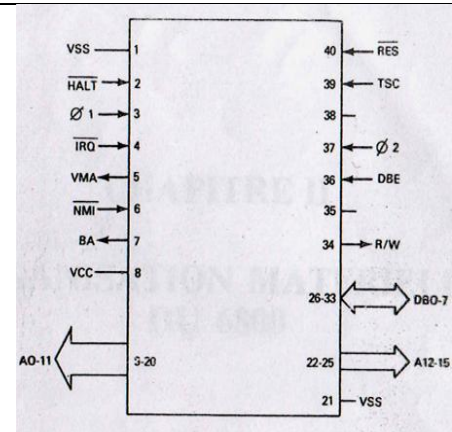


**Contrôle Systèmes à microprocesseurs ... (01h30min)**  
**Un seul document est autorisé est le jeu d'instructions**

**Questions de cours :** (08 points)

Le 6800 possède **deux** lignes d'interruption prioritaire et normale.

1. Qu'est ce qu'une interruption **prioritaire** ?
2. Donner la broche du circuit ci-contre qui la représente.
3. Donner alors son vecteur.
4. Donner les étapes suivies lors d'une **interruption prioritaire** sous forme de **schéma**.
5. Qu'est ce qu'une interruption **normale** ?
6. Donner la broche du circuit ci-contre qui la représente.
7. Donner alors son vecteur.
8. Donner les étapes suivies lors d'une **interruption normale** sous forme de **schéma**.



**Microprocesseur Motorola 6800**

**Exercice 1 :** (06 points)

On veut écrire un programme qui réalise l'addition (**F23D + 34F0**) illustrée par la figure ci-contre :

Pour faire ceci, vous suivez les étapes suivantes :

- Charger un registre par : **Partie basse 1**.
- Additionner avec : **Partie basse 2** en considérant la retenue intermédiaire..
- Ranger le résultat dans l'adresse [0007].
- Utilisez le **registre d'index** et un **branchement** afin de créer la boucle.
- Recharger le registre par : **Partie haute 1**.
- Additionner avec : **Partie haute 2** en considérant la retenue intermédiaire.
- Ranger le résultat dans l'adresse [0006].
- En fin de ce programme on veut **considérer la retenue finale**, vous procédez comme suit :
- Tester le **flag C**.
- Si **C = 1** alors : la case mémoire [0005] est incrémentée.
- Si non : Rien faire.

Partie haute 1	F2	0000
Partie basse 1	3D	0001
Partie haute 2	34	0002
Partie basse 2	F0	0003
		0004
		0005
		0006
		0007

1 ← Retenue intermédiaire  
 F2 3D  
 34 F0  
 1 27 2D  
 ↑  
 Retenue finale

**Exercice 02 :** (06 points)

Ecrire un programme en Assembleur 6800 qui permet de calculer les vingt (20) premiers termes de la série définit par:

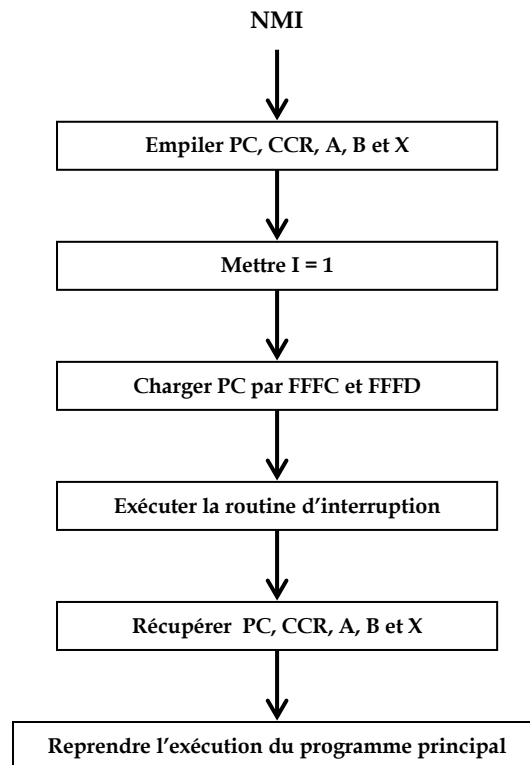
$$U_{n+1} = U_n + 2 \quad \text{avec : } U_1=1$$

les termes de la série doivent être rangés à partir de l'adresse mémoire [0000].

