Licence 1

Informatique, Développement d'Application

Cours : Introduction aux Réseaux

Séquence 2 : Aspects mathématiques des réseaux

UVS
UNIVERSITÉ VIRTUELLE DU SÉNÉGAL

1.2 - Aspects mathématiques des réseaux

1. Présentation binaire des données
2. Bits et octets
3. Système de numération décimal
4. Système de numération à base 2
5. Conversion de nombres décimaux en nombres binaires à 8 bits
6. Conversion de nombres binaires à 8 bits en nombres décimaux
7. Représentation de nombres binaires à 32 bits par une notation décimale à quatre octets avec points de séparation
8. Système hexadécimal
9. Logique booléenne ou binaire
10. Adresses IP et masques de réseau

1.2.1 - Présentation binaire des données

Les ordinateurs gèrent et stockent les données à l'aide de commutateurs électroniques pouvant prendre deux états :

- « En fonction » (ON)
- « Hors fonction » (OFF)

1.2.1 - Présentation binaire des données

Ils ne peuvent traiter que les données répondant à ce format à deux états, dit binaire. Les chiffres 0 et 1 servent à représenter les deux états possibles d'un composant électronique de l'ordinateur.

•Valeur 1 : État En fonction

•Valeur 0 : État Hors fonction

Ces valeurs sont appelées chiffres binaires ou bits.

1.2.1 - Présentation binaire des données

Code ASCII associé aux caractères A à H

FIGURES

1

2

Clavier	Code binaire
A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000

Copyright sur l'intégralité du contenu © 2003 Cisco Systems, Inc. Tous droits réservés.

1.2.1 - Présentation binaire des données

Représentation binaire des données

FIGURES

1

2

Caractère	Valeur ASCII	Binaire
0	48	00110000
9	57	00111001
A	65	01000001
Z	90	01011010
a	97	01100001
z	122	01111010

1.2.2 - Bits et octets

Unités de stockage des données

FIGURE

1

Unités	Définition	Octets*	Bits*	Exemples
Bit (b)	Chiffre binaire, égal à 1 ou 0	1	1	En fonction/Hors fonction ; Ouvert/Fermé ; +5 volts ou 0 volt
Octet (o)	8 bits	1	8	La lettre X en code ASCII
Kilo-octet (Ko)	1 kilo-octet = 1 024 octets	1000	8,000	Email type = 2 Ko Rapport de 10 pages = 10 Ko
Méga-octet (Mo)	1 méga-octet = 1 024 kilo-octets = 1 048 576 octets	1 million	8 millions	Disquettes = 1,44 Mo RAM standard = 32 Mo Cédérom = 650 Mo
Gigaoctet (Go)	1 gigaoctet = 1 024 méga-octets = 1 073 741 824 octets	1 milliard	8 milliards	Disque dur type = 40 Go ou plus
Téraoctet (To)	1 téraoctet = 1 024 gigaoctets = 1 099 511 627 778 octets	1 billion	8 billions	Quantité théorique de données transmissibles par fibre optique en une seconde

* Nombre d'octets ou de bits habituel ou approximatif

Copyright sur l'intégralité du contenu © 2003 Cisco Systems, Inc. Tous droits réservés.

1.2.2 - Bits et octets

- Le chiffre binaire 0 est représenté par une tension électrique de 0 volts.

- Le chiffre binaire 1 est représenté par une tension électrique de +5 volts.

- Les ordinateurs sont conçus pour traiter des groupes de huit bits, chaque groupe constituant un octet.

1.2.2 - Bits et octets

- Dans un ordinateur, un octet représente un emplacement de mémoire adressable unique.
- Chacun de ces emplacements correspond à une valeur ou un caractère unique de données, tel qu'un caractère du code ASCII. Les huit commutateurs susceptibles de prendre chacun la valeur 0 ou 1 peuvent donner lieu à un nombre total de 256 combinaisons.

1.2.3 - Système de numération décimall

Système de numération à base 10

FIGURE

1

Valeur des rangs	1000 100 10 1
$\text{Base}^{\text{Exposant}}$	$10^3 = 1000$ $10^2 = 100$ $10^1 = 10$ $10^0 = 1$
Nombre de symboles	10
Symboles	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Fondement	Les êtres humains ont généralement 10 doigts

Copyright sur l'intégralité du contenu © 2003 Cisco Systems, Inc. Tous droits réservés.

UVS
UNIVERSITE VIRTUELLE DU SENEGAL

1.2.3 - Système de numération décimal

- Un système de numération consiste en un ensemble de symboles dont l’utilisation est définie par des règles.
- Le système de numération est le plus largement utilisé.
- C’est le système décimal ou système à base 10.

1.2.3 - Système de numération décimal

- La base 10 utilise les dix symboles 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9, qu'elle permet de combiner pour représenter toutes les valeurs numériques possibles.

1.2.3 - Système de numération décimal

- Le système de numération décimal repose sur les puissances de 10.
- Dans un nombre, que l'on doit lire de droite à gauche, chaque chiffre est multiplié par le nombre de base 10, élevé à une certaine puissance (l'exposant).
- La puissance à laquelle on élève la base 10 dépend du rang occupé par le chiffre à gauche de la virgule décimale.

1.2.3 - Système de numération décimal

La lecture d'un nombre décimal en partant de la droite fournit les informations suivantes :

- le tout premier rang à droite (rang numéro 0) correspond à 10^0, qui est égal à 1 ;
- le deuxième rang (rang n° 1) correspond à 10^1 qui est égal à 10 ;
- le troisième rang (rang n° 2) correspond à 10^2, qui est égal à 100 ;

1.2.3 - Système de numération décimal

- ainsi de suite jusqu'au septième rang (rang n° 6), qui correspond à 10^6, c'est-à-dire à 1 000 000.
- Ces règles s'appliquent quel que soit la longueur d'un nombre.

1.2.3 - Système de numération décimal

Exemple:

- $2134 = (2 \times 10^3) +$
  $(1 \times 10^2) +$
  $(3 \times 10^1) +$
  (4×10^0)

- Cette révision du système décimal vise à faciliter la compréhension des systèmes à base 2 et à base 16, qui fonctionnent selon les mêmes règles.

UNIVERSITE VIRTUELLE DU SENEGAL

Système de numération à base 2

FIGURE

1

Valeur des rangs	128 64 32 16 8 4 2 1
Base^{Exposant}	$2^7 = 128$ $2^6 = 64$ $2^5 = 32$ $2^4 = 16$ $2^3 = 8$ $2^2 = 4$ $2^1 = 2$ $2^0 = 1$
Nombre de symboles	2
Symboles	0, 1
Fondement	Les systèmes à transistors à deux états de tension (binaire discret) sont multiples, efficaces, peu coûteux, à faible encombrement et présentent une certaine immunité aux interférences (bruit).

