IESN **Annéeacadémique 2019-2020**

NUMERATION

Première TI

L. BAGHDADI - I. JACQUEMAIN

Table des matières

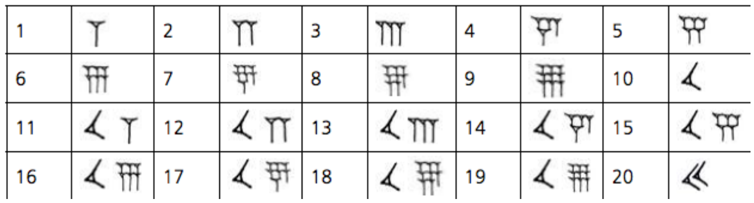
1 SYSTEMES DE NUMERATION

1.1 Introduction

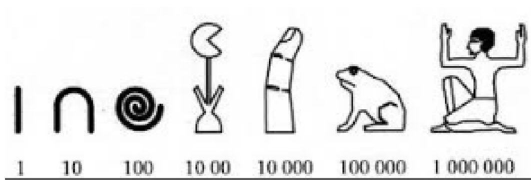
Un système de numération est un système qui permet de représenter les nombres à l'aide de symboles appelés chiffres.

Les hommes ont réalisé une longue démarche scientifique pour arriver au système de numération actuel : Voici quelques exemples de systèmes :

1. **Le système babylonien**, qui utilise la répétition de séquences de symboles en forme de clous pour représenter les chiffres.



2. **le système égyptien** : qui utilise des spectrogrammes pour représenter chaque puissance de 10.



3. **Chiffres et nombres romains**, sont représentés à l'aide de symboles combinés entre eux. La numération romaine est un système de numération additive.

I	II	III	IV	V	VI	VII	VIII	IX
1	2	3	4	5	6	7	8	9
X	XI	XII	XX	XXX	XL	L	LX	
10	11	12	20	30	40	50	60	

4. **le système horaire** : Un jour = 24 heures ; une heure = 60 minutes ; une minute = 60 secondes.

Horloge de Venise indiquant deux fois les heures de 1 à 12.

1.2 le système décimal

***Le système décimal est un système positionnel de base 10** :*

∘ positionnel c'est à dire que la valeur d'un chiffre dépend de la position qu'il occupe dans le nombre ;

∘ de base 10 c'est à dire qu'il utilise dix symboles pour écrire les nombres. Ces symboles sont 0,1,2,3,4,5,6,7,8,9.

L'écriture de symboles n'est pas suffisante pour définir un système de numération. Il faut définir en plus les lois d'assemblage de ces symboles pour former des nombres et des règles arithmétiques de base.

1) **Nombres entiers** : Soit, par exemple, le nombre entier 538.

$$538 \ = \ 5.100 + 3.10 + 8.1 \ = \ 5.10^2 + 3.10^1 + 8.10^0$$

- Les chiffres sont : 5, 3 et 8
- Chaque chiffre a un poids, la puissance de 10 qui lui correspond dans le nombre : 5 à un poids de 2, 8 à un poids de 0.
- Les poids, dans le système décimal expriment l'unité (10^0), la dizaine (10^1), la centaine (10^2), le millier (10^3), la dizaine de milliers (10^4) etc...
- les poids sont dans l'ordre croissant de la droite vers la gauche.

Poids forts $< - - - - - - - -$ Poids faibles ;

2) **Nombres fractionnaires** : Soit le nombre à virgule 0,527.

$$0,527 \ = \ 0,5 + 0,02 + 0,007 \ = \ 5.10^{-1} + 2.10^{-2} + 7.10^{-3}$$

Les poids sont des puissances négatives de 10.

1.3 le système binaire

Pourquoi ?

- L'algèbre de Boole (1815-1864) exclusivement basée sur les deux symboles 1 et 0.
- Développement de machines à calculer (à circuits intégrés) : le courant passe ou le courant ne passe pas !

***Le système binaire est un système positionnel de base 2**, dont les chiffres sont 0 et 1.*

- C'est le système de numération interne d'un ordinateur.
- Chaque symbole ou chiffre dans ce système est appelé : bit pour **bi**nary digi**t**.

1) **Nombres entiers** : Soit, par exemple, le nombre entier 110101.

Ce nombre est écrit sur 6 bits.

Calculons sa valeur décimale : Pour cela utilisons le fait que c'est un système positionnel, donc chacun des chiffres 0 ou 1 doit être relié à une puissance de 2 (la base). d'où

$$110101 \;=\; 1.2^5 + 1.2^4 + 0.2^3 + 1.2^2 + 0.2^1 + 1.2^0 = \; 32 + 16 + 0 + 4 + 0 + 1 \; = 53$$

Donc

$$(110101)_2 \;=\; 53$$

2) **Nombres fractionnaires** : Soit le nombre à virgule 0,1101.

Calculons sa valeur décimale :

$$0,1101 = 1.2^{-1} + 1.2^{-2} + 02^{-3} + 1.2^{-4} \;=\; \frac{1}{2} + \frac{1}{4} + 0 + \frac{1}{16} = 0,3625$$

Donc

$$(0,1101)_2 \;=\; 0,3625$$

3) **Exercices** : Donner la valeur décimale des nombres binaires suivants :

$$a)\ 10110101 \qquad b)\ 100011 \qquad c)\ 111111$$
$$d)\ 0,111 \qquad e)\ 1101,011 \qquad f)\ 011110,1110$$

4) **Remarques** :

(i) Si le nombre est formé d'une suite de n 1 successifs, alors sa valeur décimale est facile à calculer : $(111......1)_2 \;=\; 2^n - 1$

ex : $11111 = 2^5 - 1 = 31$ en effet $11111 = 16 + 8 + 4 + 2 + 1 = 31$

(ii) La valeur décimale de tout nombre binaire dont le bit le plus faible est un 1 (respectivement, un 0) est un nombre impair (resp. pair).

(iii) Un nombre binaire de 8 bits est appelé un octet, de 16 un demi-mot et de 32 un mot.

(iv) Pour le moment nous ne considérons que les nombres positifs (sans signe), on parle de "*binaire naturel*" ; plus tard nous intégrerons le signe négatif, et nous parlerons de "*binaire signé*"

1.4 Conversions : binaire et décimal

1.4.1 Cas d'un nombre entier

∇ Passer du binaire au décimal.

Il suffit de développer selon les puissances de 2.

Ex : $(10110,101)_2 \ = \ 2^4+2^2+2+2^{-1}+2^{-3} \ = \ 16+4+2+0,5+0,125 \ = \ 22,625_{10}$
D'où

$$(10110,101)_2 \ = \ 22,625_{10}$$

∇ Passer du décimal au binaire .

A) Soit en utilisant les puissances de 2.

Ex : $67 \ = \ 64+3 \ = \ 2^6+2+1 \ = \ (1000011)_2$
Ex : $500 \ = \ 256+128+64+32+16+4 \ = \ (111110100)_2$

2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}
1	2	4	8	16	32	64	128	256	512	1024	2048

B) Soit en effectuant des divisions successives par 2 qu'on arrête lorsque le quotient est nul. L'écriture binaire sera alors formée par les restes de ces divisions.

Ex : $79_{10} = (....?...;)_2$

$79 : 2 \ = \ 39$ et il reste 1
$39 : 2 \ = \ 19$ et il reste 1
$19 : 2 \ = \ 9$ et il reste 1
$9 : 2 \ = \ 4$ et il reste 1
$4 : 2 \ = \ 2$ et il reste 0
$2 : 2 \ = \ 1$ et il reste 0
$1 : 2 \ = \ 0$ et il reste 1 et stop car le quotient est nul.

Pour obtenir l'écriture en binaire on recopie les restes en partant du bas vers le haut.

D'où : $79_{10} = (1001111)_2$

• Autre façon de poser les divisions par 2 :

79	1
39	1
19	1
9	1
4	0
2	0
1	1
0	

D'où : $79_{10} = (1001111)_2$

• Autre exemple : $83_{10} = (....?...;)_2$

83	1
41	1
20	0
10	0
5	1
2	0
1	1
0	

D'où : $83_{10} = (1010011)_2$

1.4.2 Cas d'un nombre fractionnaire

∇ Passer du binaire au décimal. Il suffit de développer selon les puissances de 2.

Ex : $(0,101)_2 \ = \ 2^{-1} + 2^{-3} \ = \ 0,5 + 0,125 \ = \ 0,625_{10}$
D'où

$$(0,101)_2 \ = \ 0,625_{10}$$

∇ Passer du décimal au binaire .

Il faut faire une suite de multiplications successives par 2, et l'écriture binaire sera constituée des parties entières des réponses. Il se peut que cette suite de multiplications soit infinie (sans arrêt), dans ce cas le nombre de bits disponibles fixera l'arrêt de la suite de multiplication.

Ex : $0,6875_{10} = (....?...)_2$

$$
\begin{array}{llll}
0,6875 * 2 & = & 1,375 & \text{on garde le } 1 \\
0,375 * 2 & = & 0,75 & \text{on garde le } 0 \\
0,75 * 2 & = & 1,5 & \text{on garde le } 1 \\
0,5 * 2 & = & 1,0 & \text{on garde le } 1 \text{ et stop car la partie à droite de la virgule du résultat} \\
 & & & \qquad \text{de cette multiplication est nul.}
\end{array}
$$

Pour obtenir l'écriture en binaire on recopie les 1 et 0 qu'on a gardé, en partant du haut vers le bas.

D'où : $0,6875_{10} = (0.1011)_2$

• Autre façon de poser les multiplications par 2 :

$$
\begin{array}{c|c|l}
0{,}6875 & 1{,}375 & 1 \\
0{,}375 & 0{,}75 & 0 \\
0{,}75 & 1{,}5 & 1 \\
0{,}5 & 1{,}0 & 1 \\
\text{stop} & &
\end{array}
\qquad \text{D'où : } 0,6875_{10} = (0.1011)_2
$$

• Autre exemple, cyclique : $0,2_{10} = (....?...;)_2$

$$
\begin{array}{c|c|l}
0{,}2 & 0{,}4 & 0 \\
0{,}4 & 0{,}8 & 0 \\
0{,}8 & 1{,}6 & 1 \\
0{,}6 & 1{,}2 & 1 \\
0{,}4 & 0{,}8 & 0 \\
0{,}8 & 1{,}6 & 1 \\
0{,}6 & 1{,}2 & 1 \\
\text{stop} & &
\end{array}
\qquad \text{D'où : } 0,2_{10} = (0.00110011.....)_2 = 0,\overline{0011}
$$

Remarque : Plus le nombre de bits disponibles est grand, plus la conversion est précise. Prenons, par exemple, la conversion du nombre décimal 0,2 sur un nombre différent de bits :

$0,2 \ = (0.0011)_2 \ = \ 2^{-3} + 2^{-3} \ = \ 0,1875$
$0,2 \ = (0.00110011)_2 \ = \ 2^{-3} + 2^{-3} + 2^{-7} + 2^{-8} \ = \ 0,19921875$
$0,2 \ = (0.001100110011)_2 \ = \ 2^{-3} + 2^{-3} + 2^{-7} + 2^{-8} + 2^{-11} + 2^{-12} \ = \ 0,1999511719$

- Encore un autre exemple, cette fois infini : $0,71_{10} = (....?...;)_2$

0,71	1,42	1
0,42	0,84	0
0,84	1,68	1
0,68	1,36	1
0,36	0,72	0
0,72	1,44	1
0,44	0,88	0
...		

