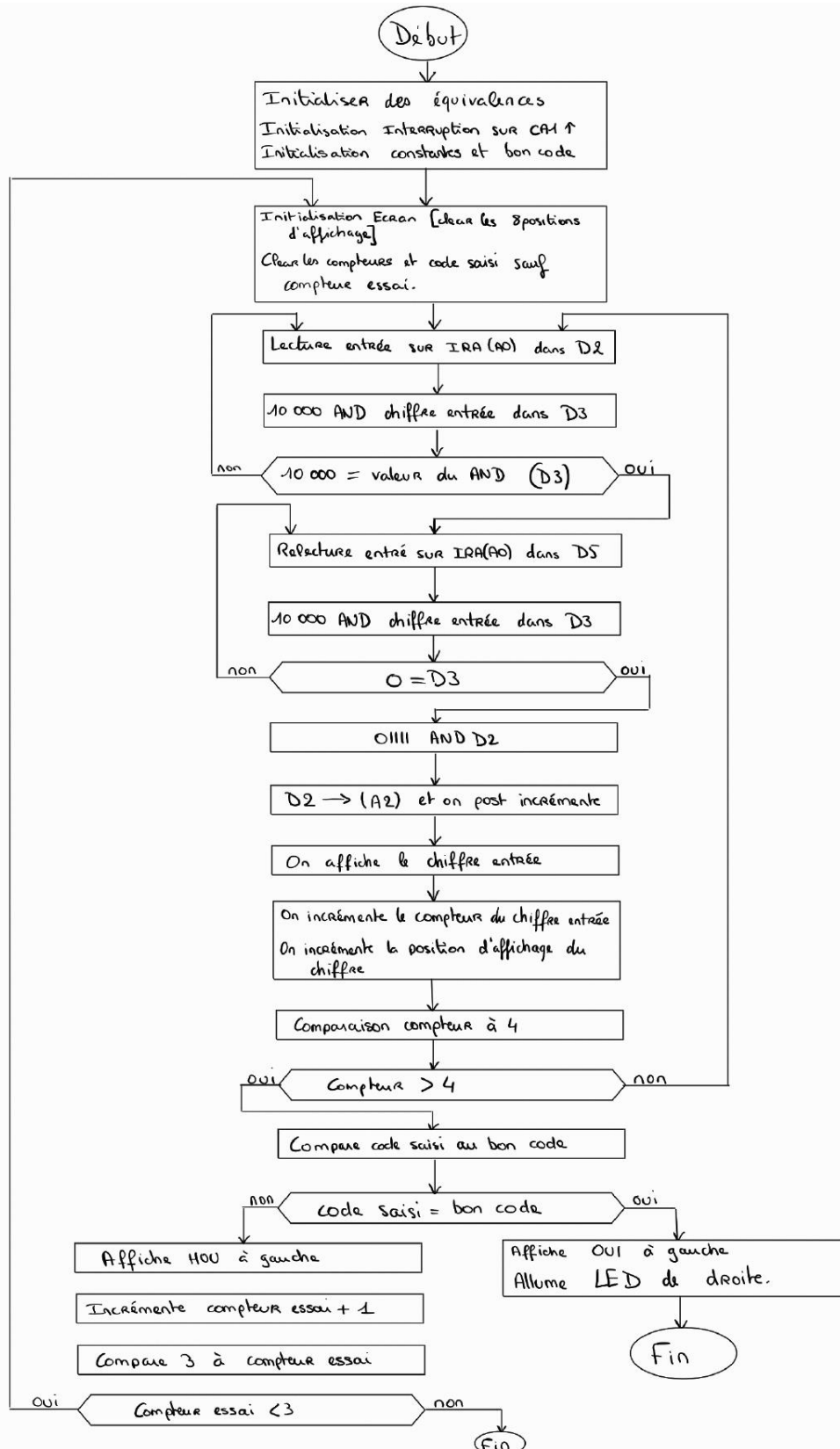
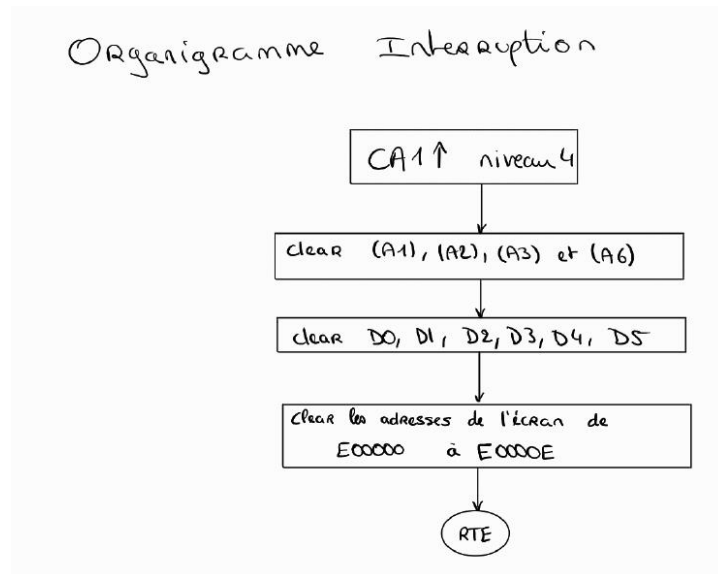