Copyright sur l'intégralité du contenu © 2003 Cisco Systems, Inc. Tous droits réservés.

1.2.4 - Système de numération à base 2

- Le système de numération utilisé par les ordinateurs pour reconnaître et traiter les données : le système binaire ou système à base 2.

1.2.4 - Système de numération à base 2

- Le système binaire utilise seulement deux symboles, le 0 et le 1. Dans un nombre binaire lu de droite à gauche, le rang de chaque chiffre correspond à 2, le nombre de base, élevé à la puissance (exposant) 0, puis à la puissance 1 et ainsi de suite jusqu'à la puissance 7.

1.2.4 - Système de numération à base 2

- Les valeurs respectives des rangs, de droite à gauche, sont donc les suivantes :
  - 2^0, 2^1, 2^2, 2^3, 2^4, 2^5, 2^6 et 2^7
  - ou 1, 2, 4, 8, 16, 32, 64 et 128.

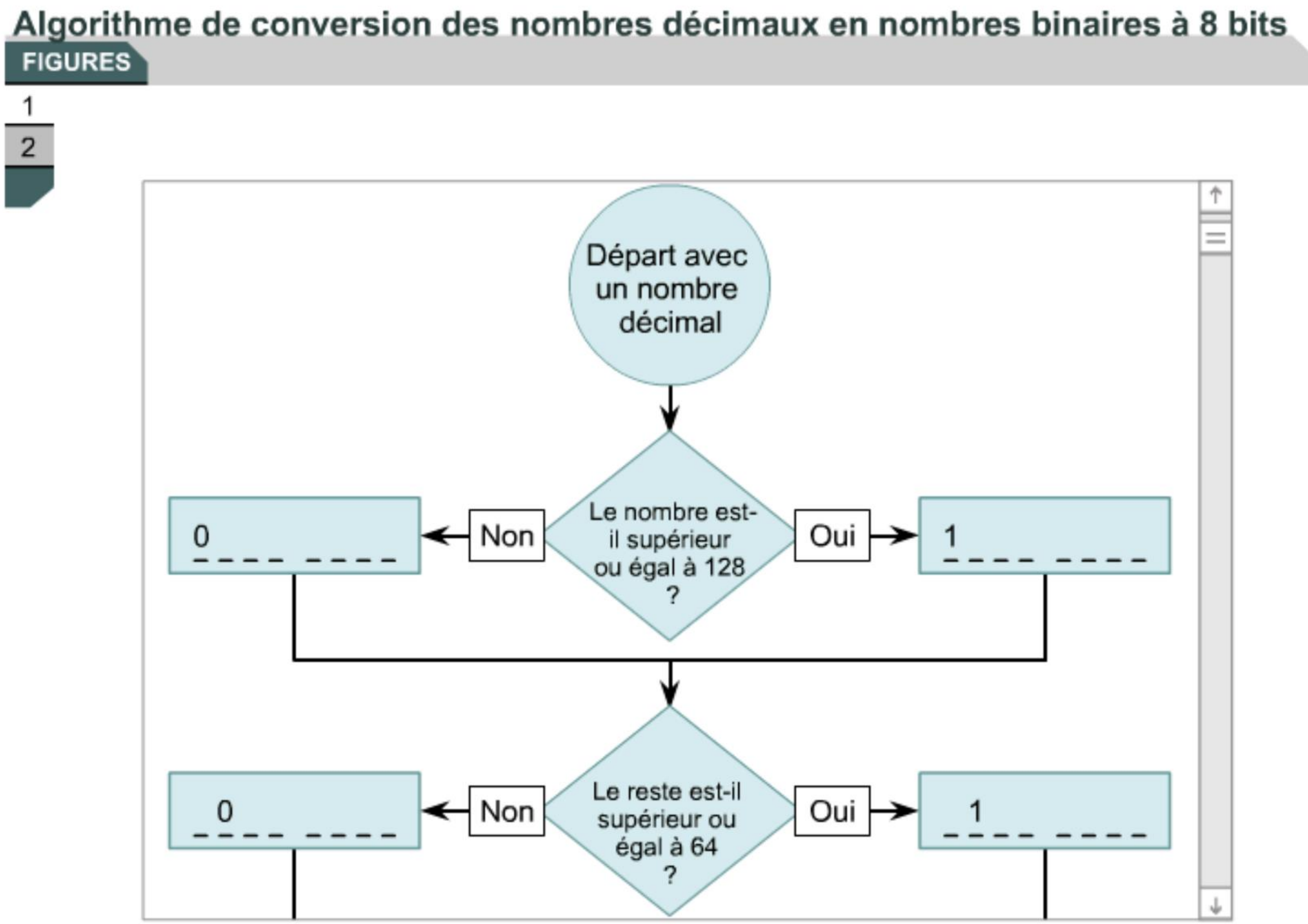
1.2.4 - Système de numération à base 2

Exemple:

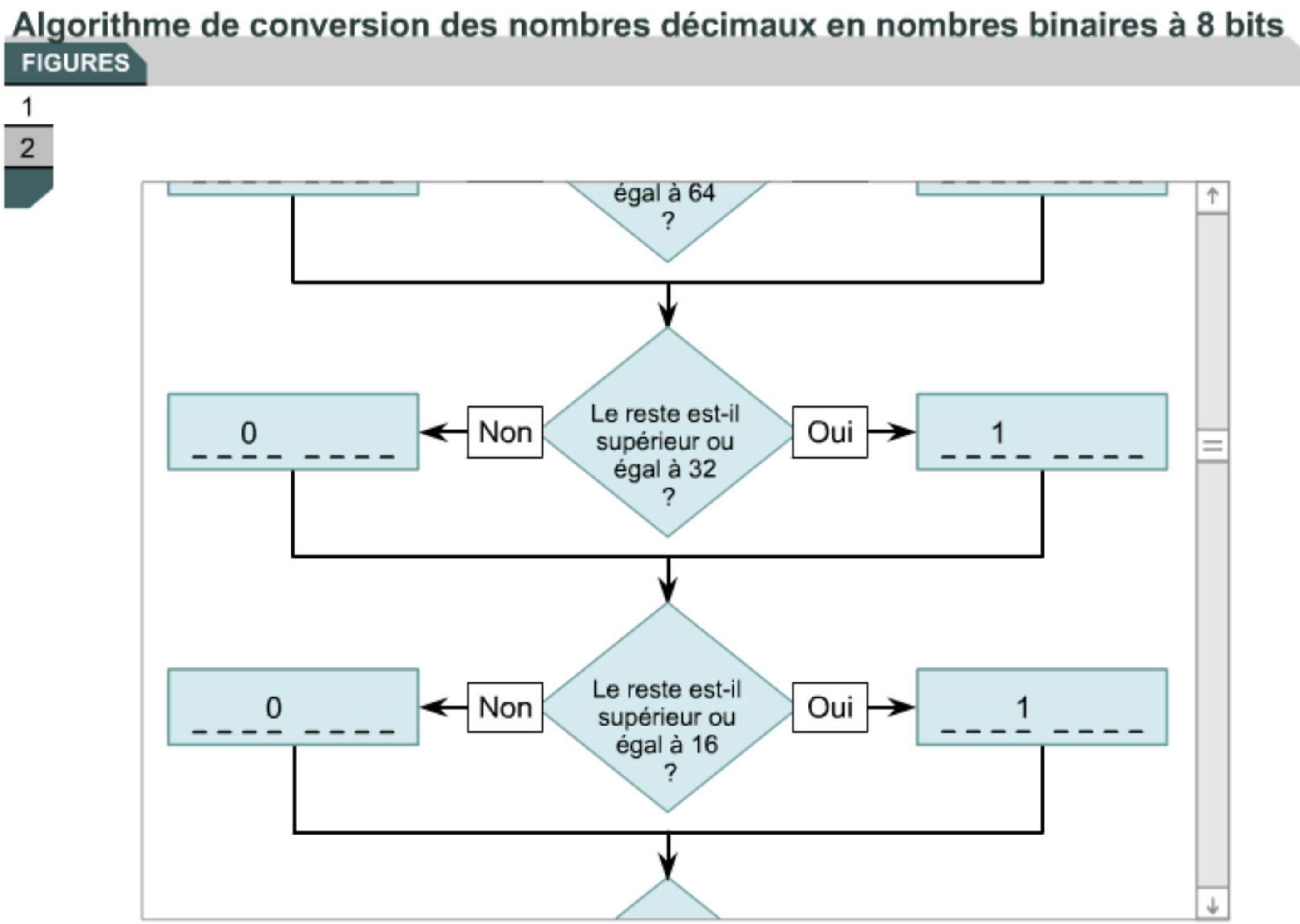
- $10110_2 = (1 \times 2^4 = 16) +$
  $(0 \times 2^3 = 0) +$
  $(1 \times 2^2 = 4) +$
  $(1 \times 2^1 = 2) +$
  $(0 \times 2^0 = 0) =$
  $22\ (16 + 0 + 4 + 2 + 0)$

Cet exemple montre que le nombre 10110 en base 2 est égal au nombre décimal 22.

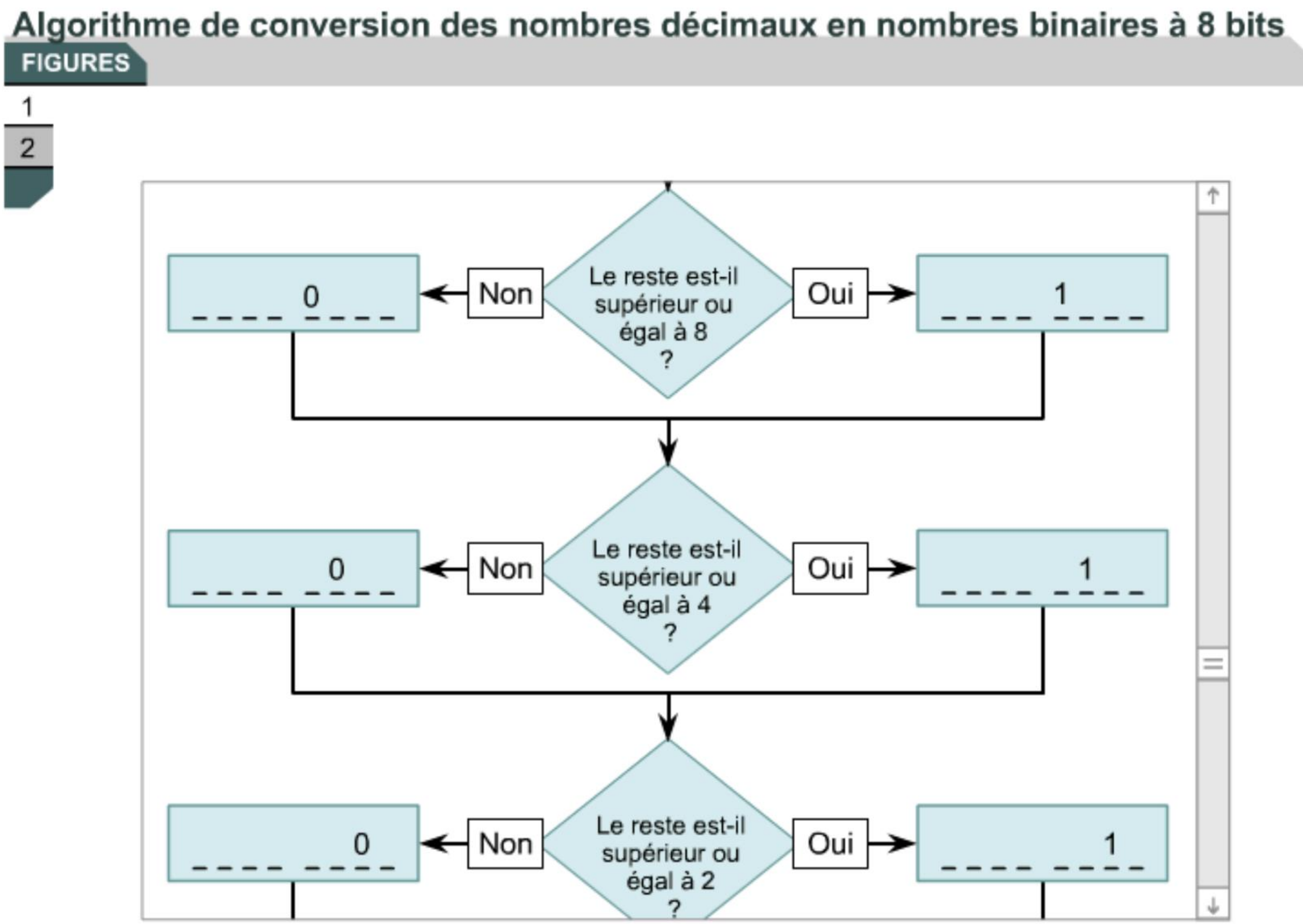
1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits



1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits

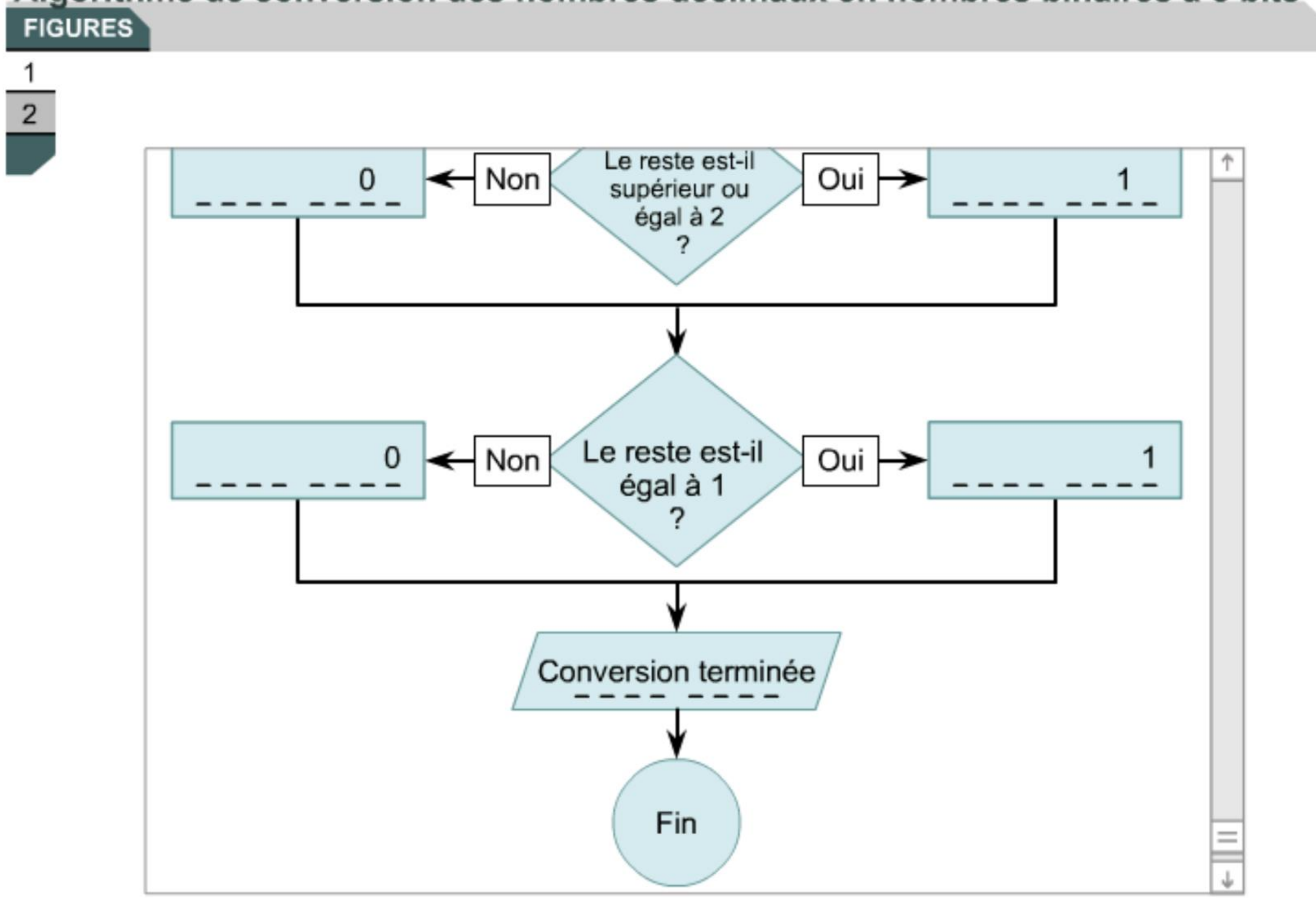


1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits



1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits

Algorithme de conversion des nombres décimaux en nombres binaires à 8 bits



1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits

Il existe plusieurs façons de convertir les nombres décimaux en nombres binaires. L’organigramme de la figure décrit une méthode parmi les différentes disponibles. Il est recommandé aux étudiants de choisir une méthode et de s’exercer à l’utiliser aussi longtemps que nécessaire pour trouver le résultat correct à chaque fois.

1.2.5 - Conversion de nombres décimaux en nombres binaires à 8 bits

1.2.5 Conversion de nombres décimaux en nombres binaires à 8 bits

Exercice de conversion

Utilisez l'exemple ci-après pour convertir le nombre décimal 168 en un nombre binaire:

- 128 est inférieur à 168 donc le bit le plus à gauche du nombre binaire est 1. 168 – 128 = 40.
- 64 n'est pas inférieur ou égal à 40 donc le deuxième bit en partant de la gauche est 0.
- 32 est inférieur à 40 donc le troisième bit en partant de la gauche est 1. 40 – 32 = 8.
- 16 n'est pas inférieur ou égal à 8 donc le quatrième bit en partant de la gauche est 0.
- 8 est égal à 8 donc le cinquième bit en partant de la gauche est 1. 8 – 8 = 0.
- Les trois derniers bits de droite sont donc tous égaux à 0.

Cet exemple montre que le nombre décimal 168 est égal au nombre 10101000 en base 2.

Travail Pratique 1.2.5 Conversion de nombres décimaux en nombres binaires

Ce TP vous permet d'apprendre et de vous entraîner à convertir des valeurs décimales en valeurs binaires.

