CHAPITRE

8

Les modes d'adressage

I- Introduction :

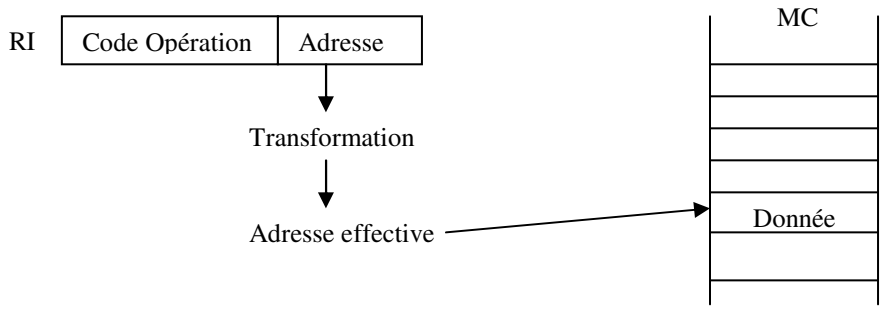
II- Les modes d'adressage

III- Influence du mode d'adressage sur le cycle d'exécution d'une instruction

I- Introduction :

Une instruction est formée d'une actions et des opérandes nécessaire à son exécution. Cependant le programmeur doit indiquer au microprocesseur comment accéder aux données des opérandes. La façon dont le microprocesseur doit suivre pour trouver la donnée en mémoire s'appelle mode d'adressage.

L'**adressage** est une technique permettant la correspondance entre le contenu de la zone d'adresse de l'instruction qui se trouve dans le registre d'instruction et l'adresse réelle qui sera envoyée sur le bus d'adresse (adresse effective).



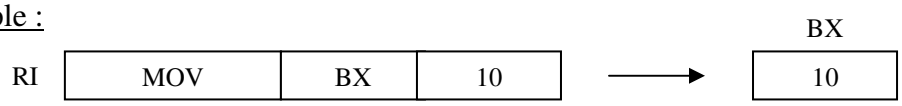
Les opérandes sont des données, devant être stockées en mémoire. Les manières d'accéder à ces données sont multiples.

I- Les modes d'adressage :

1. Adressage immédiat:

On place les données directement dans l'instruction.

Exemple :



=>Pas besoin d'une adresse effective

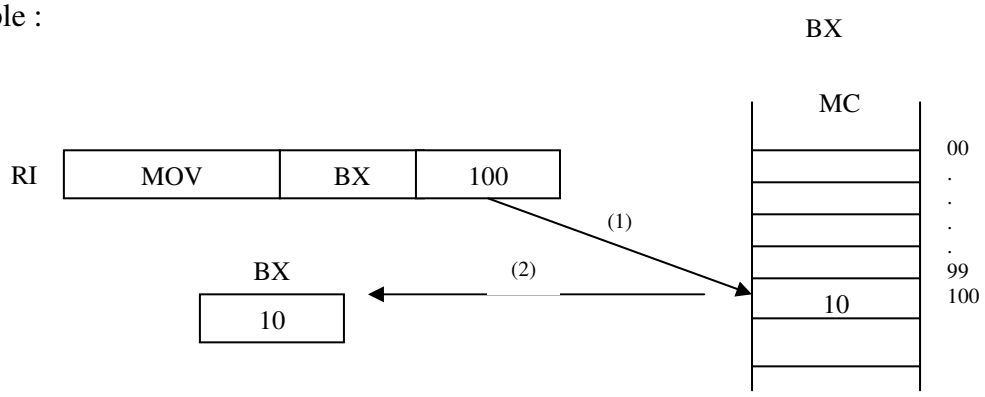
Problème: on est limité en codage de donnée ;

Avantage : c'est le mode d'adressage le plus rapide vue qu'il ne nécessite pas un accès à la mémoire.

2. Adressage direct:

On place dans l'instruction l'adresse d'une case de la mémoire principale qu'on va aller chercher ensuite ;

Exemple :



:Pointer versl'adresse case 100.

:Ecrire dans BX.

=> 100 est l'adresse effective.

