



Faculté
Polydisciplinaire
de Khouribga

Université Sultan
Moulay Slimane
Béni Mellal



Cours d'électronique Numérique

Chapitre 1 : Systèmes de numération et codage

FPK, Filière SMI S3

Pr. A. LAHRECH

Année 2020-2021



Systemes de numération et codage



FPK

Filière SMI S3

1.1 Electronique analogique/Electronique numérique

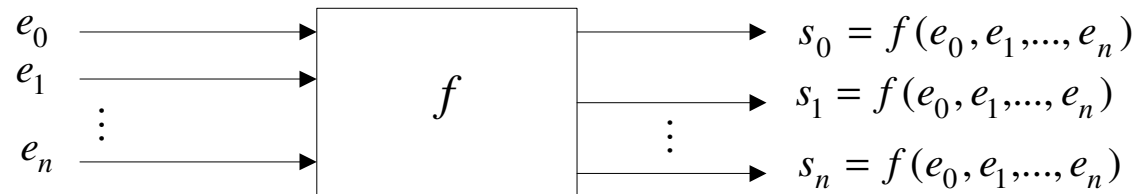
- L'électronique analogique est la discipline traitant des systèmes électroniques opérant sur des grandeurs analogique (tension, courant) à variation continue.
- L'électronique numérique et logique est la discipline qui traite et manipule les signaux discrets (ou binaires) à deux état 0 (état logique bas) et 1 (état logique haut).

1.2 Avantages des systèmes numériques

- La numérisation d'un signal permet une bonne immunité au bruit.
- Les systèmes numériques sont plus simples à concevoir qu'en analogique
- Possibilité de stocker et/ou de transmettre des grandes quantités d'information
- Maitrise de la précision des traitements : choix du codage de l'information à traiter (nombre de bits).
- Possibilité de construction de systèmes programmable

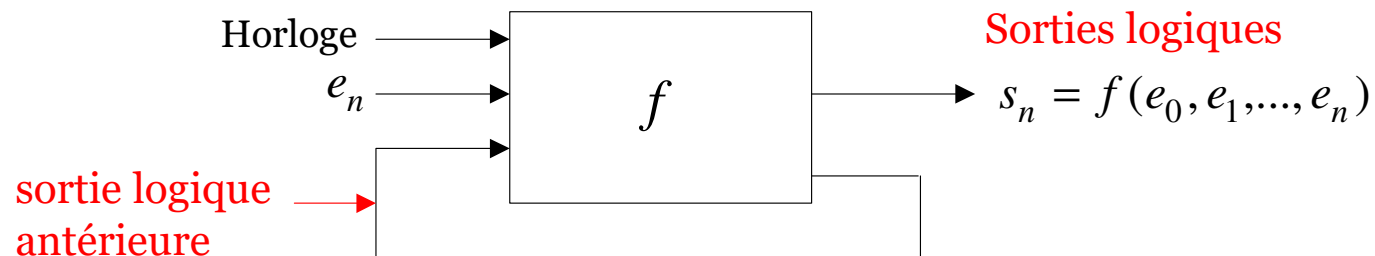
1.3 Type de systèmes numériques

a) Système combinatoire



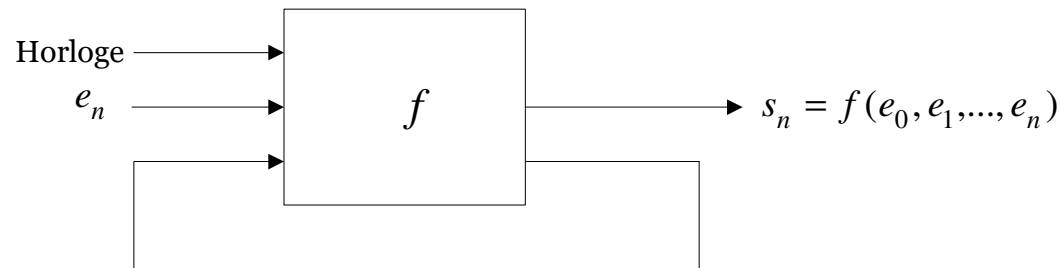
- Les sorties sont déterminées uniquement en fonction des variables d'entrée (pas d'état interne).

b) Système séquentiel



- Dans un système séquentiel, l'état de sortie ne dépend pas uniquement de la combinaison des entrées à un instant donné, mais aussi de valeurs passées de sorties ou de variables internes.

1.4 Système séquentiel synchrone



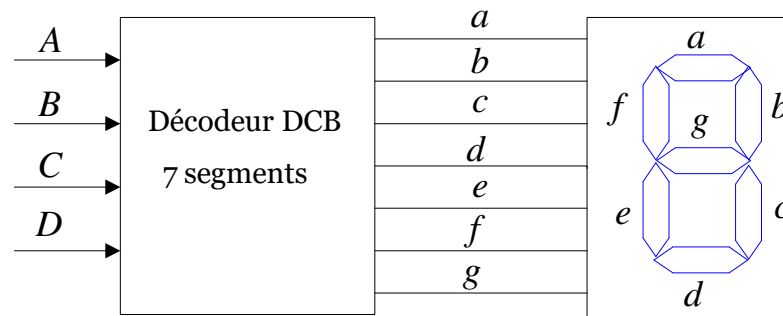
- La sortie du système est mis à jours à des instants précis (connus grâce au signal d'horloge (le système séquentiel est dit synchrone). Le changement d'état est commandé par un signal d'horloge, les informations évoluent en fonction du temps.