D'où : $0,71_{10} = (0.1011010.....)_2$

1.4.3 Cas d'un nombre réel positif

Un nombre réel peut toujours se décomposer en la somme d'un nombre entier et d'un nombre fractionnaire. il suffit donc de scinder le nombre en deux et d'appliquer les deux méthodes précédentes.

Ex : $83,6875_{10} = 83 + 0,6875$
Nous avons déjà : $83_{10} = (1010011)_2$ et $0,6875_{10} = (0.1011)_2$

D'où : $83,6875_{10} = (1010011.1011)_2$

1.5 Précisions et erreurs

En général, la conversion d'une base humaine (décimal) vers une base machine (binaire) ne donne que des valeurs approchées de la valeur réelle.

Nous verrons plus tard quelques solutions : soit augmenter le nombre de bits, soit utiliser l'écriture en virgule flottante simple, double, voire quadruple précision.

Dans la majorité des cas, et en particulier pour les calculs scientifiques , il n'est pas toujours nécessaire d'avoir une précision absolue, cependant il est important de connaître l'erreur commise lors de la représentation d'une valeur décimale en binaire.

On définit deux types d'erreurs :

∘ **L'erreur absolue** : c'est la différence qu'il y a entre la valeur réelle (celle que l'on devrait avoir représentée dans la machine) et la valeur effectivement représentée par les poids binaires dans la machine ;

$$\text{Erreur absolue} \;=\; \Delta V \;=\; |\text{val réelle} - \text{val machine}|$$

Ex : Soit $n = 0,1_{10}$ valeur réelle.

Après conversion en binaire : $n = 0,1_{10} = (0.000110011)_2$; sa valeur machine est donc de $0,099609375$. d'où

$\Delta V \;=\; 0,1 - 0,099609375 = 3,90625.10^{-4}$ de l'ordre du dix-millième !

∘ **L'erreur relative** : c'est le rapport entre l'erreur absolue et la valeur réelle (exprimée en %).

$$\text{Erreur relative} \;=\; \frac{\Delta V}{V} \;=\; \frac{|\text{val réelle} - \text{val machine}|}{\text{val réelle}}$$

Ex : Si $n = 0,1_{10}$, alors $\frac{\Delta V}{V} \;=\; 0,39\%$

Exercice : Quelles sont l'erreur absolue et l'erreur relative commises sur la conversion en binaire sur 8 bits du nombre décimal 3,14 ?

Rép : $3,14_{10} \;=\; (11,001000)_2$.

D'où $\Delta V \;=\; 3,14 - 3,125 = 0,015$ et $\frac{\Delta V}{V} \;\simeq\; 0,48\%$

1.6 Echelle des nombres entiers en binaire naturel

$\longrightarrow$ Le plus grand nombre que l'on peut écrire, en binaire naturel, sur n bits a pour valeur décimale : $2^n - 1$

Sur 2 bits : $11 \ = \ 3 \ = \ 2^2 - 1$
Sur 3 bits : $111 \ = \ 7 \ = \ 2^3 - 1$
Sur 4 bits : $1111 \ = \ 15 \ = \ 2^4 - 1$
etc...
Sur 10 bits : $1111111111 \ = \ 2^{10} - 1 = 1023$.

$\longrightarrow$ Le plus petit nombre que l'on peut écrire, en binaire naturel, sur n bits a pour valeur décimale : 0

Exercice1 : Combien faut-il de bits pour écrire les nombres suivants :

$a)$ 9 $b)$ 32 $c)$ 49 $d)$ 1200 $e)$ 53.10^5 $f)$ $6,022136.10^{23}$ (Nbre d'Avogadro)

Rép : a) 4 ; b) 6 ; c) 6 ; d) 11 e) 23 f) 79.

Exercice2 : Combien de nombres peut-on écrire avec 2, 3, 10, n bits ?

Rép : 4, 8, 1024 et 2^n nombres possibles.

1.7 Système Hexadécimal

Le système hexadécimal est un système positionnel de base 16*, dont les chiffres sont 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.*

Avec $A_{16} = 10_{10}$; $B_{16} = 11_{10}$; $C_{16} = 12_{10}$; $D_{16} = 13_{10}$; $E_{16} = 14_{10}$; $F_{16} = 15_{10}$;

Le système hexadécimal est particulièrement commode et permet un compromis entre le code binaire des machines et une base de numération pratique à utiliser pour l' humain. En effet, chaque chiffre hexadécimal correspond exactement à quatre chiffres binaires (ou bits), rendant les conversions très simples et fournissant une écriture plus compacte.

1.8 Conversions : hexadécimal, décimal et binaire

1.8.1 Convertir un nombre décimal en hexadécimal

Ex : Soit $542,90625_{10}$, il faut le convertir en hexadécimal ; Pour cela, il faut d'abord le décomposer en une partie entière 542 et une partie fractionnaire $0,90625$.

Pour la partie entière il faut effectuer des divisions successives par 16 et garder à chaque fois le reste. L'écriture en binaire sera formée des restes pris du bas vers le haut.

542	14
33	1
2	2
0	
stop	

D'où : $542_{10} = (21E)_{16}$

Pour la partie fractionnaire, il faut effectuer des multiplications successives par 16 et garder à chaque fois la partie entière de la réponse. L'écriture en binaire sera formée par les parties entières prises du haut vers le bas.

0,90625	14,5	14
0,5	8,0	8
stop		

D'où : $0,90625_{10} = (0,E8)_{16}$

et enfin, $542,90625_{10} = (21E,E8)_{16}$

Exercice : Ecrire le nombre décimal 1034,9648 en hexadécimal sur 24 bits.

Rép : $40A,F6F$

1.8.2 Convertir un nombre hexadécimal en décimal

Ex : Soit le nombre hexadécimal 1A0D,3B. Qu'elle est sa valeur décimale ?
Il suffit de développer selon les puissances de 16 (la base) :

$$\begin{aligned}1A0D,3B &= 1.16^3 + 10.16^2 + 0.16 + 13.16^0 + 3.16^{-1} + 11.16^{-2} \\ &= 16^3 + 10.16^2 + 13 + \frac{1}{16} + \frac{1}{16^2} = \\ &= 6669,230469\end{aligned}$$

Donc : $1A0D,3B_{16} \;=\; 6669,230469_{10}$

Exercice : Ecrire le nombre hexadécimal 8AD5,0F01 en décimal.

Rép : $35541,05861_{10}$

1.8.3 Convertir un nombre binaire en hexadécimal

Ex : soit le nombre binaire 1110101001. Quelle est sa valeur en hexadécimal ?

Pour cela il faut observer que pour écrire les chiffres hexadécimaux en binaire, il faut avoir 4 bits :

Système décimal	Système binaire	Système hexadécimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

D'où, pour convertir un nombre binaire en hexadécimal, il faudra le "découper" en tranches de 4 bits successifs, des poids faibles vers les poids forts puis de lire en hexadécimal chaque groupe de 4 bits.

$$\begin{aligned} 1110101001,111011_2 &= 0011 \;\; 1010 \;\; 1001 \;,\; 1110 \;\; 1100 = 3 \;\; A \;\; 9 \;,\; E \;\; C \\ &= 3A9,EC_{16} \end{aligned}$$

1.8.4 Convertir un nombre hexadécimal en binaire

il faut développer chaque symbole en binaire sur quatre bits.

Ex : $8AD5,7_{16} \;=\; 1000\;1010\;1101\;0101\;,\;0111 \;=\; 100010101101,0111_2$

2 APPLICATION AUX ADRESSES IP

Ref : http ://www.commentcamarche.net/contents/523-adresse-ip

2.1 Qu'est-ce-qu'une adresse IP ?

Sur internet, les ordinateurs communiquent entre eux grâce au **protocol IP**(Internet Protocol), qui utilise des adresses numériques composées de 4 nbres entiers entre 0 et 255. En binaire, l'écriture se fait sur 4 octets.

Elles sont notées sous forme : $* * * . * * * . * * * . * * *$

ex : 194 . 153 . 205 . 26

Chaque ordinateur d'un réseau possède une adresse IP unique sur ce réseau.

C'est l'**ICANN** (Internet Corporation for Assigned Names and Numbers), depuis 1998, qui est chargée d'attribuer des adresses IP publiques, càd les adresses IP des ordinateurs directement connectés sur le réseau publique internet.

2.2 Déchiffrement d'une adresse IP :

Une adresse IP est une adresse 32 bits, généralement notée sous forme de 4 entiers séparés par des points. On distingue deux parties dans l'adr. IP :

$\mapsto$ La partie de gauche désigne le réseau
(appelée IDréseau, adresse réseau, (net-ID))

$\mapsto$ La partie de droite désigne les ordinateurs de ce réseau
(appelée IDd'hôte, adresse client, (host-ID))

Rq : Plus le nbre de bits réservés au réseau est petit, plus celui-ci peut contenir d'ordinateurs.

ex : réseau 58.0.0.0 $\Rightarrow$
les ordi de ce réseau peuvent avoir des adr.IP allant de 58.0.0.1 à 58.255.255.254.
Il pourra contenir $256^3 - 2 = 16777214$ ordi.
Par contre, le réseau 194.26.0.0, pourra contenir $256^2 - 2 = 65534$ ordi.

2.3 Adresses particulières :

◇ `adresse réseau` : Tous les bits machines -clients- (à droite) sont à 0.
ex : 50.0.0.0 ou 194.26.0.0

◇ `adresse machine` (adr. clients) : Tous les bits réseau (à gauche) sont à 0.
ex : 0.0.0.1 ou 0.0.94.210

◇ `adresse de diffusion` (broadcast) : Tous les bits clients (à droite) sont à 1.
ex : 50.255.255.255.

◇ `adresse de diffusion limitée` (multicast) : Les bits réseau (à gauche) sont à 1.
ex : 255.0.0.0.

◇ `adresse de rebouclage` (loopback) : 127.0.0.1 $\longmapsto$ la machine locale.

2.4 Les classes d'adresses :

A. *Adresse IP de classe A* :

Δ Le 1er octet représente le réseau et le bit de poids le plus fort est à 0.
Δ rés. disponibles en cl.A : de 0.0.0.0 à 126.255.255.255 (2^7 possibilités réseaux)
Δ Les 3 octets de droite $\rightarrow$ ordi. ($2^{24} - 2 = 16777214$ ordi possibles)

0 – – – – – – – . – – – – – – – – –. – – – – – – – – –. – – – – – – – – –

Rem : Les adresses IP commençant par 127 sont utilisées pour faire des tests particuliers. L'adresse réseau 127.0.0.0 est réservée pour les communications en boucle locale.

