

TD Architecture et Système 1:

Assembleur 68000 -1-

Les suffixes en langage assembleur 68000 :

« .B » correspond à 8 bits de données.

« .W » correspond à 16 bits de données.

« .L » correspond à 32 bits de données.

Réservation de données :

int x= correspond à DC.L valeur 32bits initialisée

int t[..] correspond à DS.L tableau

Dans le code :

Move correspond à une copie

Add, sub, muls, ... correspondent aux opérations de l'UAL.

Bra correspond à un branchement inconditionnel.

Beq, bne, bgt, bge, ble, blt correspondent à des branchements conditionnels.

RTS correspond au retour d'une fonction ou à la fin d'un programme.

Registres :

De données : de D0 à D7 (utilisé pour stocker des variables intermédiaires).

D'adresses : de A0 à A6.

Exercice 1 :

Que fait le programme assembleur suivant :

```

                DATA
V :             DC.L  2
U :             DC.L  9
T :             DC.L  1
X :             DC.L  4

                CODE
                MOVE.L    X, D0
                CMP.L     #2 , D0
                BNE       SINON
                MOVE.L    U,T
                BRA       FIN
SINON :        MOVE.L    V,T
FIN :          RTS
```

Exercice 2 :

Programmez en assembleur (la partie CODE) la fonction f1() donnée ci-dessous :

```
f1()
{
    if ((y*(2x+4)) == 0)
        t = u ;
    else
        t = v ;
```

```
}
```

Définissez la partie DATA qui vous permettra de tester votre programme.

Exercice 3 :

Programmez en assembleur 68000 les fonctions suivantes :

- 1) f2(){if ((x == 0) || (y == 0)) z = 3 ;}
- 2) f3() {if (x == 14) z = 5 ;
else if (x == 27) z = 6 ;
else if (x == 8) z = 7 ;
else z=-1 ;}.
- 3) f4() { switch (x) { case 1 : z=5 ; break;
case 2: z=2 ; break;
default: z=7; }}
- 4) f5() { while (x < 2) { z = z+x; x++;}}
- 5) f6() { do { z = z + x; } while (z < 5); }
- 6) f7() { for(i=5; i<=10; i++) { z = z + x; }}
- 7) f8() { for(i=5; i<=10; i++) { if (i<5) z = z + x; else z=z++;}}

Exercice 4 :

Définir un tableau tab1 de 10 octets et initialiser les différents éléments du tableau à 1.

```
char tab1[10] ;  
for (i=0 ; i<10 ; i++)  
    tab1[i] = 1 ; //tab[i] = i
```

Exercice 5:

Définir un tableau tab3 de 10 mots longs et initialiser les différents éléments du tableau comme suit :

```
int tab3[10] ;  
for (i=0 ; i<10; i++)  
    tab3[i] = i ;
```