Compte rendu TP 4

Partie 1 : Digicode avec combinaison préenregistrer

Organigramme :





Programme :

```

*-----
* Title       : TP4
* Written by  : ERRARD et DACCACHE
* Date       : 03/04/2024
* Description: TP4_code_intégrer
*-----
vial equ $f0441      *initialisation des equivalences
ora equ $1E          *output port A
ira equ $1E          *input port A
rpc equ $18          *registre de controle des peripheriques
ifr equ $1A          *registre d'interruption valide
ier equ $1C          *registre d'indicateur d'interruption
sr equ $14           *registre d'etat
ddra equ $6          *status des ports A0 a A7 (entree ou
sortie)
orb equ $0           *output port B
ddrb equ $4          *status des ports B0 a B7 (entree ou
sortie)

    ORG    $1000
START:                ; first instruction of program

* Put program code here

    lea vial,A0        *chargement de vial dans l'adresse de
registre A0
    move.l pin,D1      *on met le pin predeterminer dans D1
    move.l #inter1,$70 *on choisit l'adresse d'interruption
(niveau 4)
    move.b #$11,rpc(A0) *on met a 1 le bit numero 0 de rpc qui
correspond a un front montant sur CA1
    move.b #%10000010,ier(A0) *on met ier a 1000 0010, pour autorise
les interruption sur CA1
    move.w #$2300,sr    *on met les 8 bits de gauche de SR a
0010 0011, pour changer le masque d'interruption a 3 et autorisé mode
superviseur
    move.b #$00,ddra(A0) *on met les pates de A en mode Lecture
(0F0443 pour l'affichage dans hardware)
  
```

```

        move.b #$FF,ddrb(A0)          *on met les pates de B en mode
ecriture(0F0441 pour l'affichage dans hardware)

        move.l #$4000,A1              *on met 4000 en hexa dans A1 (qui va
pointer a l'adresse 4000 de la memoire, qui contient le bon pin)
        move #$4020,A6                *on met 4020 en hexa dans A6 (qui va
pointer a l'adresse 4020 de la memoire, qui contient le nombre d'essai)
        clr.w (A6)                    *on efface le contenu de A6
        move #$4010,A4                *on met 4010 en hexa dans A4 (qui va
pointer a l'adresse 4010 de la memoire, qui contient la position du numero
a affiche)

*-----init pour reinitialiser notre programme apres reset (interruption
CA1)-----
init:   move.l #$4000,A2              *on met 4000 en hexa dans A2 (qui va
pointer a l'adresse 4000 de la memoire, qui nous permettra de post-
incrementer)
        clr.l (A2)                    *on efface le contenu de l'adresse dont
A2 est notre pointeur
        clr.l (A4)                    *on efface le contenu de l'adresse dont
A4 est notre pointeur
        clr.l D0                      *on efface le contenu de D0
        clr.l D2                      *on efface le contenu de D2
        clr.l D3                      *on efface le contenu de D3
        clr.l D4                      *on efface le contenu de D4
        clr.l D5                      *on efface le contenu de D5
        clr.w $E00000                 *on efface le contenu de l'adresse
E00000 (position 1 de gauche)
        clr.w $E00002                 *on efface le contenu de l'adresse
E00002 (position 2 de gauche)
        clr.w $E00004                 *on efface le contenu de l'adresse
E00004 (position 3 de gauche)
        clr.w $E00006                 *on efface le contenu de l'adresse
E00006 (position 4 de gauche)
        clr.w $E00008                 *on efface le contenu de l'adresse
E00008 (position 5 de gauche)
        clr.w $E0000A                 *on efface le contenu de l'adresse
E0000A (position 6 de gauche)
        clr.w $E0000C                 *on efface le contenu de l'adresse
E0000C (position 7 de gauche)
        clr.w $E0000E                 *on efface le contenu de l'adresse
E0000E (position 8 de gauche)

*-----lecture du nombre entree-----
lecture: move.b ira(A0),D2             *on met la valeur entree dans D2
        move.w #%10000,D3             *on met 10000 en binaire dans D3
        and D2,D3                     *on realise une operation ET (AND)
entre D2 et D3 pour savoir si le bouton 4 est active
        cmp #%10000,D3                *on compare 10000 en binaire a D3 pour
savoir si le bouton 4 est active
        beq test1                     *si oui on pass au code de l'etiquette
test1
        bne lecture                   *sinon on passe au code de l'etiquette
lecture attendre la validation du nombre entree par l'utilisateur

test1:  move.b ira(A0),D5              *si bouton 4 active on met la valeur
entree dans D5
        move.w #%10000,D3             *on met 10000 en binaire dans D3
        and D5,D3                     *on realise une operation ET entre D5 et
D3 pour effacer le nombre entree et garder 1 (si activer sinon 0)