B. *Adresse IP de classe B* :

Δ Les deux 1ers octets représentent le réseau et les deux 1ers bits sont 10.
Δ rés. disponibles en cl.B : de 128.0.0.0 à 191.255.255.255 (2^{14} possibilités rés.)
Δ Les 2 octets de droite $\rightarrow$ ordi. $2^{16} - 2 = 65534$ ordi possibles)

10 – – – – – –. – – – – – – – – –. – – – – – – – – –. – – – – – – – – –

C. *Adresse IP de classe C* :

Δ Les trois 1ers octets représentent le réseau et les trois 1ers bits sont 110.
Δ Les réseaux disponibles en classe C vont de 192.0.0.0 à 223.255.255.0 (2^{21} possibilités réseaux)
Δ L'octet de droite $\rightarrow$ ordi. ($2^8 - 2 = 254$ ordi possibles)

110 − − − − − . − − − − − − − − −. − − − − − − − − −. − − − − − − − − −

C. *But de la subdivision en classes* :

Le but de la division des adresses IP en trois classes A,B et C est de faciliter la recherche d'un ordinateur sur le réseau. En effet avec cette notation il est possible de rechercher dans un premier temps le réseau que l'on désire atteindre puis de chercher un ordinateur sur celui-ci. Ainsi, l'attribution des adresses IP se fait selon la taille du réseau.

classes	Nbre réseaux poss.	Nbre max ; d'ord.
A	126	16 777 214
B	16384	65 534
C	2 097 152	254

5. Adresses IP réservées :

Il arrive souvent dans une entreprise qu'un seul ordi soit relié à internet. C'est par son intermédiaire que les autres ordi ; du réseau accèdent à internet (proxy ou passerelle)
Dans ce cas, seul cet ordi a besoin de reserver une ad.IP de l'ICANN. Mais les autres ordi. ont tout de même besoin d'une ad.IP pour pouvoir communiquer en interne.
Ainsi, l'ICANN a réservé une poignée d'adresses dans chaque classe pour permettre d'affecter une ad.IP aux ordi. d'un réseau local rélié à internet sans risquer de créer des conflits d'ad.IP dans le réseau.

∇ Ad.IP privées de cl. A : de 10.0.0.1 à 10.255.255.254 (vastes réseaux privés comprenant des milliers d'ordi.)

∇ Ad.IP privées de cl. B : de 172.16.0.1 à 172.31.255.254 (réseaux privés de taille moyenne.)

∇ Ad.IP privées de cl. C : de 192.168.0.1 à 192.168.0.254 (petits réseaux privés.)

2.5 Masques :

2.5.1 Masque réseau :

Un masque réseau (en anglais netmask) se présente sous la forme de 4 octets séparés par des points (comme une adresse IP), il comprend (dans sa notation binaire) des zéros aux niveau des bits de l'adresse IP que l'on veut annuler et des 1 au niveau de ceux que l'on désire conserver. On peut dire que c'est un séparateur entre la partie réseau et la partie machine dans une adresse IP.

Le premier intérêt d'un masque de sous-réseau est de permettre d'identifier simplement le réseau associé à une adresse IP. Par exemple, le réseau associé à l'adresse 34.56.123.12 est 34.0.0.0, car il s'agit d'une adresse IP de classe A.

Masques correspondant à chaque classe d'adresse :

∇ Pour une adresse de Classe A, seul le premier octet doit être conservé. Le masque possède la forme suivante 11111111.00000000.00000000.00000000, c'est-à-dire 255.0.0.0 en notation décimale ;

∇ Pour une adresse de Classe B, les deux premiers octets doivent être conservé, ce qui donne le masque suivant 11111111.11111111.00000000.00000000, correspondant à 255.255.0.0 en notation décimale ;

∇ Pour une adresse de Classe C, avec le même raisonnement, le masque possédera la forme suivante 11111111.11111111.11111111.00000000, c'est-à-dire 255.255.255.0 en notation décimale

2.5.2 Masque sous réseau :

On peut vouloir subdiviser un réseau donné en plusieurs sous réseaux. On dédie une partie de la zone client pour noter les sous réseaux. Le nombre de sous-réseaux dépend du nombre de bits attribués en plus au sous-réseau.

En classeA :

8	$24-x$	x
Réseau	Sous réseau	Clients

En classeB :

16	$16-x$	x
Réseau	Sous réseau	Cls

En classeC :

24	$8-x$	x
Réseau	S-réseau	Cls

∇ *Comment reconnaitre une adresse masque ?*

Un masque est constitué de quatre octets :

- Dans son écriture binaire, les 1 se suivent et les 0 se suivent.

Ex1 : 11111111.11111111.00000000.00000000 est une adresse masque réseau (Cl B).
Ex2 : 11111111.11111111.11110000.00000000 est un masque de sous réseau (Cl B).

- Dans son écriture en décimal, on ne peut avoir que l'un des 8 nombres suivants : 255 ; 254 ; 252 ; 248 ; 240 ; 224 ; 192 ;128 et 0.

Ex1 : 255. 255.0.0 est un masque réseau (Cl B).
Ex2 : 255.255.240.0 est un masque de sous réseau (Cl B).
Ex3 : 255.253.240.0 n'est pas une adresse masque.

Exemple : Soit l'adresse IP 192 . 168 . 25 . 36.
On voit bien que c'est une adresse de classe C, donc les trois premiers octets sont réservés au réseau.
Comment savoir s'il-y-a une subdivision en sous réseaux ?
Il faut avoir le masque.
- Soit par ex le masque : 255.255.255.224. Où se trouve la subdivision ?

	192	168	25	36
AdIP	11000000 .	10101000 .	00011001 .	001|00100
M	11111111 .	11111111 .	11111111 .	111|00000
	255	255	255	224
	Réseau			*SR* | *Cls*

- Quelle est l'adresse réseau ? S'obtient, à partir de l'Ad IP, en mettant des 0 dans la partie clients (et sous réseau) :

Rép : 192.168.25.0

- Quelle est l'adresse sous réseau ? S'obtient, à partir de l'Ad IP, en mettant des 0 dans la partie clients seulement.

Rép : 192.168.25.32

3 ARITHMETIQUE DE l'ORDINATEUR

3.1 Arithmétique en binaire naturel

3.1.1 Addition

Dans tout système de numération positionnel, les symboles sont utilisés de façon cyclique et la longueur d'un cycle est égale à la base. Pour compter, dès qu'on a fini un cycle, on en rajoute un à gauche et on continue.

Dans le système décimal :

$$\text{Pour compter : cycle 1} \left\{ \begin{array}{c} 1 \\ 2 \\ 3 \\ \ldots \\ 9 \end{array} \right. \qquad \text{cycle 2} \left\{ \begin{array}{l} 10 \\ 11 \\ 12 \\ \ldots \\ 99 \end{array} \right. \qquad \text{cycle 3} \left\{ \begin{array}{l} 100 \\ 101 \\ 102 \\ \ldots \\ 999 \end{array} \right.$$

Pour additionner :

+	0	1	2	...
0	0	1	2	
1	1	2	3	
2	2	3	4	
...				

On fait de même en binaire, sauf que nous avons moins de symboles à utiliser. les cycles sont beaucoup plus courts :

$$\text{Pour compter : cycle 1} \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right. \qquad \text{cycle 2} \left\{ \begin{array}{l} 10 \\ 11 \end{array} \right. \qquad \text{cycle 3} \left\{ \begin{array}{l} 100 \\ 101 \\ 110 \\ 111 \end{array} \right.$$

Pour additionner :

+	0	1
0	0	1
1	1	10

Exemples d'additions :

$$
\begin{array}{ccccc}
 & \boldsymbol{1} & \boldsymbol{1} & & \\
 & & 1 & 1 & 0 \\
+ & & 1 & 1 & 1 \\
\hline
= & 1 & 1 & 0 & 1
\end{array}
\qquad
\begin{array}{ccccccc}
 & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & \boldsymbol{1} & & \\
 & & 0 & 1 & 1 & 1 & 0 \\
+ & & 1 & 0 & 1 & 1 & 1 \\
\hline
= & 1 & 0 & 0 & 1 & 0 & 1
\end{array}
$$

Exercice : Effectuer l'opération suivante en binaire naturel, puis vérifier en décimal :

$$a)\ 1101011 + 1010110 \qquad b)\ 01001110 + 00110101$$

Rép : a) 11000001 b) 10000011

3.1.2 Soustraction

La soustraction peut se faire en respectant les règles suivantes :

−	0	1
0	0	(∗)
1	1	0

pour (*), on a 0 - 1 a effectuer. pour cela il faut faire un emprunt de 1 à la colonne juste à gauche pour faire 10 - 1 = 1.

Deux cas peuvent se présenter :
- Il y a un 1 dans la colonne immédiatement à gauche et le calcul peut être effectué ;
- il n' y a pas de 1 dans la les colonne immédiatement à gauche, il faut remonter plus loin, sinon pas moyen d'effectuer l'opération,

Exemples :

$$
\begin{array}{cccc}
 & & \boldsymbol{1} & \\
 & 0 & \boldsymbol{0} & \boldsymbol{1} \\
 & \cancel{1} & \cancel{1} & 0 \\
- & 0 & 1 & 1 \\
\hline
= & 0 & 1 & 1
\end{array}
\qquad
\begin{array}{ccccc}
 & & \boldsymbol{1} & \boldsymbol{1} & \\
 & & & \boldsymbol{0} & \boldsymbol{1} \\
 & \cancel{1} & 0 & \cancel{1} & 0 \\
- & 0 & 1 & 1 & 1 \\
\hline
= & 0 & 0 & 1 & 1
\end{array}
$$

Techniquement, l'addition et la soustraction en binaire, dans la machine, sont effectuées par un même circuit qui ne fait qu'additionner. La soustraction est effectuée par complémentation :

$$A - B \ = \ A + (-B) \ = \ A + Compl(B)$$

il y a deux types de compléments que la machine peut effectuer :

3.1.3 le complément à 1

Le complément à 1 d'un nombre binaire s'obtient en le soustrayant à un nombre comportant le même nombre de bits tous à 1 :

Ex : Pour déterminer le complément à 1 (noté C1) du nombre 11010010, il faut le soustraire à 11111111.

$$\begin{array}{cccccccc c} & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ - & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ \hline = & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{array}$$

et donc, le complément à 1 (C1) du nombre 11010010 est le nombre 00101101.

En pratique, on utilise un inverseur qui change l'état de chaque bit.

le C1 de 10110010 est 01001101. On écrit : $\begin{array}{ll} 10110010 & \\ 01001101 & C1 \end{array}$

• <u>Soustraction par complément à 1</u> :

En binaire C1 : $A - B = A + C1(B) + \mathbf{1}$ A et B ont le même nbre de bits.

