61Systèmes logiques

Sommaire

1) Les systèmes logiques :

1.1) Définitions :

1.1.1) Système logique :

Un système logique est un ensemble de relations causales qui traduisent son fonctionnement.Le système logique utilise des signaux d'entrée et génère des signaux de sortie. Ces signaux sont du type TOR (tout ou rien) utilisant une information binaire (1 ou 0).

1.1.2) Système logique combinatoire :

Un système logique est qualifié de combinatoire lorsqu'à une combinaison des variables d'entrée ne correspond qu'une seule combinaison des variables de sortie.

1.1.3) Variable binaire :

On appelle binaire, une variable prenant ses valeurs dans l'ensemble $\{0,1\}$. Un état binaire est appelé « bit » (contraction de binary digit)

1.1.4) Equation logique :

On appelle équation logique une combinaison de plusieurs variables logiques donnant l'état d'une variable de sortie associée. Cette combinaison est réalisée à l'aide d'opérations logiques. Exemple : $S = a + (b \cdot c)$

1.1.5) Table de vérité :

La table de vérité représente l'état de la variable de sortie pour chacune des combinaisons des variables d'entrée. S'il y a n variables d'entrée, la table de vérité comporte 2^n combinaisons différentes d'entrées.

1.2) Algèbre de Boole :

1.2.1) Définition :

Soit l'ensemble B constitué des deux éléments $B = \{0,1\}$.

- 0 : élément nul
- 1 : élément identité

Cet ensemble présente une structure d'algèbre avec les lois suivantes :

- $\bar{x} = 0$ si $x = 1$ et vice versa (fonction NON ou complément)
- $x \cdot y = 1$ si et seulement si $x = 1$ et $y = 1$ (fonction ET ou produit booléen)
- $x + y = 1$ si au moins l'une des variables vaut 1 (fonction OUou somme booléenne)

1.2.2) Propriétés :

Commutativité	$x \cdot y = y \cdot x$	$x + y = y + x$
Eléments neutres	$x \cdot 1 = 1 \cdot x = x$	$x + 0 = 0 + x = x$
Eléments absorbants	$x \cdot 0 = 0 \cdot x = 0$	$x + 1 = 1 + x = 1$
Eléments symétriques	$x \cdot \bar{x} = \bar{x} \cdot x = 0$	$x + \bar{x} = \bar{x} + x = 1$
Idempotence	$x \cdot x = x$	$x + x = x$
Distributivité	$x \cdot (y + z) = x \cdot y + x \cdot z$	$x + (y \cdot z) = (x + y) \cdot (x + z)$
Absorption	$x \cdot (x + y) = x$	$x + (x \cdot y) = x$

1.2.3) Théorèmes de De Morgan :

Premier théorème : Le complément d'une somme logique est égal au produit des termes complémentés :

$$\overline{x+y}=\bar{x}\cdot\bar{y}$$

Deuxième théorème : Le complément d'un produit logique est égal à la somme des termes complémentés :

$$\overline{x\cdot y}=\bar{x}+\bar{y}$$

Généralisation : $\overline{x_1\cdot x_2\cdot\ldots\cdot x_n}=\bar{x}_1+\bar{x}_2+\ldots+\bar{x}_n$ et $\overline{x_1+x_2+\ldots+x_n}=\bar{x}_1\cdot\bar{x}_2\cdot\ldots\cdot\bar{x}_n$

1.3) Fonctions logiques :

Tout opérateur logique ou toute fonction logique peuvent être définis et représentés de diverses manières par des :

- Expressions mathématiques (littérales ou symboliques) : $S=a$ OU b ; $S=a+b$
- Tables de vérité
- Représentations graphiques (logigrammes, chronogrammes, schémas à contacts)

1.3.1) Table de vérité, équations logiques:

Une fonction logique à n variables est entièrement définie quand on connaît sa valeur 0 ou 1 pour les 2^n combinaisons possibles. Pour faire l'inventaire de toutes ces combinaisons on dresse un tableau ordonné appelé table de vérité.

