UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR
COLLÈGE STEE
SCIENCES ET TECHNOLOGIES POUR L'ÉNERGIE ET L'ENVIRONNEMENT

L2 Informatique 2024/25

INFORMATIQUE INDUSTRIELLE

Compléments
NUMÉRATION - CODAGE

<u>NUMÉRATION</u> ≡ représentation écrite des nombres

Il existe **4 types de système de numération** en électronique numérique :

- Système **binaire** *(base 2)* avec les 2 symboles {0,1} (**BI**nary digi**T** = **BIT**)
- Système **décimal** *(base 10)* avec les 10 symboles {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- Système **octal** *(base 8)* avec les 8 symboles {0, 1, 2, 3, 4, 5, 6, 7}
- Système **hexadécimal** *(base 16)*
  avec les 16 symboles {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

<u>Représentation d'un nombre N dans une base B quelconque</u>

Le nombre N dans une base B s'écrit en juxtaposant des symboles:

$$(N)_B=(A_nA_{n-1}A_{n-2}\ldots\ldots A_1A_0)_B$$

- A_n est le chiffre le plus significatif *(en binaire : **MSB** Most Significant Bit)*
- A_n est le chiffre de rang n et de poids B^n
- A_0 est le chiffre le moins significatif *(en binaire : **LSB** Least Significant Bit)*
- A_0 est le chiffre de rang 0 et de poids $B^0 = 1$

Conversion d'un nombre N écrit en base B à N écrit en décimal

$$(N)_B \Rightarrow (N)_{10}$$

$(N)_B=(A_nA_{n-1}A_{n-2} \ldots..A_1A_0)_B$

On utilise la **forme développée** de $(N)_B$:

$(N)_{10}= A_n \, . \, B^n + A_{n-1} \, . \, B^{n-1} + A_{n-2} \, . \, B^{n-2} + \ldots + A_1 \, . \, B^1 + A_0 \, . \, B^0$

(avec $B^0 = 1$)

Conversion en décimal $(N)_B \rightarrow (N)_{10}$

Exemples :

$N = (1FF)_{16} = (\ ?\ )_{10} = 1.16^2 + 15.16^1 + 15.16^0 = 256 + 240 + 15 = (511)_{10}$

$N = (777)_8 = (\ ?\ )_{10} = 7.8^2 + 7.8^1 + 7.8^0 = 448 + 56 + 7 = (511)_{10}$

$N = (111111111)_2 = (\ ?\ )_{10}$

$= 1.2^8 + 1.2^7 + 1.2^6 + 1.2^5 + 1.2^4 + 1.2^3 + 1.2^2 + 1.2^1 + 1.2^0$

$= 2^9 – 1$ (plus grand nombre binaire de 9 bits) $= 512 - 1 = (511)_{10}$

$N = (10)_B = (\ ?\ )_{10} = (B)_{10}$

$(10)_2 = (2)_{10}$ $(10)_8 = (8)_{10}$ $(10)_{16} = (16)_{10}$

Conversion décimal / binaire $(N)_{10} \Rightarrow (N)_2$

- Méthode 1 : on effectue des divisions successives par 2
- Méthode 2 : on effectue des soustractions par 2

Remarque: il faut connaître les puissances de 2
(2^{10} = 1024 = $1K_{info}$)

