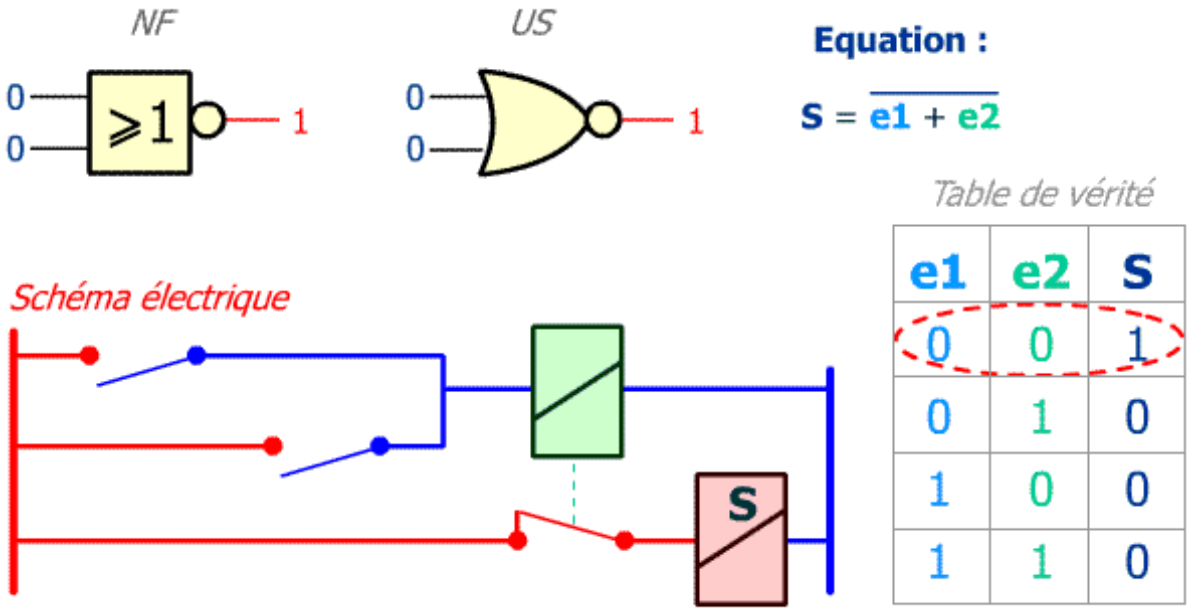
Logique combinatoire



e1	e2	S
0	0	1
0	1	0
1	0	0
1	1	0

EL KACEM EL MOSTAFA

LES SYSTEMES DE NUMERATION ET CODAGE DES INFORMATIONS

A- Systèmes de numération :

De nombreux systèmes de numération sont utilisés en électronique numérique. Les plus courants sont les systèmes de numération suivants :

- *Binaire (Base 2)*
- *Décimal (Base 10)*
- *Hexadécimal (Base 16)*
- *Octal (Base 8)*

Principe d'une base

La base est le nombre qui sert à définir un système de numération. La base du système décimal est 10 alors que celle du système octal est 8. Quel que soit la base numérique employée, elle suit la relation suivante :

$$\sum_{i=0}^{i=n}(b_i a^i)=b_i a^n+.....+b_5 a^5+b_4 a^4+b_3 a^3+b_2 a^2+b_1 a^1+b_0 a^0$$

bi : chiffre de la base de rang i et ai : puissance de la base a d'exposant de rang i

1- les bases

1.1. Système décimal : Base 10

C'est le système utilisé dans la vie courante, il est basé sur le nombre 10. Pour représenter les nombres décimaux, on utilise les chiffres de 0 à 9.
Exemple : Ecriture d'un nombre décimal

$(7348)_{10} = 7.10^3 + 3.10^2 + 4.10^1 + 8.10^0$
$= 7.1000 + 3.100 + 4.10 + 8.1$
$= 7000 + 300 + 40 + 8 = 7348$

1.2. Système binaire : Base 2

C'est le système le plus utilisé en électronique numérique ou digitale. Pour représenter les nombres binaires, on dispose uniquement de deux chiffres 0 et 1.
Tout nombre écrit dans ce système vérifie la relation suivante :

$(10\ 110)_2 = 1.2^4 + 0.2^3 + 1.2^2 + 1.2^1 + 0.2^0$
$(10\ 110)_2 = 1.16 + 0.8 + 1.4 + 1.2 + 0.1$
Donc : $(10110)_2 = (22)_{10}$

1.3. Système octal : Base 8

C'est un système qui contient 8 digits, il contient les chiffres de 0 à 7 et surtout pas de 8 et 9 !

Exemple : Ecriture d'un nombre octal

$(365)_8 = 3.8^2 + 6.8^1 + 5.8^0$

$= 3.64 + 6.8 + 5.1$

$= 192 + 48 + 5 = (245)_{10}$

1.4. Système hexadécimal : Base 16

Dans ce système, la base B vaut 16 et il y a 16 digits :

Les dix premiers digits de 0 à 9 sont les chiffres du système décimal et les digits de 10 à 15 sont les premières lettres majuscules de l'alphabet.

0,1,2,3,4,5,6,7,8,9,A,B,C,D,E et F.

Exemple : Ecriture d'un nombre hexadécimal

$(A2F)\ 16 = A.16^2 + 2.16^1 + F.16^0$

$= 10.256 + 2.16 + 15.1 = 2560 + 32 + 15 = (2607)_{10}$

	Décimal	Binaire	Octal	Hexadécimal
Base	10	2	8	16
Symboles	0 à 9	0 à 1	0 à 7	0 à F
	0	0	0	0
P	1	1	1	1
R	2	10	2	2
O	3	11	3	3
G	4	100	4	4
R	5	101	5	5
E	6	110	6	6
S	7	111	7	7
S	8	1000	10	8
I	9	1001	11	9
O	10	1010	12	A
N	11	1011	13	B
	12	1100	14	C
	13	1101	15	D
	14	1110	16	E
	15	1111	17	F
	16	10000	20	10

2- Conversion entre les bases :

2-1 Conversion des bases 2, 8 ou 16 en base 10

On exploite directement la forme polynomiale de la représentation binaire, octale ou hexadécimale du nombre.

***Exemple 1** :*

$(110101)_2 = 1x2^5+1x2^4+0x2^3+1x2^2+0x2^1+1x2^0 = 32+16+0+4+0+1 = (53)_{10}$

Exemple 2 *:*

$(305)_8 = 3x8^2+0x8^1+5x8^0 = (197)_{10}$

Exemple 3 *:*

$(61F)_{16} = 6x16^2+1x16^1+15x16^0 = (1567)_{10}$

Exemple 4 *:*

$(10110, 01)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$

$= 1 \times 16 + 0 \times 8 + 1 \times 4 + 1 \times 2 + 0 \times 1 + 0 \times 0,5 + 1 \times 0,25$

$= (22, 25)_{10}$

Exemple 5 *:* $(372, 06)_8 = 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 0 \times 8^{-1} + 6 \times 8^{-2}$

$= 3 \times 64 + 7 \times 8 + 2 \times 1 + 0 \times 0,125 + 6 \times 0,015625$

$= (250, 09375)_{10}$

Exemple 6 *:* $(FD, 2A)_{16} = F \times 16^1 + D \times 16^0 + 2 \times 16^{-1} + A \times 16^{-2}$

$= 15 \times 16 + 13 \times 1 + 2 \times 0,0625 + 10 \times 0,00390625$

$= (253, 1640625)_{10}$

2-2 Conversion de la base 10 aux bases 2, 8 et 16

Cette conversion se fait en deux parties :

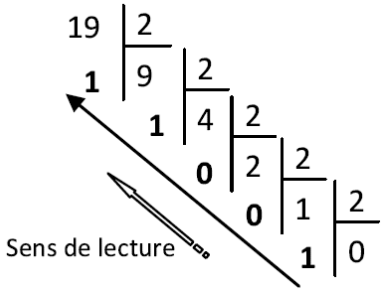
- *La partie entière.*
- *La partie fractionnaire.*

La conversion de la partie entière consiste à répéter la division par (2, 8, ou 16) du nombre décimal à convertir et noter les restes jusqu'à ce que le quotient soit 0.