On retrouve dans la colonne des entrées les 2^n combinaisons binaires possibles entre les n variables, dans la dernière colonne, on définit les valeurs des sorties pour chaque combinaison des variables d'entrées.

a	b	c	S
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Cette table de vérité permet de déterminer l'équation logique de la fonction.

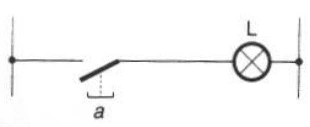
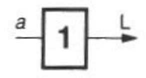
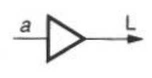
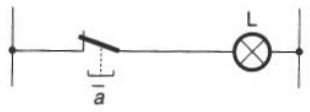
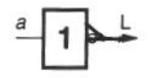
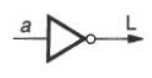
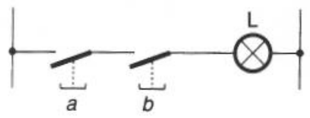
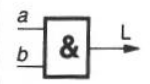
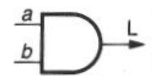
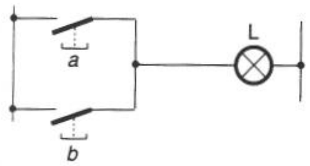
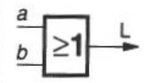
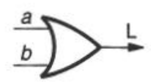
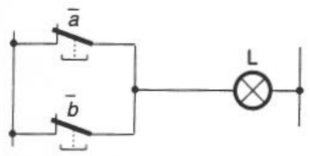
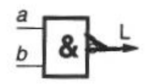
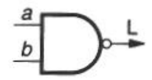
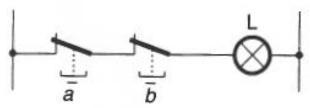
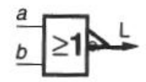
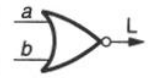
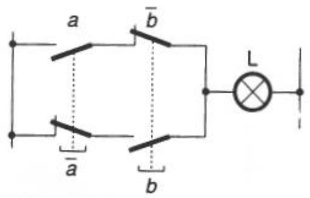
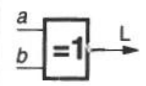
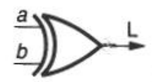
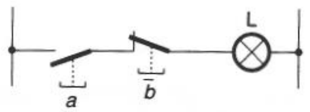
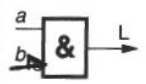
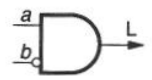
Méthode :

- Pour toutes les lignes où $S = 1$ écrire la combinaison des entrées a, b, c correspondant, à l'aide de l'opérateur ET (exemple ligne 2 : $\bar{a}\cdot\bar{b}\cdot c$)
- Associer l'ensemble des combinaisons avec l'opérateur OU.

Ici : $S=\bar{a}\cdot\bar{b}\cdot\bar{c}+\bar{a}\cdot\bar{b}\cdot c+\bar{a}\cdot b\cdot c+a\cdot\bar{b}\cdot\bar{c}+a\cdot\bar{b}\cdot c$

1.3.2) Représentation graphique des fonctions et opérateurs logiques :

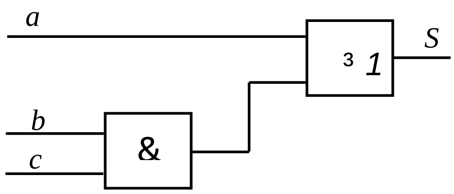
a) Représentation des fonctions :

