

TD sur les systèmes de numération - conversion binaire - hexadécimal

Exercice 1

1. Comptez jusqu'à 20 en binaire en indiquant la valeur équivalente en décimal.
2. Combien d'octets font 32 bits ?
3. Dans l'octet suivant : $(1001\ 1010)_2$, quel est le bit de poids fort, le bit de poids faible ?

Exercice 2

Faites les opérations ci-dessous, en utilisant un tableau de huit colonnes (une pour chaque bit) :

1. Calculer : $(1100\ 0110)_2 + (0010\ 0110)_2$
2. Calculer : $(1110\ 1110)_2 + (1110\ 1110)_2$: Quelle constatation faites-vous sur le résultat ?
3. Calculer : $(1110\ 1110)_2 - (1110\ 1111)_2$: Quelle constatation faites-vous sur le résultat ?

Exercice 3

1. Dans le nombre $(40\ 04)_{10}$, Qu'est-ce qui différencie le 4 de gauche de celui de droite ?
2. Convertir $(128)_{10}$ en binaire.
3. Convertir $(1100\ 0110)_2$ en décimal.

Exercice 4

1. Combien vaut $(A)_{16}$ en décimal ? Combien vaut $(4A)_{16}$ en décimal ?
2. Convertir $(510)_{10}$ en hexadécimal.
3. Convertir $(1100\ 0110)_2$ en hexadécimal.

Exercice 5

1. Complétez le tableau suivant :

Image

2. Quelles sont les valeurs minimales et maximales que l'on peut représenter sur un octet ?
3. Sachant cela, à votre avis, combien de valeurs différentes peut-on représenter sur un octet ?
4. Décalage : Il existe un opérateur (comme +, -, ...) qui consiste à décaler chaque bit d'une position vers la gauche puis à introduire un 0 à droite.

• Convertir $(75)_{10}$ en binaire. Effectuer un décalage à gauche de ce résultat.

Reconvertir en décimal. Conclusion ?

- Quel sera le problème avec $(200)_{10}$ s'il est représenté en binaire sur un octet ?
5. Combien de bits sont nécessaires pour représenter $(18\ 450)_{10}$ en binaire ?

Exercice 6

Dans un réseau, les ordinateurs fonctionnant avec le protocole TCP/IP sont identifiés par une adresse. Cette adresse est composée de quatre nombres. Par exemple, une machine pourrait avoir l'adresse 172.16.0.3. **Ceci est une adresse IP.**

Chaque nombre séparé par un point est codé sur un octet. L'adresse est donc constituée de 4 octets.

1. Sur combien de bits est codé chaque nombre d'une adresse IP ? Sachant cela, quelle est la taille en bits d'une adresse complète ?
2. Pour chaque nombre, donnez l'intervalle de valeurs possibles. En déduire le nombre d'adresses différentes que l'on peut théoriquement former.

Les adresses IP sont structurées :

- une partie identifie le réseau ;
- une partie identifie l'appareil dans le réseau.

Imaginons que l'on est sur l'appareil d'adresse IP : 172.16.1.80. À cette adresse est associé un **masque** qui a la forme : 255.255.0.0. Ce masque sert au logiciel TCP/IP fonctionnant sur la machine à calculer la partie réseau de l'adresse.

3. Convertissez en binaire l'adresse IP de la machine. Attention, on attend pour chaque nombre de l'adresse, un résultat sur 8 bits.
4. Convertissez en binaire le masque (idem ci-dessus, chaque nombre doit être sur 8 bits).
5. Réalisez un ET logique bit à bit (rappelez-vous de la table de vérité du ET) entre l'adresse IP et le masque.
6. Convertissez le résultat en décimal. Que constatez-vous ?
Imaginons que la machine 172.16.1.80 souhaite communiquer avec la machine d'adresse IP 193.252.19.3.
7. Réalisez les mêmes opérations que précédemment avec cette nouvelle adresse.
8. Peut-on dire que la machine 172.16.1.80 et 193.252.19.3 sont dans le même réseau ?