Solution du Contrôle Systèmes à microprocesseurs

Questions de cours : (08 points)

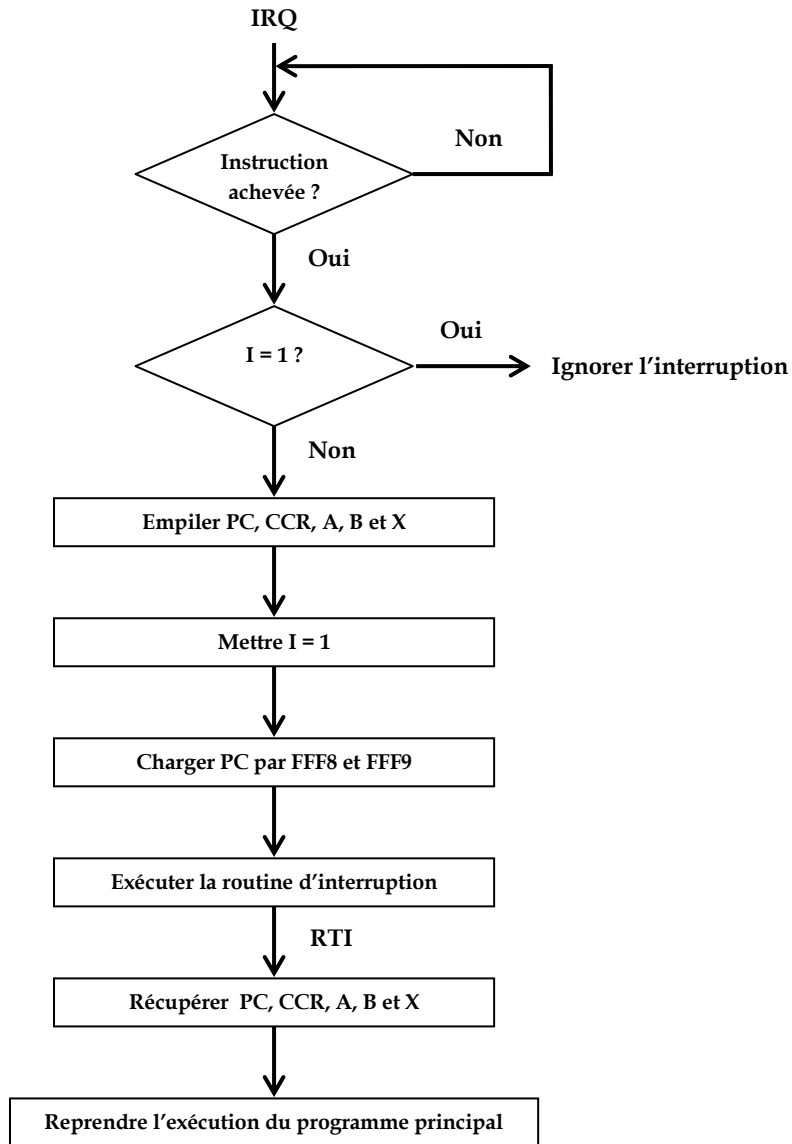
1. Interruption prioritaire :  
Une **interruption** est un arrêt temporaire de l'exécution normale d'un programme **informatique** par le microprocesseur afin d'exécuter un autre programme appelé routine d'interruption.  
L'interruption **prioritaire** appelée aussi **NMI** : Non Masquable Interruption est une interruption qui **ne peut pas être masquée par l'indicateur I**, c'est-à-dire : quelque soit la valeur de I (I = 0 ou I = 1), l'interruption est exécutée.
2. Broche du circuit ci-contre qui la représente : Broche 6 (  $\overline{\text{NMI}}$  )
3. Vecteur d'interruption : FFFC et FFFD.
4. Schéma d'une interruption prioritaire :



5. Interruption normale :  
Une **interruption** est un arrêt temporaire de l'exécution normale d'un programme **informatique** par le microprocesseur afin d'exécuter un autre programme appelé routine d'interruption.  
L'interruption **normale** appelée aussi **IRQ** : elle peut être **masquée par l'indicateur I**, c'est-à-dire : L'état de la ligne **IRQ** est testé ou ignoré par le microprocesseur selon la valeur de l'indicateur I (**Flag I** appelé **masque d'interruption du registre CCR**).
6. Broche du circuit ci-contre qui la représente : Broche 4 (  $\overline{\text{IRQ}}$  )
7. Vecteur d'interruption : FFF8 et FFF9.