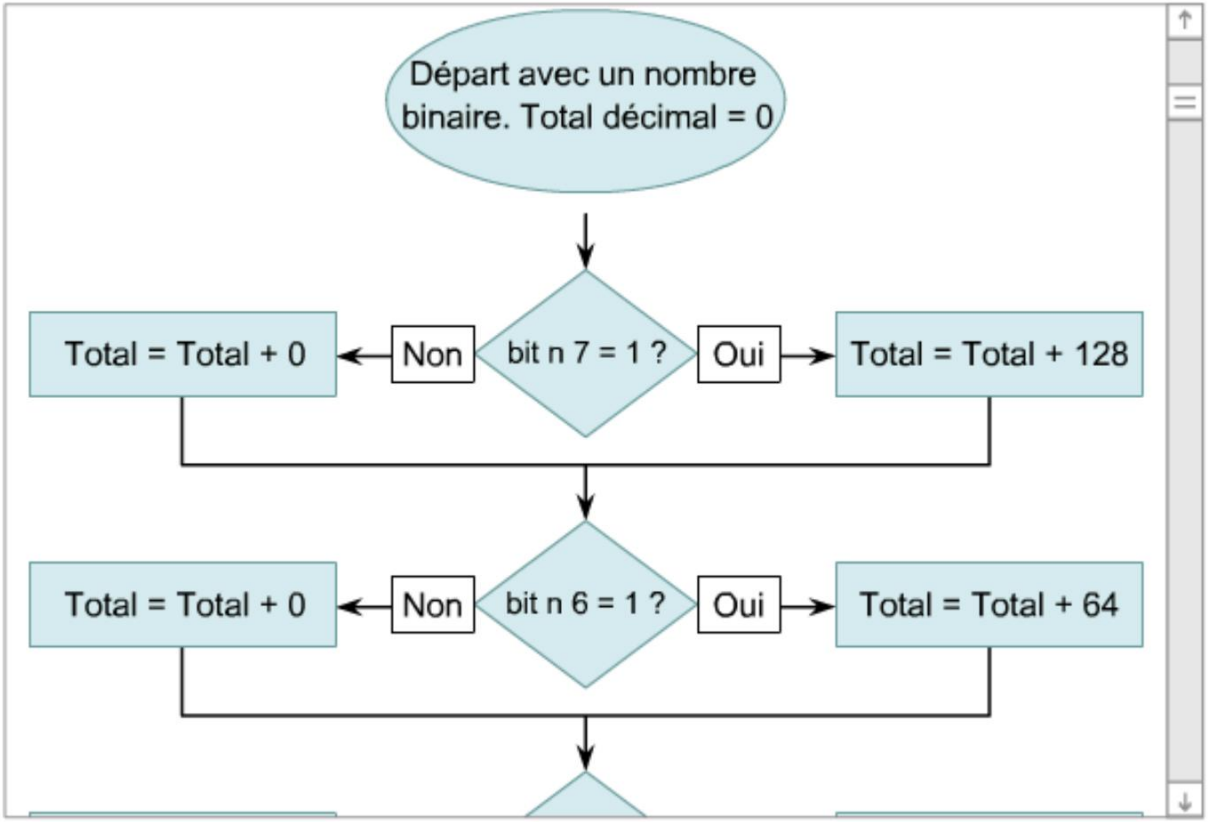
1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

Algorithme de conversion des nombres binaires à 8 bits en nombres décimaux

FIGURES

1

2



1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

Conversion des nombres binaires à 8 bits en nombres décimaux

FIGURES

1

2

Dans la zone Nombre décimal, saisissez
l'équivalent décimal du nombre binaire affiché.

Nombre binaire	Nombre décimal
00111010	58
Essayez un autre nombre	Vérifiez votre réponse

Correct

1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

- Il existe deux façons de convertir les nombres binaires en nombres décimaux. L'organigramme de la figure précédente décrit une.
- Vous pouvez également multiplier chaque chiffre binaire par le nombre de base 2, élevé à la puissance correspondant au rang de ce chiffre.

1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

Exemple:

- Convertissez le nombre binaire 01110000 en nombre décimal.

REMARQUE:

- Travaillez de droite à gauche. Gardez à l'esprit que toute valeur élevée à la puissance 0 est égale à 1.

1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

Exemple:

$$
\begin{array}{rl}
 0 \times 2^0 = & 0 \\
+\ 0 \times 2^1 = & 0 \\
+\ 0 \times 2^2 = & 0 \\
+\ 0 \times 2^3 = & 0 \\
+\ 1 \times 2^4 = & 16 \\
+\ 1 \times 2^5 = & 32 \\
+\ 1 \times 2^6 = & 64 \\
+\ 0 \times 2^7 = & 0 \\
\hline
 & 112
\end{array}
$$

1.2.6 - Conversion de nombres binaires à 8 bits en nombres décimaux

Travail Pratique 1.2.6 Conversion de nombres binaires en nombres décimaux

Ce TP vous permet d'apprendre et de vous entraîner à convertir des valeurs binaires en valeurs décimales.

1.2.7 - Représentation de nombres binaires à 32 bits par une notation décimale à quatre octets avec points de séparation

Notation entière avec points de séparation

FIGURES

1
2
3

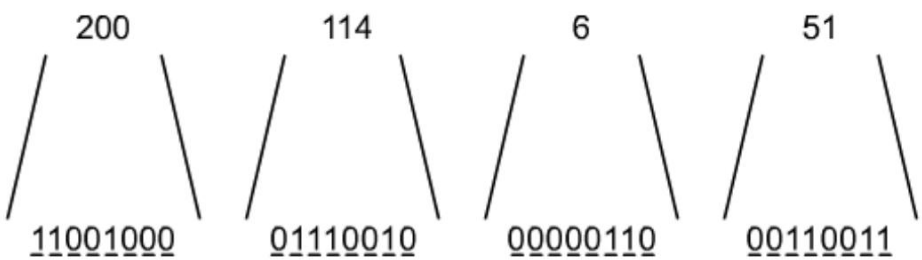
Binaire	11001000		01110010		00000110		00110011
Décimal	200	.	114	.	6	.	51
	nombre	point	nombre	point	nombre	point	nombre

1.2.7 - Représentation de nombres binaires à 32 bits par une notation décimale à quatre octets avec points de séparation

Conversion de nombres en notation entière avec points de séparation en nombres binaires