La conversion de la partie fractionnaire consiste à multiplier le nombre fractionnaire par (2, 8,16) et noter la partie entière et la partie fractionnaire obtenues. Multiplier la partie fractionnaire obtenue sur la ligne précédente par (2, 8,16), et noter le résultat (partie entière et partie fractionnaire).

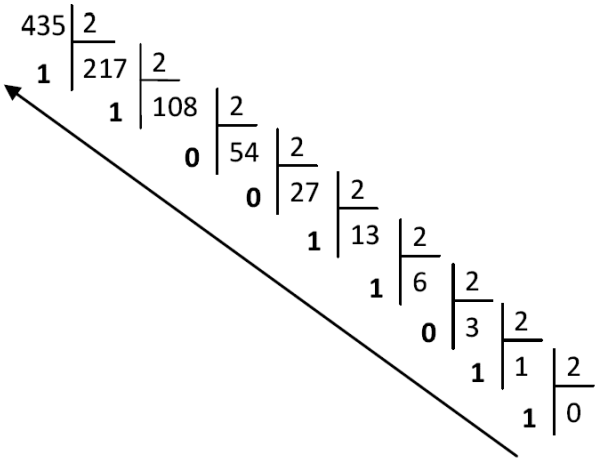
a- Base 10 vers base 2

Exemple 1 : Soit à convertir le nombre 19 en binaire.



$(19)_{10} = (10011)_2$

Exemple 2 : Ecrire $(435)_{10}$ en base deux.



$(435)_{10} = (110110011)_2$

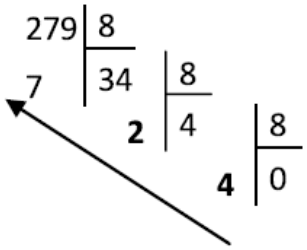
Exemple 3 *: Conversion du nombre 91,2 en base 2*

Conversion de la partie entière				***Conversion de la partie fractionnaire***	
		Position	*Reste*		
91/2	*45*	2^0	***1***	*0,2 x2*	***0**,4*
45/2	*22*	2^1	***1***	*0,4x2*	***0**,8*
22/2	*11*	2^2	***0***	*0,8x2*	***1** ,6*
11/2	*5*	2^3	***1***	*0,6x2*	***1**,2*
5/2	*2*	2^4	***1***	***0,2x2***	
2/2	*1*	2^5	***0***		
1/2	***0***	2^6	***1***		

$(91,2)_{10} = (1011011,00110011\ldots)_2$

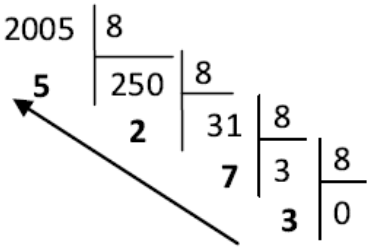
b- Base 10 vers base 8

Exemple 1 :



$(279)_{10} = (427)_8$

Exemple 2 :



$(2005)_{10} = (3725)_8$

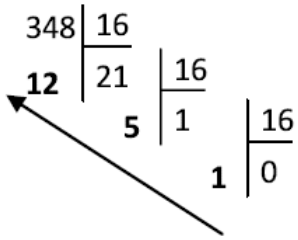
Exemple 3 *: Conversion du nombre 459,3 en base 8*

Conversion de la partie entière				***Conversion de la partie fractionnaire***	
		Position	*Reste*		
459/8	*57*	8^0	***3***	*0,3 x8*	***2**,4*
57/8	*7*	8^1	***1***	*0,4x8*	***3**,2*
7/8	*0*	8^2	***7***	*0,2x8*	***1** ,6*
				0,6x8	***4**,8*
				0,8x8	***6**,4*
				0,4x8	***3**,2*

$(459,3)_{10} = (713,231463146\ldots.)_8$

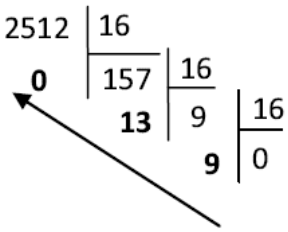
c- Base 10 vers base 16

Exemple1 : Soit à coder le nombre $(348)_{10}$ en hexadécimal.



$(348)_{10} = (15C)_{16}$

Exemple2 : Soit à coder le nombre $(2512)_{10}$ en hexadécimal.



$(2512)_{10} = (9D0)_{16}$

Exemple 3 *: Conversion du nombre 751,1 en base 16*

Conversion de la partie entière				***Conversion de la partie fractionnaire***	
		Position	*Reste*		
751/16	*46*	16^0	***15 (F)***	*0,1 x16*	***1**,6*
46/16	*2*	16^1	***14 (E)***	*0,6x16*	***9**,6*
2/16	*0*	16^2	***2***	***0,6x16***	***9** ,6*

$(751,1)_{10} = (2EF, 1999\ldots.)_{16}$

2-3 Conversion de la base binaire à la base octale et inversement

On obtient l'équivalent octal du nombre binaire en le partageant en tranche de 3 chiffres de droite à gauche pour la partie entière et de gauche à droite pour la partie

fractionnaire, puis en remplaçant chaque tranche par son équivalent octal.