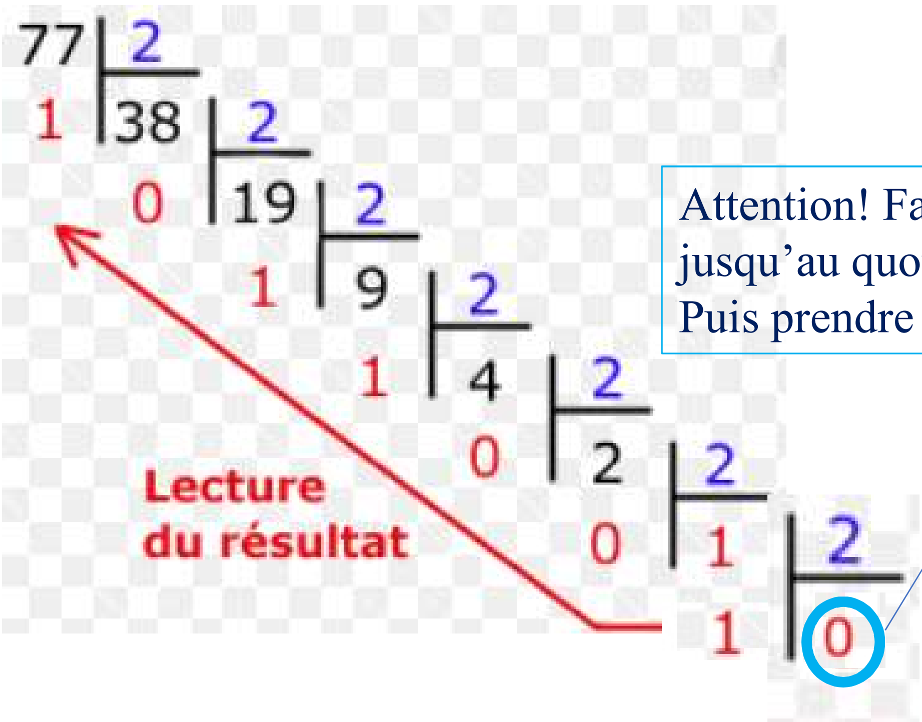
Conversion en décimal / binaire $(N)_{10} \rightarrow (N)_2$

- Méthode 1 : on effectue des divisions successives par 2

Exemples :

$N = (77)_{10} = (?)_2$

$\Rightarrow N = (\textcolor{red}{1001101})_2$



Attention! Faire des divisions successives jusqu'au quotient nul !
Puis prendre les restes du bas vers le haut

<u>Conversion décimal en binaire</u> $(N)_{10} \Rightarrow (N)_2$

- <u>Méthode 2</u> : on effectue des soustractions par 2

Exemples :

$N = (77)_{10} = (?)_2$

$N = 64 + 8 + 4 + 1 = 2^6 + 2^3 + 2^2 + 2^0$ ⇨ $N = (\mathbf{1001101})_2$

Conversion en binaire / hexadécimal $(N)_2 \Rightarrow (N)_{16}$

On procède en faisant des paquets de **4** bits en partant du LSB.

Remarque: $2^4 = 16$

Exemples :

$N = (10110100111)_2$

$$N = \underbrace{101}_{5}\ \underbrace{1010}_{A}\ \underbrace{0111}_{7}$$

$\Rightarrow N = (5A7)_{16}$

<u>Conversion en binaire / octal</u>

$(N)_2 \Rightarrow (N)_8$

On procède en faisant des paquets de **3** bits <u>en partant du LSB</u>.

Remarque: $2^3 = 8$

Exemples :

$N = (10110100111)_2$

$$N = \underbrace{\underline{010}}_{2}\ \underbrace{\underline{110}}_{6}\ \underbrace{\underline{100}}_{4}\ \underbrace{\underline{111}}_{7}$$

$\Rightarrow N = (2647)_8$

Conversion décimal / hexadécimal

$(N)_{10} \rightarrow (N)_2 \rightarrow (N)_{16}$

Passage intermédiaire par la base 2 et de donner l'équivalent en base 16 sur des paquets de 4 bits (ou faire des divisions successives par 16 !)

Exemples :

$N = (255)_{10} = (?)_{16}$

$N = 256 - 1 = 2^8 - 1 = (11111111)_2$

$N = \underbrace{1111}_{F}\ \underbrace{1111}_{F}$

⇨ $N = (FF)_{16}$

Conversion décimal / octal

$(N)_{10} \nearrow (N)_2 \searrow (N)_8$

Passage intermédiaire par la base 2 et de donner l'équivalent en base 8 sur des paquets de 3 bits (ou faire des divisions successives par 8 !)

