SYSTEMES DE NUMERATION ET CODES

-Bien que le système de numération binaire soit le plus important de ceux utilisés dans les circuits numériques, il ne faut pour autant pas négliger l'importance des autres systèmes (décimal, octal, hexadécimal).

-Dans un système numérique, il peut arriver que trois ou quatre de ces systèmes de numération cohabitent, d'où l'importance de pouvoir convertir un système dans un autre.

-Par exemple, lorsque nous composons un nombre décimal sur notre calculatrice (ou sur le clavier de notre ordinateur), les circuits internes convertissent ce nombre décimal en une valeur binaire.

-De même, notre calculatrice (ou notre ordinateur) calcule la réponse à un problème au moyen du système binaire, puis convertit cette réponse en des valeurs décimales avant de les afficher.

-Voyons donc comment s’effectuent ces conversions, avant de découvrir d’autres codes binaires utilisés pour représenter divers genres d’information.

1. Conversion binaire – décimal

Tout nombre binaire peut être transformé en son équivalent décimal en additionnant les poids des diverses positions où se trouve une valeur 1.

Exemples:

1	1	0	1	1	Binaire
2^4	2^3	2^2	2^1	2^0	$= 16+8+0+2+1= (27)_{10}$

$11011_2 = 27_{10}$

10110101_2

1	0	1	1	0	1	0	1
2^7	0	2^5	2^4	0	2^2	0	2^0

$=128+32+16+4+1$

$=181_{10}$

2- Conversion décimal – binaire

Une méthode convenant bien aux petits nombres consiste à exprimer le nombre décimal comme une somme de puissance de 2, puis à inscrire des 1 et des 0 vis –à – vis des positions binaires appropriées.

Exemples:

$45_{10} = 32 + 8 + 4 + 1 = 2^5 + 0 + 2^3 + 2^2 + 0 + 2^0$

$= 1\ 0\ 1\ 1\ 0\ 1_2$

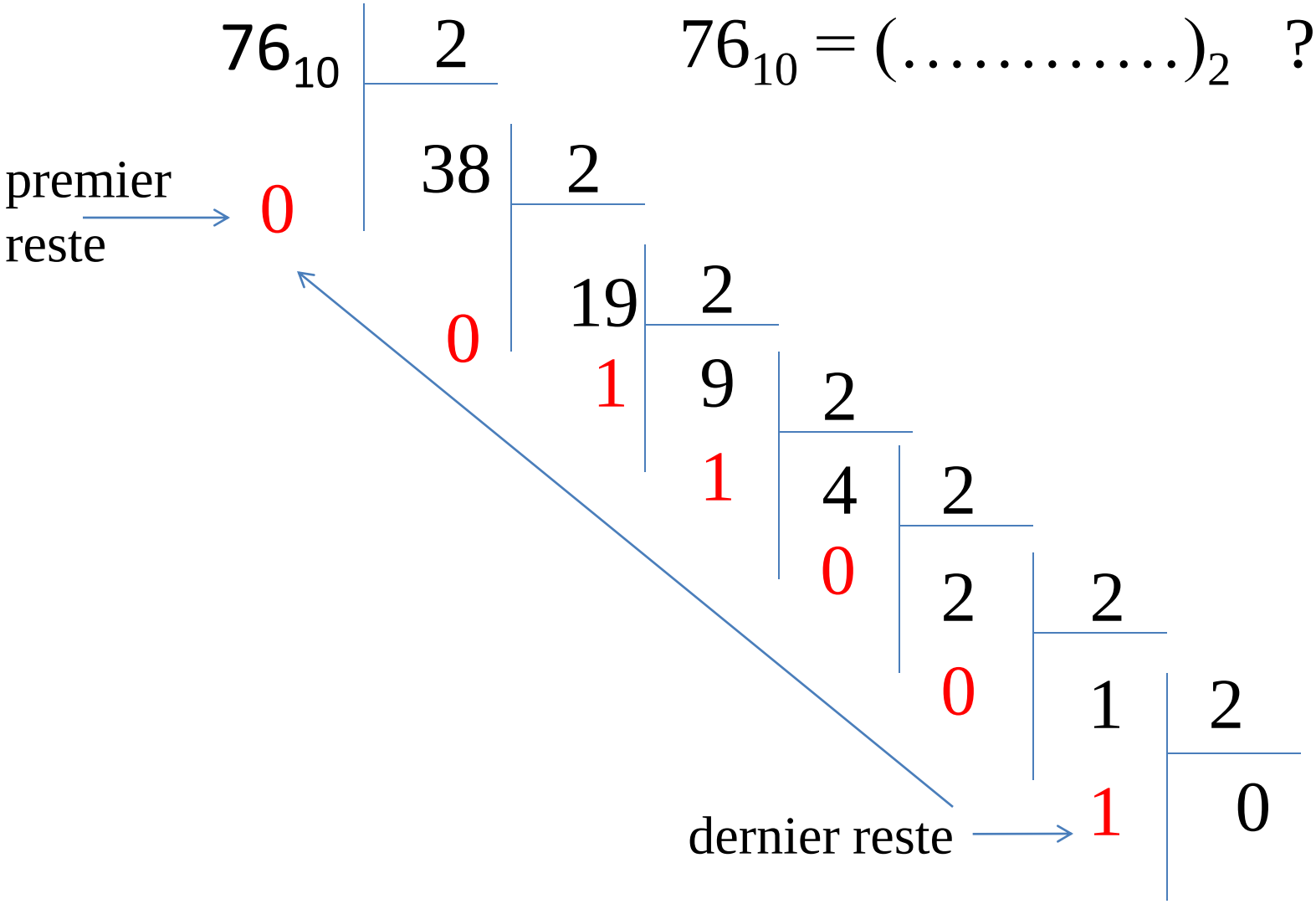
$25_{10} = 16 + 8 + 1 = 2^4 + 2^3 + 0 + 0 + 2^0$

