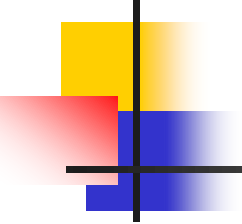
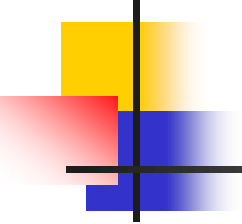


Code, codage et transcodeur

(Suite Chapitre I)

- 
- Principe du codage
 - Le code binaire naturel
 - Les codes décimaux
 - Convertisseur BCD décimal
 - Convertisseur décimal BCD
 - Code 2421
 - Code plus 3
 - Code AIKEN
 - Code GRAY
 - Code complément à 9
 - Codage des caractères
 - Code ASCII

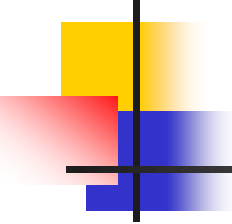
Code, codage et transcodeur



Dans le traitement des signaux numériques, les transcodeurs sont utilisés pour adapter les différents codes numériques les uns par rapport aux autres.

Ils effectuent la conversion d'une numération déterminée en une autre. La conversion d'un résultat de calcul binaire dans le code à 7 segments pour un afficheur numérique est un exemples parmi d'autres.

Définitions

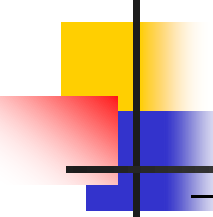
- 
- **Le codage est l'opération qui transforme une information (écriture décimal, position angulaire, vitesse...) en écriture binaire dans un code de notre choix.**
 - **Un code est un est un langage composé de différents symboles ou mots.**
 - **Un mot est un ensemble de caractères numériques ou alphanumériques.**
 - **Le transcodage est le passage d'un code à un autre.**

Code binaire pur

- Ce code correspond au système de numération binaire et fait référence au code binaire naturel.
- Les caractères sont des bits, les mots sont formés par une association ou combinaison de bits et avec n bits nous pouvons former au maximum 2^n mots.

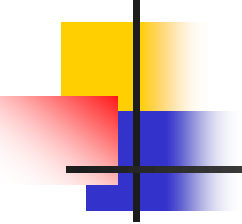
Code binaire pur

Code DCB



Nombre décimal	Code binaire m = 4				Code DCB N = 10			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	0	1	1	1
8	1	0	0	0	1	0	0	0
9	1	0	0	1	1	0	0	1
10	1	0	1	0	Pseudo-tétrades			
11	1	0	1	1				
12	1	1	0	0				
13	1	1	0	1				
14	1	1	1	0				
15	1	1	1	1				

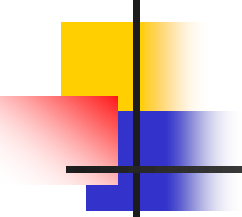
Code 2421

- 
- Parmi la variété de codes possibles, le code 2421 est également utilisé.

Dans ce code, les poids affectés aux variables binaires ne sont plus 8, 4, 2, 1 mais 2, 4, 2, 1.

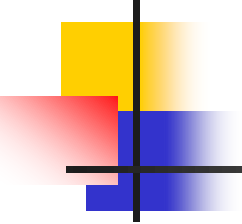
Le tableau suivant nous présente une comparaison entre le code 2421 et le 8421.

Code 2421



Nombre décimal	Code 8421				Code 2421			
	(8) D	(4) C	(2) B	(1) A	(2) D'	(4) C'	(2) B'	(1) A'
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	0	1	1	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1

Code 2421

- 
- Dans ce code il existe deux manières de représenter certains chiffres, ainsi :

7 peut s'écrire :
0111 ou 1101

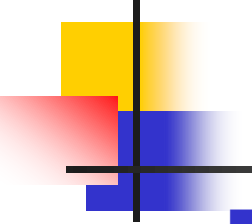
ainsi que 5 :
0101 ou 1011



Remarques sur le code 2421

- Nous pouvons remarquer également que le 9 en décimal est représenté par quatre bits à 1 dans le **code 2421**. Ce qui veut dire que l'on ne peut pas représenter une valeur supérieure à 9 avec 4 bits en 2421 contrairement au code 8421 qui peut représenter 16 valeurs différentes.

code 2421 AIKEN

- 
- **Le code Aiken étant fréquemment utilisé dans les automatismes industriels, il est par conséquent nécessaire de le convertir aux interfaces en code 8421 et inversement.**

En étudiant le code 2421 nous avons observé qu'il était possible de représenter certains chiffres de deux manières.