1.5 Système séquentiel synchrone

- La sortie du système est mis à jours à des instants non précisées (pas de signal d'horloge (le système est dit système asynchrone). La sortie logique peut changer d'état à tout moment quand une ou plusieurs entrées changent ;

Exemple d'un système combinatoire :

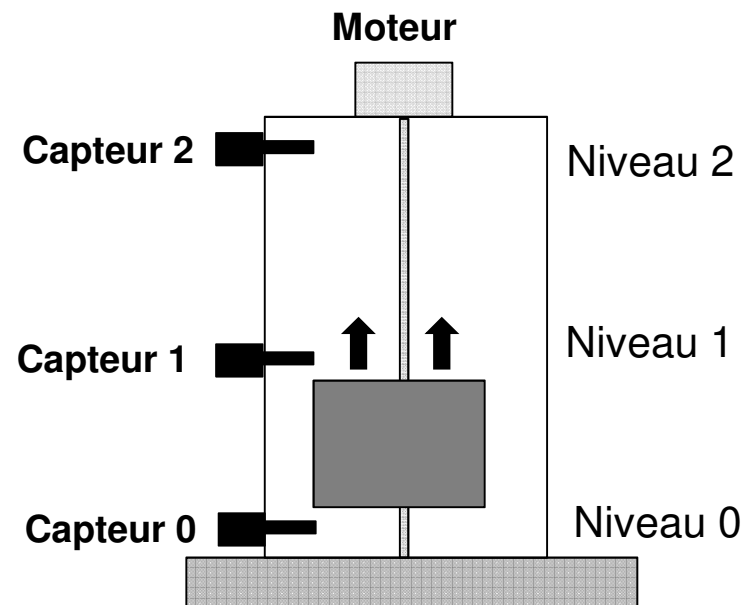
- Afficheur LED 7-segments d'un chiffre compris entre 0 et 9



0 1 2 3 4 5 6 7 8 9

- Entrée : le chiffre à afficher, codé sous forme binaire (ABCD);
- Sortie : segment allumé ou éteint, 7 variables logiques de sortie, a, b, c, d, e, f, g .
- Par exemple pour afficher le chiffre 6, il faut éteindre le segment b , les autres étant allumés.

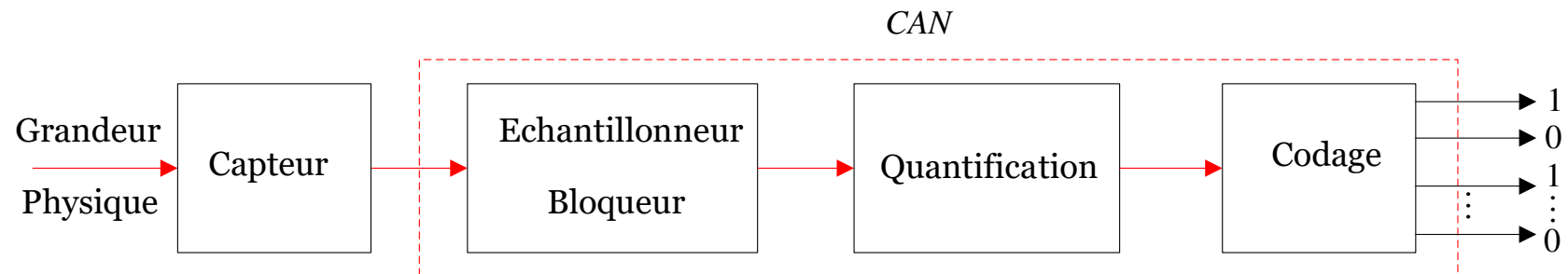
Exemple d'un système séquentiel : Système d'ascenseur



- Entrée : appel étage interne ou externe 0, 1 et 2, états des capteurs présence étage 0, 1 et 2. L'information analogique est convertie en information numérique
- Sortie : ordre de commander le moteur (descente ou monté)

1.6 Génération d'une donnée numérique

- Comment passe-t-on d'une grandeur analogique à une grandeur numérique ?
- On peut établir le schéma type d'une chaîne de traitement numérique



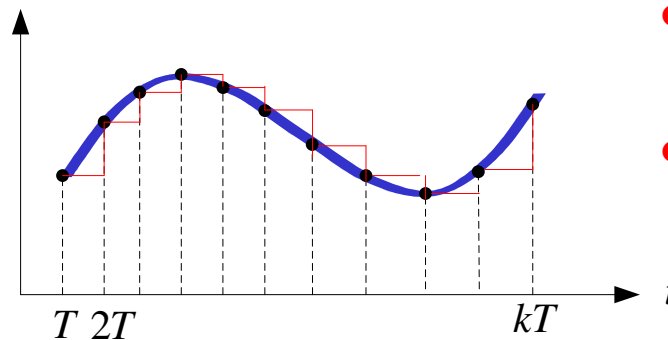
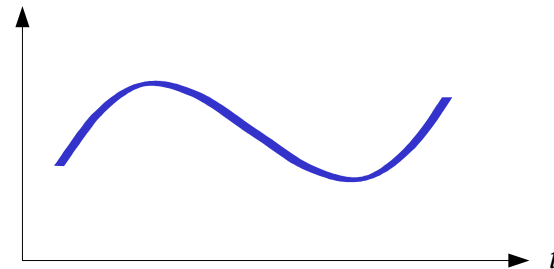
- On trouve toujours dans une telle chaîne trois opérations principales :
 - Echantillonnage
 - Quantification
 - Passage en binaire ou Codage

Etudions les trois opérations principales :

a) L'échantillonnage

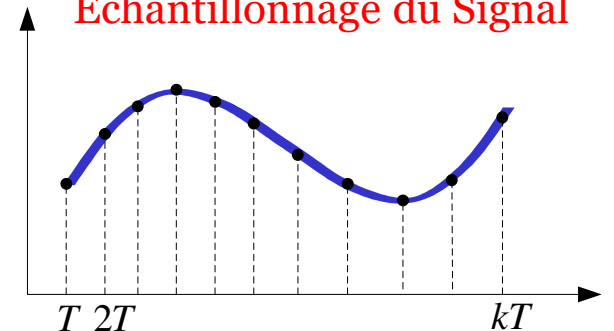
- L'échantillonnage est une opération qui consiste à prélever à intervalles de temps réguliers des mesures sur un signal continu (toutes les T secondes). Le signal obtenu est alors sous forme d'une suite d'échantillons, mais encore analogiques.