$= 1\ \ 1\ 0\ 0\ 1_2$

Pour les grands nombres décimaux, la méthode consiste à répéter la division par 2 du nombre décimal et à reporter les restes pour chaque division jusqu'à ce que le quotient soit 0.

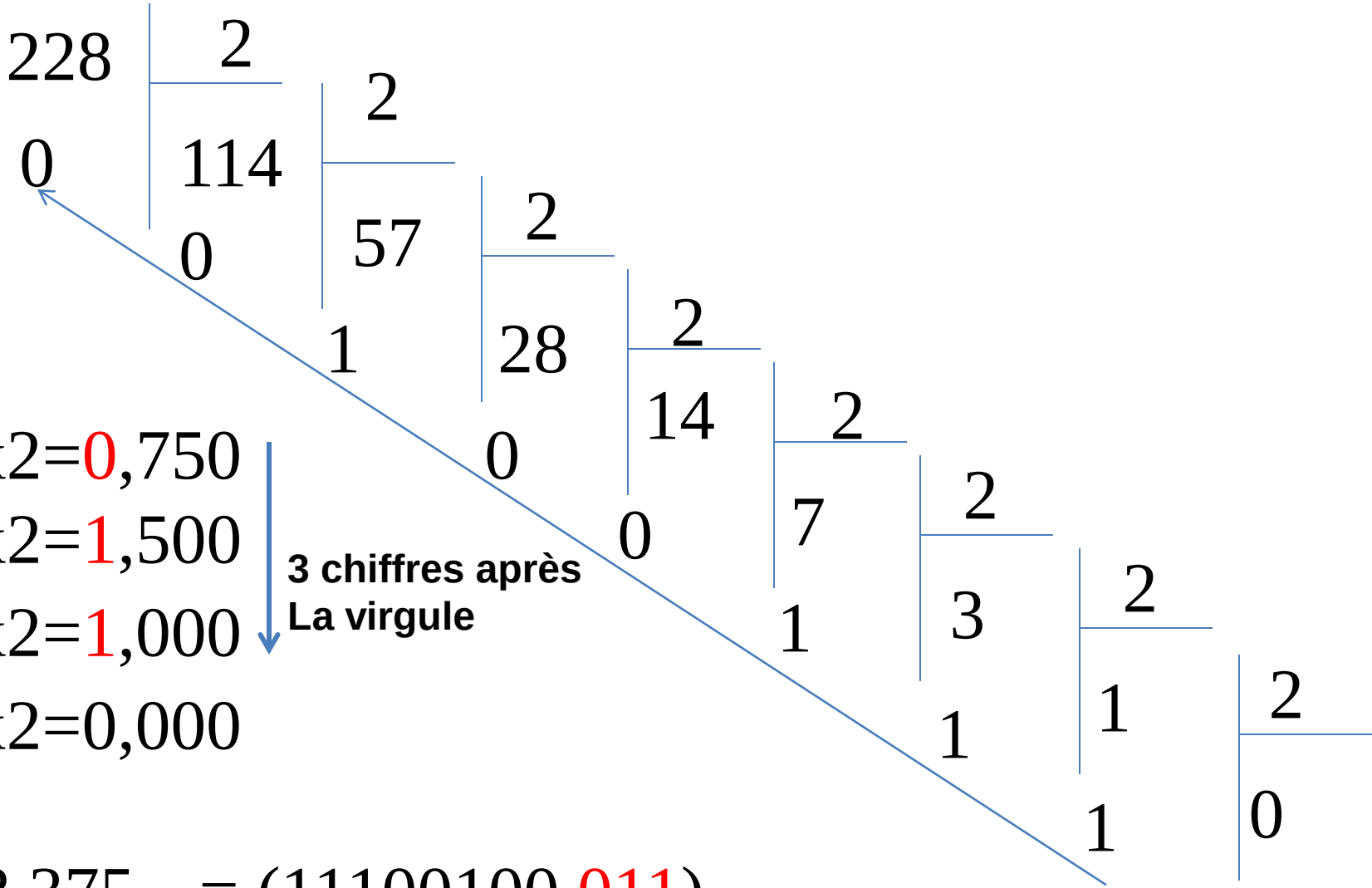
Le nombre binaire résultant s'obtient en écrivant le premier reste à la position du bit de poids le plus faible et le dernier reste à la position du bit de poids le plus fort.

