

TD5 : Programmation Assembleur

Exercice 1

Sachant que le bus d'adresse du processeur est de 16 bits avec un alignement à l'octet, quelle est la taille de l'espace mémoire maximum que celui-ci peut adresser ?

1. Quelles solutions existent pour adresser une plus grande zone mémoire ?
2. Où sont effectués les calculs ?
3. A quoi servent les registres suivants du processeur :
 - i. PC/IP (ou CO/PI)
 - ii. IR (ou RI)
 - iii. SP (ou PP)
 - iv. Accumulateur
4. Quelle tâche réalise le séquenceur dans un processeur ? l'ordonnanceur ?

Exercice 2

Comme cela a été dit en cours, à chaque processeur est associé un langage machine. Voici un langage assembleur du langage machine du Motorola 68000

La mémoire est une mémoire de mot de 8 Bits.

L'instruction MOVE S, D transfère la source S sur la destination D

Le \$ signifie valeur en Hexa

Le # signifie adressage immédiat

Les registres Di, Ai sont des adressages direct registre, les registres sont de 32 bits

Les adresses mémoire sont des entiers.

Le () signifie adressage indirect

le -() est un adressage pré-décrémenté (La valeur d'incrément est de 1 ou 2 suivant le nombre d'octets transférés)

Le () + est un adressage post-incrémenté

Et le n(,) est un adresse indirect indexé avec déplacement

Les instructions manipulent des mots de 2 octets par défaut, de 1 octet si le « .B » est spécifié derrière l'instruction MOVE. Les bits de poids faible sont ceux qui sont modifiés.

Exécutez ce programme et pour chaque instruction montrez les changements intervenus sur les contenus des registres et de la mémoire. Les registres sont initialisés à \$00000000.

```
MOVE    #$0010,D0
MOVE    #$10FB,$2000
MOVE    #$2002,A0
MOVE    -(A0),D1
MOVE.B  (A0)+,D2
MOVE.B  (A0)+,D2
MOVE    #$8FFF,16(A0,D0)
```

Exercice 3

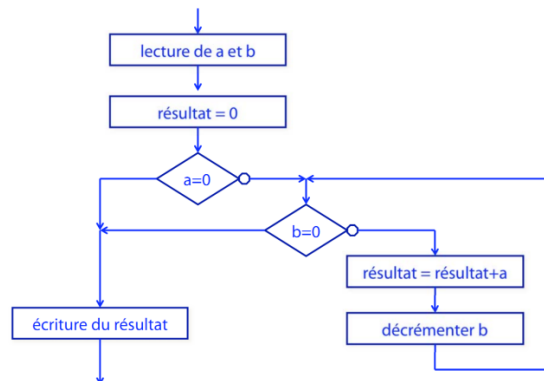
Supposez que vous voulez multiplier deux variables a et b, stockées dans les positions de mémoire M[20] et M[21], respectivement, pour affecter cette valeur à la variable c, stockée à la position de mémoire M[40] C'est-à-dire, vous voulez effectuer l'opération:

c = a*b ou: M[40] = M[20]*M[21]

Le processeur possède 8 registres (R0...R7). Le registre R0 contient initialement la valeur 0. Les instructions du langage machine du processeur sont:

LOAD Rd, M[adr] $Rd \leftarrow M[adr]$
STORE M[adr], Rs $M[adr] \leftarrow Rs$
ADD Rd, Rs1, Rs2 $Rd \leftarrow Rs1 + Rs2$
SUB Rd, Rs1, Rs2 $Rd \leftarrow Rs1 - Rs2$
DEC R $R \leftarrow R - 1$
JMP adr sauter à adr
JZ Rs, adr si Rs = zero alors sauter à adr

Avant d'écrire le programme, nous devons trouver un algorithme réalisant la tâche voulue. Un algorithme possible serait:



1. Donnez le code correspondant à cet algorithme qui commence à l'adresse b'0000
2. Donnez le code correspondant à cet algorithme qui commence à l'adresse b'0010
3. Comment étendre le jeu d'instruction pour écrire un programme indépendant de l'adresse de départ. Réécrire le programme. Quelle différence avec l'utilisation d'un assembleur et d'un label.