<table>
<tr><th>Fonction</th><th>Équation</th><th>Table de vérité</th><th>Schéma électrique</th><th>Symbole N.F.</th><th>Symbole américain</th></tr>
<tr><td>OUI</td><td>$L = a$</td><td><table><tr><th>a</th><th>L</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>NON</td><td>$L = \bar{a}$</td><td>action physique / niveau logique<table><tr><th>a</th><th>$\bar{a}$</th><th>L</th></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>ET</td><td>$L = a \cdot b$</td><td><table><tr><th>a</th><th>b</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>OU</td><td>$L = a + b$</td><td><table><tr><th>a</th><th>b</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>NAND
ET-NON-</td><td>$L = \overline{a \cdot b}$</td><td><table><tr><th>a</th><th>b</th><th>$a \cdot b$</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td><td>
$L = \overline{a \,.\, b} = \bar{a} + \bar{b}$</td><td></td><td></td></tr>
<tr><td>NOR
OU-NON</td><td>$L = \overline{a + b}$</td><td><table><tr><th>a</th><th>b</th><th>$a + b$</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table></td><td>
$L = \overline{a + b} = \bar{a} \,.\, \bar{b}$</td><td></td><td></td></tr>
<tr><td>OU
exclusif</td><td>$L = a \oplus b$
$L = a\bar{b} + \bar{a}b$</td><td><table><tr><th>a</th><th>b</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td><td></td><td></td><td></td></tr>
<tr><td>NON
inhibition</td><td>$L = a \cdot \bar{b}$</td><td><table><tr><th>a</th><th>$\bar{b}$</th><th>L</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td><td></td><td></td><td></td></tr>
</table>

b) Logigramme :

Le logigramme est une représentation graphique d’une fonction logique à l’aide de symboles logiques (appelés également « portes logiques »).

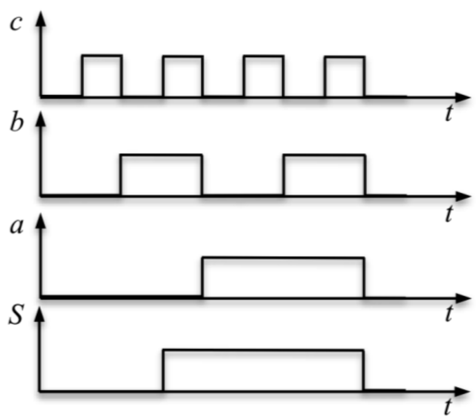
<u>Exemple</u> : Pour $S = a + (b \cdot c)$ voici le tracé du logigramme avec une porte « ET » et une porte « OU »



c) chronogrammes :

C’est un tracé qui montre l’évolution temporelle (axe des abscisses) de la sortie en fonction de celles des entrées.

<u>Exemple</u> : chronogrammes pour la fonction : $S = a + (b \cdot c)$



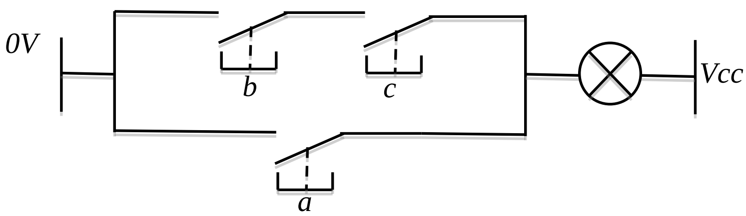
Sur le chronogramme, l’instant où une variable a passe de « *0* » à « *1* » est qualifié de front montant , il est noté$\uparrow a$.

A l’inverse l’instant ou une variable passe de « *1* » à « *0* » est qualifié de front descendant, il est noté$\downarrow a$.

d) Schéma à contacts :

La réalisation électrique d’un système combinatoire est représenté par un schéma à contacts plus approprié. D’une manière générale, les entrées sont représentées par des interrupteurs (contacteurs) et les sorties par des lampes.