FIGURES

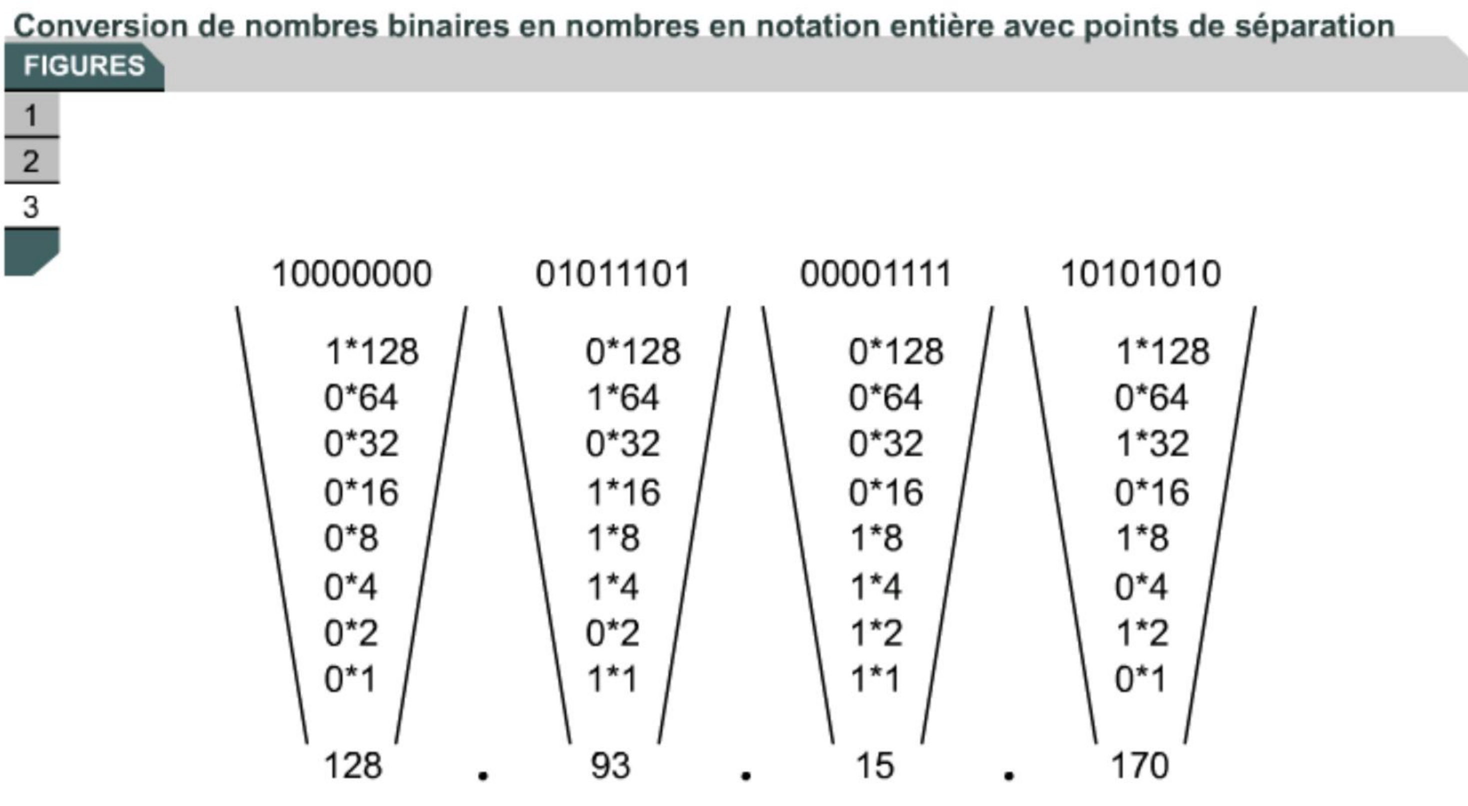
1
2
3



Conversion d'un nombre en notation entière avec points de séparation en un nombre binaire.

Copyright sur l'intégralité du contenu © 2003 Cisco Systems, Inc. Tous droits réservés.

1.2.7 - Représentation de nombres binaires à 32 bits par une notation décimale à quatre octets avec points de séparation



Alassane.Diop@uadb.edu.sn

1.2.8 - Système hexadécimal

Systèmes de numération binaire et hexadécimal

FIGURES

1
2
3
4
5

Décimal	Binaire	Hexadécimal
0	00000000	00
1	00000001	01
2	00000010	02
3	00000011	03
4	00000100	04
5	00000101	05
6	00000110	06
7	00000111	07
8	00001000	08
9	00001001	09
10	00001010	0A
11	00001011	0B
12	00001100	0C
13	00001101	0D
14	00001110	0E
15	00001111	0F
16	00010000	10
32	00100000	20
64	01000000	40
128	10000000	80
255	11111111	FF

1.2.8 - Système hexadécimal

Systèmes de numération binaire et hexadécimal

FIGURES

1
2
3
4
5

Binaire	Hexadécimal	Binaire	Hexadécimal
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

1.2.8 - Système hexadécimal

Systèmes de numération binaire, hexadécimal et décimal

FIGURES

1
2
3
4
5

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

1.2.8 - Système hexadécimal

Exemple de conversion du format binaire au format hexadécimal

FIGURES

1
2
3
4
5

Conversion d'un nombre binaire en nombre hexadécimal

100100100010111110111110111001001

Équivaut à :

0001 0010 0100 0101 1111 0111 1101 1100 1001

Équivaut à :

1 2 4 5 F 7 D C 9

D'où :

100100100010111110111110111001001 au format binaire

= 1245F7DC9 au format hexadécimal

1.2.8 - Système hexadécimal

Exemple de conversion du format hexadécimal au format binaire

FIGURES

1
2
3
4
5

Conversion d'un nombre hexadécimal en nombre binaire

0x2102

Équivaut à :

2	1	0	2
0010	0001	0000	0010

D'où :

2102 au format hexadécimal équivaut à : 0010 0001 0000 0010 au format binaire

1.2.8 - Système hexadécimal

Le système de numération hexadécimal ou système à base 16 est couramment utilisé pour représenter les nombres binaires sous une forme plus lisible. Les ordinateurs exécutent tous les calculs en mode binaire. Toutefois, il existe plusieurs situations dans lesquelles la valeur binaire renvoyée par un ordinateur est exprimée sous sa forme hexadécimale pour une lecture plus facile.

Ainsi, le registre de configuration des routeurs Cisco requiert souvent des conversions de valeurs hexadécimales en valeurs binaires et inversement. Les routeurs Cisco possèdent un registre de configuration d'une longueur de 16 bits. Un nombre binaire de 16 bits peut donc être représenté comme un nombre hexadécimal à quatre chiffres. Par exemple, le nombre binaire 0010000100000010 est égal à 2102 dans le système hexadécimal. Les nombres hexadécimaux ont souvent comme préfixe 0x. On écrira par exemple le nombre hexadécimal 2102 de la façon suivante : 0x2102.

Tout comme les systèmes binaire et décimal, la notation hexadécimale repose sur l'utilisation de symboles, de puissances et de rangs. Les symboles utilisés sont les chiffres 0 à 9 et les lettres A à F.

1.2.8 - Système hexadécimal

Pour représenter n'importe quelle combinaison de quatre chiffres binaires, il suffit d'un seul symbole hexadécimal, contre un à deux symboles décimaux. La représentation d'une combinaison de huit chiffres binaires requiert deux chiffres en format hexadécimal, tandis qu'elle peut en nécessiter trois dans le système décimal. Par ailleurs, la représentation de quatre bits par un ou deux chiffres décimaux peut être source de confusion. Par exemple, le nombre binaire à 8 bits 01110011 est égal à 115 après conversion vers le système décimal. Dans ce cas, la représentation décimale est-elle 11 et 5 ou 1 et 15 ? Si l'on considère 11 et 5, le nombre binaire représenté est alors 1011 0101, qui n'est pas le nombre initial converti. En revanche la représentation hexadécimale est 1F, qui donne toujours 00011111 après reconversion en base 2.