```

```

        cmp #$0,D3                *on compare 0 a D3 pour savoir si on a
eteint le bouton 4
        beq suite1                *si le bouton 4 est eteint c.a.d que
l'utilisateur a valider son choix de chiffre et on passe au code de
l'etiquette suite1
        bne test1                *sinon on relance le code de
l'etiquette test1 pour attendre la validation du choix de l'utilisateur

suite1  move.w #%01111,D3         *si le bouton 4 est desactive, on met
01111 en binaire dans D3
        and D3,D2                *on realise une operation ET entre D3
et D2 pour retirer le 1 situe sur le 5eme bit de la valeur saisi
        move.b D2,(A2)+          *on met ce chiffre dans l'adresse que
contient A2 et on post incremente A2 (ex:si A2=4000,apres le move A2=40001)
        bra affichage            *on saute au code de l'etiquette
affichage qui nous permettra d'afficher le chiffre entree sur l'ecran

                                         *apres avoir afficher le chiffre saisi
et l'avoir positionner on passe a l'etape suivante
cont    add #1,D4                *on incremente de 1 D4 (compteur du
nombre entree)
        add #$1,(A4)             *on incremente de 1 le le contenue de
A4 pour passer a la position suivante de l'affichage
        cmp #$4,D4              *on compare D4 a 4 pour verifier si on
a saisi les 4 chiffres
        bgt test2               *si D4 plus grand que 4 on passe au
code de l'etiquette test2 c.a.d qu'on a entree 4 chiffre
        blt lecture              *sinon on reprend le code de
l'etiquette lecture pour entree le chiffre suivant jusqu'a se qu'on a saisi
4 chiffres

*-----verification du code entree-----
test2   cmp.l (A1),D1            *on compare le contenue de l'adresse
dans A1 a D1
        beq aff_oui             *si elle est egale on passe au code de
l'etiquette aff_oui pour afficher que le code est bon
        bne aff_hou             *le code entree est faux on passe au
code de l'etiquette aff_hou

aff_oui move.b zero,$E00000      *on affiche 0 qui correspond a la
lettre O dans la premiere case de gauche de l'afficheur
        move.b let_U,$E00002     *on affiche U dans la deuxieme case de
gauche de l'afficheur
        move.b let_I,$E00004     *on affiche I dans la troisieme case de
gauche de l'afficheur
        move.b #$F,orb(A0)       *on allume les 4 de droite sous
l'afficheur pour indiquer que le code est bon
        bsr stop                *on passe au code de l'etiquette stop
pour arreter le programme

aff_hou move.b let_H,$E00000     *on affiche H dans la premiere case de
gauche de l'afficheur
        move.b zero,$E00002     *on affiche 0 qui correspond a la
lettre O dans la deuxieme case de gauche de l'afficheur
        move.b let_U,$E00004     *on affiche U dans la premiere case de
gauche de l'afficheur
        move.b #$0,orb(A0)       *on eteint les led qui son sous
l'afficheur pour indiquer que le code est faux
        add #$1,(A6)            *on incremente de 1 le contenu de
l'adresse contenu dans A6 (qui contient le nombre d'essai)

```

```

        cmp #$3, (A6)                *on compare le contenu de l'adresse
contenu dans A6 a 3
        blt initi                    *si c'est plus petit donc on a encore
le droit d'essayer donc on passe au code de l'etiquette initi pour
reinitialiser notre code
        bsr stop                    *sinon on a plus le droit d'essayer et
on passe au code de l'etiquette stop pour arrter le programme qui affichera
*HOU et le 3eme mauvais code entree
*-----Affichage des nombres sur le LCD-----
affichage  cmp #$0,D2                *on compare 0 a D2
        beq aff_zero                *si D2 == 0 on affiche 0
        cmp #$1,D2                *on compare 1 a D2
        beq aff_un                  *si D2 == 1 on affiche 1
        cmp #$2,D2                *on compare 2 a D2
        beq aff_deux                *...
        cmp #$3,D2
        beq aff_trois
        cmp #$4,D2
        beq aff_quatre
        cmp #$5,D2
        beq aff_cinque
        cmp #$6,D2
        beq aff_six
        cmp #$7,D2
        beq aff_sept
        cmp #$8,D2
        beq aff_huit
        cmp #$9,D2
        beq aff_neuf
*----Fin boucle de comparaison et passage a l'etiquette d'affichage
specifique au chiffre contenu dans D2-----
*----Debut Affichage et positionnement-----
*----Affiche 0-----
aff_zero  cmp #$0, (A4)              *on compare 0 au contenu de l'adresse
mise dans A4
        beq aff_pos10              *si egale a 0 on affiche a la position
5 de gauche
        cmp #$1, (A4)              *on continue la comparaison jusqu'a 3
pour afficher les 4 chiffres a droite en passant aux
        beq aff_pos20              *codes des etiquettes aff_posXX avec
(XX = position, nombre a afficher)
        cmp #$2, (A4)              *...
        beq aff_pos30
        cmp #$3, (A4)
        beq aff_pos40
        rts                        *apres avoir afficher le numero sur la
case specifique on retourne a notre programme
aff_pos10 move.b zero,$E00008        *on met la constante defini par
'zero'dans l'adresse E00008 en hexa pour afficher '0' dans la case 5 de
l'afficheur
        bra cont                    *on passe au code de l'etiquette cont
pour continuer notre code apres l'affichage
aff_pos20 move.b zero,$E0000A        *on refait de meme pour les 4 positions
et pour les 9 autres chiffres
        bra cont                    *...
aff_pos30 move.b zero,$E0000C
        bra cont
aff_pos40 move.b zero,$E0000E
        bra cont
*----Affiche 1-----