Signal analogique



Echantillonnage + bloqueur

Echantillonnage du Signal



- **Contrainte théorique : théorème de Shannon :**
- Pour un échantillonnage sans perte d'information, la fréquence d'échantillonnage doit être au moins égale au double de la fréquence maximal présente dans le signal : $f_e \geq 2f_{\max}$



Systemes de numération et codage

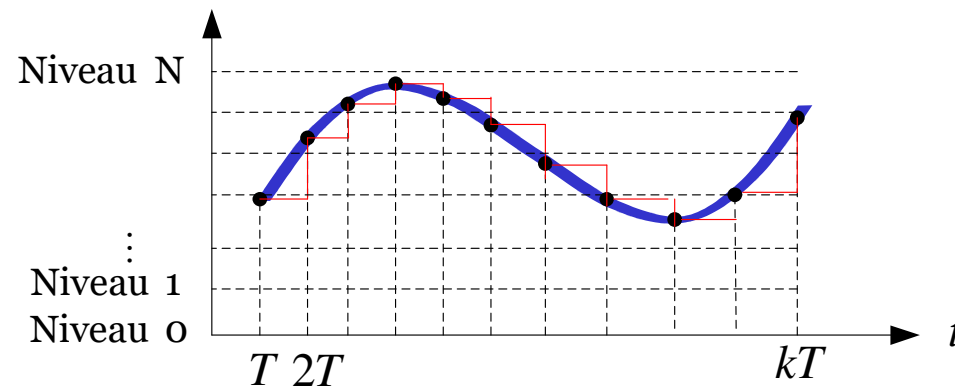


FPK

Filière SMI S3

b) La quantification

- L'opération de quantification consiste à partager l'étendue des variations du signal en N niveaux équidistant et à arrondir les valeurs des échantillons au niveau le plus proche.

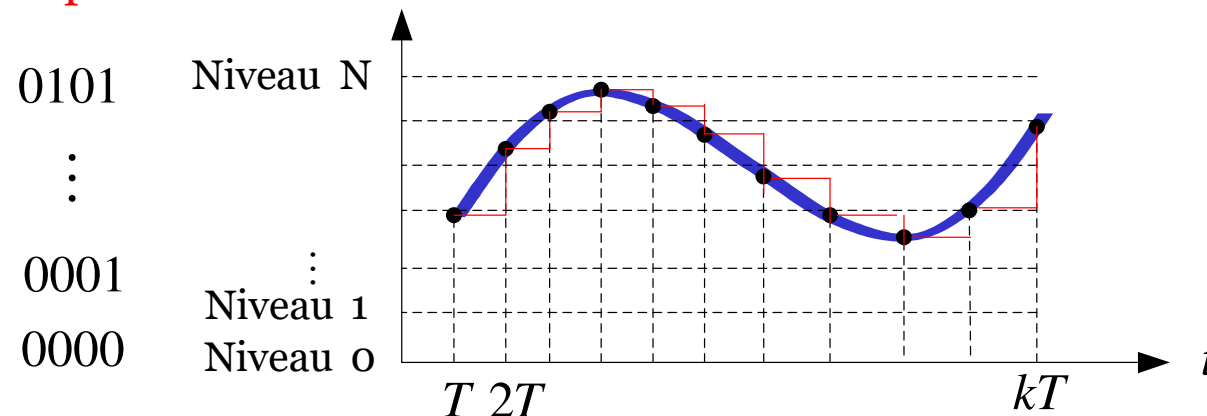


- L'intervalle entre chaque niveau se nomme le « pas de quantification »
- Plus N est grand (plusieurs centaines ou plusieurs milliers), plus la précision sera élevée.
- On arrondit alors la valeur des échantillons précédents au niveau le plus proche (on commet donc une erreur de $\pm 1/2$ du pas de quantification). 9

c) Le codage

- L'opération de codage consiste à attribuer un nombre binaire à chaque niveau de quantification. La valeur binaire sera codée avec un nombre de bits (8 bits pour des petits nombres, 16, 32, 64 bits ou plus pour des grands nombres).

Code binaire
correspondant



- Chaque niveau de tension est codé sur p bits, chaque bit pouvant prendre deux valeurs (0 ou 1). Donc un convertisseur à p bits possède 2^p niveaux de quantification



Systemes de numération et codage



d) Résolution du convertisseur analogique numérique

- C'est le plus petit changement de la pleine échelle exprimée en %.

Pour n bits, la résolution $R = \frac{100}{2^n} \%$

Exemple :

- Un signal analogique varie entre 0V et 10V. On désire avoir une résolution de 5 mV. Combien de bits doit avoir le convertisseur CAN ?

$$\frac{10V}{5 \cdot 10^{-3}} = 2000 \text{ (Niveaux de quantification)}$$

$$10 \text{ bits} \Rightarrow 2^{10} = 1024 \text{ niveaux (pas suffisant)}$$

$$11 \text{ bits} \Rightarrow 2^{11} = 2048 \text{ niveaux}$$

$$\Rightarrow \text{CAN avec 11 bits}$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.0 Système de numération

- Le Système de numération décrit la façon avec laquelle les nombres sont représentés.
- En électronique numériques, les systèmes les plus utilisés sont : le système binaire, le système octal, le système décimal et le système hexadécimal.