Exemples :

$N = (255)_{10} = (?)_8$

$N = 256 - 1 = 2^8 - 1 = (11111111)_2$

$N = \underbrace{11}_{3}\ \underbrace{111}_{7}\ \underbrace{111}_{7}$

⇨ $N = (377)_8$

Codes pondérés

- **C**ode **B**inaire **N**aturel ***(CBN)*** :
  - ⇨ ***Pondération : 8 4 2 1*** sur 4 bits

Exemples de conversion décimal-binaire :

$N = (13)_{10} = 8 + 4 + 1 = 2^3 + 2^2 + 2^0 = (1101)_2$

$N = (93)_{10} = 64 + 16 + 8 + 4 + 1 = 2^6 + 2^4 + 2^3 + 2^2 + 2^0 = (1011101)_2$

Codes décimaux binaires pondérés

- **Code BCD** (ou DCB) : **Décimal Codé Binaire**
  - *code décimal binaire* (*sert à coder les chiffres 0,1,…,9*)
  - **Principe** : chaque chiffre du nombre décimal est codé sur **4 bits en CBN**
  - ***Pondération: 1,2, 4, 8, 10, 20, 40, 80, 100, 200, 400, 800 etc…***

Exemples de conversion décimal-binaire :

$N = (13)_{10} = \underbrace{0001}_{1}\ \underbrace{0011}_{3}$

$N = (93)_{10} = \underbrace{1001}_{9}\ \underbrace{0011}_{3}$

Attention ! : les « 0 » à gauche sont nécessaires en BCD
Il y a 6 combinaisons interdites :
1010; 1011; 1100; 1101; 1110; 1111

Codes décimaux binaires NON pondérés

- **Code Excess 3** (XS3 ou Stibitz):

➡ code décimal binaire utilisé pour la représentation des nombres en base 10

Decimal	8 4 2 1 BCD	+ 3	XS 3
0	0000	+0011	0011
1	0001	+0011	0100
2	0010	+0011	0101
3	0011	+0011	0110
4	0100	+0011	0111
5	0101	+0011	1000
6	0110	+0011	1001
7	0111	+0011	1010
8	1000	+0011	1011
9	1001	+0011	1100

$(2)_{XS3} = (7)'$

$(8)_{XS3} = (1)'$

Le codage en Excess 3 permet d'optimiser les calculs avec des nombres négatifs

Exemple de conversion décimal binaire en Code Excess 3 :

Soit le nombre décimal à coder en Excess 3:

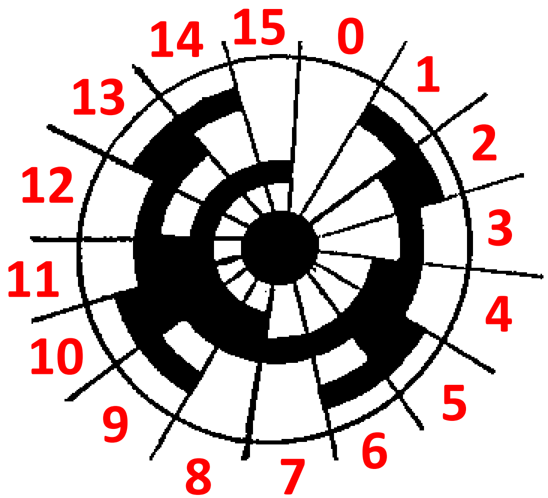
$$N = (12587) \quad = \mathbf{(\underset{1}{0100}\ \underset{2}{0101}\ \underset{5}{1000}\ \underset{8}{1011}\ \underset{7}{1010})_{XS3}}$$

➡ Pas de pondération pour le code XS3

Codes NON Pondérés

❑ Code Binaire Réfléchi ou code Gray :

- Le code Gray sert souvent dans des situations où d'autres codes, comme le Code Binaire Naturel, peuvent produire des résultats ambigus ou erronés au moment de transitions entraînant le changement de plusieurs bits dans le code.