```

```
aff_un    cmp  #$0, (A4)
          beq  aff_pos11
          cmp  #$1, (A4)
          beq  aff_pos21
          cmp  #$2, (A4)
          beq  aff_pos31
          cmp  #$3, (A4)
          beq  aff_pos41
          rts
aff_pos11 move.b un,$E00008
          bra  cont
aff_pos21 move.b un,$E0000A
          bra  cont
aff_pos31 move.b un,$E0000C
          bra  cont
aff_pos41 move.b un,$E0000E
          bra  cont
*----Affiche 2-----
aff_deux  cmp  #$0, (A4)
          beq  aff_pos12
          cmp  #$1, (A4)
          beq  aff_pos22
          cmp  #$2, (A4)
          beq  aff_pos32
          cmp  #$3, (A4)
          beq  aff_pos42
          rts
aff_pos12 move.b deux,$E00008
          bra  cont
aff_pos22 move.b deux,$E0000A
          bra  cont
aff_pos32 move.b deux,$E0000C
          bra  cont
aff_pos42 move.b deux,$E0000E
          bra  cont
*----Affiche 3-----
aff_trois cmp  #$0, (A4)
          beq  aff_pos13
          cmp  #$1, (A4)
          beq  aff_pos23
          cmp  #$2, (A4)
          beq  aff_pos33
          cmp  #$3, (A4)
          beq  aff_pos43
          rts
aff_pos13 move.b trois,$E00008
          bra  cont
aff_pos23 move.b trois,$E0000A
          bra  cont
aff_pos33 move.b trois,$E0000C
          bra  cont
aff_pos43 move.b trois,$E0000E
          bra  cont
*----Affiche 4-----
aff_quatre cmp  #$0, (A4)
          beq  aff_pos14
          cmp  #$1, (A4)
          beq  aff_pos24
          cmp  #$2, (A4)
          beq  aff_pos34
          cmp  #$3, (A4)
```

```
        beq aff_pos44
        rts
aff_pos14 move.b quatre,$E00008
        bra cont
aff_pos24 move.b quatre,$E0000A
        bra cont
aff_pos34 move.b quatre,$E0000C
        bra cont
aff_pos44 move.b quatre,$E0000E
        bra cont
*----Affiche 5-----
aff_cinque  cmp #$0, (A4)
        beq aff_pos15
        cmp #$1, (A4)
        beq aff_pos25
        cmp #$2, (A4)
        beq aff_pos35
        cmp #$3, (A4)
        beq aff_pos45
        rts
aff_pos15 move.b cinq, $E00008
        bra cont
aff_pos25 move.b cinq, $E0000A
        bra cont
aff_pos35 move.b cinq, $E0000C
        bra cont
aff_pos45 move.b cinq, $E0000E
        bra cont
*----Affiche 6-----
aff_six  cmp #$0, (A4)
        beq aff_pos16
        cmp #$1, (A4)
        beq aff_pos26
        cmp #$2, (A4)
        beq aff_pos36
        cmp #$3, (A4)
        beq aff_pos46
        rts
aff_pos16 move.b six, $E00008
        bra cont
aff_pos26 move.b six, $E0000A
        bra cont
aff_pos36 move.b six, $E0000C
        bra cont
aff_pos46 move.b six, $E0000E
        bra cont
*----Affiche 7-----
aff_sept  cmp #$0, (A4)
        beq aff_pos17
        cmp #$1, (A4)
        beq aff_pos27
        cmp #$2, (A4)
        beq aff_pos37
        cmp #$3, (A4)
        beq aff_pos47
        rts
aff_pos17 move.b sept, $E00008
        bra cont
aff_pos27 move.b sept, $E0000A
        bra cont
aff_pos37 move.b sept, $E0000C
```

```
        bra cont
aff_pos47 move.b sept,$E0000E
        bra cont
*----Affiche 8-----
aff_huit  cmp #$0,(A4)
        beq aff_pos18
        cmp #$1,(A4)
        beq aff_pos28
        cmp #$2,(A4)
        beq aff_pos38
        cmp #$3,(A4)
        beq aff_pos48
        rts
aff_pos18 move.b huit,$E00008
        bra cont
aff_pos28 move.b huit,$E0000A
        bra cont
aff_pos38 move.b huit,$E0000C
        bra cont
aff_pos48 move.b huit,$E0000E
        bra cont
*----Affiche 9-----
aff_neuf  cmp #$0,(A4)
        beq aff_pos19
        cmp #$1,(A4)
        beq aff_pos29
        cmp #$2,(A4)
        beq aff_pos39
        cmp #$3,(A4)
        beq aff_pos49
        rts
aff_pos19 move.b neuf,$E00008
        bra cont
aff_pos29 move.b neuf,$E0000A
        bra cont
aff_pos39 move.b neuf,$E0000C
        bra cont
aff_pos49 move.b neuf,$E0000E
        bra cont
*----Fin Affichage et positionnement-----
*----Code le 1'interruption sur front montant de CA1-----
inter1  move.l #$4000,A2
        clr.l (A1)
        clr.l (A2)
        clr.l (A4)
        clr.l (A6)
        clr.l D0
        clr.l D2
        clr.l D3
        clr.l D4
        clr.l D5
        clr.w $E00000
        clr.w $E00002
        clr.w $E00004
        clr.w $E00006
        clr.w $E00008
        clr.w $E0000A
        clr.w $E0000C
        clr.w $E0000E
        bset #1,ifr(A0)
        rte
```



```
*----Fin du Code le l'interruption sur front montant de CA1-----  
  
stop      move.b #9,D0  
          TRAP #15  
  
          SIMHALT          ; halt simulator  
  
* Put variables and constants here  
pin       dc.b 6,9,1,2  
zero      dc.b $3F  
un        dc.b $06  
deux      dc.b $5B  
trois     dc.b $4F  
quatre    dc.b $66  
cinque    dc.b $6D  
six       dc.b $7D  
sept      dc.b $07  
huit      dc.b $7F  
neuf      dc.b $6F  
let_U     dc.b $3E  
let_I     dc.b $30  
let_H     dc.b $76  
off       dc.b $00  
  
          END      START      ; last line of source
```

