Culture informatique

Cours n° 3 :

Les opérations logiques

addition bin.

Addition binaire

L'addition est réalisée bit à bit. Il suffit de savoir que :

0 + 0 = 0,

0 + 1 = 1 + 0 = 1

1 + 1 = 10

11 + 1 = 100

A	B	Somme	Retenue
0	0	0	0
0 1	1 0	1	0
1	1	0	1

et d'effectuer éventuellement une retenue comme dans le cas d'une addition décimale.

$$\begin{array}{r} 1\,1\,0\,1\,1 \\ +\quad 1\,1\,0\,1 \\ \hline 1\,0\,1\,0\,0\,0 \end{array}$$

Autre exemple :

$$\begin{array}{r} 1\,0\,1\,1\,0\,1 \\ +\quad 1\,0\,0\,1\,0 \\ \hline 1\,1\,1\,1\,1\,1 \end{array}$$

addition bin.
soustraction bin.

Soustraction binaire

Il y a deux méthodes pour effectuer des soustractions binaires, la méthode directe et la méthode par le complément.

Méthode directe :

On peut opérer comme dans la soustraction décimale.

```
  1 0 1 0 0 1 1
-   1 0 1 1 0 1
---------------
  0 1 0 0 1 1 0
```

Nous verrons la méthode par le complément un peu plus loin.

Multiplication binaire

Il suffit de bien tenir compte des décalages.

$$
\begin{array}{rr}
 & 1\,0\,1\,1 \\
\times & 1\,1\,0\,1 \\
\hline
 & 1\,0\,1\,1 \\
 & 1\,0\,1\,1 \\
+ & 1\,0\,1\,1 \\
\hline
 & 1\,0\,0\,0\,1\,1\,1\,1
\end{array}
$$

Division binaire

$$\begin{array}{r|l} 1\,1\,1\,0\,1 & 1\,0\,1 \\ \cline{2-2} 1\,0\,1 & 1\,0\,1 \\ \cline{1-1} 1\,0\,0\,1 & \\ 1\,0\,1 & \\ \cline{1-1} 1\,0\,0 & \end{array}$$

Il suffit en fait de soustraire 101 lorsqu'on le peut, et d'abaisser le chiffre suivant :

$$11101 = 101 \times 101 + 100$$

Exercice 1 :

a) Additionnez les nombres suivants :

01101111	11011001
10101010	01101100

b) Effectuez les soustractions suivantes :

1101 – 101

10001 – 1111

c) Effectuez les multiplications suivantes :

1000 x 11

10101 x 1010

d) Effectuez les divisions suivantes :

100110 / 101

100100 / 11

addition bin.
soustraction bin.
multiplication bin
division bin.
exercices
opérations hexa.

Opérations en hexadécimal

On pourra s'essayer à construire les tables d'addition, de soustraction ou de multiplication en base 16, et effectuer quelques opérations.

En pratique, on convertira en binaire ou en décimal avant d'effectuer les opérations. On écrira ensuite le résultat dans la base de départ.

Exercice 2 :

a) Effectuez les additions suivantes en hexadécimal :

1FA + A2D

ABC + BCD

b) Effectuez les soustractions suivantes en hexadécimal :

62 – 23

D123 - 1FCB

Complément à 1

Si on prend un nombre A, le complément à 1 de A est le nombre qu'il faut ajouter à A pour que le résultat soit égale à 1.

$A + \bar{A} = 1.$

Le complément à 1 de 0 est 1 car 0 + 1 = 1

Le complément à 1 de 1 est 0 car 1 + 0 = 1

Le complément à 1 de A = 10110010 est $\bar{A}$ = 01001101 obtenu en remplaçant chaque chiffre de A par son complément à 1.

Prendre le complément à 1 d'un nombre binaire revient à remplacer les 0 par des 1 et réciproquement.

Si le complément à 1 de A est $\bar{A}$ alors le complément à 1 de $\bar{A}$ est A.

Complément à 2

Le complément à 2 d'un nombre A sera son complément à 1 auquel on ajoute 1.

Complément à 2 de A = $\bar{A}$ + 1.

On a vu que le complément à 1 de A = 10110010 est $\bar{A}$ = 01001101. Le complément à 2 de A est donc :

01001101 + 1 = 01001110.

Notons que la somme de A et de son complément à 2 est remarquable :

10110010 + 01001110 = 100000000

Les nombres signés

Dans notre système numérique nous utilisons le signe "-" pour préciser qu'un nombre est négatif. Mais lorsqu'on veut représenter un nombre négatif en base binaire on ne dispose pas du signe "-". Logique puisque le code machine se résume à des 1 et des 0.

En informatique, les nombres binaires sont tous de la même longueur (en général des mots de 16 bits, 32 bits...).

On utilise le bit de poids fort (bit le plus à gauche) du nombre pour contenir la représentation de son signe : s'il vaut 1 le nombre est négatif, s'il vaut 0 le nombre est positif.

Mais cela ne suffit pas, en effet, si on se contentait de cette notation (avec 8 bits), -3 s'écrirait 10000011.

Or si on essaye de faire 3 + (-3) : 00000011 + 10000011 = 1000 0110 donc en décimal -6 et pas 0.

C'est pour remédier à ce problème que l'on utilise la notation en complément à deux. Les nombres positifs sont représentés comme attendu, en revanche les nombres négatifs sont obtenus en prenant le complément à 2.

Le bit de signe est automatiquement mis à 1 par l'opération d'inversion.

Ainsi pour obtenir -3, on prend le complément à 2 de 3 :

complément à 1 + 1 : 11111100 + 1 = 11111101

Si on essaye de nouveau de faire 3 + (-3) :

00000011 + 11111101 = 100000000

sur 8 bits, on obtient bien le résultat 0.

Maintenant que l'on sait comment coder les nombres négatifs, il est tout naturel pour faire des soustractions, d'ajouter le complément à 2 d'un nombre plutôt que de soustraire ce nombre.

Exemple avec 4 bits.

Représentations possibles avec 4 bits	Interprété comme un nombre non signé (NS)	Interprété comme un nombre signé (S)
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	-8
1001	9	-7
1010	10	-6
1011	11	-5
1100	12	-4
1101	13	-3
1110	14	-2
1111	15	-1

Soustraction binaire

Méthode par le complément

Comme A – B = A + (-B), la deuxième méthode pour soustraire un nombre est de lui ajouter son négatif. Or le négatif d'un nombre binaire est son complément à 2. Aussi pour soustraire un nombre B à A il suffit d'ajouter à A le complément à 2 de B.

A = 5 B = -3 : le complément à 2 de -3 (sur 4 bits) : 1101

0101	le résultat est bien positif et vaut 0010 c'est
+ 1101	à dire 2.
(1)0010	

A = -2 B = -4 : le complément à 2 de A : 1110

1110	le complément à 2 de B : 1100
+ 1100	le résultat est bien négatif et vaut 1010
(1)1010	c'est à dire -6

Exercice 3 :

a) On travaille avec 8 bits, effectuez les soustractions suivantes et convertissez en décimal pour vérifier votre résultat :

01100100 – 00101000

00010100 – 00100011

b) Convertissez les nombres suivants en binaires puis faite les soustractions :

-10 – 30

-20 – 5