

Correction TP ASM N°2

Interrupteurs, diodes et afficheurs 7 segments

Objectif : Mise en œuvre de différents périphériques de l'environnement IDE68K.

Langage utilisé : Langage assembleur

1^{ère} étape : Recherche des adresses

A l'aide de la documentation déterminer les adresses permettant d'accéder aux différents éléments.

1)- Compléter le tableau suivant : D'après la documentation

Périphérique	@ du registre sur 8 bits	@ du registre sur 16 bits
Interrupteurs	\$E001	\$E000
Diodes	\$E003	\$E002
Afficheur le + à droite	\$E017	\$E016

2^{ème} étape : Programmation

1)- Interrupteurs, diodes

Cahier des charges :

☞ Lire l'état des interrupteurs et écrire cette valeur sur les diodes.

Algorithme :

```

Debut
|   Faire
|       Lecture des interrupteurs
|       Ecriture sur les diodes
|   Tant que(infini)
Fin
  
```

Programme :

```

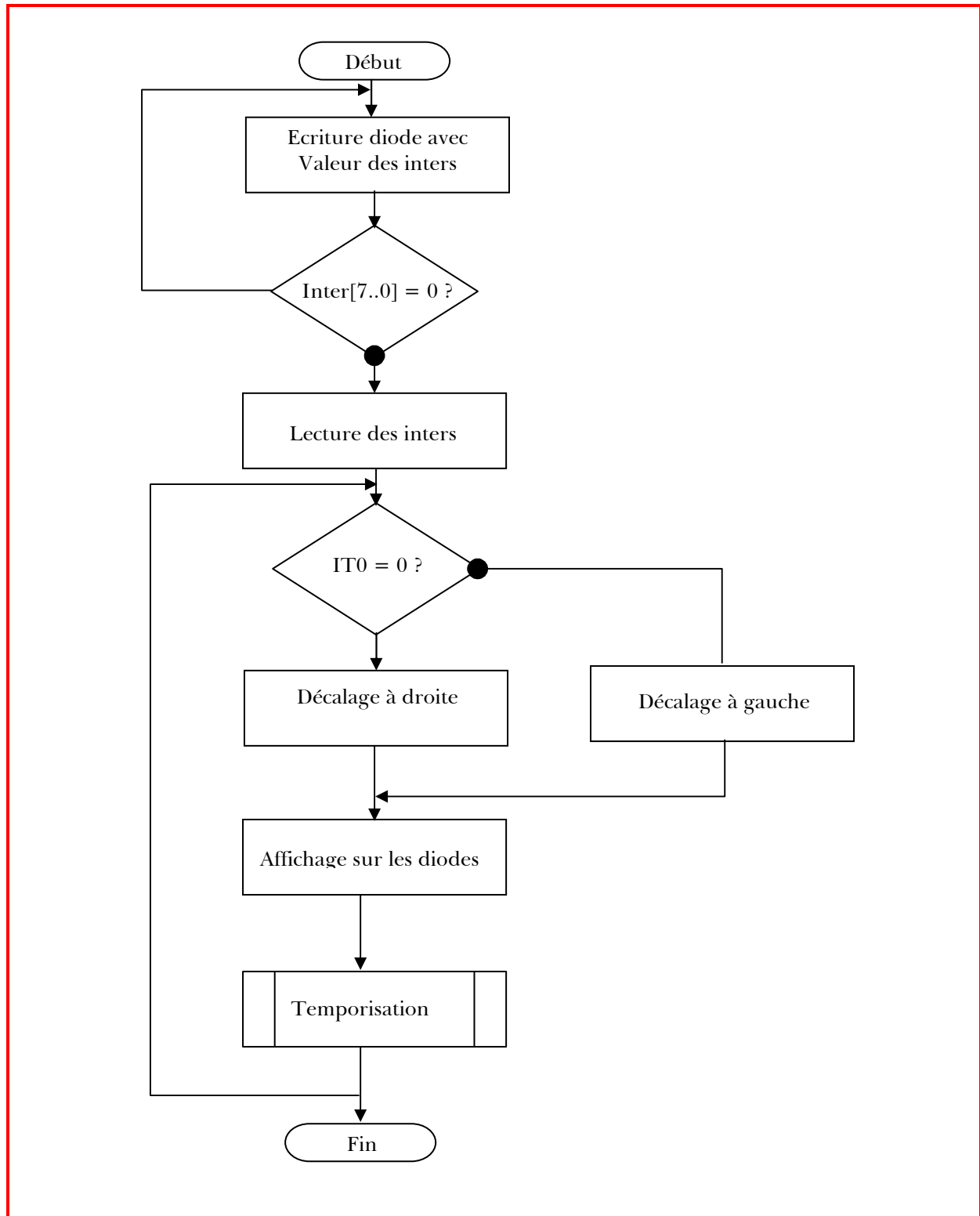
org      $400
debut    move.b    $e001,$e003 ;lecture des inter et écriture sur les
diodes
        bra      debut      ;retour au debut
        trap     #15
        DC.W     0
        end      debut
  