Disque de codage Gray (4 bits)
16 secteurs d'angle

- **code <u>continu</u>** :
  Les combinaisons successives du code Gray sont <u>adjacentes</u>
  c'est-à-dire qu'1 seul bit change d'une combinaison à la suivante

- **code <u>cyclique</u>** : **la dernière combinaison est adjacente à la première**

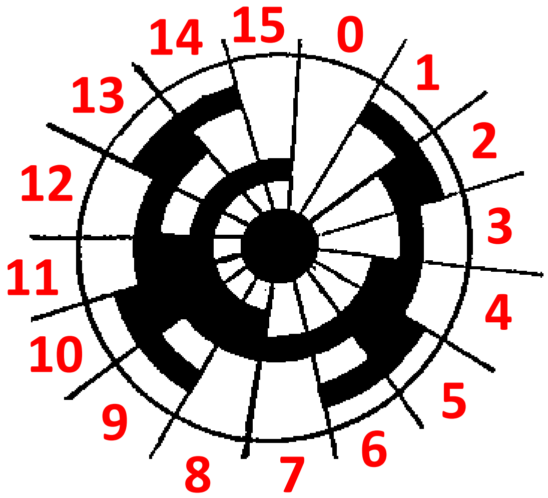
- très utilisé dans les **capteurs de position linéaire ou angulaire**

Code Binaire Réfléchi ou code Gray :

- Le disque de codage Gray présente des parties transparentes et d’autres opaques empêchant la lumière de passer (partie blanche correspond à état 0 et partie noire à l'état 1).

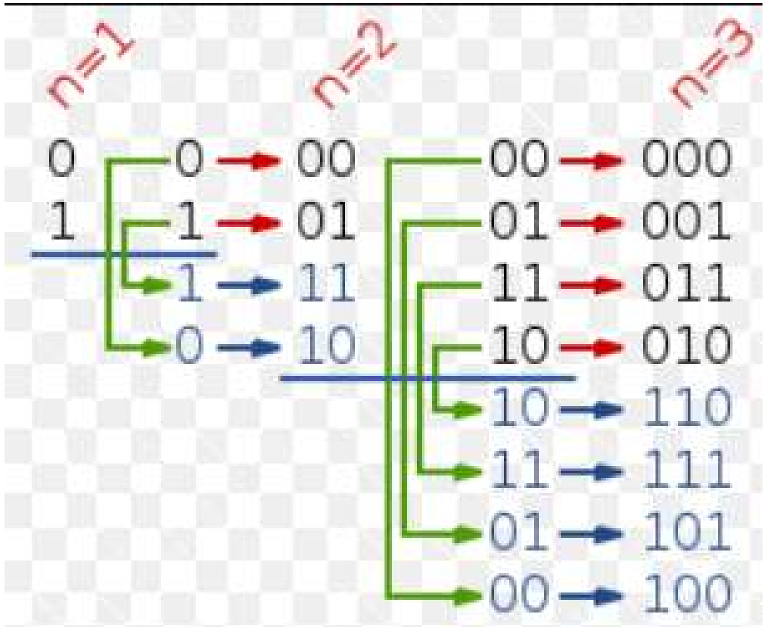
- Ce disque solidaire par exemple de l'axe d'un moteur permettra de connaître la position angulaire de l'arbre.

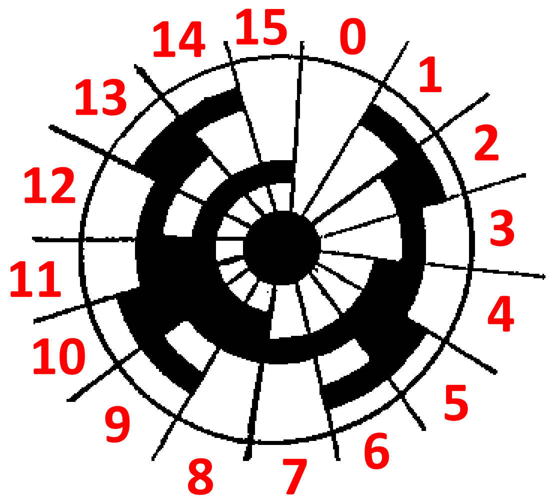
- Il est positionné entre 4 diodes émettrices et 4 photo-capteurs permettra un codage Gray de 16 positions.