Exemple 1 : $(111000110101)_2 = 111/000/110/101$
$= 7 / 0 / 6 / 5$
$= (7065)_8$

Exemple 2 : $(011010101110,001011100)_2 = 011/010/101/110, 001/011/100$
$= 3/2/5/6, 1/3/4$
$= (3256,134)_8$

Exemple 3 : $(2\ 4\ 3,2\ 1)_8 = 2/4/3,2/1 = 010/100/011,010/001$
$= (010100011,010001)_2$

2- 4 Conversion de la base 2 à la base 16

La conversion de la base 2 à la base 16 (et inversement) se fait aisément, la base 16 étant un multiple entier de la base 2. Elle permet de représenter sous une forme réduite un nombre binaire.
*$2^4 = 16$ => un groupe binaire de 4 bits est **transcodable** directement en un digit hexadécimal.*
***Méthode :** On divise le nombre binaire en tranches de 4 bits (à partir du LSB). Chacun des quartets est ensuite converti en un digit hexadécimal par simple sommation pondérée.*

Exemple 1 : $(111000110101)_2 = 1110 / 0011 / 0101$
$= E / 3 / 5$
$= (E35)_{16}$

Exemple 2 : $(D4C7)_{16} = D / 4 / C / 7$
$= 1101 / 0100 / 1100 / 0111 = (1101\ 0100\ 1100\ 0111)_2$

Exemple 3 : $(0011110100101110\ ,11010100)_2 = 0011/1101/0010/1110,1101/0100$
$= 3/D/2/E, D/4$
$= (3D2E, D4)_{16}$

Exemple 4 : $(B\ 3\ 0,\ 1\ A)_{16} = 1011/0011/0000,0001/1010$
$= (101100110000,00011010)_2$

B- Les Codes

Un système électronique traite les informations en binaire de différentes natures. Par exemple, en traitement de texte, on manipule des caractères ; qui sont codés en un nombre binaire. Cette association s'appelle "Codage" de l'information binaire et permet d'utiliser plusieurs codes suivant le domaine d'application. L'opération inverse s'appelle "Décodage" ou "Transcodage". On étudie en particulier : Le code binaire pur, le code GRAY, le code BCD, le code ASCII et le code à barre.

1- Les codes

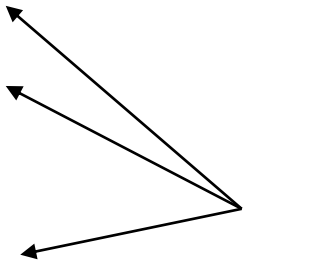
Le code binaire pur

Valeur décimale	Code binaire			
	A3	A2	A1	A0

0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
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

Le code GRAY

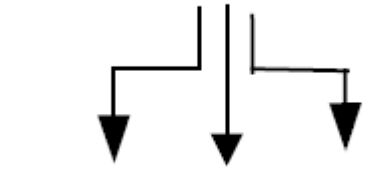
Valeur décimale	**Code binaire**				**Code GRAY**			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0



Code B C D : (Binary coded décimal) en français (Décimal codé Binaire)

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

Exemple : $(115)_{10}$



$= (0001\ 0001\ 0101)_{BCD}$

Le code ASCII :

Le code ASCII (American Standard Code for Information Interchange) est un code qui représente les caractères éditables ou non éditables : éditables parce que l'on peut les éditer comme le caractère "A" et non éditables comme le cratère "Escape" ou "Return". Il est codé sur 7 bits en binaire

D	O	H	C	D	O	H	C	D	O	H	C	D	O	H	C
0	000	00	nul	32	040	20	sp	64	100	40	@	96	140	60	‘
1	001	01	soh	33	041	21	!	65	101	41	A	97	141	61	a
2	002	02	stx	34	042	22	“	66	102	42	B	98	142	62	b
3	003	03	etx	35	043	23	#	67	103	43	C	99	413	63	c
4	004	04	eot	36	044	24	$	68	104	44	D	10	144	64	d
5	005	05	enq	37	045	25	%	69	105	45	E	101	145	65	e
6	006	06	acq	38	046	26	&	70	106	46	F	102	146	66	f
7	007	07	bel	39	047	27	`	71	107	47	G	103	147	67	g
8	010	08	BS	40	050	28	(	72	110	48	H	104	150	68	h
9	011	09	HT	41	051	29	)	73	111	49	I	105	151	69	i
10	012	0A	LF	42	052	2A	*	74	112	4A	J	106	152	6A	j
11	013	0B	VT	43	053	2B	+	75	113	4B	K	107	153	6B	k
12	014	0C	FF	44	054	2C	’	76	114	4C	L	108	154	6C	l
13	015	0D	CR	45	055	2D	-	77	115	4D	M	109	155	6D	m
14	016	0E	SO	46	056	2E	.	78	116	4E	N	110	156	6E	n
15	017	0F	SI	47	057	2F	/	79	117	4F	O	111	157	6F	o
16	020	10	dle	48	060	30	0	80	120	50	P	112	160	70	p
17	021	11	dc1	49	061	31	1	81	121	51	Q	113	161	71	q
18	022	12	dc2	50	062	32	2	82	122	52	R	114	162	72	r
19	023	13	dc3	51	063	33	3	83	123	53	S	115	163	73	s
20	024	14	dc4	52	064	34	4	84	124	54	T	116	164	74	t
21	025	15	nak	53	065	35	5	85	125	55	U	117	165	75	u
22	026	16	syn	54	066	36	6	86	126	56	V	118	166	76	v
23	027	17	etb	55	067	37	7	87	127	57	W	119	167	77	w
24	030	18	can	56	070	38	8	88	130	58	X	120	170	78	x
25	031	19	em	57	071	39	9	89	131	59	Y	121	171	79	y
26	032	1A	sub	58	072	3A	:	90	132	5A	Z	122	172	7A	z
27	033	1B	esc	59	073	3B	;	91	133	5B	[	123	173	7B	{
28	034	1C	fs	60	074	3C	<	92	134	5C	\	124	174	7C	|
29	035	1D	gs	61	075	3D	=	93	135	5D	]	125	175	7D	}
30	036	1E	rs	62	076	3E	>	94	136	5E	^	126	176	7E	~
31	037	1F	us	63	077	3F	?	95	137	5F	-	127	177	7F	del

hex			0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
	dec		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		bin	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	0	0000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	1	0001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	2	0010	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	3	0011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	4	0100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	5	0101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	6	0110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	7	0111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

2- Conversion entre code

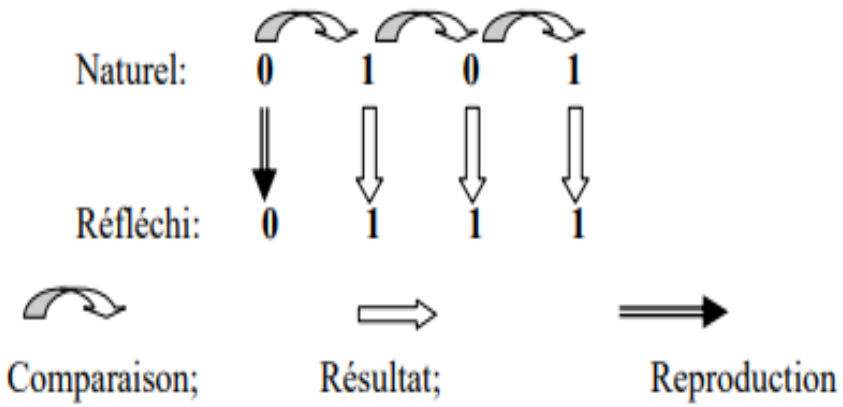
Somme en binaire :

0 + 0 = 0

1 + 0 = 0 + 1 = 1

1 + 1 = 0 report 1

Conversion du code binaire (naturel) en code de Gray (binaire réfléchi)