```

2)- Rotation de la valeur sur les diodes : utilisation de l'instruction ROR ou ROL

Cahier des charges :

☞ Modifier le programme précédent de manière à ce que :

- Lorsque l'interrupteur 0 vaut '0' décaler la valeur vers la droite
- Lorsque l'interrupteur 0 vaut '1' décaler la valeur vers la gauche

Algorigramme :

Programme :

```

debut      org      $400
           move.b   $e001,$e003
           move.b   $e001,D0
           cmpi.b   #0,D0           ;tant que la valeur des inter
           beq      debut           ;égale 0 on reboucle au début
           ;lecture de l'état de IT0
tstIT0     move.b   $e001,D1
           ;isolation IT0 dans C
           ror.b    #1,D1
           ;branchement à DecD si IT0 = 0
           BCC      DecD
DecG        rol.b   #1,D0           ;décalage à gauche
           bra      aff
DecD        ror.b   #1,D0           ;décalage à droite
Aff         move.b   D0,$e003       ;affichage
           bsr      tempo           ;temporisation
           bra      tstIT0          ;branchement au début
           trap     #15
           DC.W     0

           ;boucle de temporisation
tempo      move.l   #$Fffff,D2
bcl         subi.l   #1,D2
           bne      bcl
           rts

           end      debut

```

3)- Compteur de 0 à 9**Cahier des charges :**

☞ Ecrire un programme permettant d'afficher les valeurs de 0 à 9 sur l'afficheur 7 segments le plus à droite.

Algorithme :

Initialiser un tableau de valeur pour l'affichage des valeurs sur 7 segments

```

Debut
|   Initialisation pointeur sur tableau de valeurs
|Faire
|   Initialisation index sur 10 (0 → 9)
|   Faire
|       afficher valeur pointée
|       Incrémenter l'index de 1
|       Temporisation
|   Tant que (index < 10)
|Tant que (infini)
Fin

```

Programme :

```

AFF      EQU      $E017

Debut    ORG      $400

        move.w    #tab,A0          ;initialisation pointeur
enc      move.w    #0,D0           ;initialisation de l'index
bcl      move.b    0(A0,D0),AFF     ;affichage valeur pointée
        bsr      tempo
        addi.b     #1,D0
        cmpi.b     #10,D0
        bne      bcl              ;tant que index < 10
        bra      enc

        Trap      #15
        DC.W      0

        ;temporisation
tempo    move.l     #$1ffffff,D1
bcll     subi.l     #1,D1
        bne      bcll
        rts

        ;init tableau
tab      DC.b      $3F,$06,$5B,$4F,$66,$6D,$7D,$07,$7F,$6F

        End      Debut

```

Cahier des charges :

☞ Modifier le programme précédent, de manière à ce que la valeur affichée s'incrémente à chaque passage à '1' de l'interrupteur 0.

Algorithme :

Initialiser un tableau de valeur pour l'affichage des valeurs sur 7 segments

```

Debut
|   Initialisation pointeur sur tableau de valeurs
|Faire
|   Initialisation index sur 10 (0 → 9)
|   Faire
|   |   afficher valeur pointée
|   |   Temporisation
|   |   Faire
|   |   |   tester ITO
|   |   |   Tant que (IT = 0)
|   |   |   Incrémenter l'index de 1
|   Tant que (index < 10)
|Tant que (infini)
Fin

```

Programme :

```

AFF      EQU          $E017

Debut    ORG          $400

        move.w        #tab,A0          ;initialisation pointeur
enc       move.w        #0,D0          ;initialisation de l'index
bcl      move.b        0(A0,D0),AFF    ;affichage valeur pointée
        bsr          tempo ;permet d'éviter l'incrémentation automatique

        ;test de l'état d'IT0
tstIT0    move.b        $e001,D1
        ror.b         #1,D1
        BCC          tstIT0
        ;Si IT0 = 1
        addi.b        #1,D0
        cmpi.b        #10,D0
        bne          bcl              ;tant que index < 10
        bra          enc

        Trap          #15
        DC.W          0

        ;temporisation
tempo     move.l        #$1ffffff,D1
bcll      subi.l        #1,D1
        bne          bcll
        rts

        ;init tableau
tab       DC.b          $3F,$06,$5B,$4F,$66,$6D,$7D,$07,$7F,$6F

        End          Debut

```