Ex : $(101101)_2 - (11001)_2 \;=\; (101101)_2 - (011001)_2 \;=\; 101101 + 100110$

Posons l'opération pour l'effectuer :

$$\begin{array}{cccccccc} & & 1 & 0 & 1 & 1 & 0 & 1 \\ + & & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline = & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ + & & & & & & & \mathbf{1} \\ \hline = & \cancel{1} & 0 & 1 & 0 & 1 & 0 & 0 \end{array}$$

D'où : $(101101)_2 - (11001)_2 \;=\; 0101011$. Vérifions en décimal : $45 - 25 = 20$

Remarque : Pour le moment nous ne pouvons faire que des soustractions d'un nombre grand moins un nombre plus petit, car nous ne savons pas encore écrire un nombre négatif.

3.1.4 le complément à 2

Le complément à 2 (noté C2) d'un nombre s'obtient en calculant le complément à 1 et en lui additionnant 1.

Ex : C2(10110110) = C1(10110110) + 1 = 01001001 + 00000001 = 01001010

En pratique, pour obtenir le complément à 2 d'un nombre binaire, on laisse inchangés tous les bits en commençant par la droite jusqu'au premier 1 inclus et on change tous les autres bits qui suivent.

Ex : $\begin{array}{llcll} 1101101 & & \qquad & 10110100 & \\ 0010011 & C2 & \qquad & 01001100 & C2 \end{array}$

• Soustraction par complément à 2 :

En binaire C2 : $A - B = A + C2(B)$ A et B ont le même nbre de bits.

Ex : $(1110110)_2 - (1011010)_2 \ = \ (1110110)_2 + (0100110)_2$
Posons l'opération pour l'effectuer :

$$\begin{array}{cccccccccc} & \boldsymbol{1} & & & \boldsymbol{1} & \boldsymbol{1} & & & \\ & & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ + & & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ \hline = & \cancel{1} & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{array}$$

D'où : $(1110110)_2 - (1011010)_2 \ = \ (0011100)_2$. Vérifions en décimal : $118 - 90 = 28$

Exercice : Effectuer l'opération suivante $215_{10} - 108_{10}$ en binaire C2, puis vérifier en décimal :

Rép : $(11010111)_2 - (01101100)_2 \ = \ (01101011)_2$. En décimal : $215 - 108 = 107$

3.2 Arithmétique en binaire normalisé : Nombres signés

Dans l'ordinateur, l'unité d'information est le bit, qui peut prendre la valeur 0 ou 1. Un seul bit ne peut contenir beaucoup d'information, c'est pourquoi on regroupe les bits en mots. Ceux ci peuvent avoir différentes longueurs dont les plus courants sont de 8 bits (octet ou byte), de16 bits (mot), de 32 bits (double mot) et de 64 bits (quadruple mot). rappelons qu'avec n bits nous pouvons écrire 2^n nombres de 0 à $2^n - 1$.

la représentation des nombres entiers et réels, se fait en tenant compte de la longueur des mots dans la mémoire de la machine (dépend du fabricant !)

3.3 Représentation normalisée des nombres réels

Dans une mémoire d'ordinateur, les nombres ne sont pas représentés en binaire naturel, mais en binaire normalisé, c'est à dire que tous les nombres sont écrits en utilisant le même nombre de bits et le bit tout à fait à gauche est réservé au signe (bit signe).

Si le nombre est positif, le bit signe est à 0

Si le nombre est négatif, le bit signe est à 1.

Pour une écriture sur un octet (8 bits) on a :

$$signe- > \lfloor\;\lfloor\qquad\qquad\qquad\rfloor < -nombre$$

Il y a trois façons différentes de représenter les nombres signés :
1- En signe-grandeur (SG), la plus naturelle (analogue à ce qu'on fait en décimal) ;
2- En complément à 1 (C1), facile puisqu'il suffit d'inverser les bits ;
3- En complément à 2 (C2), qui permet d'effectuer facilement les soustractions.

Dans la suite, nous effectuerons les écritures des nombres signés sur un octet. Le travail est identique sur un nombre quelconque de bits.

3.4 Représentation en Signe-Grandeur (SG)

Dans cette représentation, dès que le nombre de bits à utiliser est fixé (normalisé), le bit de poids le plus fort représente le bit signe (0 -positif ; 1-négatif) et le reste représentera la valeur absolue du nombre, exactement comme en décimal, si n est un nombre réel, alors

$$n \;=\; \pm|n|$$

nbre décimal	bin. naturel	bin. normalisé SG
+5	$(101)_2$	$(00000101)_2$
-5	$-(101)_2$	$(10000101)_2$

• Pour déterminer la valeur décimale d'un nombre écrit en binaire normalisé SG :

1. si le nombre est positif (bit signe à 0), il suffit de le développer selon les puissances de 2 et calculer sa valeur.
2. si le nombre est négatif (bit signe à 1), il faut revenir au nombre positif (changer le bit signe), et faire de même qu'en 1.

   Bien sûre, on peut aussi juste lire la valeur absolue du nombre (càd les 7 bits de droite).

Ex1 : $(01110010)_2$ est un nombre positif, sa valeur décimale est : $64+32+16+2 = 114$ donc $(01110010)_2 \;=\; +114$

Ex2 : $(11010111)_2$ est un nombre négatif, le nombre positif qui lui correspond est $(01010111)_2$ et sa valeur décimale est : $64+16+4+2+1 = 87$
donc $(11010111)_2 \;=\; -87$

Remarques :

(i) Le plus grand nombre (positif) que l'on peut écrire, sur un octet, en notation binaire normalisée SG est : $(01111111)_2 \; = \; +127 \; = \; 2^7 - 1$

(ii) Le plus petit nombre (négatif) que l'on peut écrire, sur un octet, en notation binaire normalisée SG est : $(11111111)_2 \; = \; -127 \; = \; -2^7 + 1$

(iii) Dans une notation SG nous obtenons deux écritures pour le zéro (+0 et -0) :

$$0 \; = \; (00000000)_2 \; = \; (10000000)_2$$

(iv) En SG nous pouvons écrire $2^8 - 1 = 255$ nombres différents.

3.4.1 Addition SG : Deux nombres de même signe

Il faut poser l'opération, effectuer l'addition comme en binaire naturel, puis détecter les dépassements (report en provenance du bit de plus grande puissance) et enfin assigner au résultat le signe commun aux deux nombres.

Ex : Effectuer les additions suivantes en binaire normalisé SG.

$$a)\ (51)_{10} + (74)_{10} \qquad b)\ (-63)_{10} + (-42)_{10} \qquad c)\ (115)_{10} + (54)_{10}$$

Les nombres sont donnés en décimal, donc il faut d'abord les convertir en binaire naturel puis les écrire en binaire normalisé SG, puis poser et effectuer l'addition et enfin donner la réponse (ne pas oublier de la vérifier en décimal !).

3.4.2 Addition SG : Deux nombres de signes contraires

Même consignes que pour le cas précédent.

Ex : Effectuer l'addition suivante en binaire normalisé SG : $(74)_{10} + (-51)_{10}$

3.5 Représentation en Complément à 1 (C1)

Le bit le plus à gauche représente toujours le bit signe, mais les nombres entiers négatifs sont représentés par leurs compléments à 1. La représentation des nombres par C1 est intéressante car un circuit pour complémenter à 1 est facile à construire.

nbre décimal	bin. naturel	bin. normalisé C1
+5	$(101)_2$	$(00000101)_2$
-5	$-(101)_2$	$(11111010)_2$

• Pour déterminer la valeur décimale d'un nombre écrit en binaire normalisé C1 :

1. si le nombre est positif (bit signe à 0), il suffit de le développer selon les puissances de 2 et calculer sa valeur.
2. si le nombre est négatif (bit signe à 1), deux possibilités sont à notre disposition :
   - soit revenir au nombre positif (en passant au complément à 1), et faire de même qu'en 1.
   - soit développer directement selon les puissances de 2 mais en faisant attention à associer au bit signe un signe moins et à ajouter 1 à la fin.

Ex1 : $(01110010)_2$ est un nombre positif, sa valeur décimale est : $64+32+16+2 = 114$ donc $(01110010)_2 = +114$

Ex2 : $(11010111)_2$ est un nombre négatif, le nombre positif qui lui correspond est $(00101000)_2$ et sa valeur décimale est : $32 + 8 = 40$
donc $(11010111)_2 = -40$

On peut aussi développer selon les puissances de 2, sans oublier le signe moins pour le bit signe et le +1 à la fin :
$(11010111)_2 = \mathbf{-128} + 64 + 16 + 4 + 2 + 1\mathbf{+1} = -40$

Attention, c'est le même nombre que nous avons pris comme exemple en SG, mais il n'a plus la même valeur décimale maintenant qu'il est pris en C1.

Remarques :

(i) Le plus grand nombre (positif) que l'on peut écrire, sur un octet, en notation binaire normalisée C1 est : $(01111111)_2 = +127$

(ii) Le plus petit nombre (négatif) que l'on peut écrire, sur un octet, en notation binaire normalisée C1 est : $(10000000)_2 = -127$

(iii) Dans une notation C1 nous obtenons deux écritures pour le zéro (+0 et -0) :

$$0 = (00000000)_2 = (11111111)_2$$

(iv) En C1 nous pouvons écrire $2^8 - 1 = 255$ nombres différents.

3.5.1 Addition C1 : Deux nombres de même signe

Ex : Effectuer les additions suivantes en binaire normalisé C1.

$a)\ (27)_{10} + (37)_{10} \quad b)\ (59)_{10} + (119)_{10} \quad c)\ (-27)_{10} + (-10)_{10} \quad d)\ (-17)_{10} + (-12)_{10}$

Les nombres sont donnés en décimal, donc il faut d'abord les convertir en binaire naturel puis les écrire en binaire normalisé C1, puis poser et effectuer l'addition et enfin donner la réponse (ne pas oublier de la vérifier en décimal !).

3.5.2 Addition C1 : Deux nombres de signes contraires

Même consignes que pour le cas précédent.
Ex : Effectuer les additions suivantes en binaire normalisé C1 :

$a)\ (106)_{10} + (-37)_{10} \quad b)\ (-107)_{10} + (+26)_{10} \quad c)\ (17)_{10} + (-12)_{10} \quad d)\ (-17)_{10} + (12)_{10}$

3.6 Représentation en Complément à 2 (C2)

Le bit le plus à gauche représente toujours le bit signe, mais les nombres entiers négatifs sont représentés par leur compléments à 2.

nbre décimal	bin. naturel	bin. normalisé C2
+5	$(101)_2$	$(00000101)_2$
-5	$-(101)_2$	$(11111011)_2$

• Pour déterminer la valeur décimale d'un nombre écrit en binaire normalisé C2 :

1. si le nombre est positif (bit signe à 0), il suffit de le développer selon les puissances de 2 et calculer sa valeur.
2. si le nombre est négatif (bit signe à 1), deux possibilités sont à notre disposition :
   - soit revenir au nombre positif (en passant au complément à 2), et faire de même qu'en 1.
   - soit en développant directement selon les puissances de 2 mais en faisant attention à associer au bit signe un signe moins.