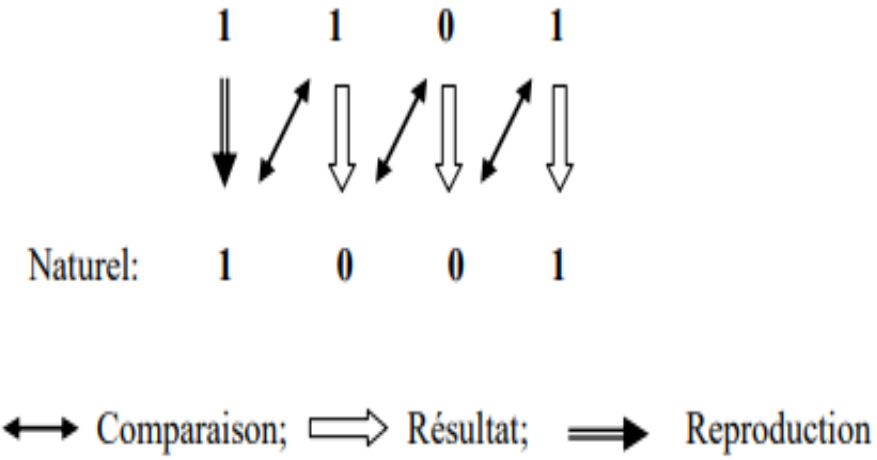


Somme sans report

Exemple :

Nombre Binaire N	**Code Gray**
N = 0101	**0111**
N = 0111	**0100**
N= 001101	**001011**
N= 0100100	**0110110**

Conversion du code de Gray au code binaire



Somme sans report

Exemple :

Code Gray	1101	00010110	10010110	01100111
Code Binaire	**1001**	**00011011**	**11100100**	**01000101**

Exercices : réalisez les conversions suivantes

1- (1251)8 = (……) 10

2- (17,3)8 = (……) 10

3- (96)10 = (……) 8

4- (19,25)10 = (……) 8

5- (10110)2 = (……) 8

6- (100011011000,1101)2 = (……) 8

7- (674)8 = (……) 2

8- (152)8 = (……) 2

9- (FEE) 16 = (……) 10

10- (AC, 2)16 = (……)10

11- (86,31)10 = (……)16

12- (122)10 = (……)16

13- (11011)2 = (……)16

14- (101110111,1100)2 = (……)16

15- (CAFE)16 = (……)2

16- (A25,5E)16 = (……)2

17- (128)10 = (……)BCD

18- (942)10 = (……)BCD

19- (10100110111)BCD = (……)10

20- (1000100111)BCD = (……)10

21- (10111)2 = (……)Gray

22- (11111011101)2 = (……)Gray

23- (10111)Gray = (……)2

24- (110110)Gray = (……)2

C- Opérations arithmétiques avec la base binaire.

L'addition en binaire : Les règles de base sont :

Exemple :

```
   1 1 1
   1101
+
   1011
  11000
```

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 reporte 1

La soustraction en binaire : Les règles de base sont :

Exemple :

```
    11011
-
      110
  10101
```

0 - 0 = 0
0 - 1 = 1 Emprunte 1
1 - 0 = 1
1 - 1 = 0

La multiplication en binaire : Les règles de base sont :

Exemple :

```
    101
  *
    110
   000
  101
 101
 11110
```

*0 * 0 = 0*
*0 * 1 = 0*
*1 * 0 = 0*
*1 * 1 = 1*

La division en binaire : Les règles de base sont :

Exemple :

```
 1010      /  10
-10        |  101 |
 001
-00
   10
  -10
   00
```

0 / 0 = Indéterminé
0 / 1 = 0
1 / 0 = Impossible
1 / 1 = 1

Exercices d'application :

Effectuer les opérations arithmétiques suivantes :

Exemple 1 :

0011	*1101*	*110*	*1111/11*
+	*-*	***	
1101	*0010*	*11*	

Exemple 2 :

10110110	*1010111*	*1001*	*100000/110*
+	*-*	***	
1011101	*10101*	*1100*	

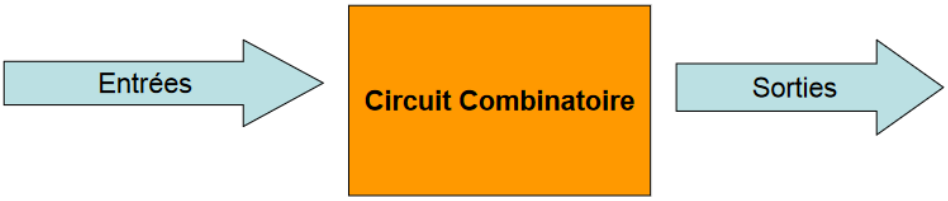
Exemple 3 :

1110111			*1000010/1011*
+			
101101	*10110101*	*1011*	
+	*-*	***	
1101	*1110101*	*1100*	

ALGEBRE DE BOOLE ET FONCTIONS LOGIQUES

Introduction :

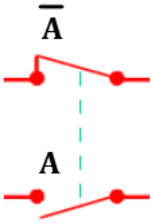
Tous les récepteurs ou automatismes peuvent s'écrire sous forme d'équation en utilisant des variables booléienne qui ont deux états 0 ou 1 donc ce sont des valeurs binaires.
Les fonctions de sortie s'expriment selon des expressions logiques des seules variables d'entrée
Un circuit combinatoire est défini par une ou plusieurs fonctions logiques.



Les variables de sortie sont soit actives (=1) ou inactives (=0)
Exemple (lampe, moteur, alarme, sonnerie, relais, etc…)
Les variables d'entrées sont soit non actionnées (=0) ou actionnées (=1)

Exemple (Interrupteur, bouton poussoir, capteur, etc…)

1- Principales fonctions logiques



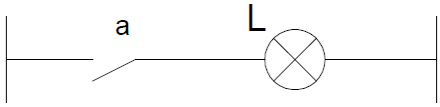
Entrée : porte soit un nom ou alphabet

A : contact NO *$\bar{A}$: Contact NF se lit A barre*

Sortie :

L= 0 *L= 1*

1-1- Fonction Égalité « OUI »

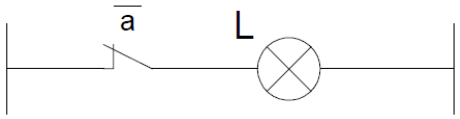


On écrit : F = a ou L = a

La lampe est à l'état 1 (allumée) si, et seulement si, a est à l'état 1 (fermé).

Conclusion : La sortie est à l'état 1 si, et seulement si l'entrée est à l'état 1.

1-2- Fonction inverse « NON »

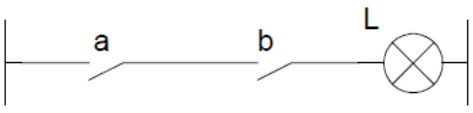


On écrit : $F = \overline{a}$

La lampe L est à l'état 1 (allumé) si, et seulement si, il n'y a pas d'action sur la variable a (donc fermé).

Conclusion : La sortie est à l'état 1 si, et seulement si, l'entrée est à l'état 0.

1-3- Fonction produit logique « ET »

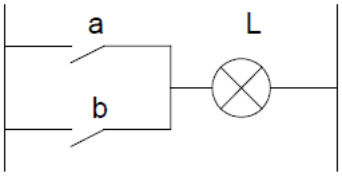


On écrit : F = a • b

La lampe L est à l'état 1 (allumé) si, et seulement si, a ET b sont à l'état 1 (fermés).

Conclusion : La sortie est à l'état 1 si, et seulement si, toutes les entrées sont à l'état1.

