

TD 1 - Représentation de l'Information

EXERCICE 1

Remplissez le tableau suivant.

Considérez tout les nombres binaires comme non signés.

Utilisez la représentation en complément à 1 pour les décimaux négatifs.

| Décimal | Binaire | Héxadécimal | Longeurs en bits |
|---------|---------------------|-------------|------------------|
| 6 | 110 | 6 | 3 |
| 13 | 1101 | D | 4 |
| 123 | 111 1011 | 7B | 7 |
| 210 | 1101 0010 | D2 | 8 |
| 128 | 1000 0000 | 80 | 8 |
| 630 | 10 0111 0110 | 276 | 10 |
| 44 252 | 1010 1100 1101 1100 | ACDC | 16 |
| -48 | 1101 0000 | D0 | 8 |
| 275 | 1 0001 0011 | 113 | 9 |
| 48 879 | 1011 1110 1110 1111 | BEEF | 16 |
| 47 758 | 1011 1010 1011 1110 | BABE | 16 |

EXERCICE 2

Donnez le pseudocode d'un algorithme convertissant un nombre binaire donné sous la forme d'un tableau de bit (le bit 0 étant stocké à la case 0 du tableau, le bit N dans la case N-1) en nombre décimal stocké sous la forme d'un entier.

Algo: BIN2DEC(Int BITS[], Int N)

Debut

Valeur : Int \leftarrow 0

Pour I de N à 0

Valeur = Valeur*2 + BITS[I];

Fin

EXERCICE 3

On rappelle que les nombres réels peuvent être représentés par un nombre dit à **virgule fixe**, c'est à dire un nombre à virgule où les chiffres à gauche de la virgule sont représentés par des puissances de deux positives et ceux à droite de la virgule par des puissance de deux négatives.

On note ces formats $Q\langle n,m \rangle$ où n représente le nombre de bits utilisés pour la partie entière et m le nombre de bits représentant la partie décimale. On considérera ici que ces nombres sont tous positifs ou nul.

3.1 Donnez, pour chacun des formats suivants, la plus petite valeur non nulle représentable et la plus grande valeur représentable:

- $Q\langle 2,2 \rangle$
 - PPV = 0,01 = 0,25
 - PGV = 11,11 = 3,75
- $Q\langle 8,2 \rangle$

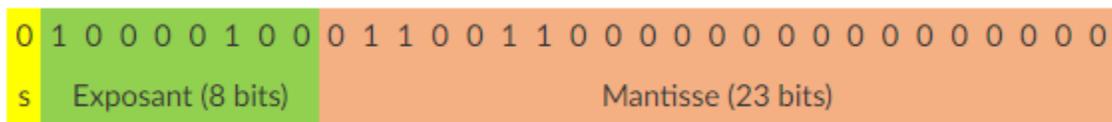
- PPV = 0000 0000, 01 = 0.25
- PGV = 1111 1111,11 = 255,75
- Q<8,8>
 - PPV = 0000 0000, 0000 0001 = 1/256 = 0,00390625
 - PGV = 1111 1111, 1111 1111 = 255, 99609375
 - Astuce PGV = 1111 1111 1111 1111 / 256 = 65535/256

3.2 Quel format Q utiliseriez vous pour représenter sans perte de précision dans un usage normal, les quantités suivantes:

- Une pression comprise entre 0,001 Pa et 5 Pa
 - PPV = 0.001 > 1/1024
 - PGV = 5 ~ 101
 - On propose Q<5,10>
- Une somme d'argent de au plus 8000 EUR au centime près
 - PPV = 0.01 > 1/128
 - PGV = 8000 ~ 8192 = 2¹³
 - On propose Q<13,7>

EXERCICE 4

On considère les nombres flottants encodés selon la norme IEEE754:



Pour convertir ce nombre en décimal, on utilise l'algorithme suivant:

$$X = (-1)^s \times 1, \text{Mantisse} \times 2^{(127 - \text{Exposant})}$$

Ainsi ici, le nombre 01000010001100011000000000000000 est égal à

$$X = (-1)^0 \times 1,011011 \times 2^{(127-(10000100)_2)} = 1 \times 1,0110011 \times 2^{(127-132)} = 1,0110011 \times 2^5$$

$$X = (101100,110)_2 = 44,75$$

4.1 Donnez Le bit de signe, la mantisse, l'exposant et la valeur décimale des nombres suivants:

$$X = (1100\ 0100\ 0100\ 0000\ 0110\ 1000\ 0000\ 0000)_2$$

$$X.SIGN = 1$$

$$X.MANTISSE = 100\ 0000\ 0110\ 1000\ 0000\ 0000$$

$$X.EXPONENT = 10001000$$

$$X.DECIMAL = -769.625$$

$$Y = (0011\ 1111\ 1001\ 1110\ 0000\ 0110\ 0101\ 0001)_2$$

$$Y.SIGN = 0$$

$$Y.MANTISSE = 001\ 1110\ 0000\ 0110\ 0101\ 0001$$

$$Y.EXPONENT = 011\ 1111\ 1$$

$$Y.DECIMAL = 1.2345697$$

$$Z = (0011\ 1110\ 1101\ 0111\ 0110\ 0100\ 1010\ 1110)_2$$

$$Z.SIGN = 0$$

$$Z.MANTISSE = 101\ 0111\ 0110\ 0100\ 1010\ 1110$$

$$Z.EXPONENT = 011\ 1110\ 1$$

$$Z.DECIMAL = 0.42069$$

4.2 Donnez un algorithme simple basé sur la manipulation de bits pour calculer la valeur absolue d'un nombre réel encodé au format IEEE754

Un nombre réel IEEE754 est de la forme 0xxx xxx si positif et 1xxx ... xxx si négatif. Prendre sa valeur absolue revient à mettre le bit de signe à 0

$$ABS(X) = X \text{ ET } 0111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111$$