Disque de codage Gray (4 bits)
16 secteurs d’angle

❑ Code Binaire Réfléchi ou code Gray :





Disque de codage Gray (4 bits)
16 secteurs d'angle

➡ **Table de correspondance CBN – Code Gray**
Construction à l'aide d'axe de symétrie

❑ Code Binaire Réfléchi ou code Gray :

➡ Table de correspondance CBN – Code Gray

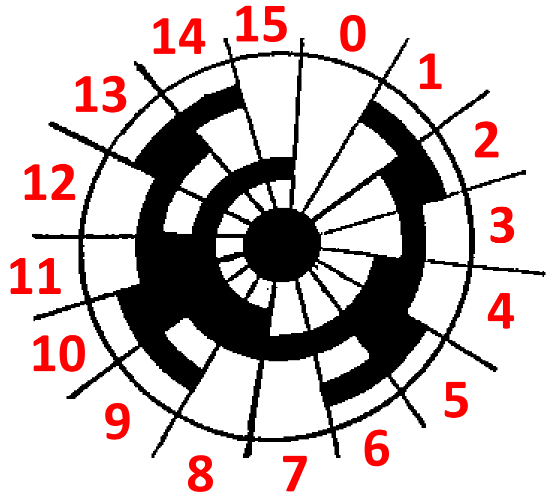
Construction à l'aide d'axe de symétrie

Nombre décimal	Code binaire pur				Code Gray			
	B_4	B_3	B_2	B_1	G_4	G_3	G_2	G_1
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

Axe 1 symétrie

Axe 2 symétrie

Axe 2 symétrie



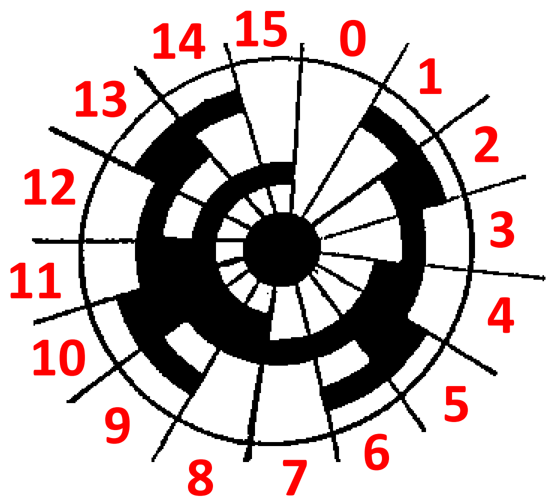
Disque de codage Gray (4 bits)
16 secteurs d'angle

❑ Code Binaire Réfléchi ou code Gray :

- Table de correspondance CBN – Code Gray
- Construction à l'aide d'axe de symétrie

CD / AB	00	01	11	10
00	0000	0001	0011	0010
01	0100	0101	0111	0110
11	1100	1101	1111	1110
10	1000	1001	1110	1010

Angle q (°)	Code Gray			
0/360	0	0	0	0
22.5	0	0	0	1
45	0	0	1	1
67.5	0	0	1	0
90	0	1	1	0
112.5	0	1	1	1
135	0	1	0	1
157.5	0	1	0	0
180	1	1	0	0
202.5	1	1	0	1
225	1	1	1	1
247.5	1	1	1	0
270	1	0	1	0
292.5	1	0	1	1
315	1	0	0	1
337.5	1	0	0	0



Disque de codage Gray (4 bits)
16 secteurs d'angle

❑ Code Binaire Réfléchi ou code Gray :

Exercice : Réaliser un transcodeur CBN ↔ code GRAY

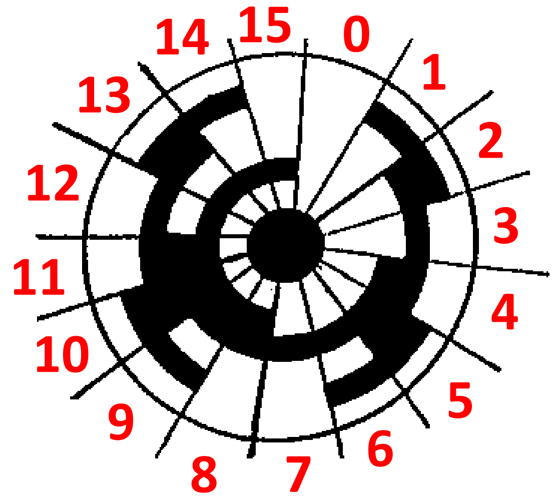
1) Compléter la table de transcodage donnant la correspondance entre le code binaire naturel et le code Gray. Ce code est-il pondéré ?
2) Donner un schéma logique simple d'un transcodeur sur 4 bits: code CBN → code Gray
3) Donner un schéma logique réalisant le transcodage inverse: code Gray → code CBN
4) En généralisant donner la conversion de N = $(11010)_{CBN}$ = $(?)_{GRAY}$