1-4- Fonction somme logique « OU »

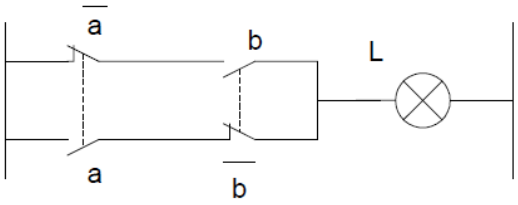


On écrit : F = a + b

La lampe L est à l'état 1 (allumé) si, et seulement si, a OU b sont à l'état 1(fermé).

Conclusion : La sortie est à l'état 1 si, et seulement si, une ou plusieurs entrées sont à l'état 1.

1-5- Fonction « OU EXCLUSIF »

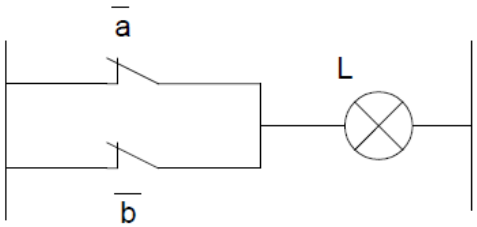


On écrit : F = a ⊕ b = $\bar{a}b + a\bar{b}$

La lampe L est à l'état 1 (allumé) si, et seulement si, il y a une action sur a ou sur b.

Conclusion : La sortie est à l'état 1 si, et seulement si, une seule entrée est à l'état 1.

1-6- Fonction « NON ET (NAND) »

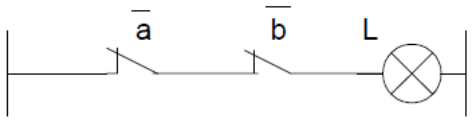


F = $\bar{a} + \bar{b}$ ou F= $\overline{a.b}$

La lampe L est à l'état 0 (éteinte) si, et seulement si, les deux variables a et b sont à l'état 1 (actionnées).

Conclusion : La sortie est à l'état 0 si, et seulement si, toutes les entrées sont à l'état 1.

1-7- Fonction « NON OU (NOR) »



F= $\bar{a}.\bar{b}$ ou F = $\overline{a+b}$

La lampe L est à l'état 1 si, et seulement si, les deux variables $\bar{a}$ et $\bar{b}$ sont à l'état 0 (non actionnées).

***Conclusion** : La sortie est à l'état 1 si, et seulement si, toutes les entrées sont à l'état 0.*

2 - Les règles de l'algèbre de Boole.

2.1- Les lois de l'algèbre de Boole :

	Lois	
Commutativité	L_1	$A \bullet B = B \bullet A$
	L_2	$A + B = B + A$
Associativité	L_3	$(A \bullet B) \bullet C = A \bullet (B \bullet C)$
	L_4	$(A + B) + C = A + (B + C)$
Distributivité	L_5	$A \bullet (B + C) = A \bullet B + A \bullet C$
	L_6	$(A + B) \bullet (A + C) = A + B \bullet C$
Absorption	L_7	$A + (A \bullet B) = A$
	L_8	$A \bullet (A + B) = A$
Expansion	L_9	$(A \bullet B) + (A \bullet \overline{B}) = A$
	L_{10}	$(A + B) \bullet (A + \overline{B}) = A$
De Morgan	L_{11}	$\overline{A \bullet B} = \overline{A} + \overline{B}$
	L_{12}	$\overline{A + B} = \overline{A} \bullet \overline{B}$
	L_{13}	$\overline{\overline{A} + \overline{B}} = A \bullet B$
	L_{14}	$\overline{\overline{A} \bullet \overline{B}} = A + B$
Similitude	L_{15}	$A + \overline{A} \bullet B = A + B$
	L_{16}	$A \bullet (\overline{A} + B) = A \bullet B$

2.2- Les Théorèmes de l'algèbre de Boole :

	Théorèmes	
Invariance	T_1	$A \bullet 0 = 0$
	T_2	$A + 1 = 1$
Élément neutre	T_3	$A \bullet 1 = A$
	T_4	$A + 0 = A$
Idempotence	T_5	$A \bullet A = A$
	T_6	$A + A = A$
Complémentarité	T_7	$A \bullet \overline{A} = 0$
	T_8	$A + \overline{A} = 1$
Involution	T_9	$\overline{\overline{A}} = A$

2.3 - Postulats de l'algèbre de Boole :

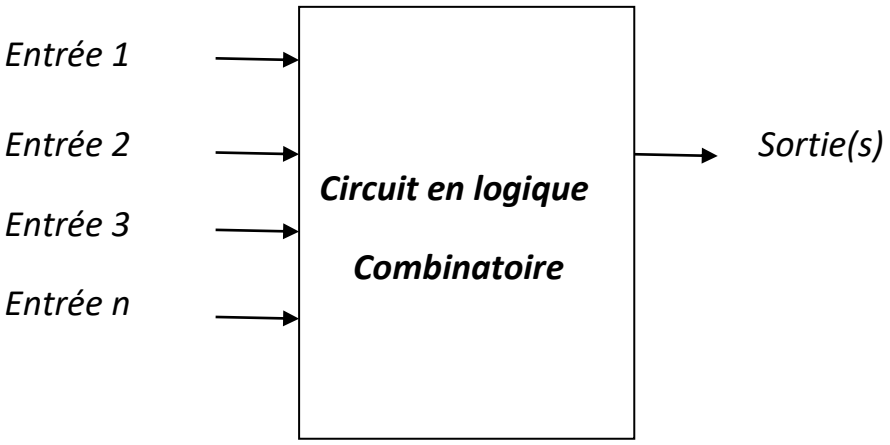
O= fil coupé

1= fil non coupé

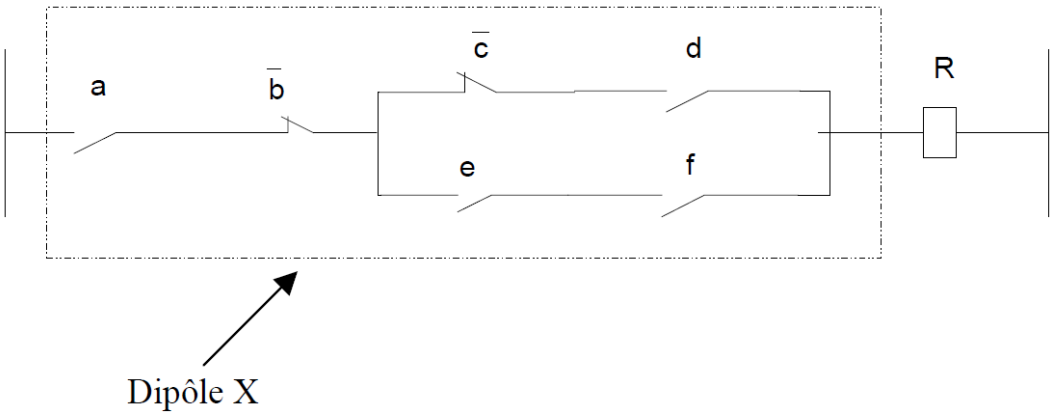
Postulats	
P_1	$0 \bullet 0 = 0$
P_2	$0 \bullet 1 = 1 \bullet 0 = 0$
P_3	$1 \bullet 1 = 1$
P_4	$0 + 0 = 0$
P_5	$1 + 0 = 0 + 1 = 1$
P_6	$1 + 1 = 1$
P_7	$\overline{0} = 1$
P_8	$\overline{1} = 0$

3- Mise en équation d'une sortie

L'équation d'un récepteur (sortie) exprime la relation conditionnelle qui existe entre ce récepteur et les entrées qui le commandent.