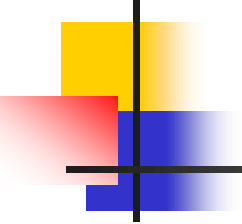
Par exemple le 7 peut s'écrire 0111 ou 1101.

Code Aiken

- C'est la deuxième représentation qui est en code Aiken.

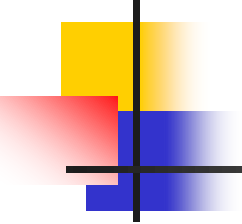
C'est un code 2421 pondéré. Pour les chiffres décimaux 0, 1, 2, 3, 4, il correspond avec le code BCD 8421, tandis que pour les nombres décimaux 5, 6, 7, 8, 9, il concorde avec les nombres 11, 12, 13, 14, 15 du code binaire pur.

Code Aiken



Nombre décimal	Code 8421				Code Aiken			
	(8) D	(4) C	(2) B	(1) A	(2) D'	(4) C'	(2) B'	(1) A'
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1

Code Aiken

- 
- **Le code aiken 2421 à la propriété d'être autocomplémentaire.**
Nous obtenons le complément à 9 des nombres codifiés en remplaçant les 0 par des 1 et les 1 par des 0

Exemple :
le complément à 9 de 6 est égale à 3.

6 = 1100
3 = 0011.

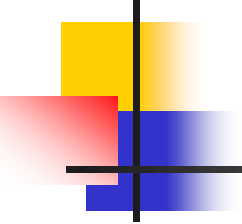


Remarques code AIKEN

- **L'écriture du chiffre 3 est complémentaire du chiffre 6 ; les 0 du chiffre 6 deviennent des 1 pour le chiffre 3 et vice versa.**

Nous remarquons également que la dernière variable pondérée 2 dans ce code est égale à 0 si le nombre est inférieur à 5 et égale à 1 quand le nombre est supérieur à 5.

Code Gray

- 
- **Le code Gray est fréquemment utilisé dans les capteurs angulaires ou de positionnement, mais aussi lorsque l'on désire une progression numérique binaire sans parasite transitoire.**
 - **Le code Gray sert également dans les tableaux de Karnaugh utilisés lors de la conception de circuits logiques.**

Code binaire réfléchi ou code Gray

- Le code Gray est un code construit de telle façon qu'à partir du chiffre 0 chaque nombre consécutif diffère du précédent immédiat d'un seul digit.
- De plus, on opère de telle manière que le digit de transformation soit d'un poids faible. Si une erreur survient lors d'une transformation d'un nombre à un autre elle est ainsi minimisée.

La construction du code Gray

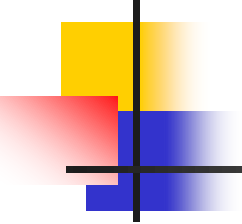
Nombre décimal	Code binaire pur				Code Gray			
	B4	B3	B2	B1	G4	G3	G2	G1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

Règle de formation du code Gray à partir du binaire pur

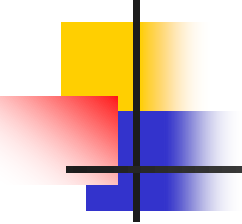
- Exemple :
soit $N = 0111$,
nous avons $2N = 1110$ (pour multiplier par 2 on effectue un décalage de la droite vers la gauche).
On effectue maintenant l'opération OU
Exclusif de N et $2N$:

$$\begin{array}{r} 0111 \\ \oplus 1110 \\ \hline 1001 \end{array}$$

Suite Binaire vers GRAY

- 
- Puis nous divisons par 2 le résultat soit $1001 / 2 = 0100$ (pour diviser par 2 on effectue un décalage de la gauche vers la droite).
 - Nous avons alors :
pour $N = 0111$ en binaire pur
correspond $n = 0100$ en code Gray.

Code excess 3

- 
- Le code excess 3 ou code plus 3 appelé aussi code STIBIZ du nom de son inventeur est un code non pondéré issu du code 8421 auquel on ajoute systématiquement 3 à chaque chiffre.