2.1 Système Binaire Naturel

- Ce système dit à base 2 comprend deux chiffres 0 et 1, appelé bit (contraction de binary digit)
- Un ensemble de plusieurs bits est appelé "mot "
- Un mot binaire de n bits permet d'obtenir 2^n combinaisons différentes et donc peut représenter 2^n états.
- Un ensemble de 8 bits est un mot de 8 bits appelé Octet (« byte » en anglais)



Systemes de numération et codage



Exemple :

- Soit N un nombre représenté en binaire par : $N = (11011, 1011)_2$

Bit de poids fort : $(11011, 1011)_2$ Bit de poids faible :
MSB (Most Significant bit) \uparrow \uparrow LSB (Least Significant bit)

- On appelle "bits de poids faible" le bit situé le plus à droite du nombre binaire et "bits de poids fort" le bit situé le plus à gauche du nombre binaire.
- A chaque bit, on affecte un poids correspond à sa position

Nombre binaire $\longrightarrow (1\ 1\ 0\ 1\ 1, 1\ 0\ 1\ 1)_2$

poids de chaque bit $\longrightarrow 2^4\ 2^3\ 2^2\ 2^1\ 2^0, 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}$

- Représentation décimale du nombre binaire : $N = (11011, 1011)_2$

$$(11011, 1011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

Ce qui donne : $(11011, 1011)_2 = (27,6875)_{10}$



Systèmes de numération et codage



2.1.1 Conversion binaire-décimal

- Cette conversion s'effectue en multipliant chaque chiffre du nombre binaire par son poids respectifs. L'équivalent décimal est ensuite obtenu par l'addition de tous les produits partiels

Par exemple :

$$(11011,1011)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4}$$

$$(11011,1011)_2 = (27,6875)_{10}$$

2.1.2 Conversion décimal-binaire

- La conversion décimal en nombre binaire se fait en effectuant des divisions successives par 2 jusqu'à ce qu'un quotient de zéro soit obtenu.

$$\text{Par exemple : } (13,71875)_{10} = (?)_2$$



Systèmes de numération et codage



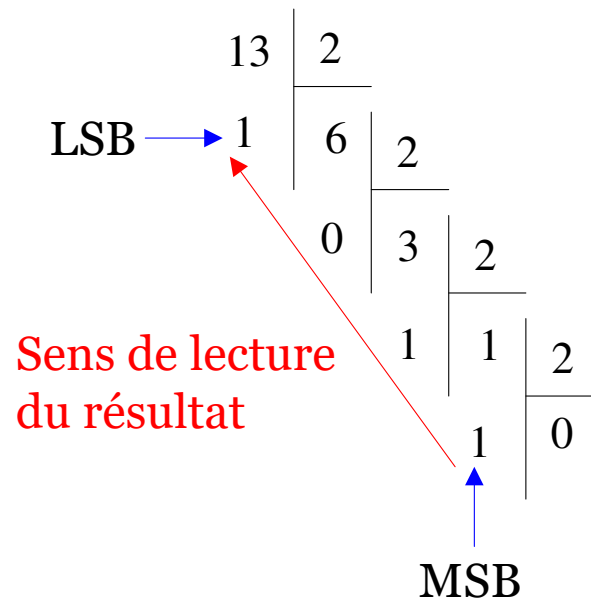
FPK

Filière SMI S3

- La conversion d'un nombre décimal qui comprend une partie fractionnaire exige l'usage de deux procédures :

a) La partie entière est convertie par des divisions successives par 2 jusqu'à ce qu'un quotient de zéro soit obtenu.

Conversion de la partie entière du nombre : $(13,71875)_{10}$



Donc : $(13)_{10} = (1101)_2$



Systèmes de numération et codage



FPK

Filière SMI S3

- b) L'équivalent binaire de la partie fractionnaire est obtenu par des multiplications successives par 2, le processus est terminé lorsque le résultat de multiplication est égal à un .

$$\begin{array}{l} 0,71875 \times 2 = 1,4375 \\ 0,4375 \times 2 = 0,875 \\ 0,875 \times 2 = 1,75 \\ 0,75 \times 2 = 1,50 \\ 0,50 \times 2 = 1,00 \end{array}$$

$(0,10111)_2$

$$(0,71875)_{10} = (0,10111)_2$$

$$\text{Donc : } (13,71875)_{10} = (1101,10111)_2$$



Systèmes de numération et codage



Exemple :

- Trouver l'équivalent binaire du nombre décimal suivant : $(31,75)_{10} = (?)_2$

$$(13)_{10} = (11111)_2$$

$$(0,75)_{10} = 0,5 + 0,25$$

$$(0,75)_{10} = \frac{1}{2} + \frac{1}{4} = 1 \times 2^{-1} + 1 \times 2^{-2}$$

$$\text{Donc : } (13,75)_{10} = (11111,11)_2$$

31		2								
1		15		2						
		1		7		2				
				1		3		2		
						1		1		2
								1		0



Systèmes de numération et codage



FPK

Filière SMI S3

2.2 Système Octal

- Le système octal est constitué de huit chiffres : 0, 1, 2, 3, 4, 5, 6, 7
- C'est un système à base 8. Il est moins utilisé dans la représentation des nombres binaires.
- Par exemple 7564,26 en base octale s'écrit $(7564,26)_8$
- Le poids de chaque chiffre d'un nombre octal est déterminé par son rang :