Ex1 : $(01110010)_2$ est un nombre positif, sa valeur décimale est : $64+32+16+2 = 114$ donc $(01110010)_2 = +114$

Ex2 : $(11010111)_2$ est un nombre négatif, le nombre positif qui lui correspond est $(00101001)_2$ et sa valeur décimale est : $32 + 8 + 1 = 41$

donc $(11010111)_2 = -41$

On peut aussi développer selon les puissances de 2, sans oublier le signe moins pour le bit signe :

$(11010111)_2 = -128 + 64 + 16 + 4 + 2 + 1 = -41$

Attention, c'est le même nombre que nous avons pris comme exemple en SG et en C1, mais il n'a plus la même valeur décimale maintenant qu'il est pris en C2.

Remarques :

(i) Le plus grand nombre (positif) que l'on peut écrire, sur un octet, en notation binaire normalisée C2 est : $(01111111)_2 = +127$

(ii) Le plus petit nombre (négatif) que l'on peut écrire, sur un octet, en notation binaire normalisée C2 est : $(10000000)_2 = -128$

(iii) Dans une notation C2, le zero décimal n'a qu'une seule écriture :

$$0 = (00000000)_2$$

(iv) En C2 nous pouvons écrire $2^8 = 256$ nombres différents.

3.6.1 Addition C2 : Deux nombres de même signe

Ex : Effectuer les additions suivantes en binaire normalisé C2.

$a)$ $(49)_{10} + (25)_{10}$ $b)$ $(82)_{10} + (61)_{10}$ $c)$ $(-18)_{10} + (-59)_{10}$ $d)$ $(-24)_{10} + (-65)_{10}$

3.6.2 Addition C2 : Deux nombres de signes contraires -soustraction-

Même consignes que pour le cas précédent.

Ex : Effectuer les additions suivantes en binaire normalisé C2 :

$a)$ $(59)_{10} + (-18)_{10}$ $b)$ $(64)_{10} + (-23)_{10}$ $c)$ $(19)_{10} + (-58)_{10}$ $d)$ $(-24)_{10} + (-65)_{10}$

Rep :

Nous pouvons aussi effectuer des soustractions. Pour cela, il suffit de savoir que soustraire un nombre revient à ajouter son complément à 2.

Ex2 : Soit à effectuer : $1101\ 0111_2 - 1000\ 1100_2$ en C2. Remarquons que ce sont deux nombres négatifs. Transformons la soustraction en addition :
$1101\ 0111_2 - 1000\ 1100_2 \ = \ 1101\ 0111_2 + 0111\ 0100_2$

Posons l'opération et vérifions en décimal :

$$\begin{array}{cccccccccc} & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} & & \mathit{1} & & \\ & & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ + & & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline = & \cancel{1} & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{array} \qquad C2 : \quad \begin{array}{cccccccc} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ \hline 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array} \qquad Verif^t : \begin{array}{cc} & (-41)_{10} \\ - & (-116)_{10} \\ \hline & (+75)_{10} \end{array}$$

D'où : $1101\ 0111_2 - 1000\ 1100_2 \ = \ 0100\ 1011_2$

3.7 Echelle des nombres entiers en binaire signé

Sur un octet (8 bits)			
	Signe Grandeur (SG)	Complément à 1 (C1)	Complément à 2 (C2)
Le plus grand	01111111=+127 $= 2^7 - 1$	01111111=+127 $= 2^7 - 1$	01111111=+127 $= 2^7 - 1$
Le plus petit	11111111=-127 $= -2^7 + 1$	10000000=-127 $= -2^7 + 1$	10000000=-128 $= -2^7$
Nombre de possibilités	$= 2^8 - 2 = 254$	$= 2^8 - 2 = 254$	$= 2^8 - 1 = 255$
Zéro	00000000 ou 10000000	00000000 ou 11111111	00000000
Sur n octets			
	Signe Grandeur (SG)	Complément à 1 (C1)	Complément à 2 (C2)
Le plus grand	011............1 $= 2^{n-1} - 1$	011.........1 $= 2^{n-1} - 1$	011.........1 $= 2^{n-1} - 1$
Le plus petit	111.........1 $= -2^{n-1} + 1$	100.........0 $= -2^{n-1} + 1$	100.........0 $= -2^{n-1}$
Nombre de possibilités	$= 2^n - 2$	$= 2^n - 2$	$= 2^n - 1$

4 ARITHMETIQUE EN HEXADECIMAL

4.1 Addition en hexadécimal naturel

Règles : 1) Pour chaque colonne d'une opération d'audition, imaginer les deux chiffres hexadécimaux à additionner tel leurs équivalents décimaux :

$$\begin{array}{cccccc} & . & . & . & 2_{16} \\ + & . & . & . & B_{16} \\ \hline = & . & . & . & D_{16} \end{array} \quad \text{en décimal} \quad \begin{array}{ccccc} & . & . & . & 2 \\ + & . & . & . & 11 \\ \hline = & . & . & . & 13 \end{array}$$

2) Si la somme de ces deux chiffres est inférieure ou égale à 15, écrire le chiffre en hexa. correspondant.

3) Si la somme de ces deux chiffres est supérieure à 15, inscrire le chiffre hexa. correspondant à la partie qui excède 16 et ajouter une retenue de 1 à la colonne immédiatement à gauche.

$$\begin{array}{ccccc} & & & \boldsymbol{1} & \\ & . & . & 4 & 8_{16} \\ + & . & . & 6 & B_{16} \\ \hline = & . & . & B & 3_{16} \end{array} \quad \text{en décimal} \quad \begin{array}{ccccc} & . & . & 7 & 2 \\ + & . & 1 & 0 & 7 \\ \hline = & . & 1 & 7 & 9 \end{array}$$

Ex :

$$\begin{array}{cccc} & & 2 & 3_{16} \\ + & & 1 & 6_{16} \\ \hline = & & 3 & 9_{16} \end{array} \qquad \begin{array}{cccc} & \boldsymbol{1} & \boldsymbol{1} & \\ & & D & F_{16} \\ + & & A & C_{16} \\ \hline = & 1 & 8 & B_{16} \end{array} \qquad \begin{array}{cccc} & \boldsymbol{1} & \boldsymbol{1} & \\ & & A & E_{16} \\ + & & 7 & C_{16} \\ \hline = & 1 & 2 & A_{16} \end{array}$$

4.2 Complément à 16

Pour obtenir le complément à 16 d'un nombre écrit en hexadécimal, on peut :

$\star$ Soit Passer par le binaire et faire un complément à 2 :

Ex : Le complément à 16 (C16) du nombre $(C9)_{16}$ est le nombre $(37)_{16}$, en effet : $C9_{16} = (1100\ 1001)_2$ et son complément à 2 est $(0011\ 0111)_2$.

⋆ Soit directement : on commence à droite du nombre, on garde tous les zéros jusqu'au premier chiffre différent de zéro. ce dernier, on le complémente à 16 et tous les autres chiffres à sa gauche on les complémente à 15 :

$$\begin{array}{cc} C & 9 \\ 3 & 7 \\ \widehat{C_{15}} & \widehat{C_{16}} \end{array}$$

Ex :

$$\begin{array}{ccc} A & 0 & B \\ 5 & F & 5 \\ \widehat{C_{15}} & \widehat{C_{15}} & \widehat{C_{16}} \end{array} \qquad \begin{array}{cccc} 3 & B & E & 0 \\ C & 4 & 2 & 0 \\ \widehat{C_{15}} & \widehat{C_{15}} & \widehat{C_{16}} & \end{array} \quad C(16) de A0B = 5F5, \;\; C(16) de 3BE0 = C420$$

⋆ Soit encore en associant à chaque chiffre la valeur qui lui manque pour obtenir 15 puis on ajoute 1 :

$$\begin{array}{ccc} & 2 & A \\ & D & 5 \\ + & & 1 \\ \hline & D & 6 \end{array}$$

Le complément à 16 de 2A est D6.

4.3 Nombres signés en hexadécimal

De même qu'en binaire signé, le nombre de bits à utiliser est fixé (normalisé) et le bit le plus à gauche représente le signe.

En général nous utiliserons deux octets (16 bits) pour écrire les nombres signés en hexadécimal, mais nous ajouterons un octet à chaque fois que nécessaire.

- Si le chiffre le plus à gauche est compris entre 0 et 7, alors le nombre signé est positif. Ex : $(2510)_{16}$ est positif.

- Si le chiffre le plus à gauche est compris entre 8 et F, alors le nombre signé est négatif. Ex : $(A531)_{16}$ est négatif.

1. *Comment convertir un nombre décimal signé en hexadécimal signé ?*

ll faut d'abord voire si le nombre donné est positif ou négatif, puis convertir la valeur absolue du nombre en hexa.

- Si le nombre donné est positif, on garde cette écriture pour le nombre (compléter par des zéros à gauche, si nécessaire, pour avoir deux octets)

- Si le nombre donné est négatif, il faut déterminer son complément à 16 sur les deux octets.

Ex : $251 = (FB)_{16}$
$+251 = (00FB)_{16}$
$-251 = (FF05)_{16}$

Exercice : Ecrire les nombres décimaux suivants en hexadécimal signé :

$$a)\ -345;\ b)\ +2117;\ c)\ -34885$$

Rep : $a)\ FEA7_{16};\ b)\ 0845_{16}\ c)\ F77BB_{16}$

2. *Comment convertir un nombre hexadécimal signé en décimal signé ?*

Si le nombre donné est positif (le chiffre le plus à gauche est ¡ 8) il suffit de le développer selon les puissances de 16. S'il est négatif, il faut d'abord le complémenter à 16 puis calculer, en développant selon les puissances de 16, la valeur du nombre positif correspondant puis écrire la valeur négatif du nombre donné :

Ex1 : $(0141)_{16} = (?)_{10}$ c'est un nbre positif d'où $(0141)_{16} = 16^2+4*16+1 = (+321)_{10}$

Ex2 : $(FA2B)_{16} = (?)_{10}$ c'est un nbre négatif d'où il faut déterminer son complément à 16 : $(05D5)_{16}$, calculer sa valeur : $(05D5)_{16} = 5*16^2+13*16+5 = 1493_{10}$ et donc $(FA2B)_{16} = (-1493)_{10}$

Exercice : Déterminer la valeur décimal des nombres suivants, écrits en hexadécimal signé :

$$a)\ 0357;\ b)\ F834;\ c)\ BC3E$$

Rep : $a)\ +855_{10}\ b)\ -1996_{10}\ c)\ -17346_{10}$

4.4 Addition de nombres signés en hexadécimal (C16)

Comme en décimal ou en binaire, l'addition s'effectue chiffre par chiffre. Si le résultat dépasse "F", on soustrait "F" et on propage un report de "1" pour le chiffre suivant, vers la gauche. Dans le cas de deux nombres négatifs, le dernier report tout à gauche est à éliminer pour obtenir la réponse finale, mais faire attention au cas de dépassement, cas où le nombre de bits réservés à l'écriture n'est pas suffisant. dans ce cas il faut prendre plus d'espace (ajouter un octet à gauche par exemple).

