

Objectifs :

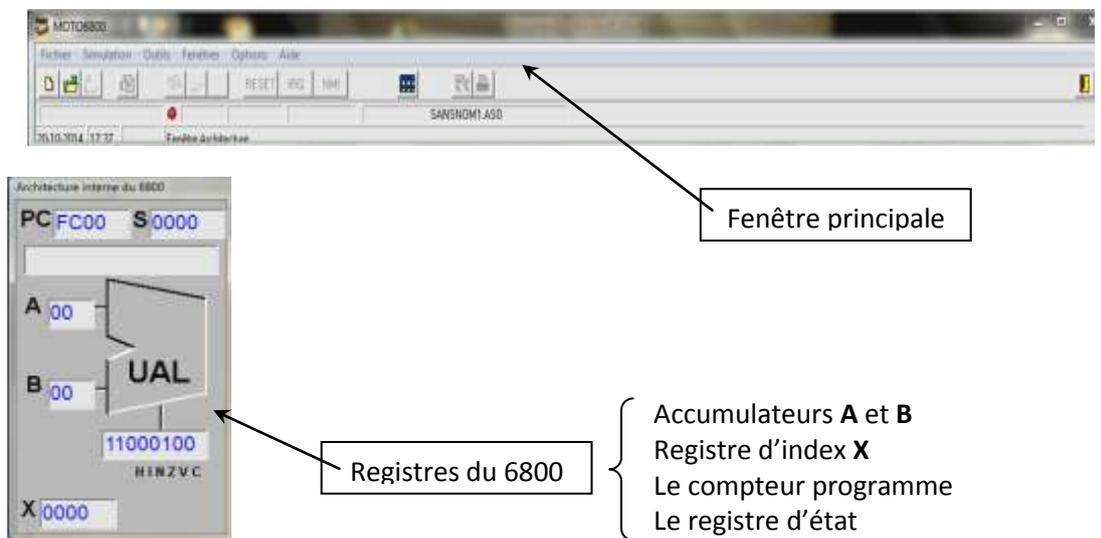
- Se familiariser avec le logiciel MOTO6800.
- Débuté avec le langage assembleur.

Matériel utilisé :

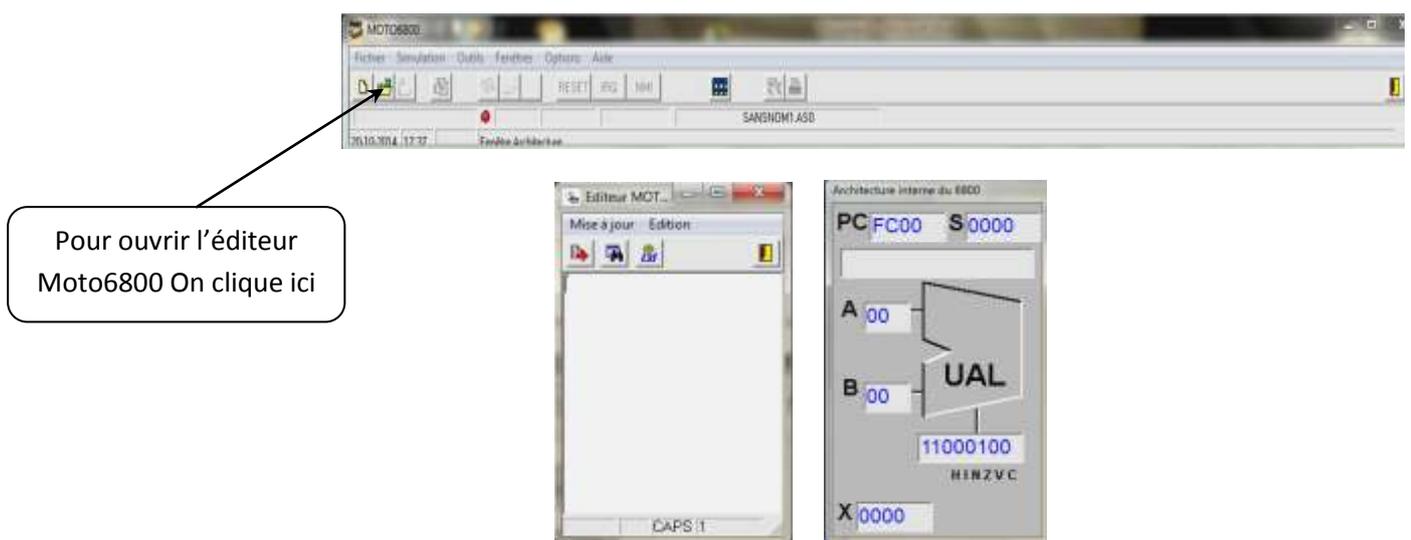
- Un PC.

A / Découvrir Moto6800 :

Ouvrez l'application Moto6800 et surfez sur les différentes fenêtres et appliquez ce qu'on explique ci-dessous.

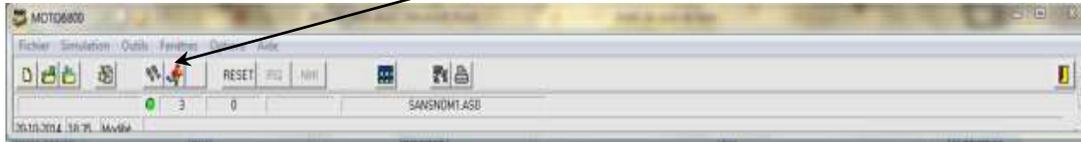


1) Editeur Moto6800 :



TP1 : Architecture et instructions du 6800 ...

