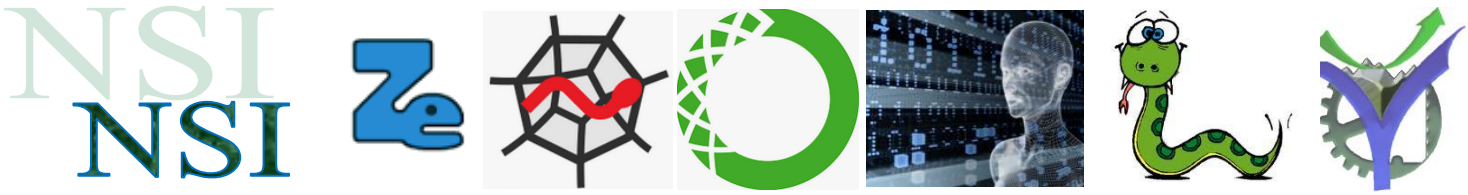


Nombre et ordinateur

Résumé :

Quelques rappels sur la base deux et la représentation des entiers dans les ordinateurs. Ces notions sont intensivement utilisées dans la suite du cours.

Sommaire

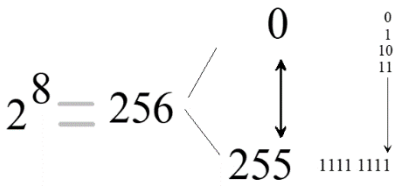
Indique une question à rédiger dans le cahier de cours.

1 Rappel sur la base 2

1.1 Généralités

La base 2 a déjà été vue. Rappelons ici les connaissances utiles pour aborder les notions de représentation des nombres en commençant avec les nombres entiers positifs.

Le système binaire utilise deux symboles notés 0 ou 1, vrai ou faux, high low, tout dépend du domaine concerné.



- 0 et 1 pour les calculs numériques ou les valeurs logiques,
- vrai ou faux dans les problèmes de raisonnements logiques,
- high et low sur les valeurs des tensions mesurées sur les opérateurs logiques des circuits électroniques.

Usuellement : 0 ⇔ faux ⇔ low et 1 ⇔ vrai ⇔ high

Attention cette représentation usuelle, 'des mathématiciens', peut conduire à des erreurs de raisonnements dans la mise en œuvre des circuits logiques 'réels' où le 0 comme le 1 peut représenter la "vérité d'une information".

Pour la représentation des nombres nous utiliserons le codage numérique pondéré voilà un exemple d'écriture sur 4 bits : $dcba_2$ avec $\{a,b,c,d\} \in \{0,1\}$ la valeur est calculée comme suit :

$$d \cdot 2^3 + c \cdot 2^2 + b \cdot 2^1 + a \cdot 2^0 = d \cdot 8 + c \cdot 4 + b \cdot 2 + a \cdot 1$$

Exemples : 1001 = 1•8+0•4+0•2+1•1 = 9 1011 = 1•8+0•4+1•2+1•1 = 11

Q1. Donner le nombre de valeurs avec un nombre de 4 bits de long, puis de 10 bits et 12 bits.

Q2. Donner les valeurs minimales et maximales avec un nombre de 4 bits de long.

Q3. Écrire les 16 chiffres en base 16 avec un format de 4 bits.

Base 10	Base 16	Base 2 sur 4 bits d	c	b	a
0	0	0	0	0	0
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				

Base 10	Base 16	Base 2 sur 4 bits d	c	b	a
8	8	1	0	0	0
9	9				
10	A				
11	B				
12	C				
13	D				
14	E				
15	F				

1.2 Les bases 2, 10 et 16

Les bases 2 et 16 sont intensivement utilisées quelques rappels utiles :

La base 2

La base 2 représente l'information la plus élémentaire dans les systèmes numériques. L'inconvénient est que la représentation de nombres importants utilise un grand nombre de bits. Dans les circuits numériques les calculs se font sur des données de plusieurs dizaines de bits de long. Les représenter en binaire est une gageure.

A ce stade il est essentiel de bien différencier la représentation d'une donnée binaire écrite par une suite de valeurs 0 et 1 de sa signification réelle qui dépend de ce que représente cette donnée et de son codage, voir suite du cours.