4.4.1 Deux nombres de même signe

Ex1 : Soit à effectuer : $345_{16} + 0B18_{16}$. Remarquons que ce sont deux nombres positifs et posons l'opération :

$$\begin{array}{ccccc} & 0 & 3 & 4 & 5_{16} \\ + & 0 & B & 1 & 8_{16} \\ \hline = & 0 & E & 5 & D_{16} \end{array} \qquad Verif^t : \quad \begin{array}{ccccc} & & 8 & 3 & 7_{10} \\ + & 2 & 8 & 4 & 0_{10} \\ \hline = & 3 & 6 & 7 & 7_{10} \end{array}$$

D'où : $0345_{16} + 0B18_{16} \ = \ 0E5D_{16}$

Ex2 : Soit à effectuer : $52C5_{16} + 2BA7_{16}$. Remarquons que ce sont deux nombres positifs et posons l'opération :

$$\begin{array}{ccccc} & & \boldsymbol{1} & & \\ & 5 & 2 & C & 5_{16} \\ + & 2 & B & A & 7_{16} \\ \hline = & 7 & E & 6 & C_{16} \end{array} \qquad Verif^t : \quad \begin{array}{ccccc} & & \boldsymbol{1} & \boldsymbol{1} & \\ 2 & 1 & 1 & 8 & 9_{10} \\ +1 & 1 & 1 & 7 & 5_{10} \\ \hline =3 & 2 & 3 & 6 & 4_{10} \end{array}$$

D'où : $52C5_{16} + 2BA7_{16} \ = \ 7E6C_{16}$

Ex3 : Soit à effectuer : $EE35_{16} + F824_{16}$. Remarquons que ce sont deux nombres négatifs et posons l'opération :

$$\begin{array}{ccccc} \boldsymbol{1} & \boldsymbol{1} & & & \\ & E & E & 3 & 5_{16} \\ + & F & 8 & 2 & 4_{16} \\ \hline =\cancel{1} & E & 6 & 5 & 9_{16} \end{array} \qquad C16 : \quad \begin{array}{cccc} 1 & 1 & C & B_{16} \\ 0 & 7 & D & C_{16} \\ \hline 1 & 9 & A & 7_{16} \end{array} \qquad Verif^t : \quad \begin{array}{ccccc} & & \boldsymbol{1} & \boldsymbol{1} & \\ - & 4 & 5 & 5 & 5_{10} \\ +- & 2 & 0 & 1 & 2_{10} \\ \hline - & 6 & 5 & 6 & 7_{10} \end{array}$$

D'où : $EE35_{16} + F824_{16} \ = \ E659_{16}$

Ex4 : Soit à effectuer : $B182_{16} + F1A3_{16}$. Remarquons que ce sont deux nombres négatifs et posons l'opération :

$$
\begin{array}{ccccc}
\boldsymbol{1} & & \boldsymbol{1} & & \\
 & B & 1 & 8 & 2_{16} \\
+ & F & 1 & A & 3_{16} \\
\hline
=\cancel{1} & A & 3 & 2 & 5_{16}
\end{array}
\qquad C16 : \quad
\begin{array}{cccc}
4 & E & 7 & E_{16} \\
0 & E & 5 & D_{16} \\
\hline
5 & C & D & B_{16}
\end{array}
\qquad Verif^{t} : \quad
\begin{array}{cccccc}
 & & & \boldsymbol{1} & \boldsymbol{1} & \\
 & -2 & 0 & 0 & 9 & 4_{10} \\
+ & - & 3 & 6 & 7 & 7_{10} \\
\hline
 & -2 & 3 & 7 & 7 & 1_{10}
\end{array}
$$

D'où : $B182_{16} + F1A3_{16} \ = \ A325_{16}$

4.4.2 Deux nombres de signes contraires

Ex1 : Soit à effectuer : $0177_{16} + FBCD_{16}$. Remarquons que ce sont deux nombres de signes contraires et posons l'opération :

$$
\begin{array}{ccccc}
 & & \boldsymbol{1} & \boldsymbol{1} & \\
 & 0 & 1 & 7 & 7_{16} \\
+ & F & B & C & D_{16} \\
\hline
= & F & D & 4 & 4_{16}
\end{array}
\qquad C16 : \quad
\begin{array}{cccc}
0 & 4 & 3 & 3_{16} \\
\hline
0 & 2 & B & C_{16}
\end{array}
\qquad Verif^{t} : \quad
\begin{array}{ccccc}
 & & 3 & 7 & 5_{10} \\
+\,- & 1 & 0 & 7 & 5_{10} \\
\hline
 & - & 7 & 0 & 0_{10}
\end{array}
$$

D'où : $0177_{16} + FBCD_{16} \ = \ FD44_{16}$

Ex2 : Soit à effectuer : $B181_{16} + 0345_{16}$. Remarquons que ce sont deux nombres de signes contraires et posons l'opération :

$$
\begin{array}{ccccc}
 & B & 1 & 8 & 1_{16} \\
+ & 0 & 3 & 4 & 5_{16} \\
\hline
= & B & 4 & C & 6_{16}
\end{array}
\qquad C16 : \quad
\begin{array}{cccc}
4 & E & 7 & F_{16} \\
 & & & \\
\hline
4 & B & 3 & A_{16}
\end{array}
\qquad Verif^{t} : \quad
\begin{array}{cccccc}
 & -2 & 0 & 0 & 9 & 5_{10} \\
+ & & & 8 & 3 & 7_{10} \\
\hline
 & -1 & 9 & 2 & 5 & 8_{10}
\end{array}
$$

D'où : $B181_{16} + 0345_{16} \ = \ B4C6_{16}$

Remarque : Soustraire un nombre revient à ajouter son complément à 16. Donc nous pouvons faire des soustractions.

Ex : Soit à effectuer : $0177_{16} - FBCD_{16}$. Avant de poser l'opération remarquer que c'est une soustraction, et qu'il faut la transformer en addition en appliquant le principe rappelé dans la remarque ci-dessus :

$$0177_{16} - FBCD_{16} \ = \ 0177_{16} + 0433_{16}$$

Posons maintenant l'opération et vérifions en décimal :

$$\begin{array}{cccccc} & 0 & 1 & 7 & 7_{16} \\ + & 0 & 4 & 3 & 3_{16} \\ \hline = & 0 & 5 & A & A_{16} \end{array} \qquad Verif^t : \qquad \begin{array}{ccccc} & + & 3 & 7 & 5_{10} \\ - & -1 & 0 & 7 & 5_{10} \\ \hline & 1 & 4 & 5 & 0_{10} \end{array}$$

D'où : $0177_{16} - FBCD_{16} \ = \ 0177_{16} + 0433_{16} \ = \ 05AA_{16}$

Exercice1 : Effectuer, sur deux octets, en signé C16, les opérations suivantes puis les verifier en décimal :

$$a)\ 1375_{16} - 0B18_{16};\ \ b)\ CB18_{16} - 02D1_{16};\, ;\ \ c)\ A104_{16} - 92E0_{16};$$

Rep : $a)$ $085D_{16}$ $b)$ $C847_{16}$ $c)$ $0E24_{16}$

Exercice2 : Effectuer, sur deux octets, en signé C16, les opérations suivantes données en décimal :

$$a)\ 2801_{10} - 85_{10};\ \ b)\ 72_{10} - 1451_{10};\, ;\ \ c)\ -3201_{10} - 1870_{10};$$

Rep : $a)$ $0A9C_{16}$ $b)$ $FA9D_{16}$ $c)$ $EC31_{16}$

5 NOMBRES A VIRGULE FLOTTANTE

Pour représenter de très grands nombres entiers, il faut beaucoup de bits. Et ça se complique encore lorsqu'on veut écrire des nombres réels, possédants un signe, une partie entière et une partie décimale. Le système des nombres à virgule flottante, qui repose sur le principe de notation scientifique, permet de représenter des nombres de très grande ou très petite valeur et des nombres réels à partie fractionnaire, sans augmenter le nombre de bits. les nombres à virgule flottante sont les nombres les plus souvent utilisés dans un ordinateur pour représenter des valeurs non entières. Ce sont des approximations de nombres réels.

Un *nombre à virgule flottante*, également appelé nombre réel, est composé de trois parties dont une réservée au *signe*, la deuxième à la *mantisse* qui représente la grandeur du nombre, et la dernière à *l'exposant* qui désigne la quantité de rangs avec laquelle la virgule décimale ou binaire est décalée.

Rappelons que pour obtenir l'écriture en notation scientifique d'un nombre, il faut déplacer la virgule décimale à la gauche de tous les chiffres et n'en garder qu'un non nul. L'exposant est alors une puissance de 10. Prenons l'exemple du nombre entier décimal qui représente la distance Terre-Soleil à peu près149,6 millions km ou encore mieux 149597870 km. Ce grand nombre est souvent écrit en notation scientifique : $1,49597870.10^8$ km

Pour les nombres à virgule flottante binaires, le format est défini par la norme ANSI/IEEE 754-1985 selon trois notations : simple précision (SP) , double précision (DP) et précision étendue (PE). Leur format est le même, à l'exception du nombre de bits. En SP les nombres sont écrits sur 32 bits, 64 en DP et 80 pour la PE.

Nous limiterons nos écritures en virgule flottante SP et quelque fois DP.

5.1 Nombres à virgule flottante simple précision

Pour écrire les nombres binaires en virgule flottante simple précision, on utilise quatre octets (32 bits) et on adopte une écriture normalisée, c'est à dire un seul chiffre non nul avant la virgule -un 1- .

Le bit signe (S) est celui situé le plus à gauche, l'exposant (E) inclut les 8 bits suivants, et la mantisse (M) comprend les 23 bits de droite :

$\longleftarrow - - - - - - - - - \quad 32 bits \quad - - - - - - - - - \longrightarrow$

S	**Exposant (E)**	**Mantisse (M)**
$1bit$	$8bits$	$23bits$

$\star$ Si le bit signe est à 1 le nombre est négatif, s'il est à 0 alors le nombre est positif.

$\star$ La virgule binaire de la mantisse se trouve à la gauche des 23 bits.

$\star$ Les 8 bits de l'exposant représentent un exposant polarisé, obtenu en additionnant 127 à l'exposant réel :

$$\text{Exposant polarisé} \ = \ 127 + \text{Exposant réel}$$

Cette polarisation permet d'écrire des nombres très grands ou très petits sans avoir besoin d'introduire un signe à l'exposant.