Exemples: Effectuer les conversions suivantes



d'où $76_{10} = 1001100_2$

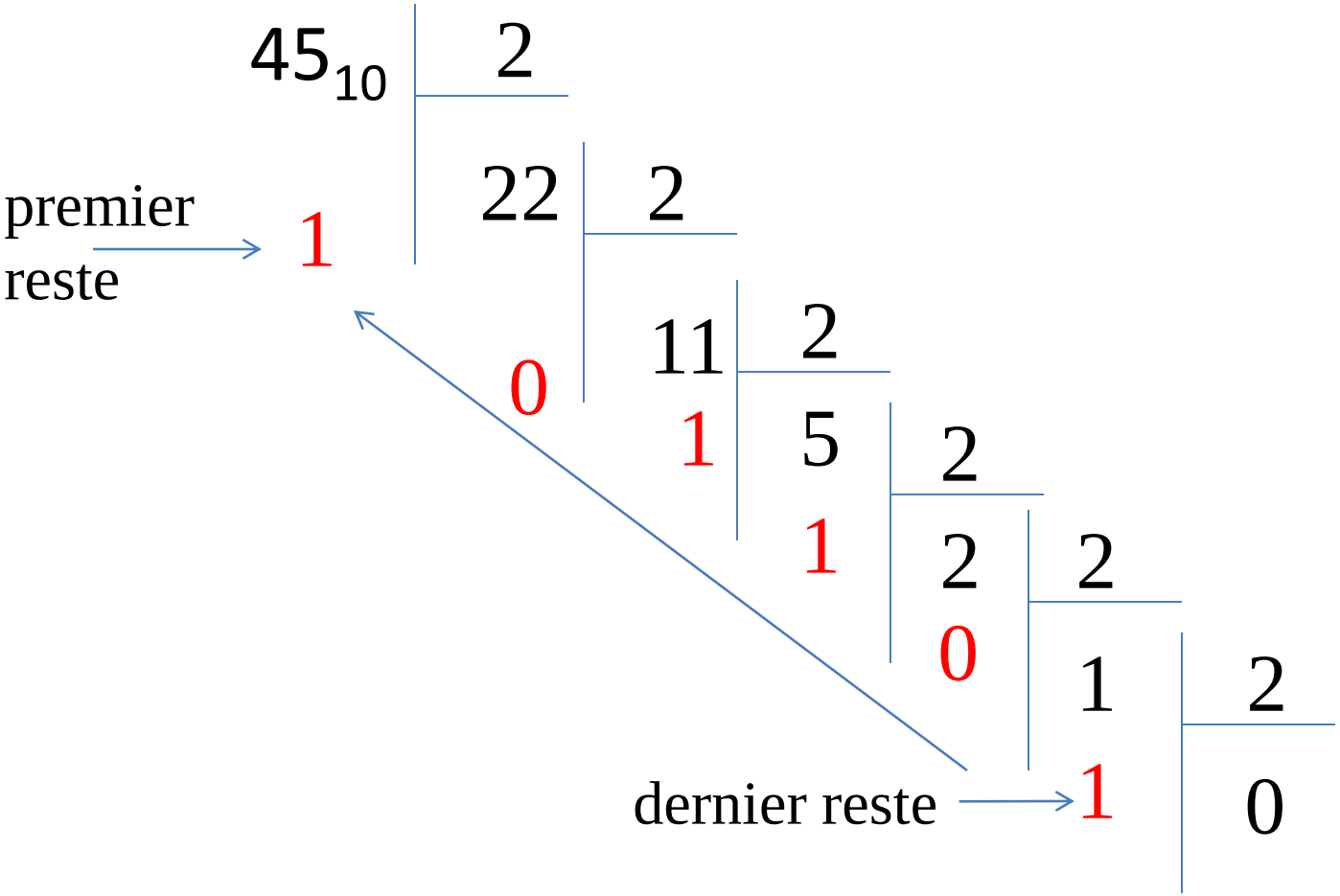
$228{,}375_{10} = (\ldots\ldots\ldots\ldots)_2$?