Nombre octal $\longrightarrow (7\ 5\ 6\ 4, 2\ 6)_8$

poids du chiffre $\longrightarrow 8^3\ 8^2\ 8^1\ 8^0, 8^{-1}\ 8^{-2}$

- La conversion d'un nombre octal en nombre décimal se fait en calculant le produit de chaque chiffre par son poids respectifs. L'équivalent décimal est ensuite obtenu par l'addition de tous ces produits partiels

$$(7564,26)_8 = 7 \times 8^3 + 5 \times 8^2 + 6 \times 8^1 + 4 \times 8^0 + 2 \times 8^{-1} + 6 \times 8^{-2}$$

$$\text{Donc : } (7564,26)_8 = (3956,34375)_{10}$$



Systèmes de numération et codage



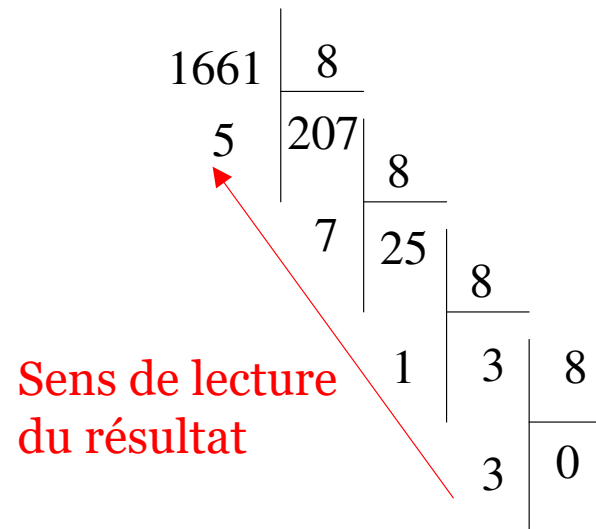
FPK

Filière SMI S3

2.2.1 Conversion décimal-octal

- La conversion d'un nombre décimal en nombre octal se réalise en effectuant des divisions successives par 8 jusqu'à ce qu'un quotient de zéro soit obtenu.

Par exemple l'équivalent octal du nombre décimal suivant : $(1661)_{10} = (?)_8$



Donc : $(1661)_{10} = (3175)_8$



Systèmes de numération et codage



Exemple :

- Calculer l'équivalent octal du nombre décimal suivant: $(1661,34375)_{10}$

Conversion de la Partie entière : $(1661)_{10} = (3175)_8$

Conversion de la partie fractionnaire : $(0,34375)_{10} = (?)_8$

$$\begin{array}{lcl} 0,34375 \times 8 & = & 2,75 \\ 0,75 \times 8 & = & 6,00 \\ 0,00 \times 8 & = & 0,00 \end{array}$$

$(0,260)_8$

Donc : $(1661,34375)_{10} = (3175,026)_8$



Systèmes de numération et codage



2.2.2 Conversion du système binaire au système octal

- Le passage du système binaire au système octal se fait par groupement de 3 bits.

Correspondance Binaire/octal (et inversement)

Octal	binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Par exemple : $(10110)_2 = (?)_8$

$$\left(\underbrace{010}_{2} \underbrace{110}_{6} \right)_2 = (26)_8$$

$$(10001101)_2 = (?)_8$$

$$\left(\underbrace{010}_{2} \underbrace{001}_{1} \underbrace{101}_{5} \right)_2 = (215)_8$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.3 Système Hexadécimal

- Ce système à base 16 comprend 16 symboles, dix chiffres 0 à 9 et six lettres A, B, C, D, E et F.

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

$$(15)_{10} = (F)_{16} \text{ ou } (F)_{\text{H}}$$

Nombre à deux chiffres en hexadécimal :

10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20 etc. 22



Systèmes de numération et codage



FPK

Filière SMI S3

2.3.1 Conversion du système Hexadécimal au système décimal

- Cette conversion s'effectue en multipliant chaque chiffre du nombre hexadécimal par son poids respectif. L'équivalent décimal est ensuite obtenu par l'addition de tous les produits partiels

Par exemple l'équivalent décimal du nombre hexadécimal: $(AB4F)_{16} = (?)_{10}$

$$(AB4F)_{16} = A \times 16^3 + B \times 16^2 + 4 \times 16^1 + F \times 16^0$$

$$(AB4F)_{16} = 10 \times 16^3 + 11 \times 16^2 + 4 \times 16^1 + 15 \times 16^0$$

$$(AB4F)_{16} = (43855)_{10}$$

$$(92)_{16} = 9 \times 16^1 + 2 \times 16^0$$

$$(92)_{16} = (146)_{10}$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.3.2 Conversion décimal-Hexadécimal

- Cette conversion se fait en effectuant des divisions successives par 16 jusqu'à ce qu'un quotient de zéro soit obtenu

Par exemple l'équivalent décimal du nombre hexadécimal: $(124)_{10} = (?)_{16}$

$$\begin{array}{r|l} 124 & 16 \\ \hline 12 & 7 \\ & \hline & 7 & 16 \\ & & \hline & & 0 \end{array}$$

$$(124)_{10} = (7C)_{16}$$

- De même l'équivalent hexadécimal des nombres décimaux suivants :

$$(1276)_{10} = (4FC)_{16}$$

$$(716)_{10} = (2CC)_{16}$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.3.3 Conversion du code binaire au code Hexadécimal

- Le passage du code binaire au code hexadécimal se fait par groupement de 4 bits. Chaque quadruplet ainsi formé correspond à un chiffre hexadécimal

Exemple : $(11\ 1011\ 1111\ 1100)_2 = (?)_{16}$

$$(11\ 1011\ 1111\ 1100)_2 = (0011\ 1011\ 1111\ 1100) = (3\ B\ F\ C)_{16}$$

- La conversion d'un nombre hexadécimal en nombre binaire s'effectue en représentant chaque chiffre du nombre hexadécimal par son équivalent binaire de 4 bits.