Correction

Solution Exercice 1

1.

Binaire	Décimal
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15
10000	16
10001	17
10010	18
10011	19
10100	20

2. 32 bits font 4 octets puisque $8 \text{ bits} = 1 \text{ octet}$

3. Dans $(1001\ 1010)_2$ le bit de poids fort est le 1 le plus à gauche, le bit de poids faible est le 0 le plus à droite.

Solution Exercice 2

1.

Retenues					1	1		
	1	1	0	0	0	1	1	0
+0	0	1	0	0	1	1	1	0
	1	1	1	0	1	1	0	0

2.

Retenues	1	1		1	1	1		
	1	1	1	0	1	1	1	0
+1	1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	0	0

Le résultat ne tient pas sur 8 bits mais sur 9. On verra un peu plus tard que ce peut être une source de problèmes lorsque l'on programme.

3.

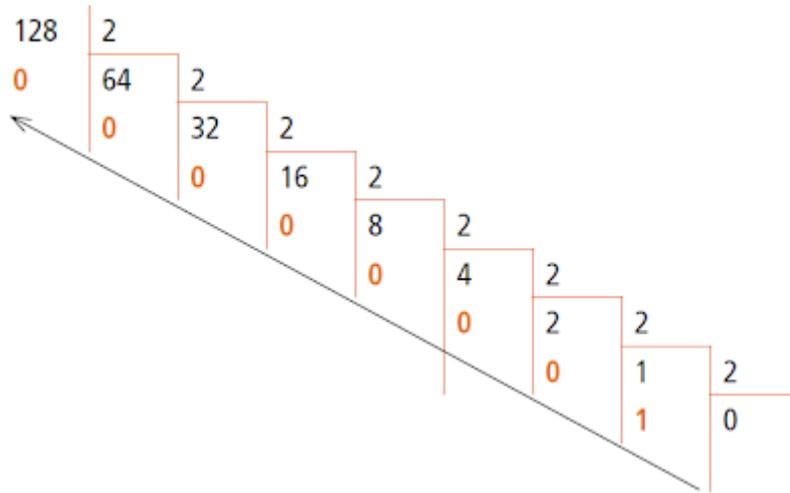
Retenues	-1	-1	-1	-1	-1	-1	-1	
	1	1	1	0	1	1	1	0
	-1	1	1	0	1	1	1	1
-1	1	1	1	1	1	1	0	1

Drôle de résultat, vous ne trouvez pas ? En fait, le résultat est négatif, puisque le deuxième nombre est plus grand que le premier. Nous reviendrons sur ce problème dans la séquence suivante en répondant à la question : comment les nombres binaires négatifs sont-ils stockés dans la mémoire de l'ordinateur ?

Solution Exercice 3

1. Dans le nombre $(40\ 04)_{10}$ Qu'est-ce qui différencie le 4 de gauche de celui de droite ? Eh bien, c'est son poids !

2.



$$(128)_{10} = (1000\ 0000)_2$$

3.

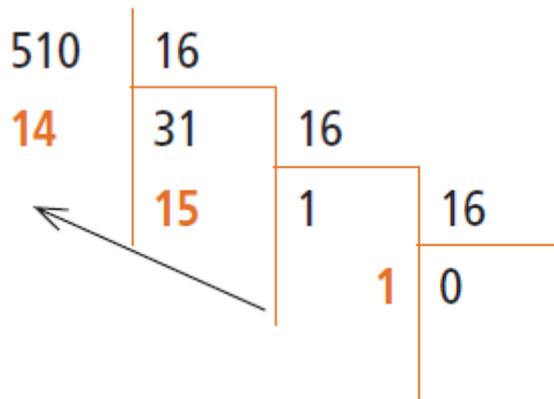
Rang	7	6	5	4	3	2	1	0
Nombre	1	1	0	0	0	1	1	0
Poids	2^7 (128) ₁₀	2^6 (64) ₁₀	2^5 (32) ₁₀	2^4 (16) ₁₀	2^3 (8) ₁₀	2^2 (4) ₁₀	2^1 (2) ₁₀	2^0 (1) ₁₀
Valeur	1 x 128	1 x 64	0 x 32 +	0 x 16 +	0 x 8 +	1 x 4 +	1 x 2 +	0 x 1

$$(1100\ 0110)_2 = (198)_{10}$$

Solution Exercice 4

1. $(A)_{16} = (10)_{10}$, $(4A)_{16} = (74)_{10}$

2.



$$(510)_{10} = (1FE)_{16}$$

3.

