



LMI-TD02: Instructions arithmétiques, instructions de contrôle et de test, écriture en mémoire vidéo.

1 But du TD

Manipulation des instructions : div, idiv, neg.

Manipulation de l'instruction de saut inconditionnel : jmp.

Manipulation des instructions de saut conditionnel : jcc et , éventuellement loop.

Manipulation de l'instruction de test : cmp.

Ecrire directement en l'adresse B8000h.

2 Division entière

2.1 Entiers non signés

Ex1. Soient v1, v2, v3 et v4 quatre variables de taille un octet. Ecrivez un code qui stocke dans v3 le résultat de la division entière des contenus des variables v1 et v2 et dans v4 le reste de cette division.

Testez votre code dans td avec les valeurs suivantes :

- v1 : 21 et v2 : 4 ;
- v1: 4 et v2 : 21 ;
- v1: 250 et v2 : 1.

Ex2. Même question que l'Ex1 mais cette fois la variable v1 fait deux octets de long.

Testez votre code dans td avec les valeurs :

- v1 : 3 000 et v2 : 27 ;
- v1: 10 000 et v2 : 5 ;
- v1 : 10 000 et v2 : 127 ;
- v1 : 10 000 et v2 : -127.

Peut-être rencontrez-vous un problème avec le second couple de valeurs proposées.

Comment le résoudre ?

Qu'observez-vous pour le quatrième couple de valeurs test ?

2.2 Entiers signés

Ex3. Modifiez votre code de l'Ex2 de sorte qu'il permette la division d'entiers positifs ou négatifs, en d'autres termes : d'entiers signés.

Testez votre code dans td avec les valeurs :

- v1 : 10 000 et v2 : 127 ;
- v1 : -10 000 et v2 : 127 ;
- v1 : 10 000 et v2 : -127 ;
- v1 : -10 000 et v2 : -127.

2.3 Exercice récapitulatif : volume du cône de révolution

Ex4. Le volume V , d'un cône de révolution de base de rayon r et de hauteur h vaut :

$$V = (\pi * r^2 * h) / 3$$

où $\pi = 3.141592654 \dots$

Ecrivez un code qui stocke dans la variable `volume` le volume du cône de révolution dont le rayon de la base (entier positif) est contenu dans la variable `rayon` et la hauteur (entier positif) dans `hauteur`. Ces deux dernières variables ont pour taille un octet. Prenez $\pi = 3$ et choisissez au mieux la taille de la variable `volume`. Testez dans `td` avec les valeurs : `rayon=6` et `hauteur=15`.

Comment augmenter la précision jusqu'à deux décimales ($\pi = 3.14$) ? Ecrivez le code source correspondant.

3 Comparaisons et sauts

Ex5. Soient `a` et `valAbs` deux variables de taille un octet. Ecrivez un code qui stocke dans

`valAbs` la valeur absolue du contenu de `a`.

Testez votre code dans `td` avec les valeurs suivantes :

- `a` : 25 ;

- `a` : -30 ;

- `a` : 200.

Qu'observez-vous ?

Ex6. Soient `a`, `aMin` et `aMax` trois variables de taille deux octets et `estIn` une variable de taille un octet. On suppose avoir toujours `aMin < aMax`. Ecrivez deux codes qui stockent dans

`estIn` :

- la valeur 1 si `aMin ≤ a < aMax` ;

- la valeur 0 sinon.

Le premier code ne manipule que des entiers non signés. Le second est adapté aux entiers signés.

Testez vos codes dans `td`.

4 Boucles

Ex7. Soient `N` et `somme` deux variables de taille deux octets. On considère le contenu de `a` toujours positif. Ecrivez un code qui stocke dans `somme` la somme des entiers positifs de 0 à `N`

Testez votre code dans `td` avec les valeurs suivantes :

- `N` : 25 ;

- `N` : 361 ;

- `N` : 400.

Qu'observez-vous ?

Modifiez votre code de sorte que la valeur nulle (0) soit stockée dans la variable `somme` si le résultat du calcul effectué n'est pas stockable sur deux octets.

5 Ecrire directement en l'adresse B8000h

Ex8 : Ecrivez un code assembleur qui permet d'afficher un message donné à l'écran. On écrira directement en l'adresse B8000h (ne pas utiliser les services d'interruption). Ecrivez un autre code qui utilisera les services d'interruption.

6 Exercice d'évaluation : Rechercher un min et un max parmi trois nombres; effectuer une division entière ; affichage des résultats à l'écran.

Ex9 : Soient 3 variable v1, v2, et v3 de taille 2 octets. Ecrivez un programme qui recherche le plus petit et le plus grand parmi ces 3 variables, puis qui réalise les opérations suivantes :

- si l'une des 2 variables recherchées (le min ou le max) est nulle, le programme stocke la valeur 1 dans une variable estNul et il se termine.
- si l'une des 2 variables recherchées (le min ou le max) est négative, le programme stocke la valeur 1 dans une variable estNeg, puis il réalise la division entière de la variable (le min ou le max) qui est la plus grande en valeur absolue par celle qui est plus petite en valeur absolue. Le quotient de cette division est stocké dans une variable Quotient et le reste dans une variable Reste.
- si les 2 variables recherchées (le min et le max) sont positives, le programme stocke la valeur 1 dans une variable estPos, puis il réalise la division entière de la variable (le min ou le max) qui est la plus grande par celle qui est plus petite. Le quotient de cette division est stocké dans une variable Quotient et le reste dans une variable Reste.

Le programme affichera à l'écran (sans faire usage des services d'interruption) le contenu des variables Quotient et Reste avant de se terminer.