<u>Exemple</u> : schéma à contacts pour la fonction : $S = a + (b \cdot c)$



2) Système de numération : Codage de l'information :

2.1) Base de numération:

2.1.1) Définitions :

- Transcodage : Opération qui consiste à remplacer un mot codé par un mot dans un type de codage différent.
- Code pondéré : Code pour lequel chaque caractère est affecté d'un « poids ». exemple : binaire, décimal, hexadécimal,…
- Code détecteur d'erreurs : Pour détecter des transmissions de code avec des erreurs, on ajoute parfois un ou plusieurs bits permettant d'effectuer une vérification. Exemple : On ajoute un bit appelé « bit de parité » de façon à ce que le nombre de bits égal à « 1 » d'un mot binaire soit paire. Si ce n'est pas le cas pour un code « lu », c'est qu'il est donc erroné.
- Code auto-correcteur : Code pour lequel l'information lue peut être identifiée comme « bonne». Dans le cas contraire cette information peut s'auto-corriger. Exemple: code de Hamming.
- Code à barres : Code représenté graphiquement par une succession de barres noire et d'espaces blanc de grande ($bit = 1$) ou de faible largeur ($bit = 0$) . il existe de nombreux codes à barres du fait de leur simplicité de lecture par lecteur laser et de leur fiabilité.

2.1.2) Bases de numérisation usuelle , les codes pondérés :

Une base B de numération est un ensemble de B symboles. Les trois systèmes de base « B » utilisés en sciences industrielles sont les bases 10, 2 et 16 appelés respectivement décimale, binaire et hexadécimale.

Tout nombre entier positif peut être représenté par une expression de la forme :

$$N_{(B)} = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} ... + a_1 \cdot b^1 + a_0 = \sum_{i=0}^{n} a_i \cdot b^i$$

avec $a_i \in \{0,1,\ldots,b-1\}$ et $a_n \neq 0$

On utilise la notation condensée équivalente : $N_{(B)} = (a_n a_{n-1} .. a_0)_B$

<u>Exemples</u> : $N1_{(16)} = (E2F)_{16}$; $N2_{(10)} = (3807)_{10}$; $N3_{(2)} = (1010)_2$

Ce sont des codes pour lesquels chaque caractère est affecté d'un « poids ».

<u>Exemple</u> : $N2_{(10)} = (3807)_{10} = 3 \cdot 10^3 + 8 \cdot 10^2 + 0 \cdot 10^1 + 7 \cdot 10^0$

Les nombres binaires sont souvent composés d'un grand nombre de bits. On préfère généralement les exprimer dans les systèmes octal($b = 8$) et hexadécimal($b = 16$), car la conversion avec le système binaire est simple.

Décimal	Binaire	Hexadécimal	Octal
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17

2.1.3) Changement de base :

L'objectif est de mettre en place la correspondance entre l'expression d'un nombre dans une base(ex : B_{10}) et son expression dans une autre base (ex : B_2).

Il existe deux méthodes :

- Pratiquer des divisions successives .
- Créer un masque avec les puissances successives de la base désirée. On commence ensuite par placer le chiffre adéquat sous les digits de poids immédiatement inférieur au nombre à transcrire, et ainsi de suite avec des poids de plus en plus faibles jusqu'à "*1*".

Exemples :

- Méthode du masque :

$(1854)_{10}$ à transcoder en binaire naturel $(N)_2$

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	
2048	*1024*	*512*	*256*	*128*	*64*	*32*	*16*	*8*	*4*	*2*	*1*	*poids*
	1	*1*	*1*	*0*	*0*	*1*	*1*	*1*	*1*	*1*	*0*	

1854-1024 = 830 ; 830-512 = 318 ; 318-256 = 62 ; 62-32 = 30 ; 30-16 = 14 ; 14-8 = 6 ; 6-4 = 2 ; 2-2 = 0.