Rappelons les dénominations usuelles :

Un bit (bit) : l'élément le plus simple 0 ou 1. Un octet (byte) : une suite de 8 bits.

Un mot (word) : deux octets.

La base 16

Cette base est souvent utilisée pour écrire de manière plus concise des données en base 2. Le passage de la base 2 vers la base 16 se fait sans calcul par simple substitution.

Exemples :

- $1001_2 = 9_{16}$ ou \$9 ou 0x9 00011011 = 0001 1011 = $1B_{16}$ ou \$1B ou 0x1B
- 1110011 = 111 0011 = 73_{16} ou \$73 ou 0x73 \$40A = 0100 0000 1010 = 010000001010_2

A noter : l'indication de la base est parfois omise si aucune ambiguïté n'est possible.

L'intérêt de la base 16 est également la possibilité de calculer 4 fois plus rapidement puisque pour chaque chiffre en base 16 on manipule en réalité 4 bits.

2 Les préfixes binaires normalisés (source Wikipédia)

Préfixes binaires (préfixes CEI)

Nom	Symbole	2^{10a} = facteur	a
kibi	Ki	2^{10} = 1 024	1
mébi	Mi	2^{20} = 1 048 576	2
gibi	Gi	2^{30} = 1 073 741 824	3
tébi	Ti	2^{40} = 1 099 511 627 776	4
pébi	Pi	2^{50} = 1 125 899 906 842 624	5
exbi	Ei	2^{60} = 1 152 921 504 606 846 976	6
zébi	Zi	2^{70} = 1 180 591 620 717 411 303 424	7
yobi	Yi	2^{80} = 1 208 925 819 614 629 174 706 176	8

A connaître :

⇦

$$2^8 = 256 \quad 2^{10} = 1024 \quad 2^{12} = 4096$$

3 Les formats de données numériques dans un ordinateur

Les données numériques dans un ordinateur sont disponibles avec plusieurs types possibles. Après observation du tableau ci-dessous représentant les types disponibles en langage c répondre aux questions suivantes :

Q4. C'est quoi un type ?

Q5. Qu'est-ce qui différencie un type dans la mémoire de l'ordinateur ?

Type de donnée	Signification	Taille (en octets)	Plage de valeurs acceptée
char	Caractère	1	-128 à 127
unsigned char	Caractère non signé	1	0 à 255
short int	Entier court	2	-32 768 à 32 767
unsigned short int	Entier court non signé	2	0 à 65 535
int	Entier	2 (sur processeur 16 bits) 4 (sur processeur 32 bits)	-32 768 à 32 767 -2 147 483 648 à 2 147 483 647
unsigned int	Entier non signé	2 (sur processeur 16 bits) 4 (sur processeur 32 bits)	0 à 65 535 0 à 4 294 967 295
long int	Entier long	4	-2 147 483 648 à 2 147 483 647
unsigned long int	Entier long non signé	4	0 à 4 294 967 295
float	Flottant (réel)	4	$3.4*10^{-38}$ à $3.4*10^{38}$
double	Flottant double	8	$1.7*10^{-308}$ à $1.7*10^{308}$
long double	Flottant double long	10	$3.4*10^{-4932}$ à $3.4*10^{4932}$

4 La représentation des nombres entiers relatifs

Cette représentation est réalisée en prenant le complément à 2 ou complément vrai (CV) du nombre entier positif dont on souhaite connaitre la représentation négative.

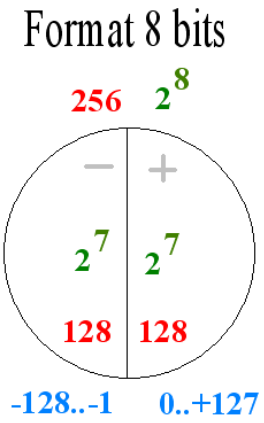
ATTENTION avec un format déterminé on IGNORE tout ce qui dépasse du format.

Dans cette représentation le bit de poids fort égal à 1 indique un nombre négatif.

Avec cette représentation soustraire un nombre entier revient à additionner sa représentation en complément vrai.

Quand on est en présence d'un nombre négatif pour connaitre sa valeur absolue il suffit de calculer le complément vrai de ce nombre.

