1. Arithmétique binaire

1.1. Association de caractères binaires

Un BIT (**BI**nary digi**T**) peut contenir 2 valeurs distinctes : `0` ou `1`.

L'association d'un bit supplémentaire permet de distinguer 2 fois plus de valeurs :

En associant un **0** aux possibilités déjà présentes :

0 0 ou **0** 1

En associant un **1** aux possibilités déjà présentes :

1 0 ou **1** 1

En généralisant, dans une variable à n bits, on peut donc stocker 2^n valeurs différentes.

Exemple

Sur 3 bits, on peut stocker $2^3 = 8$ valeurs différentes :

`000, 001, 010, 011, 100, 101, 110, 111`

A retenir

Sur 4 bits, on peut coder .... valeurs différentes.

Sur 8 bits, on peut coder .... valeurs différentes.

Sur 16 bits, on peut coder .... valeurs différentes.

1.2. Identification à des valeurs décimales

Dans le cas général d'une base n à convertir en base 10, chaque caractère représenté sera associé à un *poids* selon son rang, du plus faible à droite au plus fort à gauche, d'une valeur n× plus grande que le précédent à droite.

Dans le cas plus précis du binaire, chaque bit représenté sera associé à un *poids binaire* selon son rang, du plus faible à droite au plus fort à gauche, d'une valeur 2× plus grande que le précédent à droite.

Représentation

Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Valeur	128	64	32	16	8	4	2	1
bit	b7	b6	b5	b4	b3	b2	b1	b0
Exemple	128	64	32	16	8	4	2	1
×	0	0	1	0	1	1	0	1
=	0	0	32	0	+8	+4	0	+1
								= 45 $_{(10)}$

Exercices

Convertir en base 10 :

- 100 1110 0010 1001 $_{(2)}$
- $64223_{(7)}$
- $3A8_{(13)}$
- $4E29_{(16)}$

1.3. Représentation hexadécimale

A partir de la représentation binaire, on peut reconstituer très facilement la représentation hexadécimale par le groupement par 4 bits à partir de la droite, suivi de la conversion de chaque quartet en caractère hexadécimal.

1.4. Conversion décimal/binaire

- Énumérer les poids (puissances de 2 vers la gauche) jusqu'à dépasser la valeur à convertir,
- Pour chacun des poids inférieurs,
  - si inférieur ou égal à la valeur convertie, le soustraire et écrire un 1
  - sinon écrire un 0.

Exercices

Convertir en binaire :

- 1968 $_{(10)}$

1.5. Opérations arithmétiques

1.5.1. L'addition

Revenons aux bases de l'addition en posant les opérations :

```
  157      1 0 0 1  _ _ _ _
+  77    + _ _ _ _  _ _ _ _
 ____    ___________________
  234
```

1.5.2. La multiplication

```
   57      _ _ _ _  _ _ _ _
x  23    x _ _ _ _  _ _ _ _
 ____    ___________________
  171
 114.
 ____
```

2. Les types de données

2.1. Type entier non signé

Tous les bits de la donnée représentent un poids binaire. La valeur résultante est donc forcément entière et positive.

Sur n bits, distinguant 2^n valeurs distinctes, on pourra alors coder les valeurs `0` à `2ⁿ-1`

Poids	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
	b7	b6	b5	b4	b3	b2	b1	b0

Exemple :

Entier non signé sur 8 bits :

2^8 = 256 valeurs positives possibles : 0 à 255

2.2. Type entier signé

Il est parfois nécessaire de travailler sur des nombres négatifs. Il s'agit alors de distinguer le signe de la valeur (+ ou –) par un bit supplémentaire. La taille des mémoires des processeurs étant limitée, le bit le plus à gauche sera réquisitionné pour représenter le bit de signe :

`0` : nombre positif, `1` : nombre négatif.

Poids		2^6	2^5	2^4	2^3	2^2	2^1	2^0
	s	b6	b5	b4	b3	b2	b1	b0

On remarquera cependant que l'on peut représenter la valeur nulle par `+0` ou `-0`, ce qui entraînerait des problèmes de calcul.

Les valeurs négatives seront alors représentées en mode « complément à 2 » sur une taille (nombre de bits) dépendant du processeur utilisé (8 bits, 16 bits, 32 bits, ...).

2.3. Le format « complément à 2 »

Il s'agit de représenter les valeurs négatives par les opérations suivantes, à partir de la valeur positive :

- Complément à 1 (inverser chaque bit)
- Additionner 1

Exemple :
en considérant que l'on travaille sur des formats 8 bits :

45 (10)	**0** 0 1 0 1 1 0 1
Complément à 1 :	**1** 1 0 1 0 0 1 0
+1 :	1
= Complément à 2 :	**1** 1 0 1 0 0 1 1 = – 45 (10)

Astuce :

Autre méthode pour calculer l'opposé d'un nombre binaire (en complément à 2) :

Recopier les bits en partant de la droite jusqu'au premier 1 rencontré inclus, puis inverser tous les autres bits à sa gauche.

Exercices – Calculs sur formats 8 bits signés

Calculer l'opposé de 0(10) en format binaire complément à 2 par le calcul arithmétique et non par l'astuce.

Calculer en binaire (+45) + (–45), (+60) + (-45), (+45) + (-34)

Calculer en binaire (-126), (-127), en déduire ce que serait (-128).

Convertir un nombre négatif binaire en décimal : 10011011

3. Les calculs numériques sur API

3.1. Limitations liées au matériel

Étant données les limitations de stockage de valeurs liées aux formats des données, il faudra s'assurer, **à chaque calcul**, **final ou intermédiaire**, que le résultat entre bien dans les limites disponibles.

Ainsi, on peut être surpris parfois par des résultats obtenus.

Exemple

		s
	28000	**0**110110101100000
+	7000	**0**001101101011000
=		**1**000100010111000

Signe passé à 1, donc résultat négatif

Interprétation du résultat en décimal, en calculant l'opposé :

	1000100010111000
Inversion	**0**111011101000111
+1	1
=	**0**111011101001000 = 30536 (10)

Le résultat de l'addition 28000 + 7000 serait donc **–** 30536 !!