



Microprocesseur 6800

*Registres et modes d'adressage
Immédiat, Direct et Etendu*



Réalisé par N.NASRI

(www.blogmatlab.blogspot.com)

2010

Modes d'adressage (Immédiat, Direct/Étendu)

Adressage immédiat :

1 - Registres de données (accumulateurs A et B) :

LDA A # 25 : Charger l'accumulateur A par (25)_{hexa}.

Avant :

A : 8bit

Après:

A : 8bit

Nota:

- Dans ce cas le **registre de données A** est sur **8bits** donc l'**opérande** sera forcément sur **8 bits**.
- La présence du # indique au processeur de mettre **directement** dans **A** la **valeur** de l'opérande (25)_{hexa} (qui est une **donnée**) **sans passer par la mémoire**.
- De même pour l'accumulateur **B**.

2 - Registre d'index:

LDX # 0025 : Charger le registre d'index par (0025)_{hexa}.

Avant :

X : 16 bits

Après:

X : 16bits

Nota:

- Dans ce cas le **registre d'index** est sur **16bits** donc l'**opérande** sera forcément sur **16 bits**.
- La présence du # indique au processeur de mettre **directement** dans **X** la **valeur** de l'opérande (0025)_{hexa} (qui peut être un **adresse**) **sans passer par la mémoire**.

3 – Registre SP (pointeur de pile) :

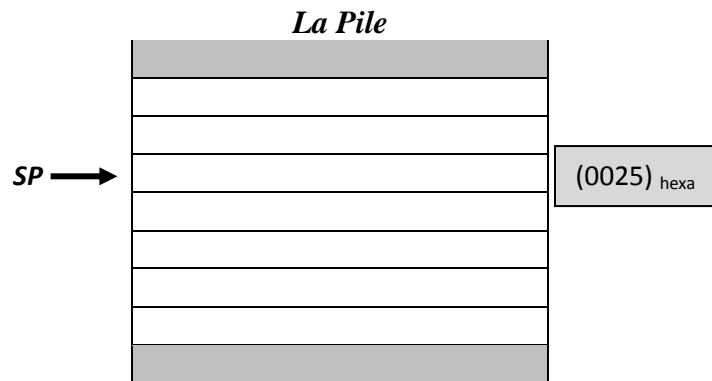
LDS # 0025 : Charger le registre SP par (0025)_{hexa}.

Avant :

SP : 16 bits

Après:

SP : 16bits



Nota:

- Dans ce cas le **registre SP** est sur **16bits** donc l'**opérande** sera forcément sur **16 bits**.
- La présence du # indique au processeur de mettre **directement** dans **SP** la **valeur** de l'opérande (0025)_{hexa} (qui est une **adresse**) sans passer par la mémoire.

Adressage Direct/Étendu :

1 - Registres de données (accumulateurs A et B) :

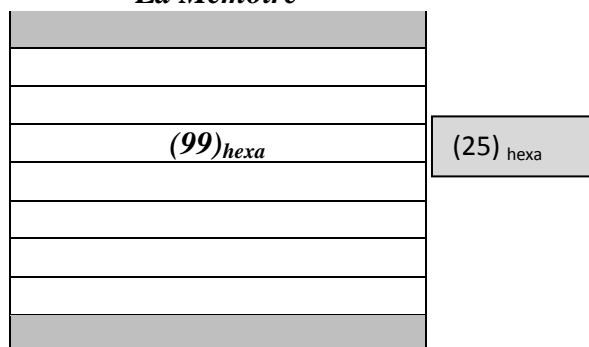
Direct :

LDA A 25 : Charger l'accumulateur A par le contenu de $(25)_{\text{hexa}}$.

Avant :

A : 8bit

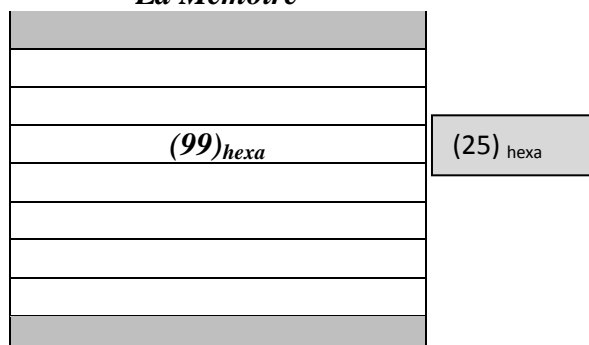
La Mémoire



Après:

A : 8bit

La Mémoire



Nota:

- Dans ce cas l'opérande (qui est une **adresse**) est sur **8 bits**.
- L'adressage Direct indique au processeur de mettre dans A **le contenu** de l'adresse $(25)_{\text{hexa}}$ (Donc le processeur passera **forcément par la mémoire**).
- De même pour l'accumulateur B.