Exemple pour un format de 8 bits pour déterminer le nombre de combinaisons possibles en positif et négatif :



Q6. Donner les valeurs possibles pour un format de 16 bits avec une représentation des nombres négatifs.

Q7. Donner les valeurs possibles pour un format de 16 bits avec des nombres uniquement positifs.

Détermination du complément à 2 d'un nombre

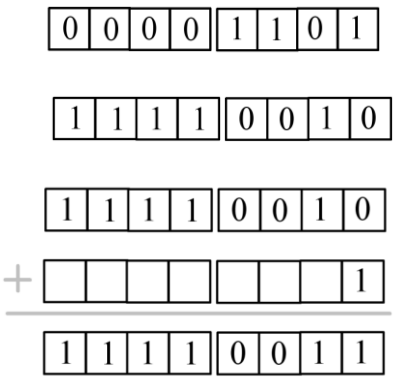
Par exemple -13 sur un format de 8 bits. (Le codage disponible va de -128 à +127)
-13 = CV(13)

Étapes du calcul à suivre impérativement

1. Codage de 13 dans le format imposé.

Puis calcul du complément vrai CV de 13

2. Calculer le complément restreint (CR) ou complément à 1
3. Ajouter 1 et ignorer tout ce qui dépasse le format



Donc la représentation de -13 dans un format de 8 bits est égale à **1111 0011 soit $F3**

Exercices à rédiger

Q8. Déterminer le codage de +35 et -35 format 8 bits

Q9. Déterminer le codage de -98 et -9 format 8 bits

Q10. Calculer en format 8 bits 35 - 9 et -35 - (-98)

Q11. Déterminer en format 8 bits ce que représente 1110 0010

5 Les types principaux dans trois langages de programmation

Type de données	Taille en octets	Plage de valeurs acceptée	c	pascal	python
Entier sur un octet	1	-128 +127	char	Shortint	NON
Entier positif sur un octet	1	0 255	unsigned char	Byte	Voir chaine de caractère
Entier court (16 bits)	2	-32768 +32767	shortint	Integer	NON
Entier long (32 bits)	4	-2147483648 +2147483647	Long int	Longint	OUI
Réel sur 32 bits	4	$3.4\ 10^{-38}$ $-3.4\ 10^{38}$	float	Single	NON
Réel sur 64 bits	8	$1.7\ 10^{-308}$ $-1.7\ 10^{308}$	double	Double	OUI

Le codage du format réel fera l'objet d'une fiche spécialisée.

6 Attention aux formats binaires dans les ordinateurs[1]

Comment sont écrites les données dans les mémoires des ordinateurs, ce qui peut rendre des échanges de données problématiques entre ordinateurs travaillant dans des représentations différentes dans le cas de l'échange de format binaire. (Dans ce cas l'échange en ASCII est préférable).

BIG ENDIAN

Quand certains ordinateurs enregistrent un entier sur 32 bits en mémoire, par exemple `0xA0B70708` en notation hexadécimale, ils l'enregistrent dans des octets dans l'ordre qui suit : `A0 B7 07 08`, pour une structure de mémoire fondée sur une unité atomique de 1 octet et un incrément d'adresse de 1 octet. Ainsi, l'octet de poids le plus fort (ici `A0`) est enregistré à l'adresse mémoire la plus petite, l'octet de poids inférieur (ici `B7`) est enregistré à l'adresse mémoire suivante et ainsi de suite.

	0	1	2	3	
...	A0	B7	07	08	...

[1] Source Wikipédia

Pour une structure de mémoire ou un protocole de communication fondé sur une unité atomique de 2 octets, avec un incrément d'adresse de 1 octet, l'enregistrement dans des octets sera `A0B7 0708`. L'unité atomique de poids le plus fort (ici `A0B7`) est enregistrée à l'adresse mémoire la plus petite.

	0	1	2	3	
...	A0	B7	07	08	...

Les architectures qui respectent cette règle sont dites ***big-endian*** ou **gros-boutistes** ou **mot de poids fort en tête**, par exemple les processeurs Motorola 68000, les SPARC (Sun Microsystems) ou encore les System/370 (IBM).

De plus, tous les protocoles TCP/IP communiquent en *big-endian*[9]. Il en va de même pour le protocole PCI Express.