0, 375x2=0,750
0, 750x2=1,500
0, 500x2=1,000
0, 000x2=0,000

3 chiffres après La virgule

$228{,}375_{10} = (11100100{,}011)_2$



d'où $45_{10} = 101101_2$

3-Système de numération octal

C’est le système de numération à base 8, utilisé dans l’ordinateur numérique.

a. Conversion octal- décimal:

L’équivalent décimal d’un nombre octal est obtenu en multipliant chaque chiffre octal par son poids positionnel et en faisant la somme.

Exemples:

$372_8 = 3x(8^2) + 7x(8^1) + 2x(8^0)$

$= 3x64+7x8+2x1 = 250_{10}$

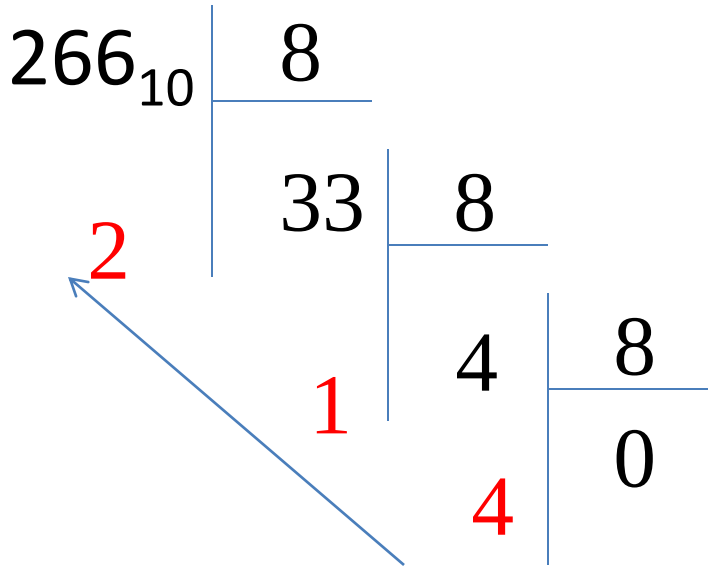
$$24{,}6_8 = 2x(8^1) + 4x(8^0) + 6x(8^{-1})$$

$$= 20{,}75_{10}$$

b. Conversion décimal – octal:

On utilise la méthode de la répétition de la division, cette fois, par 8.

Exemple:



d'où $266_{10} = 412_8$

c. Conversion octal – binaire:

Elle s'effectue en transformant chaque chiffre du nombre octal en son équivalent binaire de trois chiffres. Résumons dans le tableau ci-dessous les huit symboles octaux exprimés en binaire:

Chiffre octal	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Equivalent binaire	000	001	010	011	100	101	110	111

Exemples:

$472_8 = 100111010_2$

$5431_8 = 101100011001_2$

d. Conversion binaire – octal:

On effectue la démarche inverse du cas précédent. On effectue des groupes de trois bits du nombre binaire en partant du chiffre de poids le plus faible, puis on convertit ces triplets en leur équivalent octal.

• Exemple 1 :

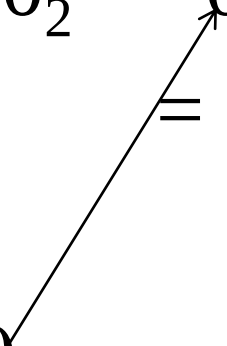
100111010_2 binaire

= 4 7 2_8 octal

- Exemple 2 :

11010110_2 = 011 010 110_2 binaire

= 3 2 6_8 octal



Ajout d'un 0

NB: Si le nombre binaire ne forme pas un nombre juste de groupes de trois, on peut ajouter un ou deux zéros à gauche du bit de poids le plus fort pour former le dernier triplet.

e. Comptage en octal:

Le chiffre le plus élevé du système octal est 7, de sorte que lorsqu'on compte au moyen du système de numération octal, on commence à 0 et on progresse jusqu'à 7, puis on revient à 0 et on augmente de 1 le chiffre immédiatement supérieur:

Exemples:

a. 65, 66, 67, 70, 71

b. 275, 276, 277, 300

Ainsi; quand on a N chiffres octaux, on peut compter jusqu'au nombre $8^N - 1$; par exemple, avec trois chiffres octaux, on peut compter de 000_8 à 777_8 , ce qui donne un total de $8^3 = 512$ nombres octaux différents, et $777_8 = 512\text{-}1 = 511_{10}$

4. Système de numération hexadécimal

C’est le système de numération à base 16 comportant 16 symboles: les dix chiffres de 0 à 9, plus les 6 lettres A, B, C, D, E, F.