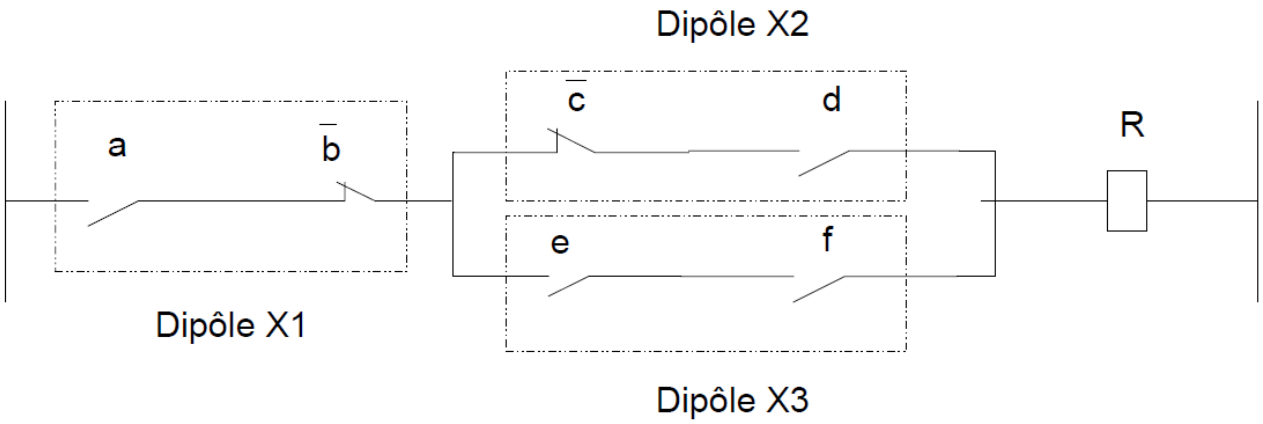


Soit le schéma à contacts ci-dessous :



Sortie : Récepteur R

Entrées : = a, $\overline{b}$, $\overline{c}$, d, e, et f



X1= a.$\overline{b}$, X2= $\overline{c}$.d , X3= e.f

R = X = X1 . (X2 + X3) = a$\overline{b}$. ($\overline{c}$d + ef)

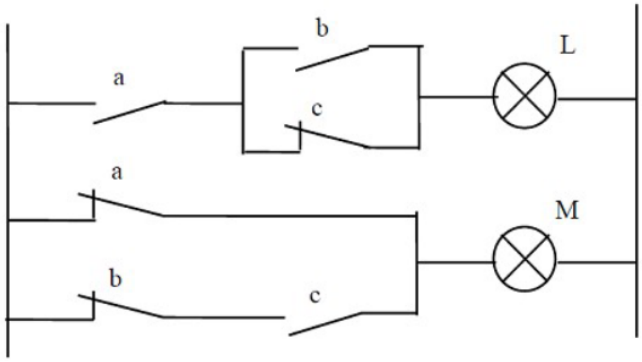
Exemple d'application :

Schématisez les équations suivantes :

L1= ABC+ $\overline{A}$C+ $\overline{B}$C

L2= AB.(C$\overline{D}$+F)

Donnez l'équation des sorties suivantes :



L= *M=*

Table de vérité

1- Définition :

On appelle table de vérité, un tableau qui indique pour chacune des combinaisons possibles des variables d'entrée la valeur de la variable de sortie.

2- Tableau des combinaisons :

Une fonction F ou S peut dépendre de n variables d'entrées.

2^n combinaisons possibles de ces n variables d'entrées.

a- Cas de 2 variables A et B (Donc 2^n combinaisons = 2^2 = 4 combinaisons 1 à 4)

N°	A	B	S
1	0	0	?
2	0	1	?
3	1	0	?
4	1	1	?

? : La sortie prend l'état 0 ou 1 selon le cahier de charge

b- Cas de 3 variables A, B et C (Donc 2^n combinaisons = 2^3 = 8 combinaisons 1 à 8)

N°	A	B	C	S
1	0	0	0	?
2	0	0	1	?
3	0	1	0	?
4	0	1	1	?
5	1	0	0	?
6	1	0	1	?
7	1	1	0	?
8	1	1	1	?

? : La sortie prend l'état 0 ou 1 selon le cahier de charge

b- Cas de 4 variables A, B, C et D (Donc 2^n combinaisons = 2^4 = 16 combinaisons 1 à 16)

N°	A	B	C	D	S
1	0	0	0	0	?
2	0	0	0	1	?
3	0	0	1	0	?
4	0	0	1	1	?
5	0	1	0	0	?
6	0	1	0	1	?
7	0	1	1	0	?
8	0	1	1	1	?
9	1	0	0	0	?
10	1	0	0	1	?
11	1	0	1	0	?
12	1	0	1	1	?
13	1	1	0	0	?
14	1	1	0	1	?

15	*1*	*1*	*1*	*0*	*?*
16	1	1	1	1	?

? : La sortie prend l'état 0 ou 1 selon le cahier de charge

Exemples :

Fonction OUI

A	*S*
0	*0*
1	*1*

Fonction NON

A	*S*
0	*1*
1	*0*

Fonction ET

A	*B*	*S*
0	*0*	*0*
0	*1*	*0*
1	*0*	*0*
1	*1*	*1*

Fonction OU

A	*B*	*S*
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*1*

Fonction OU exclusif, XOR

A	*B*	*S*
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

Fonction Non ET, NAND

A	*B*	*S*
0	*0*	*1*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

Fonction Non OU, NOR

A	*B*	*S*
0	*0*	*1*
0	*1*	*0*
1	*0*	*0*
1	*1*	*0*

3- Chronogramme :

Chronogrammes

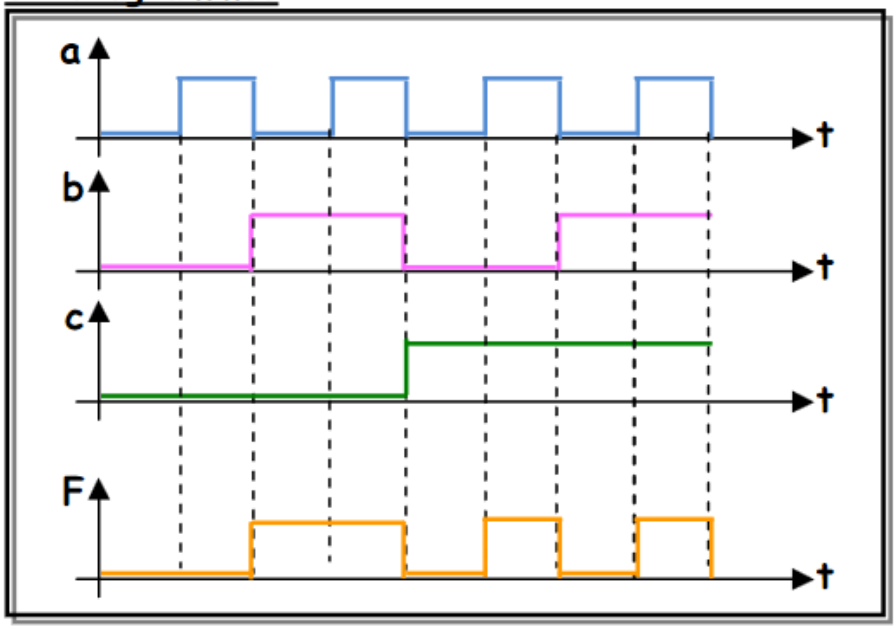


Table de vérité

c	b	a	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Equation à partir de la table de vérité