Le code excess 3 est souvent utilisé sur des unités arithmétiques qui calculent en système numérique décimal plutôt qu'en système binaire.



Code excess 3

- **Ce code permet d'effectuer les opérations arithmétiques d'addition et de soustraction avec un minimum de fonctions logiques.**
- **La complexité du code excess 3 pour d'autre applications nous dictera d'utiliser un transcodeur à l'entrée et à la sortie de l'unité arithmétique lorsque celle-ci est utilisée dans un système logique complexe.**

comparaison entre le code 8421 et le code excess 3.

Nombre décimal	Code 8421				Code plus 3			
	(8) D	(4) C	(2) B	(1) A	D'	C'	B'	A'
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0



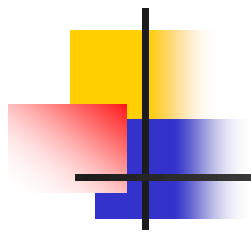
Distance de Hamming

La distance de Hamming est le nombre de bits qui changent entre deux mots d'un code.

Exemple

$$d_H(0010, 1000) = 2$$

$$d_H(000, 111) = 3$$



Distance minimale d'un code

La distance minimale d'un code est le nombre minimal de bits qui changent entre deux mots valides d'un code. La distance de Hamming du code binaire naturel est de 1 car il existe au moins deux mots ayant une distance de Hamming de 1.



Relation fondamentale

La théorie de l'information stipule que:

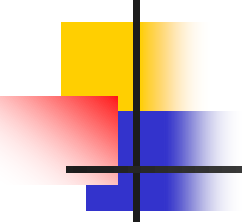
$$M - 1 = C + D$$

où: M = Distance minimale du code

D = Nombre maximal d'erreurs détectables

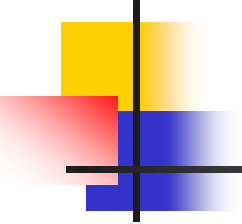
C = Nombre maximal d'erreurs corrigibles

Remarque



On ne peut pas corriger plus d'erreurs qu'on en détecte donc:

$$C \leq D$$



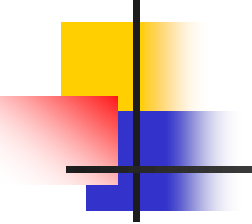
Exemple:

Par conséquent, pour qu'un code soit capable de détecter et de corriger une erreur de transmission:

$$C = D = 1$$

$$M - 1 = C + D$$

$$M = 3$$



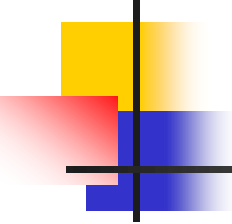
Les codes détecteurs d'erreurs, correcteurs d'erreurs

Codes p parmi n

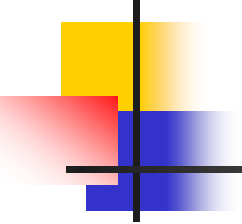
Le code p parmi n est un code à n bits dont p bits sont à 1 et (n-p) bits à 0.

Le nombre de combinaisons répondant à cette définition est égale à C_n^p .

Le code postal utilise un code 3 parmi 5



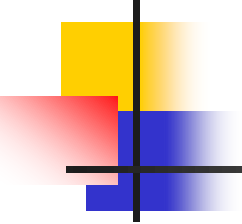
	b_4	b_3	b_2	b_1	b_0
0	0	0	1	1	1
1	0	1	0	1	1
2	0	1	1	0	1
3	0	1	1	1	0
4	1	0	0	1	1
5	1	0	1	0	1
6	1	0	1	1	0
7	1	1	0	0	1
8	1	1	0	1	0
9	1	1	1	0	0



CODE DE HAMMING.

I PRESENTATION :

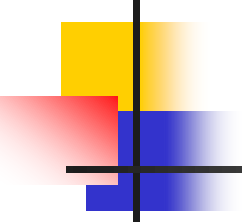
Le code de Hamming est utilisé dans les transmissions de données car il permet de détecter et de corriger une erreur survenue dans un bloc transmis.



Principe du codage.

On fixe un entier k et on code chaque bloc de $m = 2^k - k - 1$ bits de données par un bloc de $n = 2^k - 1$ bits en ajoutant donc k bits, dits de correction, à certaines positions au bloc de m bits. Le tableau suivant indique les nombres de bits de correction, de données pour différentes valeurs de k .