Résumons dans le tableau ci-dessous les rapports entre les systèmes hexadécimal, décimal et binaire:

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

a. Conversion hexadécimal – décimal:

L’ équivalent décimal d’un nombre hexadécimal est obtenu en multipliant chaque chiffre hexadécimal par son poids positionnel et en faisant la somme.

Exemples:

- $356_{16} = 3x(16^2) + 5x(16^1) + 6x(16^0)$

  $= 768 + 80 + 6 = 854_{10}$

- $2AF_{16} = 2x(16^2) + 10x(16^1) + 15x(16^0)$

  $= 512 + 160 + 15 = 687_{10}$

b. Conversion décimal- hexadécimal:

On utilise la méthode de la répétition de division, ici, par 16.

Exemples:

- 214_{10} | 16
  6 | 13 | 16
  13 | 0

d'où $214_{10} = D6_{16}$

- 687_{10} | 16
  15 | 42 | 16
  10 | 2 | 16
  2 | 0

d'où $687_{10} = 2AF_{16}$

C. Conversion hexadécimal – binaire

Elle s'effectue en transformant chaque chiffre du nombre hexadécimal en son équivalent binaire de quatre bits.

Exemples:

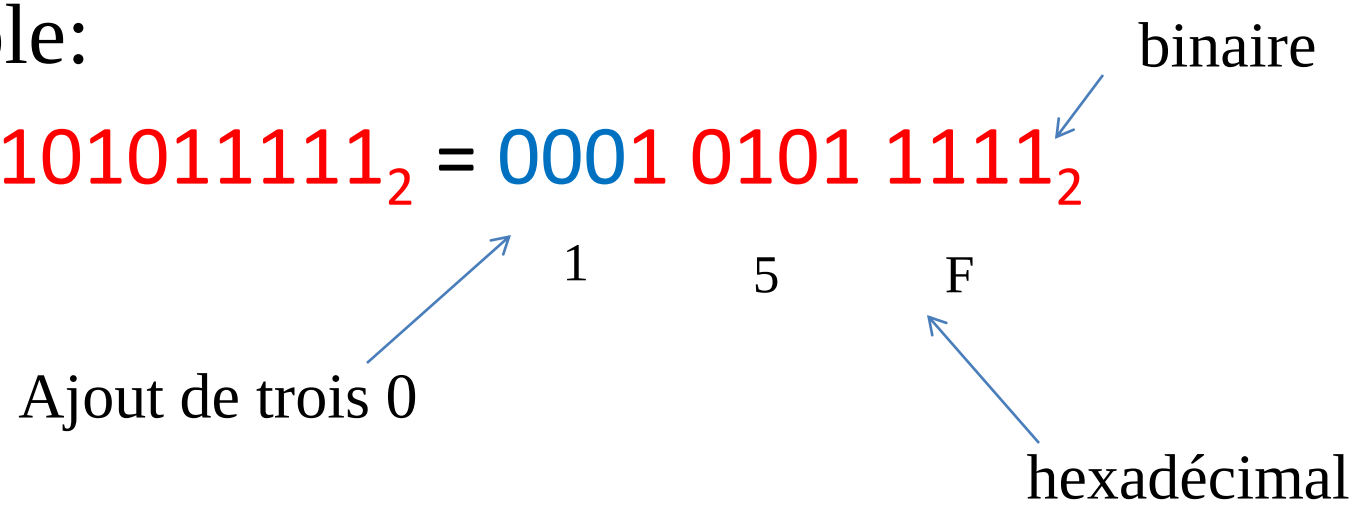
- $1A7_{16} = 0001\ 1010\ 0111_2$
- $2AF_{16} = 0010\ 10101111_2$

d. Conversion binaire – hexadécimal

Comme le système de numération octal, le nombre binaire est divisé ici en groupes de quatre bits puis on substitue à chaque groupe

son chiffre hexadécimal équivalent. Au besoin, on ajoute des zéros à gauche pour obtenir un dernier groupe de quatre bits.

Exemple:



$$101011111_2 = 15F_{16}$$

Systèmes de numération

Techniques de conversion

- Techniques pour convertir $(N)_b$ entre systèmes de numérotation bin-dec-hex:

Type de conversion	Technique de conversion
binaire → décimal	Somme pondérée des contributions
hexadécimal → décimal	
décimal → binaire	Division par la base
décimal → hexadécimal	
binaire → hexadécimal	Substitution hex-bits
hexadécimal → binaire	

e. Comptage en hexadécimal

Le comptage s'effectue en faisant croître la valeur dans une position du nombre par pas de 1 depuis 0 jusqu'à F; quand le chiffre dans une position est F, le chiffre suivant dans cette position est 0, et le chiffre immédiatement à gauche est augmenté de 1.