Avantage : c'est la façon l plus simple de spécifier une opérande en écrivant son adresse.

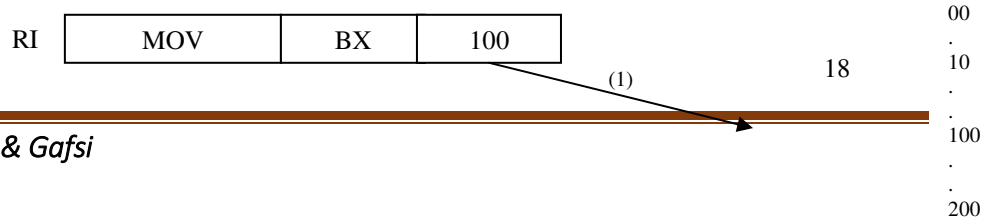
Adressage par registre:

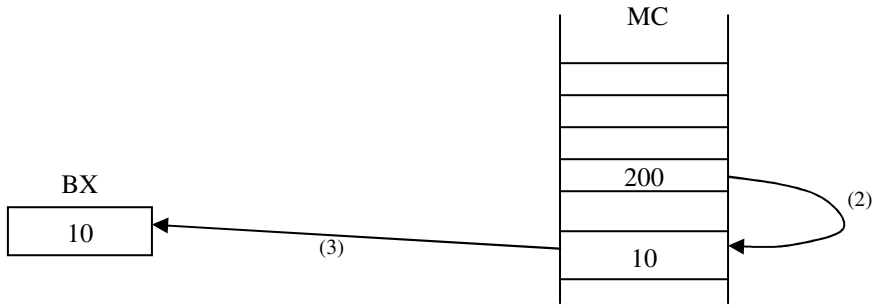
On place la donnée dans un registre et on place l'adresse du registre dans l'instruction. Cet adressage requiert un temps plus court car les registres étant plus petits et leur accès est plus rapide;

3. Adressage indirect:

On place l'adresse d'un registre ou d'une case mémoire qui contient l'adresse effective de l'opérande en mémoire principale ;

Exemple :





(1): Pointer vers l'adresse case 100.

(2): lire le contenu de la case d'adresse 100 et se déplacer suivant son contenue vers l'adresse 200.

(3) : Ecrire dans BX le contenu de l'adresse 200.

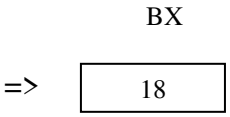
=> [100] est l'adresse effective => 200 est l'adresse effective.

Remarque :

On peut avoir plusieurs niveaux d'indirections :

Exemple avec 2 niveaux d'indirections :

On a : [[100]] est l'adresse effective => [200] est l'adresse effective => 10 est l'adresse effective.



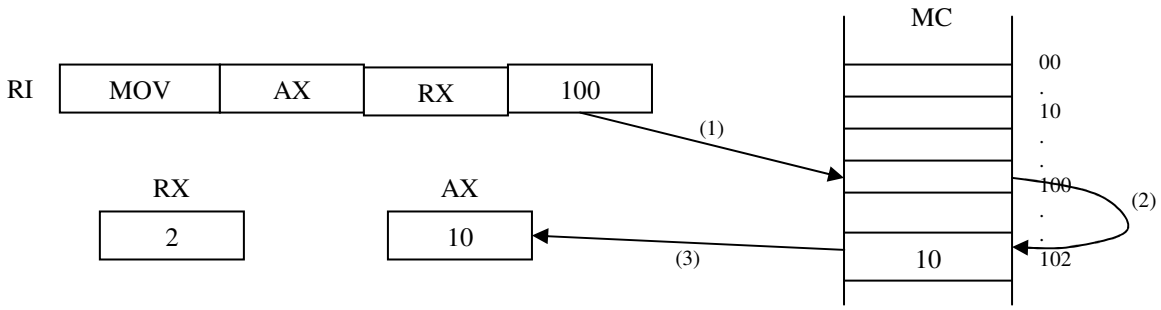
Avantage : Cette méthode permet de garder la séquentialité des instructions pour que le compteur ordinal garde son ordre.

