INF2170
Examen intra Hiver 1999

Question No 1 (5 %)

Choisissez deux nombres représentables sur 16 bits en complément à deux et qui sont tels que lorsqu'on les additionne on a bit C=1 et bit V=1.

Solution

Il faut nécessairement que les deux nombres soient négatifs. On peut choisir deux nombres quelconques tels que leur somme est inférieure à -32768. Il y a donc un grand nombre de solutions possibles. En voici une : -1 et -32768.

Question No 2 (10 %)

Écrire une suite d'instructions pour évaluer l'expression `(x+y)*x-100`, où `x` et `y` sont définis ci-dessous ; placer le résultat dans le registre `d0` (32 bits). La donnée `x` est potentiellement négative, mais le `y` est un nombre positif compris entre 0 et 255.

```
x    ds.w  1
y    ds.b  1
```

Solution

```
clr.w  d0        ; pour la conversion de y sur 16 bits
move.b y,d0      ; y sur 16 bits
add.w  x,d0      ; (x+y) ici il y a débordement possible, mais on ne le teste pas
muls.w x,d0      ; (x+y)*x
sub.l  #100,d0   ; (x+y)*x-100
```

Question No 3 (15 %)

Supposons les données suivantes stockées en mémoire.

```
x   dc.b  '1234'
y   dc.w  $FFFF
z   dc.l  12
```

Indiquez pour chaque suite d'instructions suivantes le contenu du registre d0 (32 bits) après l'exécution DE CHAQUE INSTRUCTION. (Les suites sont indépendantes.)

Solution

a) (donnez le contenu de d0 en décimal)

```
move.w y,d0  ; partie haute inconnue, partie basse -1
ext.l d0     ; -1 en décimal
add.l z,d0   ; 11 en décimal
```

b) (donnez le contenu de d0 en décimal)

```
move.l  z,d0  ; 12 en décimal
add.w   y,d0  ; 11 en décimal
```

c) (donnez le contenu de d0 en caractère)

```
move.l  x,d0              ; '1234'
add.w   #$0101,d0         ; '1245'
sub.l   #$01010101,d0     ; '0134'
add.b   #1,d0             ; '0135'
```

Question No 4 (20 %) Le registre D0.b a une valeur numérique quelconque. Le tableau tabCodes a 62 150 octets. Trouvez dans le tableau le premier octet qui contient la même valeur que D0.b et retournez en D7.l la position de l'octet trouvé. Écrivez dans l'octet trouvé la valeur hexadécimale FF. Si aucun octet du tableau ne contient la valeur contenue en D0.b, retournez la valeur hexadécimale FFFFFFFF dans D7.

Solution

```
        lea     TabCodes,a0
        clr.l   d7
        move.l  #62149,d1
cherche:cmp.b   (a0)+,d0
        beq     trouve
        add.l   #1,d7
        dbra    d1,cherche
        move.l  #-1,d7
        bra     continue
trouve: move.b  #$FF,-(a0)
continue:
         ...
         data
TabCodes ds.b 62150
         end
```

Question No 5 (10 %) Soit le code

```
move.l  #$1080C0,d0
move.l  d0,d1
mulu.w  d1,d0
jsr     decout
```

Lors de l'exécution, que devrait-t-on voir à lécran? Soyez précis.

Solution

Remarquez que la multiplication n'implique que les parties basses des registres. De plus, puisque la sous-routine `decout` n'affiche que la partie basse de `d0`, il est suffisant d'évaluer seulement cette partie du résultat de la multiplication. Le calcul de la partie basse de $80c0 \times 80c0$ ne demande que l'addition de deux nombres hexadécimaux : $80c0$ décalé à gauche de 6 bits et 7 bits, i.e. 3000 et 6000; donc 9000. C'est un nombre négatif dont le complément à deux est $7000_{16} = 7 \times 2^{12} = 7 \times 4096_{10} = 28672_{10}$.

```
move.l  #$1080C0,d0    001080C0
```

```
move.l  d0,d1          001080C0
mulu.w  d1,d0          80C0 * 80C0 = _____9000(16)
jsr     decout         affiche le résultat 9000(16) = -28672(10)
```

Question No 6 (10 %)

Nommez trois modes d'adressages et pour chacun donnez un exemple d'instruction qui contient une opérande du mode d'adressage indiqué.

Solution Il existe de nombreux mode d'adressage, en voici quelques uns.

```
direct:          move.l d1,d2
immédiat:        move.l #1,...
basé:            move.l (2,a5),...
indexé:          move.l (0,a2,d2),...
indirect:        move.l (a3),...
post-incrément:  move.l ...,(a0)+
pré-décrément:   move.l ...,-(a0)
```

Question No 7 (15 %)

Laquel (Lesquels) des instructions suivantes ne modifie pas de registre d'adresse.

Solution

```
a) movea.l D0,A0        A0 est modifié
b) move.l A0,D0         A0 n'est pas modifié
c) move.l (A0),D0       A0 n'est pas modifié
d) move.l (2,A0),D0     A0 n'est pas modifié
e) move.l D0,(A0)       A0 n'est pas modifié
f) move.l D0,(4,A0,D1)  A0 n'est pas modifié
g) move.l (A0)+,(A1)+   A0 est modifié
h) clr.l A0             A0 est modifié
```

Question No 8 (15 %)

Donnez le contenu hexadécimal des variables VAR1 à VAR5. Rappel: valeur hexadécimale ascii de 'A'est 41 et 'a' est 61.

Solution

```
VAR1  dc.l  513          00000201
VAR2  dc.l  -1024        FFFFC000
VAR3  dc.l  @147662      0000CFB2
VAR4  dc.w  %10010       0012
VAR5  dc.b  'Bien'       4269656E
```