Partie 2 : Digicode avec choix de combinaison sur la console

Programme :

```
*-----
* Title      :   TP4
* Written by :   ERRARD et DACCACHE
* Date       :   03/04/2024
* Description:   TP4_demande_de_code
*-----
vial equ $f0441      *initialisation des equivalences
ora equ $1E          *output port A
ira equ $1E          *input port A
rpc equ $18          *registre de controle des peripheriques
ifr equ $1A          *registre d'interruption valide
ier equ $1C          *registre d'indicateur d'interruption
sr equ $14           *registre d'etat
ddra equ $6          *status des ports A0 a A7 (entree ou
sortie)
orb equ $0           *output port B
ddrb equ $4          *status des ports B0 a B7 (entree ou
sortie)
CR EQU $0D           *retour chariot (carriage return)
LF EQU $0A           *Saut de ligne (Line Feed)

        ORG      $1000
START:                                ; first instruction of program

* Put program code here

        lea vial,A0      *chargement de vial dans l'adresse de
registre A0
        move.l #inter1,$70      *on choisit l'adresse d'interuption
(niveau 4)
        move.b #$11,rpc(A0)      *on met a 1 le bit numero 0 de rpc qui
correspond a un front montant sur CA1
        move.b #%10010010,ier(A0) *on met ier a 1000 0010, pour autorise
les interruption sur CA1
        move.w #$2300,sr      *on met les 8 bits de gauche de SR a
0010 0011, pour changer le masque d'interruption a 3 et autorisé mode
superviseur
        move.b #$00,ddra(A0)      *on met les pates de A en mode Lecture
(0F0443 pour l'affichage dans hardware)
        move.b #$FF,ddrb(A0)      *on met les pates de B en mode
ecriture(0F0441 pour l'affichage dans hardware)

        move.l #$4000,A1      *on met 4000 en hexa dans A1 (qui va
pointer a l'adresse 4000 de la memoire, qui contient le bon pin)
        move #$4020,A6      *on met 4020 en hexa dans A6 (qui va
pointer a l'adresse 4020 de la memoire, qui contient le nombre d'essai)
        clr.w (A6)          *on efface le contenu de A6
        move #$4010,A4      *on met 4010 en hexa dans A4 (qui va
pointer a l'adresse 4010 de la memoire, qui contient la position du numero
a affiche)
        move #$4038,A3      *on met 4038 en hexa dans A3 (qui va
pointer a l'adresse 4038 de la memoire, contient la valeur du bon code en
hex)

*-----Entree du mot de passe-----
code    lea MESSAGE1, A1      *chargement de message1 dans l'adresse de
registre A1
```

```

        move.b #14,D0      *mettre 14 dans le registre D0
        TRAP #15           *affiche du message1
        lea MESSAGE2, A1   *chargement de message2 dans l'adresse de
registre A1
        TRAP #15           *affichage du message2

        move.b #4,D0       *demande d'entree du bon code
        TRAP #15           *execution de la demande et attente de
l'appui sur le bouton enter
        move.w #$A,D6       *on met A(=10) en hex dans D6
        divu D6,D1          *on divise le nombre entree par
l'utilisateur par D6 (10)
        move.l D1, (A3)     *on met le resultat de la division (A3)
adresse 4038
        move.b $4039,$4033  *on met le contenu de l'adresse 4039 qui
represente le reste de la division par 10, dans 4033
        clr.l D1            *on efface le contenu de D1
        move.w $403A,D1     *on met le resultat de la 1ere division par
10 dans D1
        divu D6,D1          *on refait la division par 10 et on refait
les etapes precedantes pour avoir dans l'adresse 4030->4033
        clr.l (A3)          *la conversion du code entree a un code hex
qui represente les 4 chiffres dans une memoire
        move.l D1, (A3)
        move.b $4039,$4032
        clr.l D1
        move.w $403A,D1
        divu D6,D1
        clr.l (A3)
        move.l D1, (A3)
        move.b $4039,$4031
        clr.l D1
        move.w $403A,D1
        divu D6,D1
        clr.l (A3)
        move.l D1, (A3)
        move.b $4039,$4030
        clr.l (A3)
        move #4030,A3
*-----Fin de la conversion et sauvegarde du bon code dans l'adresse
4030->4033-----

*-----Meme code que le programme 1 mais au lieu de comparer D1 a code
entree sur le hardware
*-----on compare la valeur sauvegarder dans (A3) pointeur a l'adresse
4030->4033 a la valeur entree
*-----sur le hardware.
*-----init pour reinitialiser notre programme apres reset (interruption
CA1)-----
initi   move.l #$4000,A2    *on met 4000 en hexa dans A2 (qui va
pointer a l'adresse 4000 de la memoire, qui nous permettra de post-
incrementer)
        clr.l (A2)          *on efface le contenu de l'adresse dont
A2 est pointeur
        clr.l (A4)          *on efface le contenu de l'adresse dont
A4 est pointeur
        clr.l D0            *on efface le contenu de D0
        clr.l D2            *on efface le contenu de D2
        clr.l D3            *on efface le contenu de D3