Exemple : $(A9B7)_{16} = (?)_2$

$$\underbrace{1010}_A \quad \underbrace{1001}_9 \quad \underbrace{1011}_B \quad \underbrace{0111}_7$$

$$(A9B7)_{16} = (1010\ 1001\ 1011\ 0111)_2$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.4 Codages des nombres entier négatifs

- Ce codage permet la représentation des nombres entiers relatifs. Dans la représentation en binaire signé, le bit de poids fort sert à représenter le signe (0 pour un entier positif et 1 pour un entier négatif).
- Le code le plus souvent utilisé est le code complément à deux
- Sur n bits, on peut représenter les nombres allant de -2^{n-1} à $2^{n-1} - 1$.

Nombre de bits	Dynamique de codage
4	De -8 à $+7$
8	De -128 à $+127$
16	De -32768 à $+32767$
32	De $-2\,147\,483\,648$ à $+2\,147\,483\,647$

- Le codage d'un nombre entier dans un ordinateur se fait sur un nombre fixe de bits (8, 16, 32,...)



Systèmes de numération et codage



2.4.1 Code complément à deux

- Le code complément à 2 (C2 en abrégé) est un outil permettant de coder des nombres entiers négatifs et permettant les calculs arithmétiques entre deux nombres codés ainsi.
- Le complément à 1 d'un nombre binaire s'obtient en remplaçant les 1 par des 0 et vice-versa.
- Le complément à 2 s'obtient en ajoutant 1 au complément à 1.

Exemple : Représentation de +5 et -5 en valeur signée sur 4 bits

$$(+5)_{10} = (0101)_2$$

$$C1(+5) = (1010)$$

$$C2(+5) = (1011)$$

↑

Bite de signe

+ 5 -5 →

0101
+ 1011

>1 0000

└──────────┘
Débordement à
éliminer



Systemes de numération et codage



FPK

Filière SMI S3

- On doit toujours spécifier le nombre de bits lorsqu'on travaille en complément à 2. Sur 4 bits par exemple, on ne peut représenter que les nombres compris entre -7 et + 8:

Décimal	Code Complément à 2	décimal	Code Complément à 2
- 1	1111	0	0000
- 2	1110	1	0001
- 3	1101	2	0010
- 4	1100	3	0011
- 5	1011	4	0100
- 6	1010	5	0101
- 7	1001	6	0110
- 8	1000	7	0111

- Le MSB indique le signe du nombre : lorsqu'il est égal à 1 le nombre est négatif et le nombre ainsi formé est dit signé. lorsqu'il est égal à 0 le nombre est positif.



2.5 Opérations arithmétiques en binaire

a) l'addition :

- L'addition binaire est basé sur l'algorithme suivant :

$$0 + 0 = 0$$

$$0 + 1 = 1 + 0 = 1$$

$$1 + 1 = 0 \text{ avec retenue}$$

$$1 + 1 + 1 = 1 \text{ avec retenue}$$

Exemple 1 : addition de deux nombres positifs

Effectuant l'addition $(+17)_{10}$ et $(+12)_{10}$

$$\begin{array}{rcl} (+17)_{10} & & 010001 \\ + (+12)_{10} & \longrightarrow & + \underline{001100} \\ & & \quad \quad \quad \longrightarrow 011101 \end{array}$$

Bite de signe



Systemes de numération et codage



FPK

Filière SMI S3

Exemple 2 : addition de deux nombres de signes contraires

- Pour calculer $(M-N)$, on calcule la somme de M et le complément à 2 de N
- Effectuant l'addition des nombres suivants : $(+17)_{10}$ et $(-12)_{10}$

$$(+17)_{10} \rightarrow (010001)_2$$

Le complément à 2 de $(12)_{10}$ s'écrit :

$$C2(12) = (110100)$$

$$(-12)_{10} \rightarrow (110100)$$

010001
+ 110100
1 000101

→ $(+5)_{10}$

↑ ↑ Bit de signe
Débordement à éliminer

- Effectuant l'addition de $(-17)_{10}$ et de $(+12)_{10}$

Le complément à 2 de $(17)_{10}$ s'écrit :

$$(-17)_{10} \rightarrow (101111)$$

$$(+12)_{10} \rightarrow (001100)_2$$

11
101111
+ 001100
111011

↑ Bit de signe

- Cette somme est négative, le résultat est le complément à 2 du total recherché qui s'écrit : $(100101)_2 = (-5)_{10}$



Systèmes de numération et codage



FPK

Filière SMI S3

b) Multiplication en binaire

- Pour faire une multiplication en binaire, on suit les mêmes règles que le système décimal :

Exemple :

$$\begin{array}{r} 1011 \\ \times 101 \\ \hline 1011 \\ 0000 \\ 1011 \\ \hline 110111 \end{array}$$

c) Division en binaire

- La méthode est identique à celle d'une division de deux nombres décimaux

Exemple :

$$\begin{array}{r|l} 110110 & 1010 \text{ Diviseur} \\ - 1010 & \leftarrow 101 \\ \hline 00111 & \\ - 00000 & \leftarrow 101 \\ \hline 1110 & \\ - 1010 & \leftarrow 101 \\ \hline \text{Reste } 0100 & \end{array} \quad \begin{array}{l} \text{quotient} \end{array}$$



Systemes de numération et codage



FPK

Filière SMI S3

2.6 Code DCB (Binary Coded Decimal)

- Le code décimal codé binaire se compose de série de quatre bits, le nombre de chiffre représentés, se limite à dix (0 à 9)

Décimal	0	1	2	3	4	5	6	7	8	9
Code DCB	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

