

ARCHITECTURE DES ORDINATEURS

TD : 01

CALCULS EN BINAIRE ET EN HEXADÉCIMAL

Rappels

- La base hexadécimale (base 16) est, comme son nom l'indique, constituée de 16 chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
- Pour éviter les confusions entre bases, on peut noter la base en indice : $42_{16} = 66_{10}$. Par convention, la base est toujours notée en base décimale.
- Chaque chiffre hexadécimal est équivalent à un mot binaire de quatre bits. Notamment, $0_{16} = 0000_2$ et $F_{16} = 1111_2$.
- Dans beaucoup de langages informatiques, la base hexadécimale est indiquée par le préfixe « 0x ».
- Lorsqu'on compte en base 16, tout comme en base 10, on incrémente la puissance immédiatement supérieure dès qu'on atteint le plus grand chiffre d'une certaine puissance, qui repasse à 0 : $0x0 < 0x1 < \dots < 0x9 < 0xA < 0xB < \dots < 0xF < 0x10 < 0x11 < 0x12 < \dots$

Exercice 1 : Addition en binaire sur 8 bits

On considère des mots stockés en mémoire sur 8 bits. Effectuez directement les additions en binaire ci-dessous, sans passer par une autre base.

$n_1 = 00101011$	$n_4 = 01000001$	$n_7 = 11101010$	$n_{10} = 11010101$
$+ n_2 = 00101110$	$+ n_5 = 01010110$	$+ n_8 = 00100011$	$+ n_{11} = 11000111$
$n_3 =$	$n_6 =$	$n_9 =$	$n_{12} =$

Exercice 2 : Addition en hexadécimal sur 8 bits

Question 1

Convertissez en hexadécimal les additions de l'exercice précédent.

Question 2

Recalculez les quatre additions ci-dessus directement en hexadécimal.

Remarques :

- Pour accélérer les calculs, on peut utiliser les mêmes astuces qu'en base décimale. En décimal, ajouter 9 à n donne $(n - 1)$ et une retenue; ajouter 8 à n donne $(n - 2)$ et une retenue. En hexadécimal, ces chiffres sont respectivement F et E.
- Passer en hexadécimal facilite les calculs et permet de vérifier que l'on ne s'est pas trompé sur les additions binaires, que ce soit sur les bits de poids les plus forts ou les parités.

Rappels

- Pour coder les nombres négatifs, on utilise la notation dite « complément à deux » : l'opposé d'un nombre est calculé en complétant chaque bit du nombre, puis en ajoutant 1 à ce complément selon la règle de l'addition classique.
- Les propriétés de cette numérotation sont que :
 - le signe est indiqué par la valeur du bit de poids le plus fort,
 - le codage des nombres entiers naturels ne change pas sur les bits restants,
 - il n'y a qu'un seul codage pour la valeur 0 (à la différence du complément bit à bit simple, dit « complément à un »),
 - la méthode d'addition classique continue à fonctionner sans modification, que les nombres soient signés ou non.
- Selon ce codage, les nombres négatifs de plus petite valeur absolue de leurs bits autres que le bit de signe sont les plus petits en valeur relative. Les mots commençant par 0 à 7 sont positifs, et ceux commençant par 8 à F sont négatifs.
Ainsi, sur des mots de 16 bits : $0x8000 < 0x8001 < \dots < 0xFFFFE < 0xFFFF < 0x0000 < 0x0001 < 0x0002 < \dots < 0x7FFE < 0x7FFF$.

Exercice 3 : Tri de nombres signés sur 16 bits

Trier par ordre croissant : $0xF413$, $0x3F3D$, $0xF102$, $0xCF04$, $0x7F12$.

Exercice 4 : Soustraction en hexadécimal sur 8 bits

Calculez directement en hexadécimal les soustractions ci-dessous, pour des mots stockés en machine sur 8 bits.

$$\begin{array}{r} 2B \\ - 13 \\ \hline \end{array}$$

$$\begin{array}{r} 2B \\ - 3C \\ \hline \end{array}$$

$$\begin{array}{r} 23 \\ - 1B \\ \hline \end{array}$$

Exercice 5 : Complément à deux sur 8 bits

Pour chacune des soustractions de l'exercice précédent, calculez le complément à deux sur 8 bits des nombres soustraits et effectuez l'addition. Vérifiez que vous obtenez bien les mêmes résultats.

Exercice 6 : Retenue et débordement

Pour chacune des additions de l'exercice 1, y a-t-il eu retenue? Y a-t-il eu débordement?

Pour aller plus loin...

Exercice 7 : Multiplication en hexadécimal

$$\begin{array}{r} 16 \\ \times 2B \\ \hline \end{array}$$

Exercice 8 : Division en hexadécimal

$$\begin{array}{r|l} 20 & 3 \\ \hline & \end{array}$$