Un nombre binaire de huit bits peut être représenté par deux chiffres hexadécimaux. Ceci réduit les risques de confusion liés à la lecture de longues chaînes de nombres binaires et l'espace requis pour écrire les nombres binaires. Rappelez-vous qu'il est possible d'utiliser les caractères 0x pour désigner une valeur hexadécimale. Le nombre hexadécimal 5D peut donc s'écrire 0x5D.

1.2.8 - Système hexadécimal

Travail Pratique 1.2.8
Conversions hexadécimales

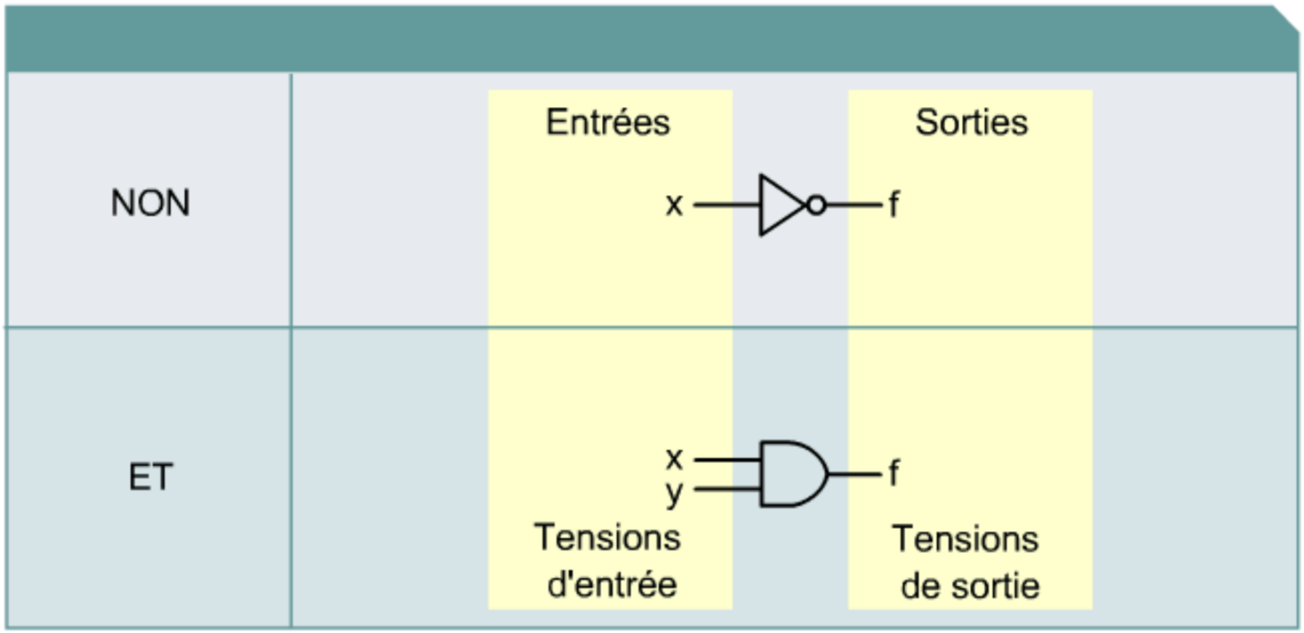
Ce TP vous permet d'apprendre à convertir des valeurs hexadécimales en valeurs décimales et binaires.

1.2.9 - Logique booléenne ou binaire

Portes logiques

FIGURES

1
2
3
4



Les portes logiques acceptent une ou deux valeurs, ou tensions, d'entrée.

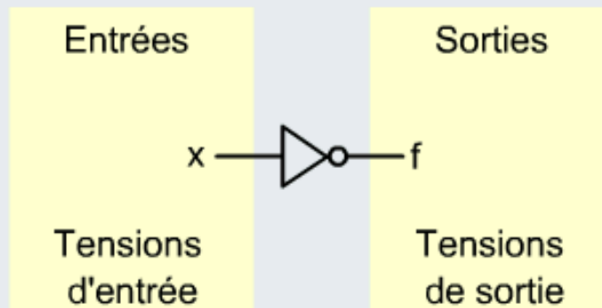
1.2.9 - Logique booléenne ou binaire

Portes logiques utilisant l'opération booléenne NON

FIGURES

1
2
3
4

NON



x	f
0	1
1	0

Si x égale 1 alors f égale 0. Sinon, f égale 1.

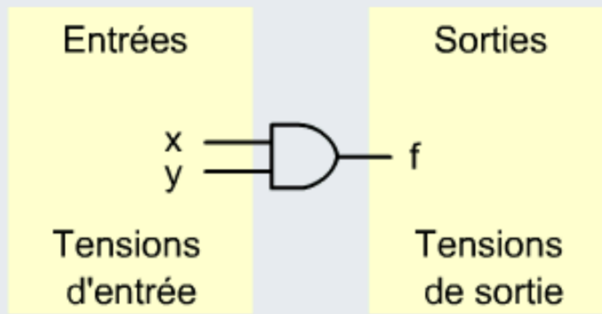
1.2.9 - Logique booléenne ou binaire

Portes logiques utilisant l'opération booléenne ET

FIGURES

1
2
3
4

ET



x	y	f
0	0	0
0	1	0
1	0	0
1	1	1

Si x égale 1 et y égale 1 alors f égale 1. Sinon, f égale 0.