4. adressage indexé:

On définit des plages d'adressage et on se contente de donner l'index de la donnée dans cette plage (plus courte) ; Ce mode utilise un ou plusieurs registres spéciaux appelés registres d'indexes.

Le déplacement fourni dans la zone adresse de l'instruction est ajouté au contenu du registre d'index spécifié.

Exemple :



(1): Pointer vers l'adresse case 100.

(2): Se déplacer suivant le contenue du registre d'index de 2 cases vers l'adresse 102.

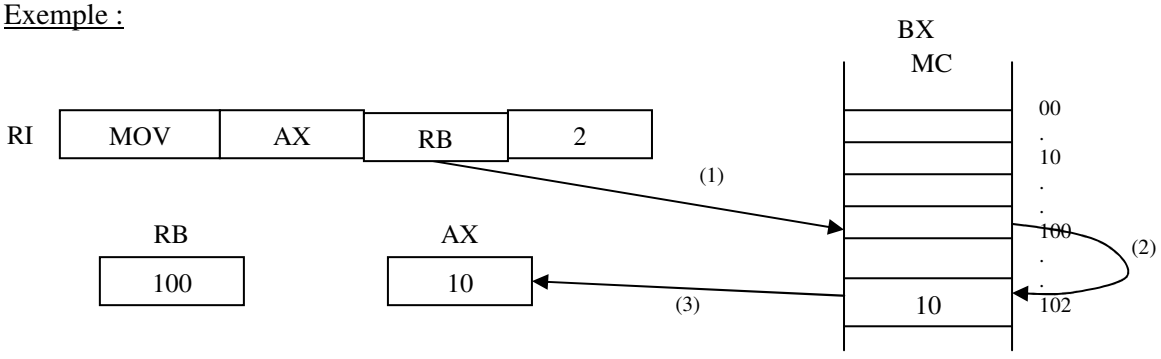
(3) : Ecrire dans AX le contenu de l'adresse 102.

⇨ 100 + [RX] est l'adresse effective => 102 est l'adresse effective.

5. Adressage basé:

La valeur indiquée dans la zone adresse de l'instruction constitue un déplacement par rapport à ne adresse mémoire dont la valeur se trouve dans un registre spécial appelé registre de base.

Exemple :



(1): Pointer vers l'adresse case 100.

(2): Se déplacer suivant le déplacement contenu dans la zone d'adresse = 2 cases.

(3) : Ecrire dans AX le contenu de l'adresse 102.

⇨ [RB]+ 2 est l'adresse effective => 102 est l'adresse effective.

Exemple :

Ce mode d'adressage est utile pour rechercher les adresses des portions d'un programme.

6. Adressage relatif:

L'adresse effective est obtenue à partir de la somme du compteur ordinal avec une consante qui se trouve dans l'instruction.

Exemple :

Instruction de branchement : RI

JMP	@

L'adresse effective = [CO] + /- @

7. Adressage implicite:

On peut trouver des action qui n'indiquent pas explicitement une opérande ou une donnée car ils se réfèrent implicitement à l'un des registre de L'unité centrale de traitement.

La donnée à traiter doit être mise d'avance dans le registre correspondant.

II- Influence du mode d'adressage sur le cycle d'exécution d'une instruction :

Lors de la compilation d'un programme, le code de l'instruction généré contient à part l'action, le mode d'adressage employé pour accéder aux opérandes.

Ainsi une même action peut être codée différemment selon le mode d'adressage employé dans l'instruction.

Exemple :

Opération d'addition :

	Instruction	Code de l'instruction
1	ADD A ,5	F9 A0B0 05
2	ADD AX ,5	F8 05

La première instruction est une addition immédiate avec une variable de la mémoire et la deuxième instruction présente une addition immédiate avec un registre.

Il est clair que la première opération d'addition sera plus coûteuse en terme de temps vu qu'elle nécessite un accès à la mémoire centrale.

D'où le mode d'adressage utilisé lors d'une opération influe le temps d'exécution de cette instruction.