❑ Code Binaire Réfléchi ou code Gray :

Exercice : Réaliser un transcodeur CBN ↔ code GRAY

1) Compléter la table de transcodage donnant la correspondance entre le code binaire naturel et le code Gray. Ce code est-il pondéré ?

Angle (°)	Code Gray				CBN			
0/360	0	0	0	0	0	0	0	0
22.5	0	0	0	1	0	0	0	1
45	0	0	1	1	0	0	1	0
67.5	0	0	1	0	0	0	1	1
90	0	1	1	0	0	1	0	0
112.5	0	1	1	1	0	1	0	1
135	0	1	0	1	0	1	1	0
157.5	0	1	0	0	0	1	1	1
180	1	1	0	0	1	0	0	0
202.5	1	1	0	1	1	0	0	1
225	1	1	1	1	1	0	1	0
247.5	1	1	1	0	1	0	1	1
270	1	0	1	0	1	1	0	0
292.5	1	0	1	1	1	1	0	1
315	1	0	0	1	1	1	1	0
337.5	1	0	0	0	1	1	1	1



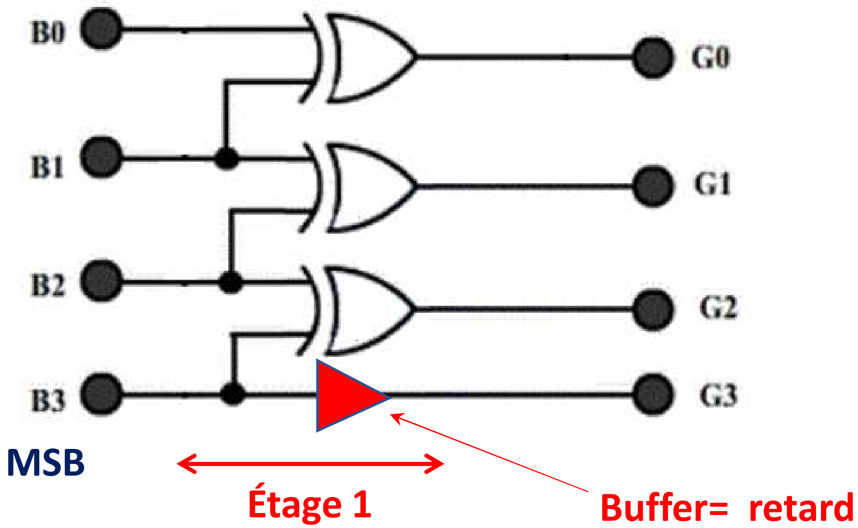
Disque de codage Gray (4 bits)
16 secteurs d'angle

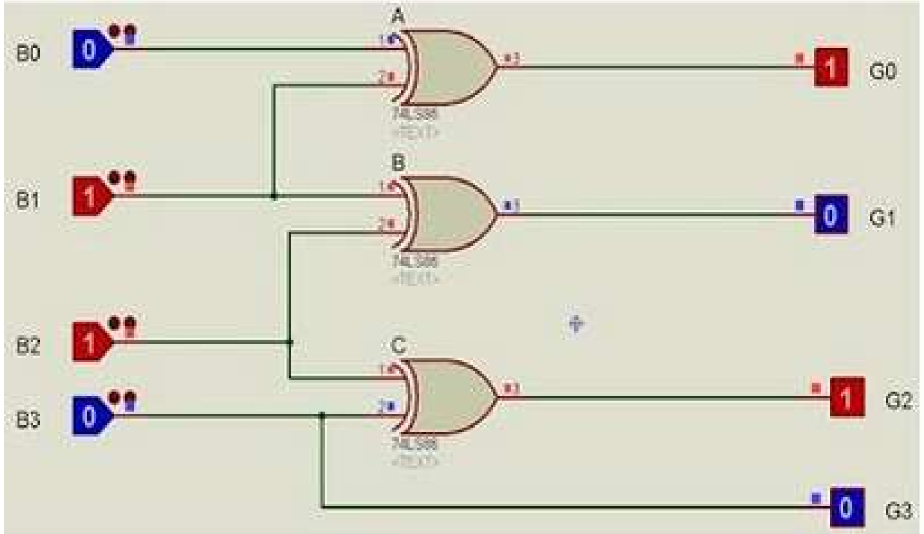
❑ Code Binaire Réfléchi ou code Gray :

