

I/ Architecture du 6800 ...

Exercice 1 :

I/

1. Quels sont les différents registres du **6800** ?
2. Quelles sont les opérations qui peuvent modifier le registre d'état ?
3. Quelle est l'utilité de ce registre (registre d'état).

II/ Vrai ou faux ?

1. Le **6800** manipule des données de **16** bits.
2. Le **6800** manipule des adresses de **16** bits.
3. Le bus de données est sur **08** bits et il est unidirectionnel.
4. Le bus d'adresses est sur **08** bits et il est unidirectionnel.

III/ Vrai ou faux ?

1. Le flag **I** se met à **1** lors de la demande d'interruption.
2. Le flag **I** se met à **1** après une requête d'interruption acceptée.
3. Le flag **I** se met à **0** lorsqu'il n'y a pas d'interruption.
4. Le flag **I** se met à **0** après une requête d'interruption acceptée.

IV/ - Vrai ou faux ?

- **Quand on réalise l'addition (FF + FF) :**

1. Le résultat est :
2. Le flag **C** est :
3. Le flag **I** reste inchangé.

- **Quand on réalise l'opération (89 + 98) :**

1. Le résultat est :
2. Le flag **C** est :
3. Le flag **I** reste inchangé.

II/ Instructions du 6800...

Exercice 2 :

I/

1. Donner le cycle d'exécution d'une instruction.
2. Quelle est la forme d'une instruction en assembleur.

II/ - Vrai ou faux ?

- **Lors de l'exécution d'une instruction :**

1. Le contenu de **SP** est copié au registre **CP**.
2. Le contenu de **SP** est envoyé sur le bus d'adresse.
3. Le contenu de **SP** est envoyé sur le bus de données.

- **Après exécution de l'instruction SBA :**

1. Le contenu de **A** est soustrait de **B**.
2. Le résultat n'est pas rangé dans **A**.
3. Le **CCR** n'est pas modifié.
4. Le flag **V** est mis à **1**.

- **Après exécution de l'instruction CBA :**

1. Le contenu de **B** est soustrait de **A**.
2. Le résultat est rangé dans **A**.
3. Le **CCR** n'est pas modifié.
4. Le flag **N** est mis à **1**.

- **L'instruction SWI :**

1. Peut remplacer l'instruction **END**.
2. Est une interruption logicielle.
3. Est utilisée pour les sous programmes.
4. Est utilisée pour diviser un programme très long.



TD2 : Architecture du 6800 et instructions ...

Après exécution de l'instruction ASR:

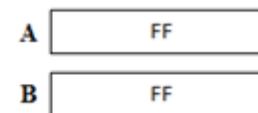
1. On obtient un résultat nul.
2. Impossible d'obtenir un résultat nul.
3. On obtient **FF** après plusieurs utilisations de **ASR**.

Après exécution de l'instruction ASL:

1. On obtient un résultat nul.
2. Impossible d'obtenir un résultat nul.
3. On obtient **FF** après plusieurs utilisations de **ASR**.

Après exécution de l'instruction LSR:

1. On obtient un résultat non nul.
2. Impossible d'obtenir un résultat non nul.
3. On obtient **FF** après plusieurs utilisations de **LSR**.

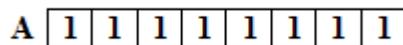


TAB
TPA
ASLB
ASLB
TAP
TBA
SWI

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

Exercice 3 : Soit la séquence d'instructions suivante :

ASL A
ASL A
LSR A



1. Décrivez le fonctionnement de chaque instruction par un schéma.
2. Quel est le contenu de l'accumulateur **A** après exécution de la séquence précédente ?
3. Quelle est la valeur de flag **C** du registre d'état ?

Exercice 4 :

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

Exercice 5 :

Ecrire un (**seul**) programme qui permet d'effectuer les opérations suivantes :

1. Effectuer une opération sur le **registre d'état** afin d'éviter les **IRQ**.
2. Charger l'accumulateur **A** avec le contenu de **[01F1]** et l'accumulateur **B** avec le contenu de **[01F2]**.
3. Incrémenter l'accumulateur **A**.
4. Soustraire la valeur **05** de l'accumulateur **B**.
5. Ranger le contenu de l'accumulateur **A** dans **X_H** (Partie **haute** du **registre d'index**) et le contenu de l'accumulateur **B** dans **X_L** (Partie **basse** du **registre d'index**).
6. Sauvegarder le contenu du **registre d'état** dans **[0100]**.
7. Exécuter le **sous programme** dont l'adresse est indiquée par le contenu du **registre d'index X**.
8. Restituer le contenu du **registre d'état**.
9. Autoriser les **IRQ**.
10. Effectuer un arrêt **logiciel**.