



## Chapitre 2 : Les modes d'adressage

Le 6809 est très complet au niveau des modes d'adressage. On appelle mode d'adressage la façon dont les instructions accèdent aux opérandes (données situées en mémoire). La plupart des instructions du 6809 supportent plusieurs types d'adressage.

### 1. Adressage inhérent (ou implicite)

#### *Adressage inhérent simple*

Dans le mode d'adressage inhérent simple, le code opération contient toute l'information nécessaire à l'exécution de l'instruction.

Exemple : CLRA,CLRB,INCA,INCB,LSRA,LSRB

#### *Adressage inhérent paramétré*

L'instruction comporte un octet supplémentaire qui précise les registres internes intervenant dans l'instruction.

Exemples :

TFR A,DP                      transfert de A dans DP

PSHS A,B,X                    sauvegarde dans la pile de A,B,X

### 2. Adressage immédiat

Dans ce mode d'adressage, le code opération est suivi directement de l'opérande. Ce type d'adressage permet de charger les registres internes du microprocesseur avec la valeur de l'opérande.

Exemple :

LDA #\$12                      charger la valeur \$10 dans l'accumulateur A

Le symbole '#' signifie immédiat dans la syntaxe assembleur.

LDX #\$E000                    charger la valeur \$E000 dans le registre d'index X

### 3. Adressage direct

Dans ce mode d'adressage, le code opération est suivi des 8 LSB de l'adresse dont les 8 MSB se trouvent dans le registre de page DP. Ce mode d'adressage est utilisé pour adresser des pages en mémoire. Le registre DP spécifie le numéro de page.

Il suffit donc d'initialiser le registre de page DP pour pouvoir travailler en adressage direct sur les 256 octets de la page choisie. A la mise sous tension, DP est mis à zéro.

#### *Instructions sur deux octets*

Le premier octet définit le code opératoire, le second les 8 LSB de l'adresse. Exemple :

LDA \$10 ou LDA <\$10          charger l'accumulateur A avec le contenu de \$E010 (DP=\$10).

Le symbole '<' signifie direct dans la syntaxe assembleur.

#### *Instructions sur trois octets*

Les deux premiers octets définissent le code opératoire, le troisième les 8 LSB de l'adresse.

Exemple :

LDY \$10          charger le registre Y avec les contenus de \$E010 et de \$E011 (DP=\$10).

### 4. Adressage étendu

Dans ce mode d'adressage, le code opération est suivi les deux octets constituant l'adresse de l'opérande proprement dit.

#### *Instructions sur trois octets*

Le premier octet définit le code opératoire, les deux autres constituent l'adresse de l'opérande.

Exemple :

LDA \$E000          charger l'accumulateur A avec le contenu de \$E000

#### *Instructions sur quatre octets*

Le code opératoire est défini par deux octets. Les deux autres constituent l'adresse de l'opérande.

Exemple :

LDY \$E000          chargement du registre Y avec le contenu de \$E000



## 5. Adressage étendu indirect

L'opérande spécifie non pas la donnée mais l'adresse de cette donnée. On parle d'indirection.

L'assembleur reconnaît la présence d'un adressage étendu indirect par mise entre [ ] de l'adresse opérande.

Exemple :

LDA [ \$E000]            charger l'accumulateur A avec le contenu dont l'adresse se trouve en \$E000 et \$E001

## 6. Adressage indexé

Le principe de l'adressage indexé est que l'instruction spécifie une base (index) plus un déplacement par rapport à cette base. On peut donc écrire :

**Adresse effective = base + déplacement**

La base peut être soit :

- un des deux registres d'index X ou Y;
- un des deux pointeurs de pile U ou S;
- le compteur programme PC (cas particulier de l'adressage relatif).

Quant au déplacement, il peut être soit :

- nul;
- codé sur 5 bits (-16 à +15);
- codé sur 8 bits (-32768 à +32767);
- variable dans le cas de l'utilisation d'un accumulateur A, B ou D.

### **Adressage indexé. Déplacement nul**

Ce mode est le mode indexé le plus rapide. Un registre (X,Y,U ou S) contient l'adresse effective de la donnée utilisée par l'instruction. Il existe deux types d'instruction :



Exemple :

Langage machine		Langage assembleur		Commentaires
Adresse	Contenu	Code op.	Opérande	
	A6 84	LDA	,X	charger l'accumulateur A avec le contenu dont l'adresse se trouve dans

**Adressage indexé. Déplacement constant**

Dans le mode d'adressage indexé, le déplacement (en complément à 2) est ajouté au contenu du registre de base pour former l'adresse effective de la donnée. Le contenu initial du registre de base n'est modifié par l'addition.

Exemple :

Langage machine		Langage assembleur		Commentaires
Adresse	Contenu	Code op.	Opérande	
	A6 88 14	LDA	+20,X	Charger l'accu. A par le contenu de l'adresse : (X) + 20

**Déplacement sur 16 bits (n,R)**

Le déplacement sur 16 bits, en deux octets, suit immédiatement le post octet (*1RR01001*).

Exemple :

Langage machine		Langage assembleur		Commentaires
Adresse	Contenu	Code op.	Opérande	
	A6 89 01 00	LDA	+256,X	Charger l'accu. A par le contenu de l'adresse : (X) + 256

*Adressage indexé. Auto-incrémentation/décrémentation*



Ce mode est utilisé pour gérer une table de données. L’auto-incrémentation/décrémentation permet d’accéder rapidement aux données successives. le registre de base contient ’adresse effective de la donnée de l’instruction.

Dans le mode auto-incrémentation, le registre de base est incrémenté de un ‘+’ ou de deux ‘++’ avant le recherche de l’instruction suivante.

Dans le mode auto-décrémentation, le registre de base est décrémenté de un ‘-’ ou de deux ‘--’ avant le recherche de l’instruction suivante.

Exemple :

Langage machine		Langage assembleur		Commentaires
Adresse	Contenu	Code op.	Opérande	
	A6 80	LDA	,X+	charger l’accumulateur A avec le contenu dont l’adresse se trouve dans X. Incrémentation par Un de X.
	A6 81	LDA	,X++	charger l’accumulateur A avec le contenu dont l’adresse se trouve dans X. Incrémentation par deux de X.
	A6 82	LDA	,-X	charger l’accumulateur A avec le contenu dont l’adresse se trouve dans X. Décrémenté par Un de X.
	A6 83	LDA	,--X	charger l’accumulateur A avec le contenu dont l’adresse se trouve dans X. Décrémenté par deux de X.