Donc : $(1854)_{10} = (11100111110)_2$

$(1854)_{10}$ à transcoder en hexadécimal $(N)_{16}$

16^3	16^2	16^1	16^0	
4096	*256*	*16*	*1*	*poids*
	7	*3*	*E*	

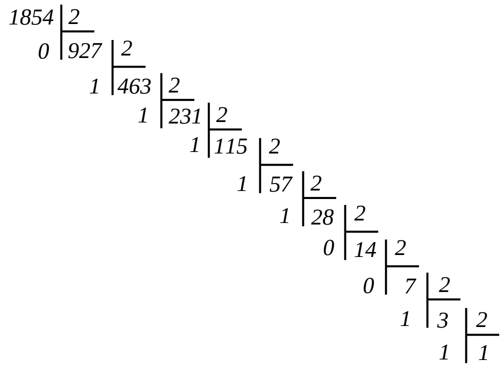
1854-7x256 = 62 ; 62-3x16 = 14 ; $(14)_{10} = (E)_{16}$

Donc : $(1854)_{10} = (73E)_{16}$

- <u>Méthode des divisions successives :</u>

$(1854)_{10}$ à transcoder en binaire naturel $(N)_2$

On effectue des divisions successives par "2".



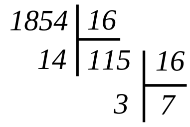
Donc : $(1854)_{10} = (11100111110)_2$

$(1854)_{10}$ à transcoder en hexadécimal $(N)_{16}$

On effectue des divisions successives par "16".

On lit , "7" puis "3" puis "14" que l'on doit transcrire en base 16.

$(1854)_{10} = (73E)_{16}$



2.2) Codage de l'information :

2.2.1) Binaire naturel :

Il se prête parfaitement au traitement des opérations arithmétiques. Il faut cependant une grande quantité de bits pour exprimer un nombre dès que celui-ci est élevé : un mot binaire de 4 bits (appelé **quartet**) ne pourra représenter qu'un nombre décimal compris entre 0 et 15 puisqu'il ne peut différencier que 2^4 nombres différents.

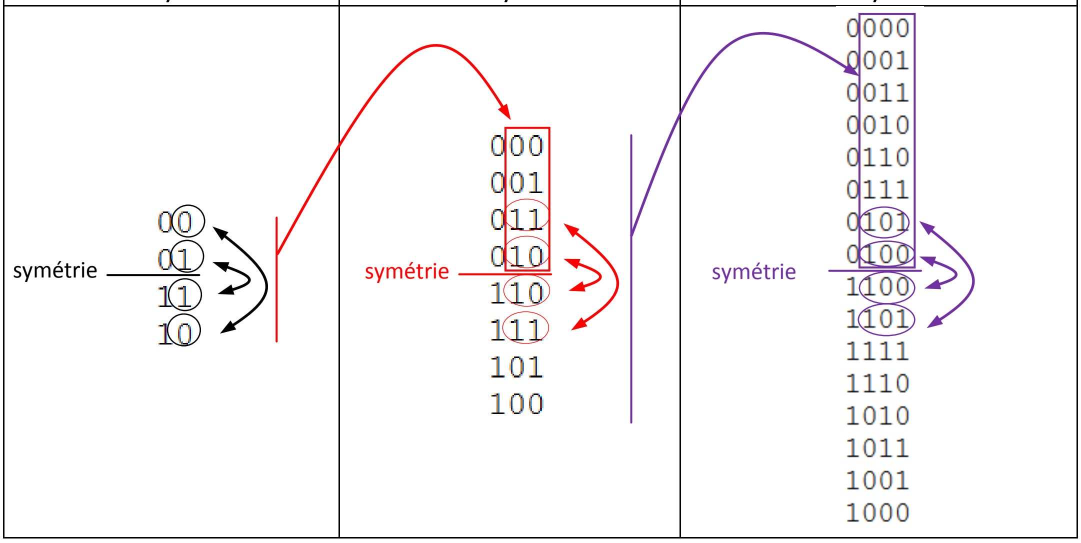
Un mot binaire de 8 bits (appelé **octet**) ne pourra représenter qu'un nombre décimal compris entre 0 et 255puisqu'il ne peut différencier que 2^8 nombres différents.