Exemples:

a. 35,36,37,38,39,3A,3B, 3C,3D,3E, 3F,40,41,42,…
b. 5A8,5A9,5AA,5AB,5AC,5AD,5AE,5AF,5B0,5B1, 5B2,…

5. Code DCB (Décimal Codé Binaire)

Le **codage** consiste à faire correspondre à des nombres, des lettres ou des mots un groupe spécial de symboles, groupe appelé **code.**

Quand on fait correspondre à un nombre décimal son équivalent binaire, on dit qu'on fait un **codage binaire pur.**

Nous savons que pour les grands nombres, les conversions de ce genre peuvent être longues et laborieuses. C'est pour cette raison qu'on utilise dans certaines situations un codage de nombres décimaux combinant certaines caractéristiques

du système binaire et du système décimal.

a.Code décimal codé binaire (DCB)

Ce code est obtenu en représentant chaque chiffre d'un nombre décimal par son équivalent binaire. Le plus élevé des chiffres décimaux étant 9, il faut donc 4 bits pour coder les chiffres (le code binaire de 9 est 1001).

Exemples:

- 7 8 5 décimal

  0111 1000 0101 DCB

- 9 8 2 décimal

  1001 1000 0010 DCB

NB: Le code DCB établissant une correspondance entre chaque chiffre d'un nombre décimal et un nombre binaire de 4 bits, il est évident que seuls les groupes binaires 0000 à1001 sont utilisés. Le code DCB ne fait donc pas usage des groupes 1010 à 1111.

Si l'une de ces combinaisons **inadmissibles** apparait , c'est généralement le signe qu'une erreur s'est produite.

b. Comparaison entre code DCB et nombre binaire:

Notons d'abord que le code DCB n'est pas un autre système de numération comme les systèmes octal, décimal, ou hexadécimal. Ce code est en fait le système décimal dont on a converti les chiffres en leur équivalent binaire.

-Un nombre DCB est différent d'un nombre binaire pur:

•Exemple:

- $785_{10} = 11\ 0001\ 0001_{2}$ (binaire) ← 10 bits
- $785_{10} = 0111\ 1000\ 0101_{DCB}$ (DCB) ← 12 bits

Le principal avantage du code DCB provient de la facilité avec laquelle on passe de ce code à un nombre décimal, et vice versa.

6- Code majoré de trois

Le code majoré de trois ou code plus trois d'un nombre décimal se trouve de la même manière que le code DCB, sauf qu'on ajoute trois (3) à chaque chiffre décimal avant d'opérer la conversion.

Exemple: Convertissons 48 en sa représentation dans le code plus trois:

4	8
+ 3	+3
7	11
↓	↓
0111	1011

Code binaire de 4 bits

Tableau de conversion des 10 Chiffres décimaux en leurs représentations DCB et code majoré de trois:

Décimal	DCB	Majoré de trois
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

NB: Les groupes non valides dans le code Majoré de trois sont: **0000, 0001, 0010, 1101, 1110, 1111.**

7- Code Gray

Ce code appartient à la catégorie des codes dits à distance minimale, du fait qu'une représentation codée ne diffère de celle qui la précède que par **un** bit. C'est un code non pondéré, c-à-dire que les positions binaires des groupes codés ne sont affectées d'aucun poids. C'est pourquoi ce code ne convient pas du tout aux calculs arithmétiques, mais se retrouve surtout dans des applications d'entrée et de sortie et dans des convertisseurs analogiques- numériques.

La principale caractéristique du code Gray est que lorsqu'on passe d'un certain chiffre au suivant, il n'y a qu'un bit qui est différent entre les deux groupes du code.

Tableau de conversion des nombres décimaux de 0 à 15 en leurs représentations en code Gray et code binaire:

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

Le code Gray est utile souvent dans des situations ou d'autres codes, comme le code binaire, peuvent produire des résultats ambigus ou erronés au moment de transitions entraînant le changement de plusieurs bits dans le code.

Exemple du passage de 0111 à 1000 dans le code binaire: les quatre bits changeant en même temps; suivant le dispositif ou le circuit qui produit les bits, il pourra y avoir une différence prononcée entre les temps de transition des divers bits. Ce qui occasionnerait un ou plusieurs états intermédiaires.

Si, par exemple, le bit de poids le plus fort change avant les autres, on observera comme code erroné la transition: 1111, avant d'atteindre 1000.

8 - Les codes alphanumériques

Un code alphanumérique reproduit tous les caractères et les diverses fonctions que l'on retrouve sur un clavier standard de machine à écrire ou d'ordinateur, à savoir des nombres, des lettres, des signes de ponctuation et des caractères spéciaux.