Après avoir dessiné la table de vérité pour chaque sortie en fonction des variables d'entrées on détermine son équation, pour cela il existe deux méthodes

Méthode 1 : Produit de sommes ou produit de maxtermes :

*On prend les sorties dont **l'état logique « 0 »***

Variable = 1 ➔ Variable

Variable = 0 ➔ Variable

Exemple : Soit la table de vérité suivant à 3 variables

Entrées			***Sortie***	
A	***B***	***C***	***S***	
0	*0*	*0*	***0***	$A+B+C$
0	*0*	*1*	***0***	$A+B+\overline{C}$
0	*1*	*0*	***0***	$A+\overline{B}+C$
0	*1*	*1*	***1***	
1	*0*	*0*	***0***	$\overline{A}+B+C$
1	*0*	*1*	***0***	$\overline{A}+B+\overline{C}$
1	*1*	*0*	***1***	

1	1	1	**1**	

$$S = (A + B + C).\ (A + B + \overline{C}).\ (A + \overline{B} + C).\ (\overline{A} + B + C).\ (\overline{A} + B + \overline{C})$$

Méthode 2 : Somme de produits ou somme de mintermes :

*On prend les sorties dont **l'état logique « 1 »***

Variable = 0 ➔ Variable

Variable = 1 ➔ Variable

Exemple : Soit la table de vérité suivant à 3 variables

Entrées			**Sortie**	
A	**B**	**C**	**S**	
0	0	0	**0**	
0	0	1	**0**	
0	1	0	**0**	
0	1	1	**1**	$\overline{A}.B.C$
1	0	0	**0**	
1	0	1	**0**	
1	1	0	**1**	$A.B.\overline{C}$
1	1	1	**1**	$A.B.C$

$$S = (A.B.C) + (A.B.\overline{C}) + (\overline{A}.B.C)$$

N.B : la méthode la plus utilisé est la somme de produit

Conclusion :

Soit un système caractérisé par la fonction F (a, b, c) définit par la table de vérité suivante :

N	**a**	**b**	**c**	**F**
0	0	0	0	**1**
1	0	0	1	**0**
2	0	1	0	**0**
3	0	1	1	**1**
4	1	0	0	**0**
5	1	0	1	**1**
6	1	1	0	**1**
7	1	1	1	**0**

La fonction F peut s'écrire :

- ✓ *Une somme des états pour lesquelles elle vaut 1.*
  - $F = \sum(0,3,5,6) = \bar{a}.\bar{b}.\bar{c} + \bar{a}.b.c + a.\bar{b}.c + a.b.\bar{c}$
- ✓ *Un produit des états pour lesquelles elle vaut 0.*
  - $F = \prod(1,2,4,7) = (a + b + \bar{c}).(a + \bar{b} + c).(\bar{a} + b + c).(\bar{a} + \bar{b} + \bar{c})$

Simplification des fonctions logiques

La traduction de la table de vérité sous forme d'équation logique donne une équation longue appelée forme normale ou canonique.

Il est donc nécessaire de la simplifier pour la rendre sous sa forme simplifiée.

Il existe deux méthodes :

Méthode algébrique

On écrit la "somme" logique des lignes où la variable de sortie prend la valeur 1.
Lorsque les états "1" sont plus nombreux que les états "0", il est avantageux d'écrire le complément de la somme logique des lignes où la variable de sortie prend la valeur 0.
Après on applique les règles de l'algèbre de Boole pour simplifier l'équation ;
Distributivité du produit par rapport à la somme :

$$A.(B+C) = A.B + A.C$$

Distributivité de la somme par rapport au produit :

$$(A+B).(A+C) = AA + AC + AB + BC = A.(1+B+C) + B.C = A + B.C$$

Factorisation :

$$(AB + A.\bar{B}) = A(B+\overline{B}) = A.1 = A$$

Loi d'absorption :

$$A + AB = A(1+B) = A$$
$$AB + \bar{A}B = B(A+\bar{A}) = B$$
$$(A + \bar{A}.B) = (A + A.B) + \bar{A}.B = A + (A.B + \bar{A}.B) = A + B.(A+\bar{A}) = A + B$$

Théorème de De Morgan :

Ce théorème d'une grande utilité, permet de calculer le complément d'une expression logique quelconque (somme de produits ou produit de sommes) :

$$\overline{X+Y} = \bar{X}.\bar{Y}$$
$$\overline{X.Y} = \bar{X} + \bar{Y}$$

Son rôle est de simplifier et optimiser la conception des structures à base d'opérateurs logiques.

D'une façon générale, Le complément d'une expression quelconque s'obtient en complémentant les variables et en permutant les opérateurs "+" et ".".

Exemple :

Soit l'expression logique F définie par : $F = (\bar{a} + b).c + a.c + (\overline{a+b})$
Le complément de F s'écrit alors :

$$\bar{F} = \overline{(\bar{a} + b).c + a.c + (\overline{a+b})}$$
$$= \overline{(\bar{a} + b).c}.\overline{a.c}.(\overline{\overline{a+b}})$$
$$= (a.\bar{b} + \bar{c}).(\bar{a} + \bar{c}).(a + b)$$
$$= (a.\bar{b}.\bar{a} + \bar{c}.\bar{a} + a.\bar{b}.\bar{c} + \bar{c}.\bar{c}).(a + b)$$

$$= (0 + \bar{c}.\bar{a} + a.\bar{b}.\bar{c} + \bar{c}).(a + b)$$
$$= \bar{c}.((\bar{a} + a.\bar{b} + 1).(a + b)$$
$$= \bar{c}.(a + b)$$

Cette méthode peut convenir pour les cas où le nombre de variables d'entrée ne dépasse pas 2 ou 3. Par fois pour simplifier une fonction, on peut ajouter un terme déjà existant.

Dans le cas où le nombre de variables devient trop important, il est plus avantageux d'utiliser une méthode graphique intitulée « Tableau de Karnaugh » permettant de trouver directement une expression simplifiée de l'équation de sortie d'une fonction logique.

Méthode graphique (Tableau de KARNAUGH)

Les tableaux de KARNAUGH sont une représentation particulière de la table de vérité ou une équation ; Sa conception permet d'obtenir de manière sûre et rapide l'équation la plus simplifiée possible.