Les puissances successives de 2(…128, 64, 32, 16, 8, 4, 2, 1) sont appelés poids binaires.

2.2.2) Binaire réfléchi (code Gray) :

Le code binaire réfléchi, aussi appelé code Gray[1] est un système binaire où deux valeurs successives ne diffèrent que d'un seul bit. Initialement prévu pour éviter les erreurs, il équipe aujourd'hui entre autres les codeurs absolus (voir §3.1.2.);

Code Gray sur 2 bits	Code Gray sur 3 bits	Code Gray sur 4 bits



[1] Frank Gray (1887-1969) physicien chercheur chez Bell labs (USA) inventeur du code qui porte son nom apparu officiellement en mars 1953 "pulse code communication" U.S. patent

2.2.3) Hexadécimal

Ce format est largement utilisé en informatique car il offre une conversion facile avec le système binaire.
Il est en effet possible de traduire du binaire vers de l'hexadécimal et réciproquement, groupe de chiffres par groupe de chiffres. Cette caractéristique s'explique par le fait que 16 (nombre de chiffres dans la base hexadécimale) est lui-même une puissance de 2 (nombre de chiffres de la base binaire)
Cette égalité revient à dire que :
« À un chiffre dans la base 16, correspondent exactement quatre chiffres dans la base 2. »

Cette facilité de conversion a conduit la notation hexadécimale à être utilisée pour noter des nombres initialement quantifiés ou à destination d'être quantifiés en binaire, l'hexadécimal étant plus compact (quatre fois moins de chiffres) et offrant une meilleure lisibilité pour l'œil humain.

Un avantage supplémentaire de la base 16 est sa concordance avec l'octet (appelé aussi byte, à ne pas confondre avec bit).

Exemple de transcodage binaire vers hexadécimal :

binaire	1.0101.1010.1010.1100.1111.0111						
regroupé par 4	1	0101	1010	1010	1100	1111	0111
regroupé en hexadécimal	1	5	A	A	C	F	7
hexadécimal	15AACF7						
(Décimal)	22719735						

2.2.4) Autres codages

a) 3 parmi 5 :

Ce code consiste à choisir 3 bits à 1 parmi 5 bits. Le nombre de combinaisons vaut :

$$\binom{5}{3} = \frac{5\,!}{3\,! \cdot (5-3)!} = 10$$

0	00111
1	01011
2	01101
3	01110
4	10011
5	10101
6	10110
7	11001
8	11010
9	11100

Il permet donc de coder les dix chiffres décimaux

- Avantages :
  - Auto-détection d'erreurs de lecture
  - Codage personnalisable (10 possibilités)
- Inconvénient :
  - La taille du codage n'est pas optimale, toute opération nécessite un décodage préalable.

Ce code est utilisé à la poste pour la lecture du code postal

b) Décimal codé binaire : DCB:

Chaque chiffre décimal est codé en binaire sur 4 bits

<u>Exemple :</u> codage de $(2014)_{10}$

2	0	1	4
0010	0000	0001	0100

16 bits sont ici utilisés ; alors que $(2014)_{10}$ en binaire naturel nécessite 11 bits: $(2014)_{10} = (11111011110)_2$

- <u>Avantages</u> :
  - le code est plus proche de la base 10,
  - les opérations peuvent être adaptées.

- <u>Inconvénient</u> :
  - la taille n'est pas optimale

Ce code est utilisé sur les afficheurs 7 segments. Chaqueafficheur reçoit le chiffrecodé en binaire sur 4 bits

C)Code ASCII(American Standard Code for Information Interchange):

Norme d'encodage informatique des caractères alphanumériques de l'alphabet latin. La norme **ASCII** (on prononce phonétiquement "aski") établit une correspondance entre une représentation binaire des caractères de l'alphabet latin et les symboles, les signes, qui constituent cet alphabet. Par exemple, le caractère "*a*" est associé à "*01100001*" et "*A*" à "*01000001*".