Conversion Décimal-code DCB

- On fait correspondre à chaque chiffre décimal un mot

par exemple l'équivalent DCB du nombre décimal : $(2583)_{10} = (?)_{DCB}$

$$(2583)_{10} = \left(\underbrace{0010}_2 \underbrace{0101}_5 \underbrace{1000}_8 \underbrace{0011}_3 \right)_{DCB}$$

- Il est important de ne pas confondre le système binaire pure avec le code DCB car le résultat peut être très différent

$$\begin{aligned} (0010010110000011)_{16} &= 1 \times 2^{12} + 1 \times 2^{10} + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^1 + 1 \times 2^0 \\ &= (9603)_{10} \end{aligned}$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.7 Code binaire réfléchi ou code GRAY

- Le code GRAY est une autre forme de la base binaire sa principale caractéristique est le changement de bits lors de l'incrémentation. En effet un seul bit à la fois change d'état lorsqu'on passe d'un nombre au suivant.
- Par exemple, le passage du chiffre 3 au chiffre 4 nécessite un changement d'état des trois bits à la fois dans le système binaire ordinaire (0011 à 0100) au lieu (0010 à 0110)
- Ce système de codage est très utilisé dans les CAN.

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



Systemes de numération et codage



FPK

Filière SMI S3

2.7.1 Conversion de la base binaire au code GRAY

Décimal	Binaire $b_3 \ b_2 \ b_1 \ b_0$	Code GRAY $g_3 \ g_2 \ g_1 \ g_0$
0	0000	0 0 0 0
1	0001	0 0 0 1
2	0010	0 0 1 1
3	0011	0 0 1 0
4	0100	0 1 1 0
5	0101	0 1 1 1
6	0110	0 1 0 1
7	0111	0 1 0 0
8	1000	1 1 0 0
9	1001	1 1 0 1
10	1010	1 1 1 1
11	1011	1 1 1 0
12	1100	1 0 1 0
13	1101	1 0 1 1
14	1110	1 0 0 1
15	1111	1 0 0 0

Axe de symétrie.

Conversion de la base
binaire au code GRAY

$$\begin{cases} g_0 = b_0 \oplus b_1 = b_0 \text{ XOR } b_1 \\ g_1 = b_1 \oplus b_2 \\ g_2 = b_2 \oplus b_3 \\ g_3 = b_3 \end{cases}$$

Conversion du code GRAY à
la base binaire

$$\begin{cases} b_0 = g_0 \oplus b_1 \\ b_1 = g_1 \oplus b_2 \\ b_2 = g_2 \oplus b_3 \\ b_3 = g_3 \end{cases}$$



Systèmes de numération et codage



FPK

Filière SMI S3

2.7.1 Conversion de la base binaire au code GRAY (suite)

- La conversion d'un nombre binaire en un nombre du code GRAY se fait de la façon suivante :
- Le bit de gauche, du poids le plus fort, reste le même qu'en binaire
- De la gauche vers la droite, chaque bit est additionné à son voisin de droite. La somme est reportée à la ligne inférieur, qui correspond au code Gray . Les retenues sont négligées.
- Le résultat en code Gray comporte toujours le même nombre de bits que la représentation binaire ordinaire

Exemple :

Donner l'équivalent en code GRAY du nombre binaire suivant :

$$(101101)_2 = (?)_{\text{Gray}}$$

	↗	↘	↗	↘	↗	↘	
	1	0	1	1	0	1	
	1	1	1	0	1	1	

Code binaire

Code Gray



Systèmes de numération et codage

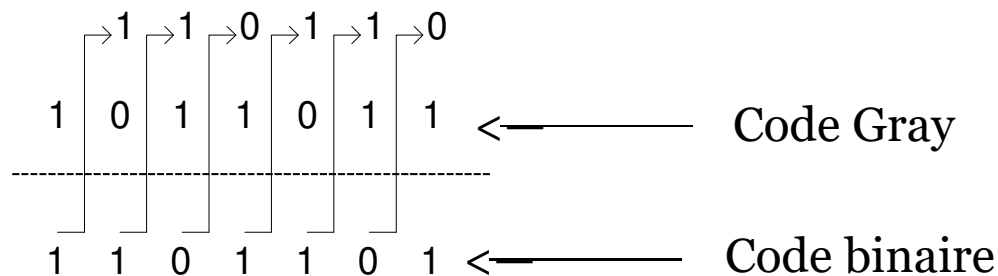


2.7.2 Conversion du code GRAY à la base binaire

- La conversion s'effectue de la façon suivante :
- Le bit de gauche, du poids le plus fort, reste le même qu'en binaire
- De la gauche vers la droite, on additionne au bit Gray le résultat binaire de la colonne précédente. Les retenues sont négligées.
- Le résultat en système binaire comporte toujours le même nombre de bits que la représentation en code Gray

Exemple :

Donner l'équivalent en binaire du code suivant : $(1011011)_{\text{Gray}} = (?)_2$





Systèmes de numération et codage



FPK

Filière SMI S3

2.8 Code ASCII

- Le code ASCII (American Standard Code for Information Interchange), ou code américain pour l'échange d'information, est un code alphanumérique qui permet de représenter des chiffres, des lettres ainsi que divers caractères spéciaux tels que = , \$, +, %. Il permet entre autres, de transformer l'information tapée sur un clavier.
- Ce code aide donc la machine à comprendre l'information qu'on veut lui transmettre lorsqu'on appuie sur une touche quelconque. C'est donc dire qu'il traduit ces caractères en langage machine.
- Dans le code ASCII, les caractères alphanumériques sont codés sur 7 bits, ce qui donne une possibilité de 256 symboles différents.