Exercice : Réaliser un transcodeur CBN ↔ code GRAY

2) Donner un schéma logique simple d'un transcodeur sur 4 bits: code CBN → code Gray

Binary to Gray converter





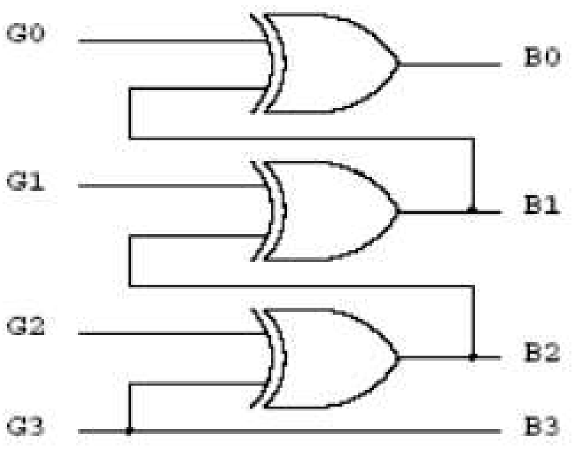
⇨ Réalisation du circuit de transcodage à l'aide de 3 portes logiques XOR à 2 entrées

❑ Code Binaire Réfléchi ou code Gray :

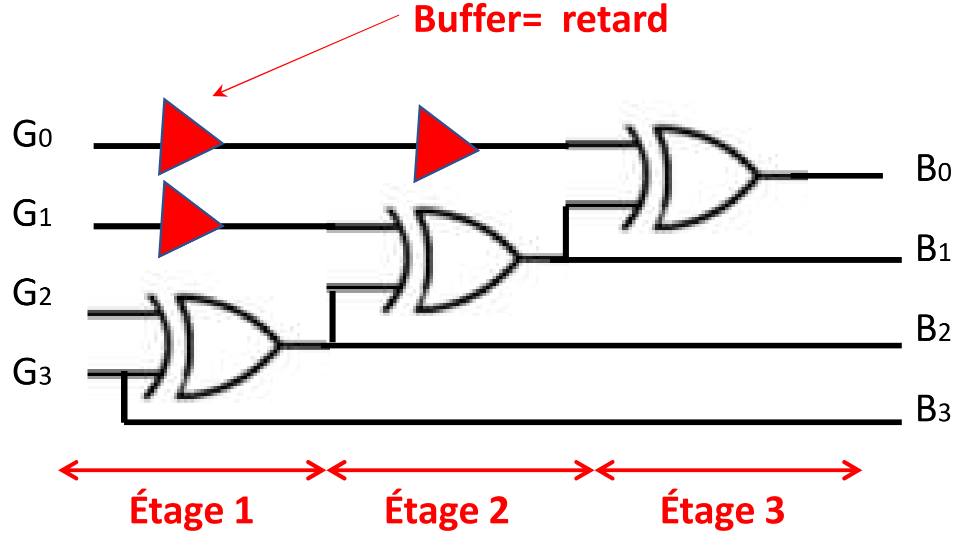
Exercice : Réaliser un transcodeur CBN ↔ code GRAY

3) Donner un schéma logique réalisant le transcodage inverse: code Gray → code CBN

Gray to binary converter



MSB



⇨ Réalisation du circuit de transcodage à l'aide de 3 portes logiques XOR à 2 entrées

❑ Code Binaire Réfléchi ou code Gray :