- **Code ASCII**: Le code alphanumérique le plus répandu est le code ASCII (American Standard Code for Information Interchange); on le retrouve dans la majorité des microordinateurs,

des mini-ordinateurs et dans beaucoup de gros ordinateurs.

Le code ASCII (aski) est un code à 7 éléments qui permet de représenter $2^7 = 128$ groupes codés. Ce qui est amplement suffisant pour reproduire toutes les lettres courantes d'un clavier et les fonctions de contrôle comme (RETOUR) et (INTERLIGNE).

Le tableau ci-dessous donne une liste partielle du code ASCII avec les équivalents octaux et hexadécimaux:

Caractère	ASCII à 7 éléments	Octal	Hexadécimal
A	100 0001	101	41
B	100 0010	102	42
C	100 0011	103	43
D	100 0100	104	44
E	100 0101	105	45
F	100 0110	106	46
G	100 0111	107	47
H	100 1000	110	48
I	100 1001	111	49
J	100 1010	112	4A
K	100 1011	113	4B
L	100 1100	114	4C
M	100 1101	115	4D
N	100 1110	116	4E
O	100 1111	117	4F
P	101 0000	120	50
Q	101 0001	121	51
R	101 0010	122	52
S	101 0011	123	53
T	101 0100	124	54
U	101 0101	125	55

Caractère	ASCII à 7 éléments	Octal	Hexadécimal
Y	101 1001	131	59
Z	101 1010	132	5A
0	011 0000	060	30
1	011 0001	061	31
2	011 0010	062	32
3	011 0011	063	33
4	011 0100	064	34
5	011 0101	065	35
6	011 0110	066	36
7	011 0111	067	37
8	011 1000	070	38
9	011 1001	071	39
Blanc	010 0000	040	20
.	010 1110	056	2E
(	010 1000	050	28
+	010 1011	053	2B
$	010 0100	044	24
*	010 1010	052	2A
)	010 1001	051	29
-	010 1101	055	2D
/	010 1111	057	2F
,	010 1100	054	2C

Le code ASCII sert à coder l'information alphanumérique transmise entre un ordinateur et ses périphériques d'entrée/sortie comme les écrans de visualisation ou les imprimantes. Un ordinateur recourt aussi à ce code pour stocker les informations envoyées par l'opérateur depuis son clavier.

Exemple 1: Déterminer les codes ASCII qui se retrouvent en mémoire d'un micro ordinateur quand un opérateur tape sur un clavier l'instruction suivante (en BASIC): GOTO 30

Solution:

G	100 0111
O	100 1111
T	101 0100
O	100 1111
(espace)	010 0000
3	011 0011
0	011 0000

Exemple 2: Le message suivant en code ASCII est mémorisé dans des emplacements consécutifs dans l'ordinateur:

101 0011 101 0100 100 1111 101 0000

Sa signification est:

S T O P

9 -Détection d’erreurs au moyen de la méthode de parité

Dans les circuits numériques la transmission de données binaires d’un point à un autre est fréquente. Par exemples:

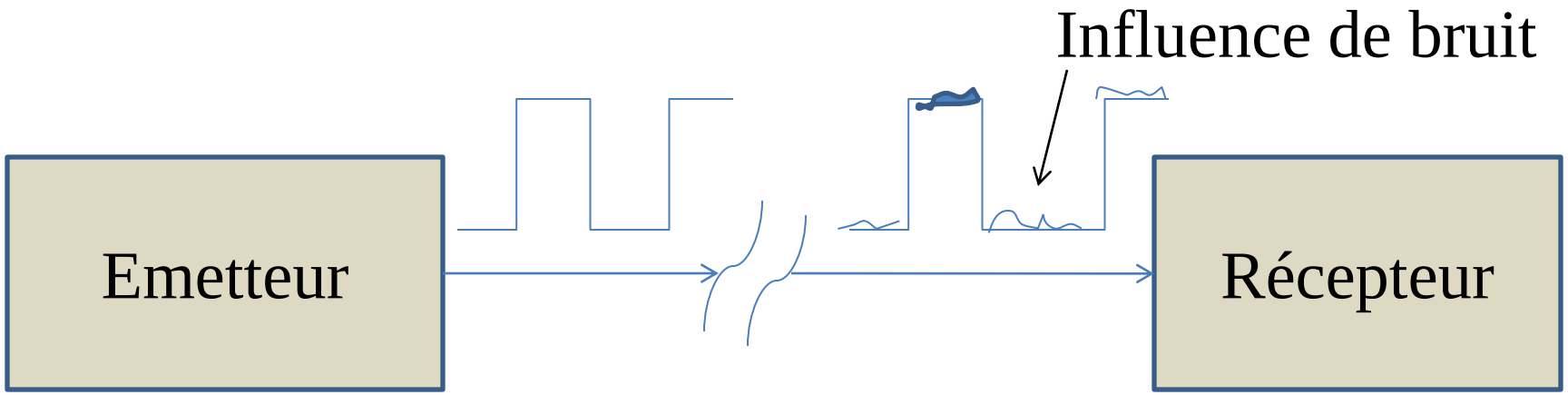
-la transmission de la voix numérisée par des faisceaux hertziens,

-le stockage de données dans une mémoire auxiliaire comme une bande magnétique ou un disque et leur récupération;

-la transmission d’informations d’un ordinateur à un terminal à distance ou à un autre ordinateur.