La norme **ASCII** permet ainsi à toutes sortes de machines de stocker, analyser et communiquer de l'information textuelle. En particulier, la quasi totalité des ordinateurs personnels et des stations de travail utilisent l'encodage **ASCII**.

Le codage **ASCII** est souvent complété par des correspondances supplémentaires afin de permettre l'encodage informatique d'autres caractères, comme les caractères accentués par exemple. Cette norme s'appelle **ISO-8859** et se décline par exemple en **ISO-8859-1** lorsqu'elle étend l'**ASCII** avec les caractères accentués d'Europe occidentale.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

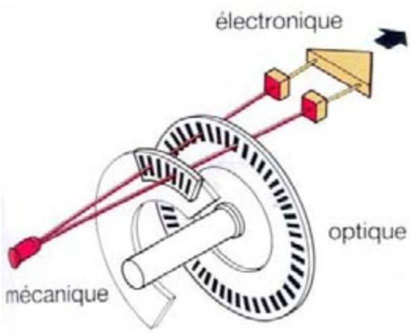
3)Exemples :

3.1) Codeurs:

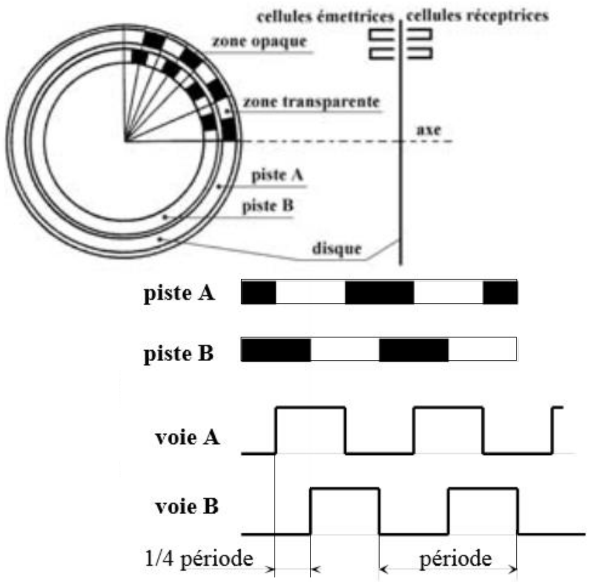
3.1.1) Codeurs incrémentaux:

C'est un capteur de position angulaire comportant :

- Un disque optique mobile avec 2 pistes (A et B) comportant chacune une succession de parties opaques et transparentes.
- Deux cellules fixes, pour chaque piste : une cellule émettrice de lumière d'un côté et une réceptrice de l'autre.



Chaque passage d'une zone transparente à une autre est détecté par les cellules réceptrices.
Les 2 pistes sont décalées d'un quart de période et la rotation du disque donne les signaux suivants :



Le comptage-décomptage des impulsions par l'unité de traitement permet de définir la position du mobile.

<u>Résolution :</u>

On note N la résolution et n le nombre de points par tour.

Pour déterminer N, trois cas peuvent se présenter :

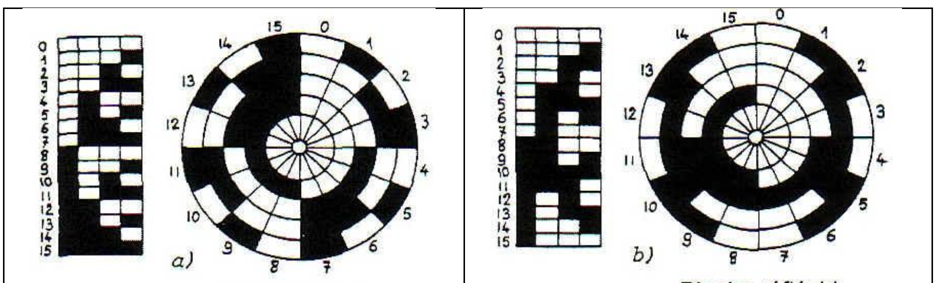
- Le système de traitement n'utilise que les fronts montants de la voie A (exploitationsimple)N est égal au nombre de points (n).
- Le système de traitement utilise les fronts descendants et montants de la voie A(exploitation double) alors N est multipliée par 2 ($N=2 \times n$).
- Le système de traitement utilise les voies A et B (exploitation quadruple) alors N est multipliée par 4 ($N=4 \times n$)