Après avoir écrit le programme dans l'éditeur. On l'exécute en cliquant ici

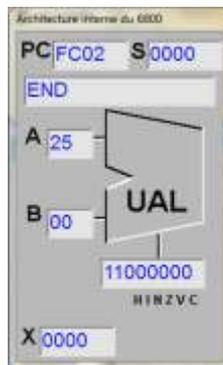


On obtient alors le résultat suivant :



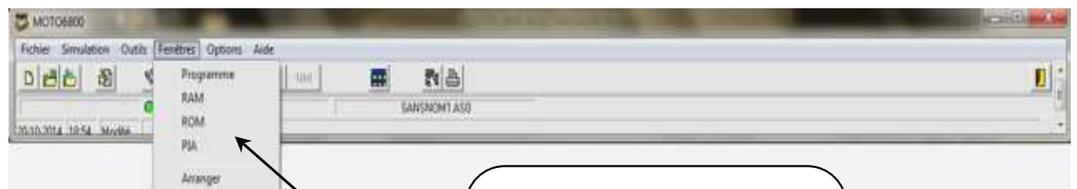
Le bouton d'exécution n'est plus rouge.

Pour pouvoir refaire l'exécution ou dans le cas d'une modification : Cliquez sur le bouton « **Mise à jour** » de la fenêtre **Editeur 6800**



Remarquez les changements dans les registres **PC** et **A**

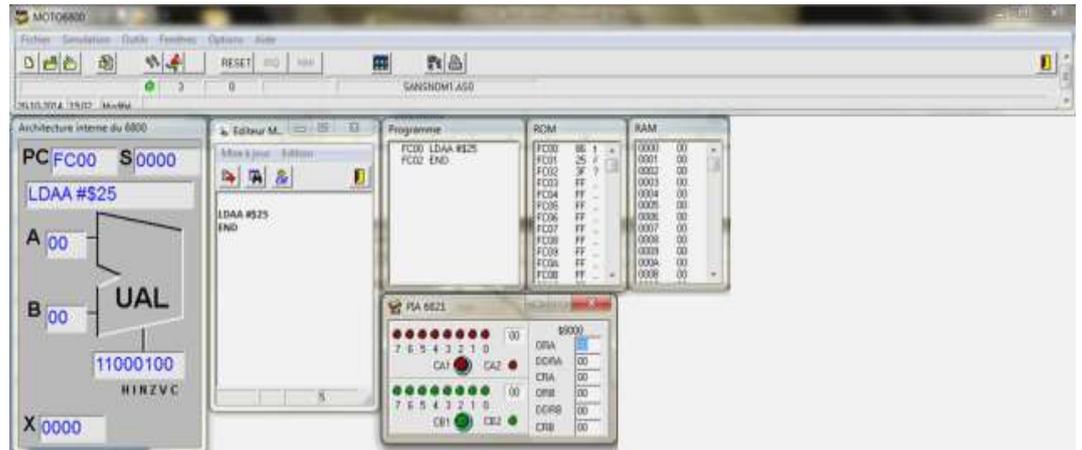
2) Le menu Fenêtre :



Ici on peut choisir les fenêtres à visualiser :

- Programme
- RAM (8 Ko)
- ROM (32 Ko)
- PIA

On peut les ouvrir toutes en même temps.



3) Ouvrir un programme existant :



Pour ouvrir un programme existant
On clique ici

Pour enregistrer un programme
On clique ici

(**Remarque :** Toutes ces actions peuvent se faire à l'aide des menus déroulant de la fenêtre principale).

B / Les registres du 6800 :

1. Ecrivez le code suivant :

```
LDAA #$25  
STAA $0000  
END
```

- Exécutez le code précédent en observant les changements sur les registres et la fenêtre **RAM**.

2. Ecrivez le code suivant :

```
LDAB #$05  
STAB $0002  
END
```

- Exécutez le code précédent en observant les changements sur les registres et la fenêtre **RAM**.
- Que remarquez-vous ?

3. Ecrivez maintenant le code suivant :

```
LDAA #$32  
STAA $0002  
END
```

TP1 : Architecture et instructions du 6800 ...

- Observez le registre **A** et la fenêtre **RAM**.
 - Que remarquez-vous ?
4. Soit le code Assembleur suivant. Après exécution, que contient ?
- Les registres **A, B**
 - Le registre **X**
 - Les cases mémoires **[0000]** et **[0012]**
 - Donner une description pour chaque instruction.

```
LDX #0010
LDAA #25
STAA 0000
LDAA #15
STAA 0012
LDAB 02,X
END
```

C / Les instructions du 6800 :

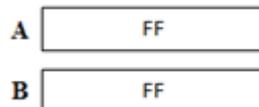
1. Soit maintenant la séquence d'instructions suivante :

```
LDAA #$FF
ASL A
ASL A
LSR A
END
```

- Quel est le contenu de l'accumulateur **A** après exécution de la séquence précédente ?
- Quelle est la valeur de flag **C** du registre d'état ?
- Executer pas à pas le code et dite ce que fait chaque instruction.
- Charger l'accumulateur **B** avec la valeur **#%11111111**
- En utilisant l'instruction de décalage adéquate, comment peut-on obtenir les valeurs suivantes :
 - **1111110** avec : flag **C == 1**
 - **1111111** avec : flag **C == 1**
- Charger maintenant l'accumulateur **A** avec la valeur **#%00001101**
- En utilisant l'instruction de décalage adéquate, comment peut-on obtenir les valeurs suivantes :
 - **00000110** avec : flag **C == 1**
 - **00000011** avec : flag **C == 0**

2. Soit le code Assembleur suivant (Le contenu des registres est illustré par la figure suivante):

```
TAB
TPA
ASLB
ASLB
TAP
TBA
SWI
```



- Que fait ce code ?
- Quel est le contenu des accumulateurs **A, B, X** et le registre d'état (**CCR**) après exécution?
- Ecrire un code qui donnera :