A chaque transition de l’information d’un dispositif (émetteur) à un autre (récepteur), il existe une probabilité non nulle qu’une erreur survienne et que le récepteur ne reçoive pas l’information envoyée par l’émetteur. Ces erreurs sont principalement dues au **bruit électrique** (fluctuations de tension et de courant parasites se produisant dans tous les appareils électriques à des degrés divers).

Exemple:



La présence de ces erreurs est désagréable et au pire désastreuse pour les équipements. Pour cette raison, on implante dans nombre de systèmes numériques une certaine forme de détection (et parfois de correction) des erreurs. L’une des méthodes les plus simples et les plus répandues est la détection d’erreur par la méthode de la parité.

Le bit de parité:

Un bit de parité est un bit supplémentaire associé à un groupe d'un code qui doit être transféré d'un endroit à un autre. Ce bit de parité est 1 ou 0, selon le nombre de 1 qu'il y a dans le groupe. Il existe deux méthodes de contrôle au moyen de la parité:

- Parité paire: Dans cette méthode de contrôle le bit de parité est fixé de sorte que le total de ‘’1’’ dans la représentation codée (en tenant compte du bit de parité) soit un nombre pair.

• Par exemple, soit à transmettre le groupe 1000011, c'est à dire le code ASCII pour le caractère ‘’C’’. On note trois ‘’1’’ dans ce groupe; il faut donc que le bit de parité soit égal à 1 pour que le nombre de ‘’1’’ soit pair.

Ainsi, le nouveau code constitué par l’ajout du bit de parité est: 11000011

bit de parité ajouté

• Autre exemple, soit à transmettre le caractère ‘’A’’, dont la représentation ASCII est 1000001. Le nombre de ‘’1’’ étant deux, c’est-à-dire pair, le bit de parité à ajouter sera 0, de sorte que le nouveau code est 01000001

- **Parité impaire:** Cette méthode de contrôle est une technique similaire à la méthode de la parité paire, sauf que dans ce cas il faut que le nombre total de ‘’1” (en comptant le bit de parité) soit un nombre impair.

Exemples:

- pour le groupe 1000001, on associe le bit de parité 1

-pour le groupe 1000011, on associe le bit de parité 0

Dans les deux méthodes de contrôle, le bit de parité fait partie intégrante de mot codé. Ainsi, le code ASCII à 7 éléments devient un code à 8 éléments.

-Le bit de parité permet de dépister les erreurs mono bit pouvant subvenir lors de la transmission d'un code entre un endroit et un autre (comme entre un ordinateur et un écran de visualisation).

Exemple:

Transmission du caractère ‘’A” en utilisant la parité impaire: ➔ code à transmettre: 11000001

Si le code reçu au niveau du récepteur est 11000000, alors, le récepteur qui détermine un nombre pair (ici 2) de ‘’1 ‘’ signale que le code reçu est erroné.

Cela suppose évidemment qu’avant la transmission, l’émetteur et le récepteur se sont accordés pour une parité impaire. Par contre, le récepteur ne peut d’aucune manière indiquer le bit erroné, puisqu’il ignore quel code devait être transmis.

NB: En pratique, la méthode de parité n'est utilisée que dans les cas ou la probabilité d'une erreur mono bit est faible et la probabilité d'avoir deux erreurs est nulle.

En effet, la méthode de parité n'est d'aucune utilité quand deux bits sont inexacts, puisque le changement de deux bits ne modifie en rien le «caractère pair ou impair» du nombre de ‘’1’’ dans la représentation codée.

Exemple: Un émetteur transmet une donnée en code ASCII à un récepteur avec un bit de parité paire. Quels sont les codes réellement envoyés par l' émetteur dans le cas du message « ALLO » ?

Solution:

+ On utilise les codes ASCII de chacun des caractères du message;

+ On compte le nombre de ‘’1’’ pour chaque code;

+ pour les nombres pairs de ‘’1’’, on inscrit un 0 comme bit de poids fort, et pour les nombres impairs de ‘’1’’, on inscrit un 1 comme bit de poids fort.

On a ainsi:

A – 0 1000001

L – 1 1001100

L – 1 1001100

0 – 1 100 1111

Bits de parité paire ajoutés