3.1.2) Codeurs absolus :

Le disque des codeurs absolus comporte un nombre « n » de pistes concentriques divisées en segments égaux alternativement opaques et transparents.

Un codeur absolu délivre en permanence un code qui est l'image de la position réelle du mobile à contrôler.

codeur 12 pistes : 4096 positions



Binaire naturel

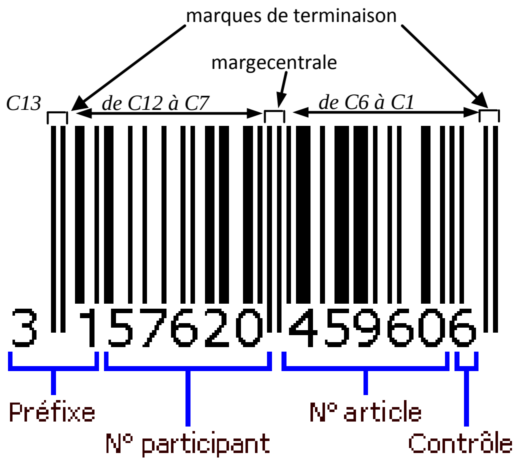
Binaire réfléchi

3.2) Codes barres :

3.2.1) Code EAN :

Le code EAN est un code :

- Numérique : chaque caractère est un chiffre de 0 à 9,
- Continu : les caractères sont représentés les uns à la suite des autres sans séparation,
- Bidirectionnel : sa lecture peut être faite dans les deux sens,
- De format fixe : il comprend 8 ou 13 chiffres (code EAN 8 ou EAN 13)



Le code est délimité par deux marques de terminaison. Entre celles-ci, le code est structuré en deux zones de même taille séparées par une marge centrale. Six chiffres sont représentés dans chaque zone, intitulés de *C1* à*C6* pour la zone de droite et de *C7* à *C12* pour la zone de gauche. Le chiffre *C13* se déduit du codage des six chiffres précédents

3.2.2) Code 2D :

Le code DataMatrixest un code bidimensionnel à haute densité, permettant de représenter une quantité importante d'informations sur une surface réduite.

Il se compose d'un motif extérieur permettant le repérage des lignes et des colonnes et d'une matrice de données située à l'intérieur.

Grâce à l'écriture redondante des informations, il offre unniveau de sécurité maximal : lecture possible d'un symbole partiellement effacé jusqu'à environ 20% de la surface

totale, au contraire d'un code unidimensionnel qui n'offre aucune sécurité si le symbole est dégradé.

La taille du DataMatrix dépend du nombre de caractèresnumériques à coder. Pour un message de 6 chiffres, lamatrice intérieure sera constituée d'un carré de 10 lignes et 10 colonnes. La surface totale du DataMatrix pour un message de 6 chiffres sera alors de $10 mm^2$, alors que le même message codé avec le code barre linéaire 39[2] occuperait plus de $30 mm^2$.

http://fr.wikipedia.org/wiki/Datamatrix

[2] Le Code 39 est utilisé, entre autres, pour le marquage des médicaments en pharmacie, en France et dans certains autres pays européens .Il a été utilisé jusqu'à récemment dans le secteur automobile, constructeurs et équipementiers, en rapport avec les normes internationales.http://fr.wikipedia.org/wiki/Code_39