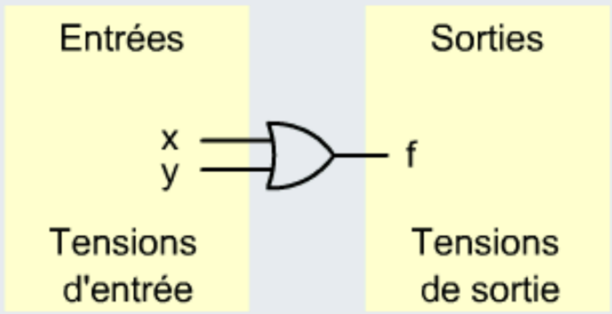
1.2.9 - Logique booléenne ou binaire

Portes logiques utilisant l'opération booléenne OU

FIGURES

1
2
3
4

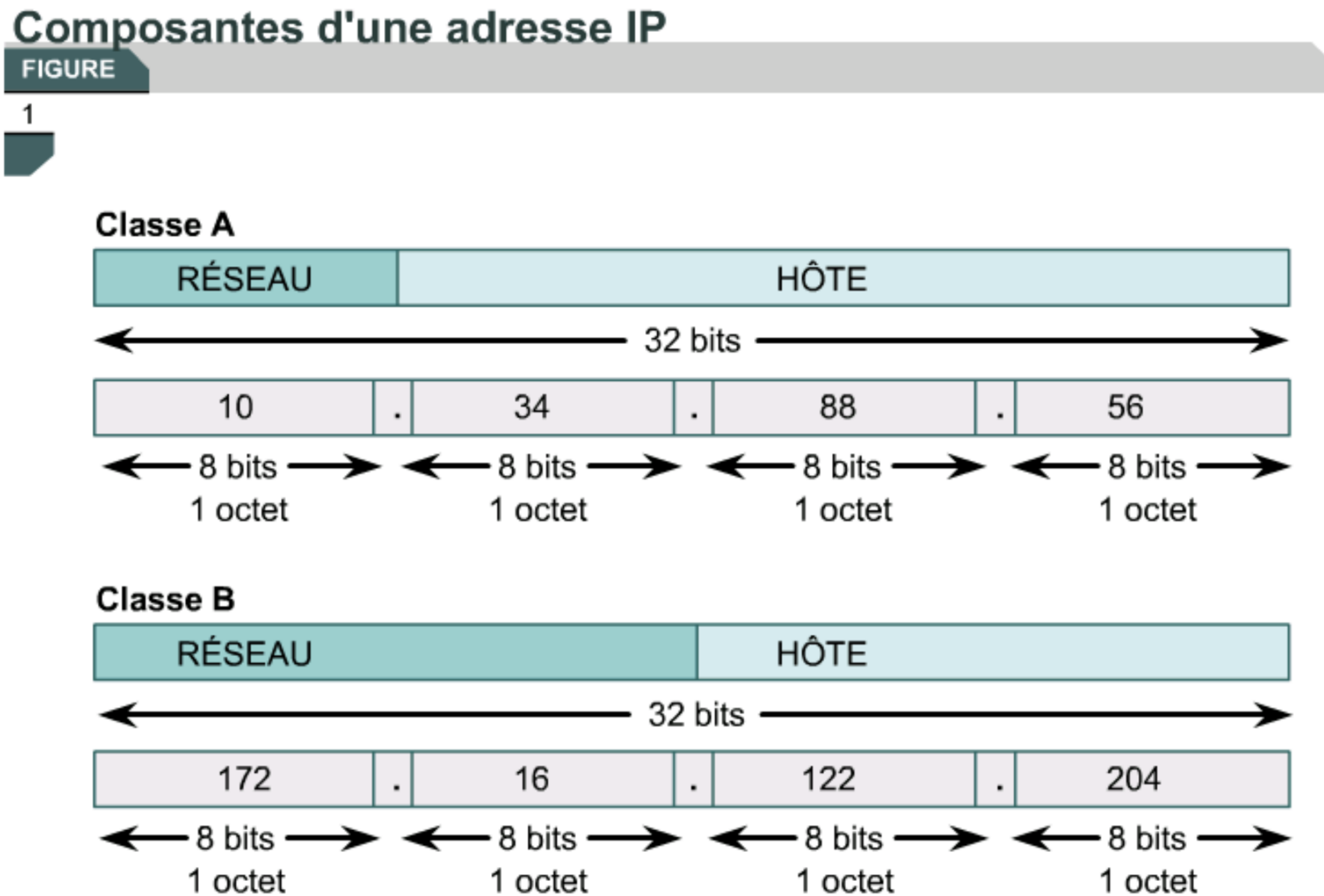
OU



x	y	f
0	0	0
0	1	1
1	0	1
1	1	1

Si x égale 1 ou y égale 1 alors f égale 1. Sinon, f égale 0.

1.2.10 - Adresses IP et masques de réseau



1.2.10 - Adresses IP et masques de réseau

Dans l'adresse IP affectée à un ordinateur inclus dans un réseau, plusieurs bits situés à gauche du numéro IP de 32 bits servent à représenter le réseau. Le nombre de bits réservés à cet effet dépend de la classe de l'adresse. Les bits restants de l'adresse IP à 32 bits identifient un ordinateur particulier sur le réseau. Le terme « hôte » est employé pour désigner un ordinateur. L'adresse IP de chaque ordinateur se compose donc d'une partie réseau et d'une partie hôte.

Pour définir la décomposition de l'adresse IP à 32 bits d'un ordinateur, un second numéro de 32 bits, appelé masque de sous-réseau, est utilisé. Ce masque fournit un guide pour l'interprétation de l'adresse IP. Il indique combien de bits sont réservés à l'identification du réseau dont fait partie l'ordinateur. La partie gauche du masque de sous-réseau est formée d'une série de 1 successifs. Tous les bits du masque qui correspondent à l'adresse du réseau ont la valeur 1, tandis que le reste du masque comporte des zéros. Les bits du masque de sous-réseau qui portent la valeur 0 identifient l'hôte.

1.2.10 - Adresses IP et masques de réseau

Les exemples ci-après sont des masques de sous-réseau.

11111111000000000000000000000000 ou, en notation entière avec points de séparation, 255.0.0.0

11111111111111110000000000000000 ou, en notation entière avec points de séparation, 255.255.0.0

Dans le premier exemple, les huit premiers bits en partant de la gauche représentent la partie réseau de l’adresse et les 24 autres bits, la partie hôte. Dans le second exemple, les 16 premiers bits représentent la partie réseau de l’adresse et les 16 autres bits, la partie hôte.

L’adresse IP 10.34.23.134 correspond, en représentation binaire, à :

00001010.00100010.00010111.10000110

1.2.10 - Adresses IP et masques de réseau

L'opération booléenne ET appliquée à l'adresse IP 10.34.23.134 et au masque de sous-réseau 255.0.0.0 permet d'obtenir l'adresse du réseau dont fait partie l'hôte:

00001010.00100010.00010111.10000110

11111111.00000000.00000000.00000000

00001010.00000000.00000000.00000000

Après conversion en notation entière avec points de séparation, cette adresse devient 10.0.0.0 et représente la partie réseau de l'adresse IP lorsque le masque 255.0.0.0 est utilisé.

L'opération booléenne ET appliquée à l'adresse IP 10.34.23.134 et au masque de sous-réseau 255.255.0.0 permet d'obtenir l'adresse du réseau dont fait partie l'hôte:

1.2.10 - Adresses IP et masques de réseau

00001010.00100010.00010111.10000110

11111111.11111111.00000000.00000000

00001010.00100010.00000000.00000000

Après conversion en notation entière avec points de séparation, cette adresse devient 10.34.0.0 et représente la partie réseau de l'adresse IP lorsque le masque 255.255.0.0 est utilisé.

Ceci illustre brièvement la signification d'un masque de sous-réseau pour une adresse IP. La réalisation d'activités supplémentaires concernant les adresses IP permettra de comprendre plus précisément l'utilité de tels masques. Au stade actuel, assurez-vous seulement de comprendre la notion de masque.