Exemple : GOTO□25

$G \rightarrow 100\ 0111$	$\square \rightarrow 0100000$
$O \rightarrow 1001111$	$2 \rightarrow 0110010$
$T \rightarrow 1010100$	$5 \rightarrow 0110101$
$O \rightarrow 1001111$	



Systemes de numération et codage



FPK

Filière SMI S3

Table du code ASCII

Decimal	Hex	Binary	Code	Code description
0	00	0000 0000	NUL	Null character
1	01	0000 0001	SOH	Start of header
2	02	0000 0010	STX	Start of text
3	03	0000 0011	ETX	End of text
4	04	0000 0100	EOT	End of transmission
5	05	0000 0101	ENQ	Enquiry
6	06	0000 0110	ACK	Acknowledgement
7	07	0000 0111	BEL	Bell
8	08	0000 1000	BS	Backspace
9	09	0000 1001	HT	Horizontal tab
10	0A	0000 1010	LF	Line feed
11	0B	0000 1011	VT	Vertical tab
12	0C	0000 1100	FF	Form feed
13	0D	0000 1101	CR	Carriage return
14	0E	0000 1110	SO	Shift out
15	0F	0000 1111	SI	Shift in
16	10	0001 0000	DLE	Data link escape
17	11	0001 0001	DC1	Device control 1 (XON)

Decimal	Hex	Binary	Code	Code description
18	12	0001 0010	DC2	Device control 2
19	13	0001 0011	DC3	Device control 3 (XOFF)
20	14	0001 0100	DC4	Device control 4
21	15	0001 0101	NAK	Negative acknowledgement
22	16	0001 0110	SYN	Synchronous idle
23	17	0001 0111	ETB	End of transmission block
24	18	0001 1000	CAN	Cancel
25	19	0001 1001	EM	End of medium
26	1A	0001 1010	SUB	Substitute
27	1B	0001 1011	ESC	Escape
28	1C	0001 1100	FS	File separator
29	1D	0001 1101	GS	Group separator
30	1E	0001 1110	RS	Record separator
31	1F	0001 1111	US	Unit separator
32	20	0010 0000	SP	Space
33	21	0010 0001	!	Exclamation point
34	22	0010 0010	"	Quotation mark
35	23	0010 0011	#	Number sign, octothorp, pound
36	24	0010 0100	\$	Dollar sign
37	25	0010 0101	%	Percent
38	26	0010 0110	&	Ampersand



Systèmes de numération et codage



FPK

Filière SMI S3

Table du code ASCII (suite)

Decimal	Hex	Binary	Code	Code description
39	27	0010 0111	'	Apostrophe, prime
40	28	0010 1000	(Left parenthesis
41	29	0010 1001)	Right parenthesis
42	2A	0010 1010	*	Asterisk, 'star'
43	2B	0010 1011	+	Plus sign
44	2C	0010 1100	,	Comma
45	2D	0010 1101	-	Hyphen, minus sign
46	2E	0010 1110	.	Period, decimal Point, 'dot'
47	2F	0010 1111	/	Slash, virgule
48	30	0011 0000	0	0
49	31	0011 0001	1	1
50	32	0011 0010	2	2
51	33	0011 0011	3	3
52	34	0011 0100	4	4
53	35	0011 0101	5	5
54	36	0011 0110	6	6

Decimal	Hex	Binary	Code	Code description
55	37	0011 0111	7	7
56	38	0011 1000	8	8
57	39	0011 1001	9	9
58	3A	0011 1010	:	Colon
59	3B	0011 1011	;	Semicolon
60	3C	0011 1100	<	Less-than sign
61	3D	0011 1101	=	Equals sign
62	3E	0011 1110	>	Greater-than sign
63	3F	0011 1111	?	Question mark
64	40	0100 0000	@	At sign
65	41	0100 0001	A	A



Systemes de numération et codage



FPK

Filière SMI S3

Table du code ASCII (suite)

Decimal	Hex	Binary	Code	Code description					
66	42	0100 0010	B	B	90	5A	0101 1010	Z	Z
67	43	0100 0011	C	C	91	5B	0101 1011	[Opening bracket
68	44	0100 0100	D	D	92	5C	0101 1100	\	Reverse slash
69	45	0100 0101	E	E	93	5D	0101 1101]	Closing bracket
70	46	0100 0110	F	F	94	5E	0101 1110	^	Circumflex, caret
71	47	0100 0111	G	G	95	5F	0101 1111	_	Underline, underscore
72	48	0100 1000	H	H	96	60	0110 0000	`	Grave accent
73	49	0100 1001	I	I	97	61	0110 0001	a	a
74	4A	0100 1010	J	J	98	62	0110 0010	b	b
75	4B	0100 1011	K	K	99	63	0110 0011	c	c
76	4C	0100 1100	L	L	100	64	0110 0100	d	d
77	4D	0100 1101	M	M	101	65	0110 0101	e	e
78	4E	0100 1110	N	N	102	66	0110 0110	f	f
79	4F	0100 1111	O	O	103	67	0110 0111	g	g
80	50	0101 0000	P	P	104	68	0110 1000	h	h
81	51	0101 0001	Q	Q	105	69	0110 1001	i	i
82	52	0101 0010	R	R	106	6A	0110 1010	j	j
83	53	0101 0011	S	S	107	6B	0110 1011	k	k
84	54	0101 0100	T	T	108	6C	0110 1100	l	l
85	55	0101 0101	U	U	109	6D	0110 1101	m	m
86	56	0101 0110	V	V	110	6E	0110 1110	n	n
87	57	0101 0111	W	W	111	6F	0110 1111	o	o
88	58	0101 1000	X	X	112	70	0111 0000	p	p
89	59	0101 1001	Y	Y	113	71	0111 0001	q	q
					114	72	0111 0010	r	r

Questions ?