to visit the main memory.

The third has an address (x) contain another address (say y) which is the memory address that stores the value. the cpu has to visit the memory twice.

So, the three instructions differ in storing, speed of implementation, and the manner of implementation. This is because the three instructions have different ***addressing mode***s .

The Big question is :

How the computer knows the type of operand

(value , address contains a value, address contains another address) ?

Well,

1- The instruction passes to (decoding cycle) after the fetch .

2- The problem explained that the instructions described so far do not give any hint about the type of operand, so the format is modified by adding another field which is called (addressing mode).

Now :

How can we determine the number of bits for the new field "addressing mode "?

The length of addressing mode field is inferred from ***the No. of addressing mode supported by the processor.*** Convert the No. of supported addressing mode to the form ***2^n ,where n is the No. of bits for (addressing mode).*** For example, if the computer supports 8 addressing modes, then the No. of bits in the addressing mode field will be :

$$8 \Rightarrow 2^3 \Rightarrow n=3$$

The no. of bits =3.

6-2 The types of addressing mode

Computer architectures differ in the number of addressing modes they support. Most RISC machines have only about five simple addressing modes, while CISC machines such as the DEC VAX supermini have over a dozen addressing modes, some of which are very complex.

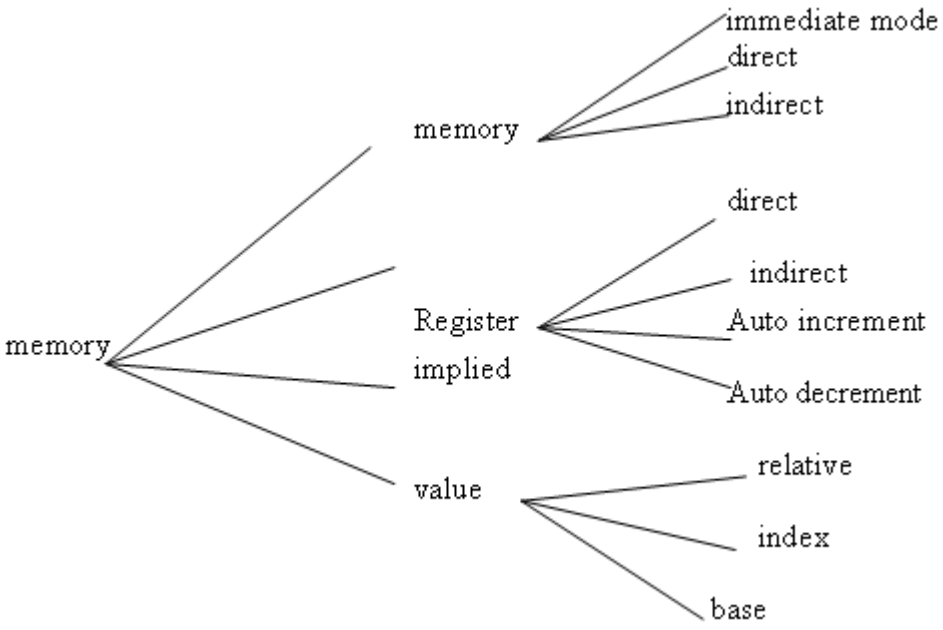


Figure 6.1 : the typical addressing modes

The IBM System/360 mainframe had only three addressing modes; a few more have been added for the System/390. This leads us to the presence of several types of addressing modes and The common types are summarizes by figure 6.1 .It's time to go into detail...

6.2.1 Memory addressing modes

As we see in the figure 6.1, there are three types fall within this category :

a) ***Immediate***: the value is stored directly in the instruction , example to this mode : add R1,500

address	Content
300	add R1,500

Memory

B) ***direct :*** the address of the value (called effective address) is stored directly in the instruction, for example : add R1,x

Address	Content
……..	
300	add R1,x
……	
1000	30

From the figure above, what is the effective address? What is the value of x?

Your Answer: ----------

---

c) ***Indirect mode*** : the instruction stores the address of the effective address , for example :add R1,[x]

address	Content
30	add R1,[x]
31	700
.....	
700	30

Memory

6.2.2 Register addressing modes

As we see in the figure 6.1, there are four types fall within this category :

a) ***direct:*** the value is stored directly in general register, like

add R1,R2

b)

Q- what is the effective address in this type ?

Your Answer: ----------

indirect: the effective address is stored directly in a general register, for example : add R1, [R2]

Q- what is the effective address in this type ?

Your Answer: ----------

c)***Auto increment*** : it is the same as (register indirect mode) but after the implementation of the operation, the address is incremented by 1, like :

add R1,[R2]+

d) Auto decrement : before the implementation of the operation the address stored in the general register is decremented by 1,like :

add R1, - [R2]

6.2.3. Value (offset)

This type of addressing mode has three subtypes but they share one Law :

EA= value + part of address (instruction)

The value may be in :

a- Pc (relative) : in the case of branch instruction and the Low will be :

EA= pc + part of address (instruction)

b- XR (index) : in the case of matrix. The Low will be :

EA= XR+ part of address (instruction)

c) BR (base) : in the case of complete memory pages from location to another. The Low will be :

EA= **B**R+ part of address (instruction)

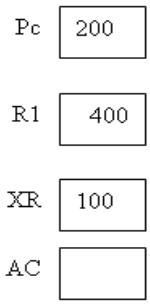
6.2.4 Implied mode

This mode for the instructions consisted of opcode only i.e. the instruction related with AC or STACK.

Q- What is the effective address in this type?

Your Answer: ----------

Example: Suppose you have the following part of memory , Find the effective address by using all types of addressing mode



address	contenet
200	Load to Ac
201	500
202	Next inst.
	. .
399	450
400	700
	. .
500	800
	. .
600	900
702	325
	. . .
800	300

SOL:

Addressing mode		EA	Operand Value
Memory	immediate	201	500
	direct	500	800
	indirect	800	300
Register	direct	-	400
	indirect	400	700
	Auto inc	400	700
	Auto dec	399	450
Value	index	600	900
	relative	702	325

Q/base and implied mode are not included in the solution, why?

Your answer :----------

6.3 instruction format again!!

A computer has a memory size 256 KW where word is 32 bit, the instruction is stored in one location. The computer has 5 addressing mode. The instruction has 3 parts:

Opcode, addressing mode ,address field.

1-Draw the instruction format and indicate the No. of bits in each field.

2- How many address lines in this computer?

Submit your answer in our group on facebook…..