Exercice : Réaliser un transcodeur CBN $\leftrightarrow$ code GRAY

4) En généralisant donner la conversion de $N = (11010)_{CBN} = (\ ?)_{GRAY}$

$$N = \mathbf{11010} = (B_4B_3B_2B_1B_0)_{CBN} = ? = (G_4G_3G_2G_1G_0)_{GRAY}$$

- MSB : $G_4 = B_4 = 1$
- $B_4 = \mathbf{1} \Rightarrow G_3 = \overline{B_3} = \overline{1} = 0$
- $B_3 = \mathbf{1} \Rightarrow G_2 = \overline{B_2} = \overline{0} = 1$
- $B_2 = \mathbf{0} \Rightarrow G_1 = B_1 = 1$
- $B_1 = \mathbf{1} \Rightarrow G_0 = \overline{B_0} = \overline{0} = 1$

$$\left.\right\} N = \mathbf{11010} = (B_4B_3B_2B_1B_0)_{CBN} = \mathbf{10111} = (G_4G_3G_2G_1G_0)_{GRAY}$$

Codes NON Pondérés

❑ Code Binaire Alphanumérique : Code ASCII

⇨ Le code **ASCII** (***A****merican* ***S****tandard* ***C****ode* ***f****or* ***I****nformation* ***I****nterchange)* permet de coder les caractères alphanumériques.

⇨ Chaque symbole est codé sur 8 bits (dont un de parité).
Ce code est utilisé par les réseaux informatiques assurant les connexions entre les ordinateurs et des organes périphériques (clavier, imprimante, visualisation …)

⇨ Ce code est **non pondéré**, sa table de correspondance permet de trouver le code ASCII des principaux caractères.

❑ Code ASCII

⇨ Table de correspondance

Exemple :
code ASCII
du caractère **A**

$b_7b_6b_5b_4b_3b_2b_1$
$= (100\ 0001)_2$
$=$ **\$41** $=$ **0x0041**

	Digits				0	1	2	3	4	5	6	7
				b_7	0	0	0	0	1	1	1	1
				b_6	0	0	1	1	0	0	1	1
				b_5	0	1	0	1	0	1	0	1
	b_4	b_3	b_2	b_1								
0	0	0	0	0	NUL	DLE	SP	0	'	P	@	p
1	0	0	0	1	SOH	DC1	!	1	A	Q	a	q
2	0	0	1	0	STX	DC2	"	2	B	R	b	r
3	0	0	1	1	ETX	DC3	#	3	C	S	c	s
4	0	1	0	0	EOT	DC4	$	4	D	T	d	t
5	0	1	0	1	ENQ	NAK	%	5	E	U	e	u
6	0	1	1	0	ACK	SYN	&	6	F	V	f	v
7	0	1	1	1	BEL	ETB	'	7	G	W	g	w
8	1	0	0	0	BS	CAN	(	8	H	X	h	x
9	1	0	0	1	TAB	EM	)	9	I	Y	i	y
A	1	0	1	0	LF	SS	*	:	J	Z	j	z
B	1	0	1	1	VT	ESC	+	;	K	[	k	{
C	1	1	0	0	FF	FS	,	<	L	~	l	|
D	1	1	0	1	CR/E0B	GS	-	=	M	]	m	}
E	1	1	1	0	SO	RS	.	>	N	^	n	\
F	1	1	1	1	SI	US	/	?	O	_	o	DEL

❑ Code ASCII

Exercice :
Quelle opération arithmétique doit-on réaliser pour convertir le code ASCII d'un caractère minuscule vers le même caractère en majuscule ? Comment réaliser la conversion inverse ?

« minuscule » = « MAJUSCULE » + $20

« MAJUSCULE » = « minuscule » **- $20**

« MAJUSCULE » = « minuscule » **+ $60**

Savez-vous représenter en binaire des nombres signés ?

$20 = $(0010\ 0000)_2$

$20 = 010 0000 (*sur format 7 bits*)

- $20 = 101 1111 + 1 = 110 0000 **= $60**

« b » = $62

« B » = $62 +$60

$62 = 0110 0010

$60 = 0110 0000

= **1 100 0010** = 100 0010 =$42