$\longrightarrow$ *Comment écrire un nombre donné dans une zone à virgule flottante simple précision ?*

Exemple1 : Ecrire le nombre décimal 5777 dans une zone à virgule flottante SP :
Pour cela il faut d'abord convertir ce nombre en binaire (naturel), décaler la virgule afin de le normaliser (avoir $1, \cdots$), puis déterminer l'exposant polarisé et enfin écrire le nombre :

$$5777 \ = \ 1011010010001_2 \ = \ 1,011010010001 \ . \ 2^{12}$$

- Bit signe $\longrightarrow$ 0 (car nbre positif)
- Exposant polarisé = 127 + 12 = 139 = $1000\ 1011_2$
- Mantisse = 011010010001

D'où :

$$+5777 \ = \ 0 \ \ 10001011 \ \ 01101001000100000000000$$

Exemple2 : Ecrire le nombre hexadécimal $7C2B_{16}$ dans une zone à VF SP :

$$7C2B \ = \ 0111\ 1100\ 0010\ 1011_2 \ = \ 1,11110000101011 \ . \ 2^{14}$$

- Bit signe $\longrightarrow$ 0 (car nbre positif)
- Exposant polarisé = 127 + 14 = 141 = $1000\ 1101_2$
- Mantisse = 11110000101011

D'où :

$$7C2B \ = \ 0 \ \ 10001101 \ \ 11110000101011000000000$$

Exemple3 : Ecrire le nombre décimal -128,109375 dans une zone à VF SP :

$$128,109375 \ = \ 1000\ 0000\ ,\ 0001\ 1100_2 \ = \ 1,000000000011100 \ . \ 2^{7}$$

- Bit signe $\longrightarrow$ 1 (car nbre négatif)
- Exposant polarisé = 127 + 7 = 134 = $1000\ 0110_2$
- Mantisse = 000000000011100

D'où :

$$-128,109375 \;=\; 1 \;\; 10000110 \;\; 00000000001110000000000$$

$\longrightarrow$ *Comment déterminer la valeur décimale d'un nombre donné écrit dans une zone à virgule flottante simple précision ?*

Pour évaluer la valeur décimal d'un nombre binaire déjà écrit en notation à virgule flottante simple précision, il faut suivre la formule suivante :

$$\text{Nombre} \;=\; (-1)^S.(1+M).2^{E-127}$$

Exemple1 : Evaluer la valeur décimale du nombre binaire écrit dans une zone à VF SP suivant :

$$1 \;\; 10010101 \;\; 01010110010000000000000$$

- Signe : négatif (car bit signe à 1)
- Exposant polarisé = 1001 0101 = 149 et Exposant réel = $149 - 127 = 22$
- Mantisse = 01010110010000000000000

D'où

$$\begin{aligned} Nbre &= -\,1,01010110010000000000000 \;.\; 2^{22} \\ &= -\,1,0101011001 \;.\; 2^{22} \\ &= -\,101\;0101\;1001 \;.\; 2^{12} \\ &= -\,(2^{10}+2^{8}+2^{6}+2^{4}+2^{3}+1).\; 2^{12} \\ &= -\,5,607424.\; 10^{6} \end{aligned}$$

5.2 Echelle des nombres en virgule flottante

• *Limites au niveau exposant* (8bits) : Tous les bits ne peuvent pas être tous à 1 (overflow) ou à 0 (underflow).

- Le plus grand exposant polarisé possible est $(11111110)_2 = 254_{10}$
$\Longrightarrow$ Le plus grand exposant réel possible est : 254 - 127 = 127.

- Le plus petit exposant (polarisé) possible est $(00000001)_2 = 1_{10}$
$\Longrightarrow$ Le plus petit exposant réel possible est : 1 - 127 = -126.
L'exposant réel varie donc entre -126 et +127.

• *Limites au niveau de la mantisse* (23bits) :
- La plus grande mantisse : $(11111111111111111111111)_2 = 1 - 2^{-23} = 0,999999881$
- La plus petite mantisse : $(00000000000000000000000)_2 = 0$

En conclusion, En virgule flottante simple précision, nous pouvons écrire des nombres :
- positifs qui vont de $1.2^{-126} \simeq 1,17.10^{-38}$ à $1.9999....2^{+127} \simeq 3,4.10^{38}$
- négaifs qui vont de $-1.9999....2^{+127} \simeq -3,4.10^{38}$ à $1.9999....2^{+127} \simeq -1,17.10^{-38}$

Echelle des nombres en virgule flottante

	Simple précision	**Double précision**
Bit de signe	1	1
Bit d'exposant	8	11
Bit de mantisse	23	52
Nombre total de bits	32	64
Codage de l'exposant	Excédant 127	Excédant 1023
Variation de l'exposant	-126 à +127	-1022 à +1023
Plus petit nombre normalisé	2^{-126}	2^{-1022}
Plus grand nombre normalisé	2^{+128}	2^{+1024}
Echelle des nombres décimaux	Entre $10^{-38} et\ 10^{+38}$	Entre $10^{-308} et\ 10^{+308}$

6 Exercices : Binaire, Hexa et Virgule Flottante

Ex1. Convertir les nombres binaires suivants en décimal :
$a)$ 11 $b)$ 100 $c)$ 111 $d)$ 1000 $e)$ 1001 $f)$ 1100 $g)$ 1011 $h)$ 1111
$i)$ 1110 $j)$ 11101 $k)$ 10111 $l)$ 11111 $m)$ 11100 $n)$ 10101 $o)$ $110011,11$
$p)$ $1000001,111$ $q)$ $1011100,10101$ $r)$ $1011010,1010$ $s)$ $1111111,11111$

Ex2. Quel est le nombre décimal maximal représentable, en binaire, si on ecrit sur :
a) 2 bits, b) 3 bits, c) 4 bits d) 10 bits, e) 2 octets, f) 100 bits, g) n bits.

Ex3. Combien faut-il de bits pour représenter les nombres décimaux suivants en binaire :
$a)17$ $b)35$ $c)49$ $d)81$ $e)14$ $f)132$ $g)205$.

Ex4. Convertir les nombres décimaux suivants en binaire naturel (sur un octet pour la 1ère série et sur 2 octets pour la 2ième) :
$a)121$ $b)29$ $c)67$ $d)0,82$ $e)0,429$ $f)5,28$ $g)257$
$h)458,71$ $i)129,8923$ $j)51,375$ $k)47,18$.

Ex5. Effectuer les additions suivantes en binaire naturel :
$a)11 + 01;$ $b)10 + 10;$ $c)101 + 11;$ $d)111 + 110;$
$e)1001 + 101;$ $f)1101 + 1011;$ $g)100101 + 1110111;$

Ex6. Dans une écriture sur un octet, déterminer le complément à 1 et le complément à 2 des nombres binaires suivants :
$a)10;$ $b)111;$ $c)1001;$ $d)1101;$ $e)11100;$ $f)10011;$ $g)10110000;$ $h)00111101;$ $i)11111111;$

Ex7. Exprimer chaque nombre décimal en un nombre binaire sur un octet en notation signe-grandeur :
$a)29;$ $b) - 85;$ $c)100;$ $d) - 123.$

Ex8. Exprimer chaque nombre décimal en un nombre binaire sur un octet en notation complément à 1 :
$a) - 34;\quad b)57;\quad c) - 99;\quad d)115.$

Ex9. Exprimer chaque nombre décimal en un nombre binaire sur un octet en notation complément à 2 :
$a)12;\quad b) - 68;\quad c)101;\quad d) - 125.$

Ex10. Déterminer la valeur décimale des nombres binaires signés en notation SG : $a)$ 10011001; $b)$ 011101

Ex11. Déterminer la valeur décimale des nombres binaires signés en notation à complément à 2 : $a)$ 10011001; $b)$ 01110100; $c)$ 10111111.

Ex12. Convertir chaque paire de nbres décimaux en binaire, à l'aide de la notation en C2, sur un octet, et les additionner :
$a)$ 33 et 15; $b)$ 56 et -27; $c)$ -46 et 25; $d)$ -110 et -84

Ex13. Effectuer les additions suivantes en binaire normalisé, sur un octet (nbrs signés en notation C2) :
a) 00010110+00110011; $b)$ 01110000+10101111; $c)$ 00110011+00010000; $d)$ 11011001+ 11100111;

Ex14. Convertir chaque nombre binaire en hexadécimal (naturel) :
$a)$ 1110; $b)$ 10111; $c)$ 1111110000; $d)$ 111001, 0110 $e)$ 111111000, 1010110101.

Ex15. Convertir chaque nombre hexadécimal en binaire (naturel) :
$a)30_{16};\quad b)A14_{16};\quad c)FB17_{16};\quad d)5C8,14_{16}\quad e)4100_{16};\quad f)8A9D_{16}.$

Ex16. Convertir chaque nombre hexadécimal en décimal (naturel) :
$a)\ 92_{16};\quad b)\ 8D_{16};\quad c)\ 5C2_{16};\quad d)\ 700_{16};\quad e)\ 1A_{16}.$

Ex17. Convertir chaque nombre décimal en hexadécimal (naturel) :
$a)\ 125;\quad b)\ 452;\quad c)\ 1295,56;\quad d)\ 12,5962.$

Ex18. Effectuer les opérations suivantes en hexadécimal naturel :
$a)$ $58 + 32$; $b)$ $AE + 7C$; $c)$$4CE1 + 3A62$; $d)$ $B0132 + 1FCA1$

Ex19. Déterminer le complément à 16 des nombres écrits en hexadécimal suivants : $a)$ $C9$; $b)$ 159; $c)$ $F7BB$; $d)$ $0EB1$; $e)$ $3A10$

Ex20. Ecrire les nombres décimaux suivants en hexadécimal signé (C16) sur 2 octets : $a)$ 45; $b)$ -45; $c)$ 2117; $d)$ -2117; $e)$ 372; $f)$ -57034

Ex21. Déterminer la valeur décimale des nombres signés écrits en hexadécimal (C16) suivants : $a)$ $C3$; $b)$ 251; $c)$ $OB3$; $d)$ 1234; $e)$ ABC;

Ex22. Effectuer les opérations suivantes de nombres écrits en hexadécimal signé (en complément à 16) puis convertir en décimal pour vérifier l'opération : $a)$ $C3 + F5$ $b)$ $345 + B18$; $c)$ $177 - BCD$; $d)$ $1C3 - 5A$;
$e)$ $048 - 5AB$; $f)$ $0AF1 - 0055$; $g)$ $0A64 - 1245$.

Ex23. Une zone à virgule flottante en SP contient l'information suivante :
1 10010100 11100110011000000000000 Quelle est la valeur de cette information en décimal ?

Ex24. Convertir en décimal les nombres stockés en virgule flottante (SP) et écrits en hexadécimal : $a)$ $(CAFE1000)_{16}$; $b)$ $C0A4E200_{16}$; $c)$ $6643E900$.

