



Rattrapage Systèmes à microprocesseurs [A répondre sur le sujet]

Questions de cours :

1. Qu'est ce que une **interruption** ?

.....
.....
.....
.....
.....

2. **Citer** les interruptions **logicielles** et **matérielles** du 6800.

.....
.....
.....
.....
.....
.....
.....

3. Que veut dire **routine** d'interruption ?

.....
.....
.....
.....

4. Que contient le **vecteur** d'interruption ?

.....
.....
.....
.....

5. Que fait le **microprocesseur** lors d'une **requête d'interruption** ?

.....
.....
.....
.....
.....
.....
.....
.....
.....

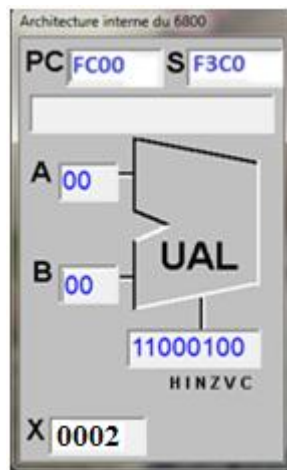
Exercice 1 :

Soit le programme suivant (Prendre en compte la **situation initiale** illustrée par les figures ci-dessous)

```

LDA A EE, X
ADD A #AB
LDX #00F1
PUSH B
PUSH A
LDA B 00F6
SBA
SWI
LDX 00F2
ADD A F0, X
PUL A
PUL B
SBA
END

```



RAM	
55	00F0
A9	00F1
54	00F2
8D	00F3
C5	00F4
18	00F5
2E	00F6

- Remplir le tableau suivant en donnant les résultats qu'obtiendrais après exécution du programme.

Instruction	Mode d'adressage	Registre et Pile		
LDA A EE, X		A =	B =	X =
ADD A #AB		A =	B =	
LDX #00F1		A =	B =	X =
PUSH B PUSH A		<div>Compléter les cases concernées + toutes les adresses.</div> <div><div><div>Pile</div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div><div>F3C0</div></div></div>		
LDA B 00F6		A =	B =	S =
SBA		A =	B =	
SWI		S =	X =	
LDX 00F2		A =	B =	X =
ADD A F0, X		A =	B =	X =
PUL A		S =		
PUL B		S =		
SBA		A =	B =	

Exercice 02 :

1. Soit le programme suivant. On vous demande de **décrire ce que fait chaque instruction**.

LDX #0000	
LDS #0F00	
loop:	
LDA A X, 00	
PSH A	
INX	
INS	
INS	
CPX #0100	
BNE loop	
END	

2. Dédurre alors ce que fait le programme.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Un seul document est autorisé est le jeu d'instructions

Solution de l'exercice 3 :

Important : Afin d'exécuter le programme sur Moto6800, les données doivent être déjà stockées en mémoire.

LDX #0000	Charger le registre X par l'adresse de la 1ere valeur (emplacement du départ).
LDS #0F00	Charger le registre SP par l'adresse d' arrivée ou sera transférée la 1ere valeur.
loop:	
LDA A X, 00	Charger l'accumulateur A par la 1ere valeur à transférer.
PSH A	Utiliser la pile pour enregistrer dans l'adresse d'arrivée.
INX	Incrémenter registre X afin de lire la valeur suivante à transférer.
INS	Incrémenter registre SP .
INS	Incrémenter registre SP encore une fois afin d'avoir l'adresse suivante.
CPX #0100	Comparer le contenu du registre X à l'adresse qui vient après la dernière valeur.
BNE loop	Refaire la boucle tant qu'on a pas dépassé la dernière valeur à transférer.
END	

- L'utilisation du registre **X** permet de faire une **lecture répétée** des valeurs présentes en mémoire.
- L'utilisation du registre **SP** permet de **gérer** les adresses de transfert.
- Incrémenter registre **SP deux fois** car **SP** se **décrémente** à chaque écriture.