$k=3$	$m=4$	$n=7$
$k=4$	$m=11$	$n=15$
$k=5$	$m=26$	$n=31$

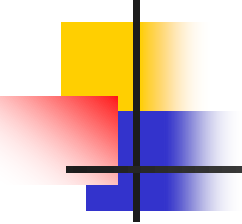


Position des k bits de correction :

Dans la suite de l'étude, on retient $k=3$.

Les k bits de correction sont placés dans le bloc envoyé aux positions d'indice une puissance de 2 en comptant à partir de la gauche. Ainsi, en notant k_1 k_2 k_3 les bits de correction et m_1 m_2 m_3 m_4 les bits de données, le bloc envoyé est :

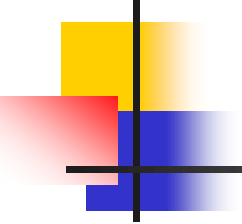
$$A = k_1 k_2 m_1 k_3 m_2 m_3 m_4.$$



Calcul des k bits de correction :

Les k bits de correction sont calculés en utilisant une matrice de parité H, représentée ci-dessous pour k=3.

$$H = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \end{matrix}$$



Remarque :

La colonne i de la matrice représente en binaire la valeur de i .

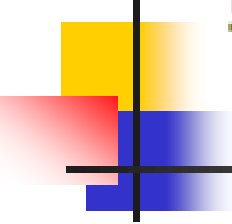
Les k bits de correction sont tels qu'en considérant le vecteur $A = \begin{pmatrix} k1 \\ k2 \\ m1 \\ k3 \\ m2 \\ m3 \\ m4 \end{pmatrix}$, on ait $H.A = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Les groupes de parité

On obtient ainsi 3 équations scalaires que doivent vérifier les k bits de correction :

$$\begin{cases} k_1 + m_1 + m_2 + m_4 = 0_{\text{modulo } 2} \\ k_2 + m_1 + m_3 + m_4 = 0_{\text{modulo } 2} \\ k_3 + m_2 + m_3 + m_4 = 0_{\text{modulo } 2} \end{cases}$$

Remarque : les opérations d'addition sont faites à modulo 2. Par exemple : $1+0=1$;
 $1+1=0$; $1+1+1=1$; etc.



Réception des données et vérification :

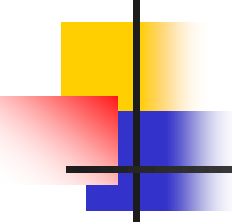
On reçoit le bloc $C = c_1 c_2 c_3 c_4 c_5 c_6 c_7$ qui peut être différent du bloc A si il y a eut des perturbations sur la ligne.

Si on considère qu'il n'y a eut qu'une seule erreur de transmission, alors on peut écrire :

$C = A + E$ où E est un bloc contenant 6 bits à 0 et 1 bit à 1.

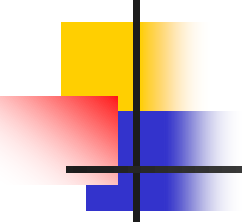
Les positions des 0 et du 1 sont inconnues dans le bloc.

Correction de l'erreur



On calcule le vecteur S tel que : $S = \begin{pmatrix} s1 \\ s2 \\ s3 \end{pmatrix} = H.C = H.(A + E) = H.A + H.E = H.E$.

Finalement, S est une des colonnes de la matrice de parité dont l'indice nous donne la position de l'erreur dans le bloc C . L'erreur est corrigée en changeant le bit considéré d'état.

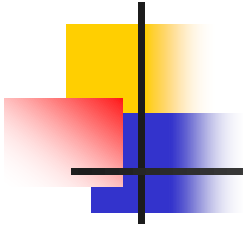


Position de l'erreur

$s_3 s_2 s_1$ est le code binaire de position de l'erreur dans le bloc C que nous obtenons à partir des équations suivantes :

$$\begin{cases} s_1 = (c_1 + c_3 + c_5 + c_7)_{\text{modulo } 2} \\ s_2 = (c_2 + c_3 + c_6 + c_7)_{\text{modulo } 2} \\ s_3 = (c_4 + c_5 + c_6 + c_7)_{\text{modulo } 2} \end{cases}$$

Si $s_3 = s_2 = s_1 = 0$, alors il n'y a pas eu d'erreur.



FIN

CHAPITRE 1

Systèmes de numération et codage