LITTLE ENDIAN

Les autres ordinateurs enregistrent `0xA0B70708` dans l'ordre suivant : `08 07 B7 A0` (pour une structure de mémoire fondée sur une unité atomique de 1 octet et d'un incrément d'adresse de 1 octet), c'est-à-dire avec l'octet de poids le plus faible en premier. De telles architectures sont dites ***little-endian*** ou **petit-boutistes** ou **mot de poids faible en tête**. Par exemple, les processeurs x86 ont une architecture petit-boutiste. Celle-ci, au prix d'une moindre lisibilité du code machine par le programmeur, simplifiait la circuiterie de décodage d'adresses courtes et longues en 1975 [réf. souhaitée], quand un 8086 avait 29 000 transistors. Elle est d'influence pratiquement nulle en 2016, aussi bien en temps d'exécution (masqué) que d'encombrement (un Haswell typique comporte 1,4 *milliard* de transistors).

	0	1	2	3	
...	08	07	B7	A0	...

Pour une structure de mémoire ou un protocole de communication fondé sur une unité atomique de 2 octets, avec un incrément d'adresse de 1 octet, l'enregistrement dans des octets sera `0708 A0B7`. L'unité atomique de poids le plus faible (ici `0708`) est enregistré à l'adresse mémoire la plus petite.

	0	1	2	3	
...	07	08	A0	B7	...

BI-ENDIAN

Certaines architectures supportent les deux règles, par exemple les architectures PowerPC (IBM), ARM, DEC Alpha, MIPS, PA-RISC (HP) et IA-64 (Intel). On les appelle ***bytesexual*** (jargon), ***bi-endian*** ou, plus rarement, **biboutistes**. Le choix du mode peut se faire au niveau logiciel, au niveau matériel ou aux deux.

Les nombres binaires pour s'entrainer

Nom :	**Note : / 20**
	Classe :

Question A.1
Comment s'écrit en base 16 (en hexadécimal) le nombre dont l'écriture binaire est `0010 1100` ?

Réponses
A 1D
B 2C
C 3C
D 3E

Question A.3
Dans quel système de numération 3F5 représente-t-il un nombre entier ?

Réponses
A binaire (base 2)
B octal (base 8)
C décimal (base 10)
D hexadécimal (base 16)

Question A.4
Quelle est la représentation binaire de l'entier positif 51 sur 8 bits ?

Réponses
A `0010 0001`
B `0010 1001`
C `0011 0001`
D `0011 0011`

Question A.5
On considère les nombres dont l'écriture en base 16 (en hexadécimal) sont de la forme suivante : un 1 suivi de 0 en nombre quelconque, comme 1, 10, 100, 1000 etc.
Tous ces nombres sont exactement :
Réponses
A les puissances de 2
B les puissances de 8
C les puissances de 10
D les puissances de 16

Question A.1
Quelle est l'écriture décimale de l'entier dont la représentation en binaire non signé est `0001 0101` ?

Réponses
A 15
B 21
C 111
D 420

Question A.4
Combien de bits faut-il au minimum pour coder le nombre décimal 4085 ?

Réponses
A 4
B 12
C 2042
D 2043

Question A.6
Quelle est la plage des valeurs entières (positifs ou négatifs) que l'on peut coder sur un octet (8 bits) en complément à 2 ?

Réponses
A −127 à 128
B −128 à 127
C −255 à 128
D −256 à 127

Question A.1
Quelle est la représentation binaire, en complément à 2 sur 8 bits, de l'entier négatif –25 ?

Réponses
A 0001 1001
B 1001 1001
C 1110 0110
D 1110 0111

Question A.3
En ajoutant trois chiffres 0 à droite de l'écriture binaire d'un entier N strictement positif, on obtient l'écriture binaire de :
Réponses
A $6 \times N$
B $8 \times N$
C $1000 \times N$
D aucune des réponses précédentes

Question A.5
Voici les écritures binaires de quatre nombres entiers positifs.
Lequel est pair ?
Réponses
A 10 0001
B 10 0010
C 11 0001
D 11 1111

Question A.1
Parmi les quatre propositions, quelle est celle qui correspond au résultat de la soustraction en écriture binaire 1010 1101 − 101 1000 ?
Réponses
A 101 0101
B 110 0001
C 100 1111
D 1 1000 0111