```

```

        clr.l D4                *on efface le contenu de D4
        clr.l D5                *on efface le contenu de D5
        clr.w $E00000           *on efface le contenu de l'adresse
E00000 (position 1 de gauche)
        clr.w $E00002           *on efface le contenu de l'adresse
E00002 (position 2 de gauche)
        clr.w $E00004           *on efface le contenu de l'adresse
E00004 (position 3 de gauche)
        clr.w $E00006           *on efface le contenu de l'adresse
E00006 (position 4 de gauche)
        clr.w $E00008           *on efface le contenu de l'adresse
E00008 (position 5 de gauche)
        clr.w $E0000A           *on efface le contenu de l'adresse
E0000A (position 6 de gauche)
        clr.w $E0000C           *on efface le contenu de l'adresse
E0000C (position 7 de gauche)
        clr.w $E0000E           *on efface le contenu de l'adresse
E0000E (position 8 de gauche)

*-----lecture du nombre entree-----
lecture move.b ira(A0),D2        *on met le valeur entree dans D2
        move.w #10000,D3        *on met 10000 en binaire dans D3
        and D2,D3               *on realise une operation ET (AND)
entre D2 et D3 pour savoir si le bouton 4 est active
        cmp #10000,D3           *on compare 10000 en binaire a D3 pour
savoir si le bouton 4 est active
        beq test1               *si oui on pass au code de l'etiquette
test1
        bne lecture             *sinon on passe au code de l'etiquette
lecture attendre la validation du nombre entree par l'utilisateur

test1  move.b ira(A0),D5        *si bouton 4 active on met la valeur
entree dans D5
        move.w #10000,D3        *on met 10000 en binaire dans D3
        and D5,D3               *on realise une operation ET entre D5 et
D3 pour effacer le nombre entree et garder 1 (si activer sinon 0)
        cmp #$0,D3              *on compare 0 a D3 pour savoir si on a
eteint le bouton 4
        beq suite1              *si le bouton 4 est eteint c.a.d que
l'utilisateur a valider son choix de chiffre et on passe au code de
l'etiquette suite1
        bne test1               *sinon on relance le code de
l'etiquette test1 pour attendre la validation du choix de l'utilisateur

suite1 move.w #01111,D3         *si le bouton 4 est desactive, on met
01111 en binaire dans D3
        and D3,D2               *on realise une operation ET entre D3
et D2 pour retirer le 1 situe sur le 5eme bit de la valeur saisi
        move.b D2,(A2)+          *on met ce chiffre dans l'adresse que
contient A2 et on post incremente A2 (ex:si A2=4000,apres le move A2=40001)
        bra affichage           *on saute au code de l'etiquette
affichage qui nous permettra d'afficher le chiffre entree sur l'ecran

        *apres avoir afficher le chiffre saisi
et l'avoir positionner on passe a l'etape suivante
cont   add #1,D4                 *on incremente de 1 D4 (compteur du
nombre entree)
        add #$1,(A4)             *on incremente de 1 le le contenue de
A4 pour passer a la position suivante de l'affichage
        cmp #$4,D4               *on compare D4 a 4 pour verifier si on
a saisi les 4 chiffres