Ex25. Soit le nombre signé (C16) : $7C2B_{16}$ L'écrire dans une zone à virgule flottante (SP)

Ex26. Soit le nombre décimal $128,109375$ L'écrire dans une zone à virgule flottante (SP)

7 Exercices : Adresses IP

Exercice 1 Soit le réseau suivant : 120.0.0.0. Ce réseau contient 8191 sous-réseaux. On souhaite maximiser le nombre de clients par sous- réseau. La passerelle par défaut est à la 255 ème adresse du sous-réseau.
1) Quelle est la classe du réseau ?
2) Quel est le masque de sous-réseau ?
3) Quelle est l'IP du client $n°$ 680 sous-réseau $n°$ 4000 ?
4) Quelle est l'adresse de broadcaste de ce sous-réseau ?
5) Quelle est l'adresse de ce sous-réseau ?
6) Quelle est l'adresse de la passerelle par défaut de ce sous-réseau ?

Exercice 2 Le client $n°$ 325 du sous-réseau $n°$ 4 a l'adresse suivante : 130.150.33.69.
1) Quel est le masque ?
2) Calculer les adresses IP des clients suivants :

$n°$ client au sein du sous-réseau	Adresse IP
1023	
279	
1221	
617	
2002	
1324	

Exercice 3 Soit l'IP du client numéro 153 dans le sous-réseau numéro 7800 : 20.243.192.153
Calculer, en maximisant le nombre de clients :
1) Le masque
2) La DG (première adresse du sous-réseau)
3) Le broadcast
4) L'adresse du sous-réseau

Exercice 4 Soit l'IP du client $n°$ 1448 du sous-réseau $n°$ 10 : 15.2.133.168
Calculer le masque, la DG (dernière adresse du sous-réseau), l'adresse de broadcast et l'adresse du sous-réseau, en maximisant le nombre de clients.

Exercice 5 Lesquelles des IP suivantes sont dans le même sous-réseau que 10.40.203.250, avec le masque 255.255.240.0 ?

Ad.IP	Mettre une croix dans l'affirmative
10.40.200.1	
10.40.207.254	
10.40.191.2	
10.40.240.25	
10.45.203.250	

Exercice 6 Quelle est l'adresse du sous-réseau qui a un client dont l'adresse IP est 200.145.90.175 et dont le masque est 255.255.255.240 ?

Exercice 7 Quelle est l'adresse IP du client numéro 150 du sous-réseau numéro 63, notée : 135.54.H1.H2, sachant que chaque sous-réseau comporte exactement 255 personnes ?

Exercice 8 Quel est le masque qui permet de mettre le client dont l'adresse IP est 200.145.90.175 et une autre dont l'adresse est 200.145.90.185 dans le même sous réseau.

Exercice 9 Quelle est l'adresse IP du client numéro 50, du sous-réseau numéro 32, notée : 135.54.H1.H2, sachant que chaque sous-réseau comporte exactement 511 personnes ?

Exercice 10 Quel est le masque qui permet de mettre le client dont l'adresse IP est 200.145.90.175 et un autre dont l'adresse IP est 200.145.90.185 dans deux sous-réseaux voisins ?

Exercice 11 Le masque de la machine-client est 255.255.255.0.
Combien y a-t-il au maximum de SR autour de lui si c'est un réseau de classe A, puis B, puis C ?

Exercice 12 Le client numéro 10 du SR numéro 6 d'un R(C) a l'IP 200.200.200.X, où X a été effacé et est irrécupérable. Cependant, l'utilisateur est certain que ce nombre, de mémoire, était supérieur à 150. Quel est ce nombre ? Ensuite, retrouver le masque qui est lui aussi disparu.

Exercice 13 Donner le masque qui met les 5 adresses suivantes dans le même plus petit SR commun 145.120.34.67 - 145.120.31.78 - 145.120.39.233 - 145.120.33.76 - 145.120.38.195

Exercice 14 InterNIC vous a affecté une adresse 202.17.69.0 de classe C. Votre réseau privé comporte actuellement cinq sous-réseaux. Chacun d'entre eux dispose d'environ 10 hôtes. Le nombre de sous-réseaux sera doublé l'an prochain. Le nombre d'hôtes de deux des sous-réseaux pourrait atteindre 14.
- Quel masque de sous-réseau devez-vous choisir ?
- Quel est le nombre maximum de sous-réseaux autorisé ?
- Quel est le nombre maximum d'hôtes par sous-réseau autorisés ?
- Quelle est l'adresse du 3ième sous réseau et des trois premières machines de ce sous-réseau ?

Exercice 15 (Janvier 2012) On souhaite créer un réseau 172.46.X.X permettant 31 sous-réseaux tout en maximisant le nombre de clients par sous-réseau.
a) Quel est le masque nécessaire ?
b) Quelle est l'adresse du sous-réseau numéro 10 ?
c) Quelle est l'adresse du client numéro 852 de ce sous-réseau ?
d) Quelle est l'adresse du broadcast de ce sous-réseau ?
e) Avec cette segmentation, combien de sous-réseaux sont utilisables ?
f) Avec cette segmentation, combien de clients sont possibles ?
g) Avec cette segmentation, combien d'adresses utiles ont été perdues ?

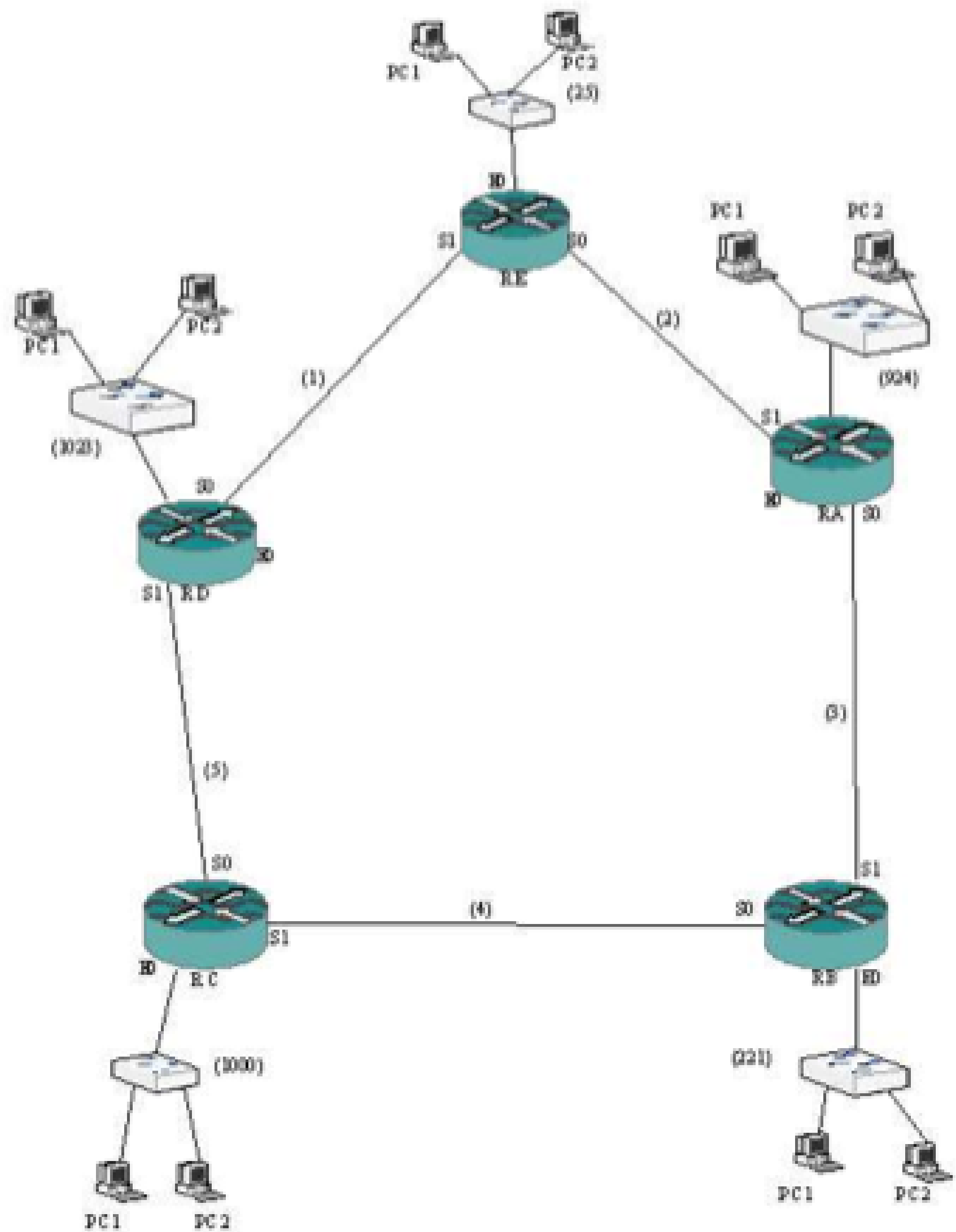
Exercice 16 (Juin 2012) Dans un réseau, l'adresse IP d'un des PC est 140.190.51.74 et le masque du sous-réseau est 255.255.240.0.
a) Quel est le numéro du sous-réseau auquel appartient ce PC ?
b) Quelle est l'adresse de ce sous-réseau ?
c) Quel est le numéro du PC (client) dans ce sous réseau ?
d) Quelle est l'adresse du dernier client du sous-réseau suivant ?
e) Le client 140.190.63.28 partage-t-il le même sous-réseau que 140.190.53.67 ?

Exercice 17 (Janv. 2016) Un client a pour adresse IP 193.222.8.98 et le masque de sous-réseau associé est 255.255.255.192.
a) Quelle est l'adresse du sous-réseau dans lequel se trouve ce client ?
b) Quel est le numéro de ce client ?
c) Quel est le numéro du sous-réseau de ce client ?
d) Quelle est l'adresse de diffusion (broadcast) ?
e) Il faut se connecter à un serveur d'adresse IP 193.222.8.171. Appartient-il au même sous-réseau que l'adresse précédente ? Justifier.

Exercice 18 (Janv. 2016) On vous attribue le réseau 125.0.0.0. On souhaite créer 1000 sous-réseaux tout en maximisant le nombre de clients par sous-réseaux (Tous les numéros de sous-réseau sont possibles).
a) Quel masque utiliser ?
b) Quelle est l'adresse du sous-réseau numéro 100 ?
c) Quelle est l'adresse du client numéro 2001 du sous-réseau numéro 101 ?
d) A quel sous-réseau (numéro) appartient le client 125.35.71.170 ?

Exercice 19 Soit le réseau 10.0.0.0.

Sur le schéma, à la page suivante, les numéros entre parenthèses sont les numéros des sous-réseaux, le PC 1 est le client 255 et le PC 2 le client 320. Sachant qu'on souhaite maximiser le nombre de clients par sous-réseau, déterminer, pour chaque sous-réseau, le masque, la dernière adresse du sous-réseau (E0), la première adresse du sous-réseau (S0), la deuxième adresse du sous-réseau (S1) et les adresses des PC1 et PC2.


PC 1
PC 2
(25)
S1
RE
PC 1
PC 2
(1)
(2)
S1
RA
S0
RD
(3)
(5)
S0
(4)
S0
S1
S1
RC
(1000)
RB
(221)
PC 1
PC 2
PC 1
PC 2

copie.png