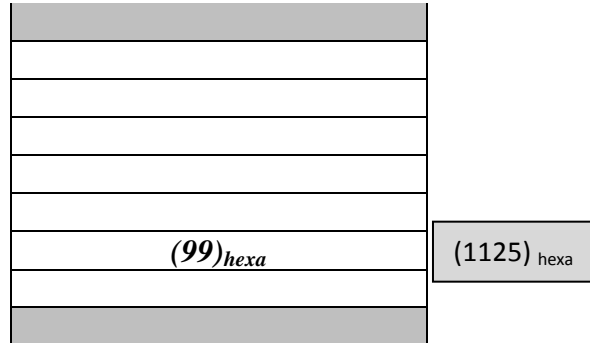
Etendu :

LDA A 1125 : Charger l'accumulateur A par le contenu de $(25)_{\text{hexa}}$.

Avant :

A : 8bit

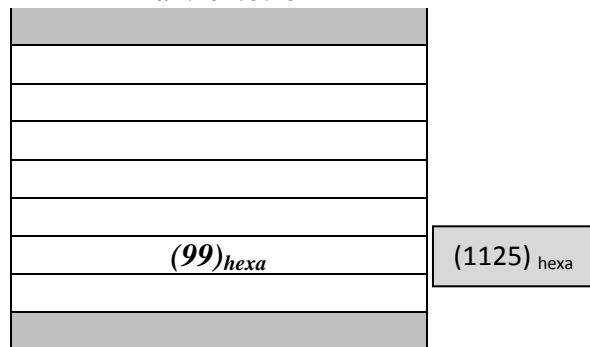
La Mémoire



Après:

A : $(99)_{\text{hexa}}$ 8bit

La Mémoire



Nota:

- Dans ce cas l'opérande (qui est une **adresse**) est sur **16 bits**.
- L'adressage *Etendu* indique au processeur de mettre dans **A le contenu** de l'adresse $(1125)_{\text{hexa}}$ (Donc le processeur passera forcément **par la mémoire**).
- Remarquez que l'adresse est sur **16 bits** alors que son contenu est sur **8 bits**.
- De même pour l'accumulateur **B**.

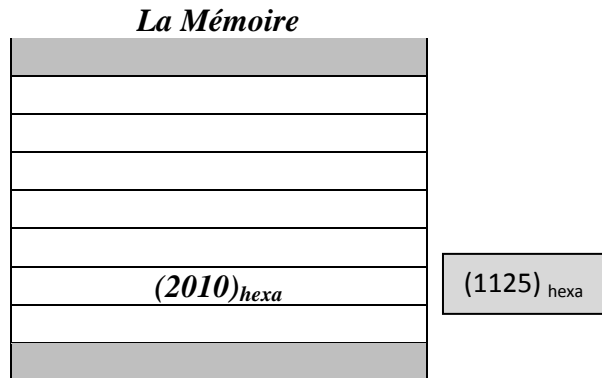
2 - Registre d'index:

Etendu :

LDX 1125 : Charger le registre d'index par le contenu de l'adresse $(1125)_{\text{hexa}}$.

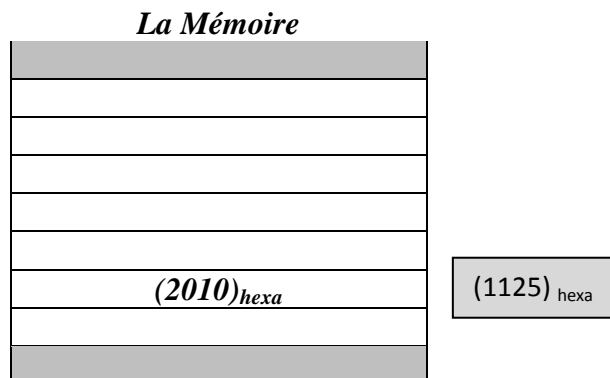
Avant :

X : 16 bits



Après:

X : 16bits



Nota:

- L'adressage *Etendu* indique au processeur de mettre dans **X** le **contenu** de l'adresse $(1125)_{\text{hexa}}$ (Donc le processeur passera forcément **par la mémoire**).