```

```

        bgt test2                *si D4 plus grand que 4 on passe au
code de l'etiquette test2 c.a.d qu'on a entree 4 chiffre
        blt lecture              *sinon on reprend le code de
l'etiquette lecture pour entree le chiffre suivant jusqu'a se qu'on a saisi
4 chiffres

*-----verification du code entree-----
test2   move.l #$4000,A1          *on met 4000 en hex dans A1
        move.l (A3),D1           *on met (A3) dans D1
        cmp.l (A1),D1           *on compare le contenu de l'adresse
dans A1 a D1
        beq aff_oui              *si elle est egale on passe au code de
l'etiquette aff_oui pour afficher que le code est bon
        bne aff_hou              *le code entree est faux on passe au
code de l'etiquette aff_hou

aff_oui move.b zero,$E00000      *on affiche 0 qui correspond a la
lettre O dans la premiere case de gauche de l'afficheur
        move.b let_U,$E00002     *on affiche U dans la deuxieme case de
gauche de l'afficheur
        move.b let_I,$E00004     *on affiche I dans la troisieme case de
gauche de l'afficheur
        move.b #$F,orb(A0)       *on allume les 4 de droite sous
l'afficheur pour indiquer que le code est bon
        bsr stop                *on passe au code de l'etiquette stop
pour arreter le programme

aff_hou move.b let_H,$E00000     *on affiche H dans la premiere case de
gauche de l'afficheur
        move.b zero,$E00002     *on affiche 0 qui correspond a la
lettre O dans la deuxieme case de gauche de l'afficheur
        move.b let_U,$E00004     *on affiche U dans la premiere case de
gauche de l'afficheur
        move.b #$0,orb(A0)       *on eteint les led qui son sous
l'afficheur pour indiquer que le code est faux
        add #$1,D7               *on incremente de 1 D7
        add #$1,(A6)             *on incremente de 1 le contenu de
l'adresse contenu dans A6 (qui contient le nombre d'essai)

        cmp #$3,(A6)            *on compare le contenu de l'adresse
contenu dans A6 a 3
        blt initi                *si c'est plus petit donc on a encore
le droit d'essayer donc on passe au code de l'etiquette initi pour
reinitialiser notre code
        bsr stop                *sinon on a plus le droit d'essayer et
on passe au code de l'etiquette stop pour arreter le programme qui affichera
*HOU et le 3eme mauvais code entree

*-----Affichage des nombres sur le LCD-----
affichage cmp #$0,D2            *on compare 0 a D2
        beq aff_zero            *si D2 == 0 on affiche 0
        cmp #$1,D2             *on compare 1 a D2
        beq aff_un              *si D2 == 1 on affiche 1
        cmp #$2,D2             *on compare 2 a D2
        beq aff_deux            *...
        cmp #$3,D2
        beq aff_trois
        cmp #$4,D2
        beq aff_quatre
        cmp #$5,D2
        beq aff_cinque

```

```

        cmp #$6,D2
        beq aff_six
        cmp #$7,D2
        beq aff_sept
        cmp #$8,D2
        beq aff_huit
        cmp #$9,D2
        beq aff_neuf
*----Fin boucle de comparaison et passage a l'etiquette d'affichage
specifique au chiffre contenu dans D2-----
*----Debut Affichage et positionnement-----
*----Affiche 0-----
aff_zero cmp #$0, (A4)          *on compare 0 au contenu de l'adresse
mise dans A4                    *si egale a 0 on affiche a la position
        beq aff_pos10          5 de gauche
        cmp #$1, (A4)          *on continue la comparaison jusqu'a 3
pour afficher les 4 chiffres a droite en passant aux
        beq aff_pos20          *codes des etiquettes aff_posXX avec
(XX = position, nombre a afficher)
        cmp #$2, (A4)          *...
        beq aff_pos30
        cmp #$3, (A4)
        beq aff_pos40
        rts                    *apres avoir affiche le numero sur la
case specifique on retourne a notre programme
aff_pos10 move.b zero,$E00008    *on met la constante defini par
'zero'dans l'adresse E00008 en hexa pour afficher '0' dans la case 5 de
l'afficheur
        bra cont                *on passe au code de l'etiquette cont
pour continuer notre code apres l'affichage
aff_pos20 move.b zero,$E0000A    *on refait de meme pour les 4 positions
et pour les 9 autres chiffres
        bra cont                *...
aff_pos30 move.b zero,$E0000C
        bra cont
aff_pos40 move.b zero,$E0000E
        bra cont
*----Affiche 1-----
aff_un   cmp #$0, (A4)
        beq aff_pos11
        cmp #$1, (A4)
        beq aff_pos21
        cmp #$2, (A4)
        beq aff_pos31
        cmp #$3, (A4)
        beq aff_pos41
        rts
aff_pos11 move.b un,$E00008
        bra cont
aff_pos21 move.b un,$E0000A
        bra cont
aff_pos31 move.b un,$E0000C
        bra cont
aff_pos41 move.b un,$E0000E
        bra cont
*----Affiche 2-----
aff_deux cmp #$0, (A4)
        beq aff_pos12
        cmp #$1, (A4)
        beq aff_pos22