(1100	0110) ₂
(C	6) ₁₆

Solution Exercice 5

1.

Décimal	Hexadécimal	Binaire
54	36	11 0110
187	BB	1011 1011
141	8D	1000 1101
164	A4	1010 0100
18	12	1 0010
511	1FF	1 1111 1111
12	C	1100

2. Sur un octet, la plus petite valeur est 0, la plus grande, c'est quand tous les bits sont à 1 soit (1111 1111)₂ ce qui donne (255)₁₀.

3. Question piège en général ! Sur un octet, on peut représenter **256** valeurs différentes (n'oublions pas le 0).

4. $(75)_{10} = (100\ 1011)_2$, décalé d'une position vers la gauche, on obtient $(1001\ 0110)_2$ ce qui donne $(150)_{10}$. Conclusion décaler un nombre binaire vers la gauche revient à multiplier par deux (décaler vers la droite, revient à diviser par 2).

Avec $(200)_{10}$, on aura un problème puisqu'il occupe déjà les 8 bits ($200 \times 2 = 400$, ce qui ne rentre pas sur un octet puisque la plus grande valeur est 255).

5. Avec 14 bits, le plus grand nombre que l'on peut représenter est $2^{14} - 1 = 16\ 383$ ce qui est insuffisant. Avec 15 bits, le plus grand nombre que l'on peut représenter est $2^{15} - 1 = 32\ 767$ ce qui convient. Il faut donc 15 bits pour représenter $(18\ 450)_{10}$ en binaire.

Solution Exercice 6

1. Si chaque nombre est codé sur un octet, cela correspond à 8 bits. Puisqu'il y a 4 nombres dans une adresse IP, le tout fait 32 bits.

2. Chaque nombre est compris entre 0 et 255 soit 256 valeurs différentes. Puisqu'il y a 4 nombres dans une adresse IP, on peut former théoriquement $256 \times 256 \times 256 \times 256$ adresses, ce qui donne 4 294 967 296, ce qui correspond à 232.

3. $(172.16.1.80)_{10} = (1010\ 1100.0001\ 0000.0000\ 0001.0101\ 0000)_2$

4. $(255.255.0.0)_{10} = (1111\ 1111.1111\ 1111.0000\ 0000.0000\ 0000)_2$

5. $(1010\ 1100.0001\ 0000.0000\ 0001.0101\ 0000)_2$ ET $(1111\ 1111.1111\ 1111.0000\ 0000.0000\ 0000)_2 = (1010\ 1100.0001\ 0000.0000\ 0000.0000\ 0000)_2$

6. $(1010\ 1100.0001\ 0000.0000\ 0000.0000\ 0000)_2 = (172.16.0.0)_{10}$ En faisant un ET logique entre l'adresse et le masque, on récupère le début de l'adresse IP. Il s'agit de la partie réseau de l'adresse.

7. $(193.252.19.3)_{10} = (1100\ 0001.\ 1111\ 1100.0001\ 0011.0000\ 0011)_2$
 $(1100\ 0001.\ 1111\ 1100.0001\ 0011.0000\ 0011)_2$ ET $(1111\ 1111.1111\ 1111.0000\ 0000.0000\ 0000)_2 = (1100\ 0001.\ 1111\ 1100.0000\ 0000.0000\ 0000)_2$
 $(1100\ 0001.\ 1111\ 1100.0000\ 0000.0000\ 0000)_2 = (193.252.0.0)_{10}$

8. De toute évidence non ! $(172.16.0.0)_{10} \neq (193.252.0.0)_{10}$