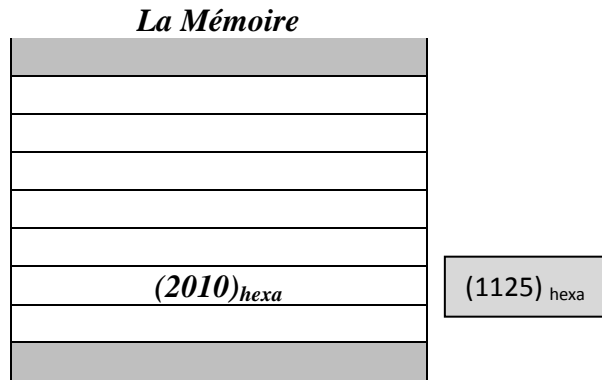
3 – Registre SP (pointeur de pile) :

Etendu :

LDS 1125 : Charger le registre **SP** par le contenu de l'adresse (1125)_{hexa}.

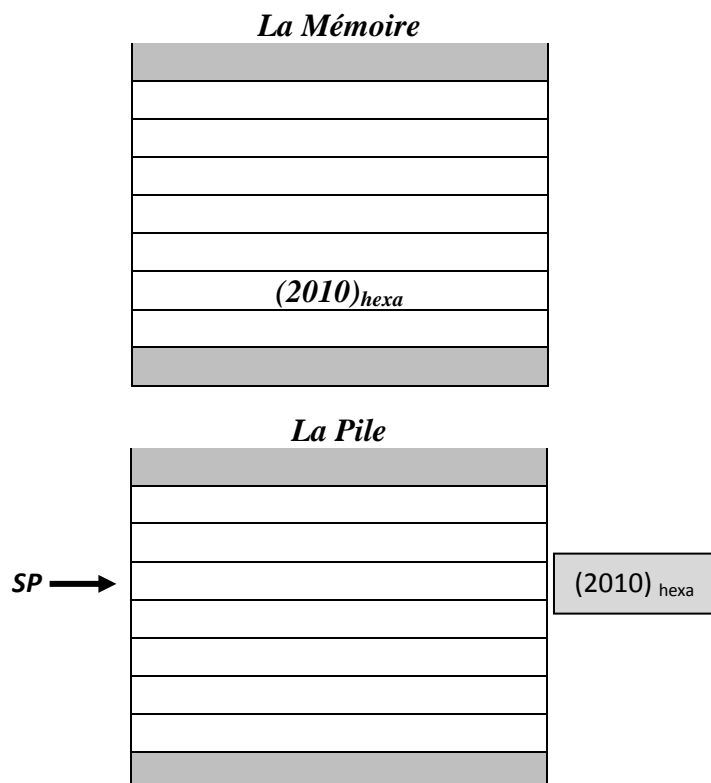
Avant :

SP : 16 bits



Après:

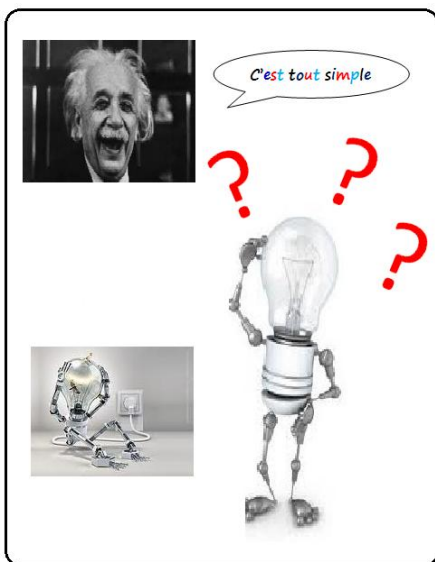
SP : 16bits



Nota:

- L'adressage *Etendu* indique au processeur de mettre dans **SP** le **contenu** de l'adresse (1125)_{hexa} (Donc le processeur passera forcément **par la mémoire**).





...

Site web :

<http://www.blogmatlab.blogspot.com>

Electronique ... C'est tout simple sur Facebook :

<http://www.facebook.com/group.php?gid=115899441759953>

Tweet this

<http://twitter.com/Matlablog>

I Youtube

<http://www.youtube.com/user/nadjibejaia#p/a>