```

```
        cmp #2, (A4)
        beq aff_pos32
        cmp #3, (A4)
        beq aff_pos42
        rts
aff_pos12 move.b deux, $E00008
        bra cont
aff_pos22 move.b deux, $E0000A
        bra cont
aff_pos32 move.b deux, $E0000C
        bra cont
aff_pos42 move.b deux, $E0000E
        bra cont
*----Affiche 3-----
aff_trois  cmp #0, (A4)
        beq aff_pos13
        cmp #1, (A4)
        beq aff_pos23
        cmp #2, (A4)
        beq aff_pos33
        cmp #3, (A4)
        beq aff_pos43
        rts
aff_pos13 move.b trois, $E00008
        bra cont
aff_pos23 move.b trois, $E0000A
        bra cont
aff_pos33 move.b trois, $E0000C
        bra cont
aff_pos43 move.b trois, $E0000E
        bra cont
*----Affiche 4-----
aff_quatre  cmp #0, (A4)
        beq aff_pos14
        cmp #1, (A4)
        beq aff_pos24
        cmp #2, (A4)
        beq aff_pos34
        cmp #3, (A4)
        beq aff_pos44
        rts
aff_pos14 move.b quatre, $E00008
        bra cont
aff_pos24 move.b quatre, $E0000A
        bra cont
aff_pos34 move.b quatre, $E0000C
        bra cont
aff_pos44 move.b quatre, $E0000E
        bra cont
*----Affiche 5-----
aff_cinque  cmp #0, (A4)
        beq aff_pos15
        cmp #1, (A4)
        beq aff_pos25
        cmp #2, (A4)
        beq aff_pos35
        cmp #3, (A4)
        beq aff_pos45
        rts
aff_pos15 move.b cinq, $E00008
        bra cont
```

```
aff_pos25 move.b cinq, $E0000A
        bra cont
aff_pos35 move.b cinq, $E0000C
        bra cont
aff_pos45 move.b cinq, $E0000E
        bra cont
*----Affiche 6-----
aff_six  cmp #0, (A4)
        beq aff_pos16
        cmp #1, (A4)
        beq aff_pos26
        cmp #2, (A4)
        beq aff_pos36
        cmp #3, (A4)
        beq aff_pos46
        rts
aff_pos16 move.b six, $E00008
        bra cont
aff_pos26 move.b six, $E0000A
        bra cont
aff_pos36 move.b six, $E0000C
        bra cont
aff_pos46 move.b six, $E0000E
        bra cont
*----Affiche 7-----
aff_sept cmp #0, (A4)
        beq aff_pos17
        cmp #1, (A4)
        beq aff_pos27
        cmp #2, (A4)
        beq aff_pos37
        cmp #3, (A4)
        beq aff_pos47
        rts
aff_pos17 move.b sept, $E00008
        bra cont
aff_pos27 move.b sept, $E0000A
        bra cont
aff_pos37 move.b sept, $E0000C
        bra cont
aff_pos47 move.b sept, $E0000E
        bra cont
*----Affiche 8-----
aff_huit cmp #0, (A4)
        beq aff_pos18
        cmp #1, (A4)
        beq aff_pos28
        cmp #2, (A4)
        beq aff_pos38
        cmp #3, (A4)
        beq aff_pos48
        rts
aff_pos18 move.b huit, $E00008
        bra cont
aff_pos28 move.b huit, $E0000A
        bra cont
aff_pos38 move.b huit, $E0000C
        bra cont
aff_pos48 move.b huit, $E0000E
        bra cont
*----Affiche 9-----
```



```
aff_neuf    cmp #0, (A4)
            beq aff_pos19
            cmp #1, (A4)
            beq aff_pos29
            cmp #2, (A4)
            beq aff_pos39
            cmp #3, (A4)
            beq aff_pos49
            rts
aff_pos19   move.b neuf, $E00008
            bra cont
aff_pos29   move.b neuf, $E0000A
            bra cont
aff_pos39   move.b neuf, $E0000C
            bra cont
aff_pos49   move.b neuf, $E0000E
            bra cont
*----Fin Affichage et positionnement-----
*----Code le 1'interruption sur front montant de CA1-----
inter1      move.l #$4000, A2
            clr.l (A1)
            clr.l (A2)
            clr.l (A4)
            clr.l (A6)
            clr.l D0
            clr.l D2
            clr.l D3
            clr.l D4
            clr.l D5
            clr.w $E00000
            clr.w $E00002
            clr.w $E00004
            clr.w $E00006
            clr.w $E00008
            clr.w $E0000A
            clr.w $E0000C
            clr.w $E0000E
            bset #1, ifr(A0)
            rte
*----Fin du Code le 1'interruption sur front montant de CA1-----
stop        move.b #9, D0
            TRAP #15

SIMHALT          ; halt simulator

* Put variables and constants here
pin          dc.b 6, 9, 1, 2
zero         dc.b $3F
un           dc.b $06
deux         dc.b $5B
trois        dc.b $4F
quatre       dc.b $66
cinque       dc.b $6D
six          dc.b $7D
sept         dc.b $07
huit         dc.b $7F
neuf         dc.b $6F
let_U        dc.b $3E
let_I        dc.b $30
let_H        dc.b $76
```

```
off      dc.b  $00
MESSAGE1  DC.B  '----- TP4 Partie 2 -----',CR,LF,0
MESSAGE2  DC.B  'Entrez le mot de passe : ',0

      END      START      ; last line of source
```

Explication de la conversion réaliser :

Lorsqu'on rentre un nombre de 4 chiffres sur la console, le programme va sauvegarder ce nombre en hexa (ex : code=8997 en hexa=2325). Vu que le nombre entré par le hardware et sauvegarder par chiffre la comparaison de sera pas bonne (c'est comme si on compare en binaire 8997 à 8997 en hexa ce qui n'est pas égale). On a procédé à faire une décomposition du nombre entré en 4 parties. On divise le nombre entré par la console par 10 pour obtenir le dernier chiffre du nombre rentré comme reste. Ensuite on sauvegarde le reste dans une adresse, puis on divise le résultat de la 1ere division à nouveau par 10 pour avoir comme reste le 3eme chiffre du nombre entré. On répète cette procédure 4 fois pour pouvoir décomposer le nombre entré en 4 chiffres et pour pouvoir le comparer au code entre par le hardware.