Solution du Contrôle Systèmes à microprocesseurs (suite)

8. Schéma d'une interruption normale.



### Solution du Contrôle Systèmes à microprocesseurs (suite)

#### Exercice 1: (06 points)

<pre> ; \$0001 DB \$F2 ; \$0002 DB \$3D ; \$0003 DB \$34 ; \$0004 DB \$F0  LDX #\$0002 CLC  Loop: LDAA \$00,X ADCA \$02,X STAA \$05,X  DEX CPX #\$0000 BNE Loop  BCC Sortir INC \$0005  Sortir: END         </pre>	<p>1 ← Retenue intermédiaire</p> <p>F2 3D</p> <p>34 F0</p> <hr/> <p>1 27 2D</p> <p>↑</p> <p>Retenue finale</p> <table border="1"> <thead> <tr> <th colspan="2">RAM</th> </tr> </thead> <tbody> <tr><td>0000</td><td>00</td></tr> <tr><td>0001</td><td>F2 0</td></tr> <tr><td>0002</td><td>3D =</td></tr> <tr><td>0003</td><td>34 4</td></tr> <tr><td>0004</td><td>F0 8</td></tr> <tr><td>0005</td><td>01</td></tr> <tr><td>0006</td><td>27 '</td></tr> <tr><td>0007</td><td>2D -</td></tr> <tr><td>0008</td><td>00</td></tr> <tr><td>0009</td><td>00</td></tr> <tr><td>000A</td><td>00</td></tr> <tr><td>000B</td><td>00</td></tr> <tr><td>...</td><td>...</td></tr> </tbody> </table>	RAM		0000	00	0001	F2 0	0002	3D =	0003	34 4	0004	F0 8	0005	01	0006	27 '	0007	2D -	0008	00	0009	00	000A	00	000B	00	...	...
RAM																													
0000	00																												
0001	F2 0																												
0002	3D =																												
0003	34 4																												
0004	F0 8																												
0005	01																												
0006	27 '																												
0007	2D -																												
0008	00																												
0009	00																												
000A	00																												
000B	00																												
...	...																												

#### Exercice 02: (06 points)

20 termes ... en hexadécimal : (14)<sub>h</sub>

<p>Architecture interne du 6800</p> <p>PC <b>FC0F</b> S <b>0000</b></p> <p><b>END</b></p> <p>A <b>29</b></p> <p>B <b>00</b></p> <p><b>UAL</b></p> <p><b>11000100</b></p> <p>HINZVC</p> <p>X <b>0014</b></p>	<p>Editeur MOT...</p> <p>Mise à jour Edition</p> <p>;</p> <p>;\$0000 DB \$01</p> <p>LDX #\$0000</p> <p>loop:</p> <p>LDAA \$00,X</p> <p>ADDA #\$02</p> <p>STAA \$01,X</p> <p>INX</p> <p>CPX #\$0014</p> <p>BNE loop</p> <p>END</p> <p>Modifié INS NUM 6</p>	<p>RAM</p> <table border="1"> <tbody> <tr><td>0000</td><td>01</td></tr> <tr><td>0001</td><td>03</td></tr> <tr><td>0002</td><td>05</td></tr> <tr><td>0003</td><td>07</td></tr> <tr><td>0004</td><td>09</td></tr> <tr><td>0005</td><td>0B</td></tr> <tr><td>0006</td><td>0D</td></tr> <tr><td>0007</td><td>0F</td></tr> <tr><td>0008</td><td>11</td></tr> <tr><td>0009</td><td>13</td></tr> <tr><td>000A</td><td>15</td></tr> <tr><td>000B</td><td>17</td></tr> <tr><td>000C</td><td>19</td></tr> <tr><td>000D</td><td>1B</td></tr> <tr><td>000E</td><td>1D</td></tr> <tr><td>000F</td><td>1F</td></tr> <tr><td>0010</td><td>21</td></tr> <tr><td>0011</td><td>23</td></tr> <tr><td>0012</td><td>25</td></tr> <tr><td>0013</td><td>27</td></tr> <tr><td>0014</td><td>29</td></tr> <tr><td>0015</td><td>00</td></tr> <tr><td>0016</td><td>00</td></tr> <tr><td>0017</td><td>00</td></tr> <tr><td>...</td><td>...</td></tr> </tbody> </table>	0000	01	0001	03	0002	05	0003	07	0004	09	0005	0B	0006	0D	0007	0F	0008	11	0009	13	000A	15	000B	17	000C	19	000D	1B	000E	1D	000F	1F	0010	21	0011	23	0012	25	0013	27	0014	29	0015	00	0016	00	0017	00	...	...
0000	01																																																			
0001	03																																																			
0002	05																																																			
0003	07																																																			
0004	09																																																			
0005	0B																																																			
0006	0D																																																			
0007	0F																																																			
0008	11																																																			
0009	13																																																			
000A	15																																																			
000B	17																																																			
000C	19																																																			
000D	1B																																																			
000E	1D																																																			
000F	1F																																																			
0010	21																																																			
0011	23																																																			
0012	25																																																			
0013	27																																																			
0014	29																																																			
0015	00																																																			
0016	00																																																			
0017	00																																																			
...	...																																																			