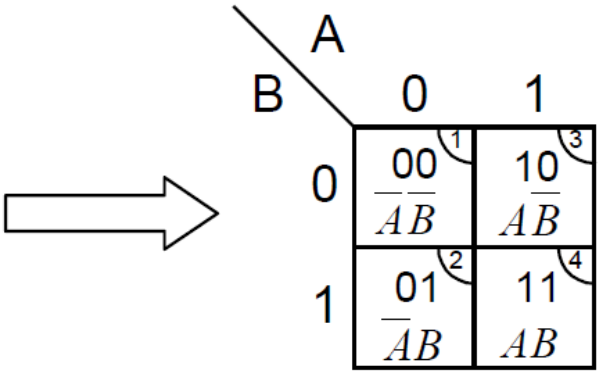
1. Construction du tableau de KARNAUGH

- *C'est un tableau de 2^n cases, n est le nombre de variables de la fonction logique.*
- *Sur les lignes et colonnes, on représente l'état des variables d'entrée codées en binaire réfléchi (code GRAY)*
- *Le tableau est construit pour une variable de Sortie*
- *Chaque case représente une combinaison des entrées où on écrit la valeur de sortie*

Exemples :

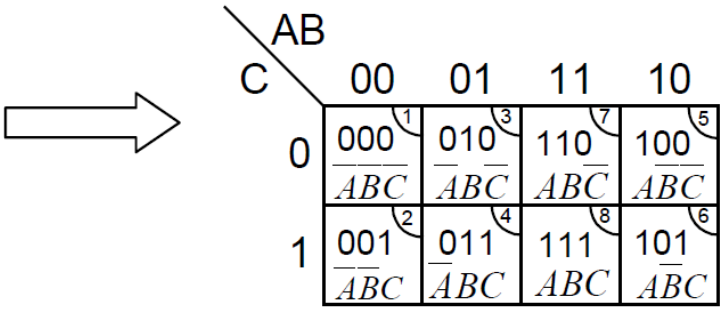
1.

	A	B	S
1-	0	0	
2-	0	1	
3-	1	0	
4-	1	1	



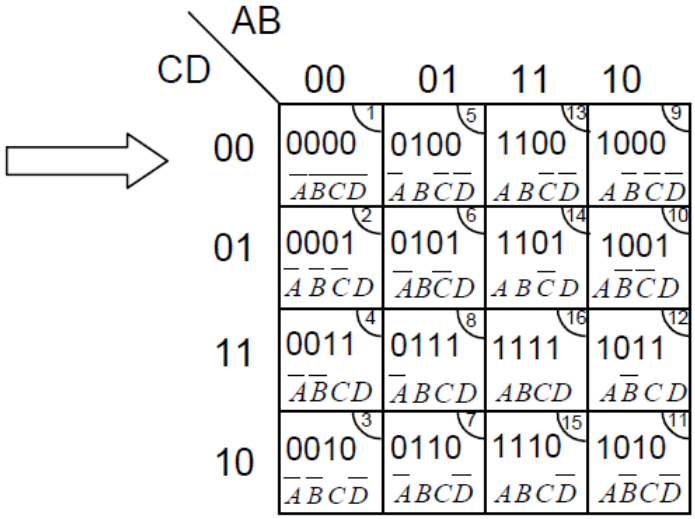
2.

	A	B	C	S
1-	0	0	0	
2-	0	0	1	
3-	0	1	0	
4-	0	1	1	
5-	1	0	0	
6-	1	0	1	
7-	1	1	0	
8-	1	1	1	



3.

	A	B	C	D	S
1-	0	0	0	0	
2-	0	0	0	1	
3-	0	0	1	0	
4-	0	0	1	1	
5-	0	1	0	0	
6-	0	1	0	1	
7-	0	1	1	0	
8-	0	1	1	1	
9-	1	0	0	0	
10-	1	0	0	1	
11-	1	0	1	0	
12-	1	0	1	1	
13-	1	1	0	0	
14-	1	1	0	1	
15-	1	1	1	0	
16-	1	1	1	1	



Passage d'une équation à la table de vérité puis au tableau de KARNAUGH

Le passage se fait en affectant une valeur de 0 pour la variable qui possède un barre et 1 pour la variable sans barre et X pour la variable non existante dans un somme de produit

Exemple : $S = ab + \bar{a}bc + \bar{b}c$

$11x + 101 + x01$

$110 + 111 + 101 + 001 + 101$

Table de vérité

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

Tableau de KARNAUGH

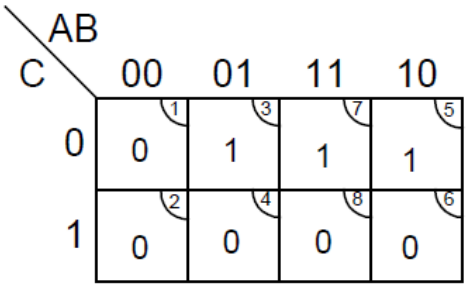
S \ ab	00	01	11	10
0	0	0	1	0
1	1	0	1	1

(c : 0 / 1)

	A	B	S
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	1

B \ A	0	1
0	0 (1)	1 (3)
1	1 (2)	1 (4)

	A	B	C	S
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	0
5	1	0	0	1
6	1	0	1	0
7	1	1	0	1
8	1	1	1	0



2. Simplification d'une équation par le diagramme de Karnaugh.

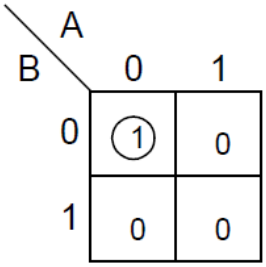
- La méthode consiste à réaliser des groupements de CASES ADJACENTES contenant des 1 ou des 0. Un groupement de 1 permet d'obtenir l'équation de S, un groupement de 0 permet d'obtenir l'équation $\overline{S}$

- Deux cases sont adjacentes lorsqu'elles sont situées côte à côte, que ce soit à l'horizontale ou à la verticale. De plus, une seule variable doit changer d'état pour que deux cases soient considérées comme adjacentes.

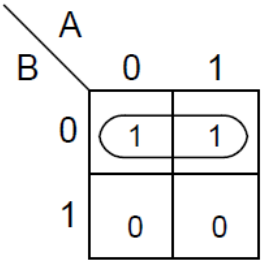
Les règles :

- *Le nombre de cases d'un groupement doit être égal à 1, 2, 4, ...2^n*
- *Les groupements doivent être les plus grands possibles et contenir la même variable*
- *Les groupements peuvent se chevaucher pour être les plus grands possibles.*
- *Dans chaque groupement on ne retient que les variables dont l'état ne change pas.*
- *Pour extraire l'équation de la fonction logique on ne retient que les variables dont l'état ne change pas à l'intérieur d'un groupement et on effectue la somme logique (OU logique) de toutes les expressions trouvées.*

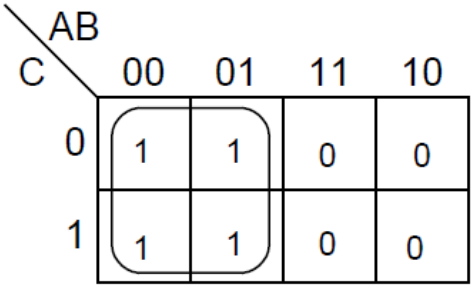
Exemples de groupements :



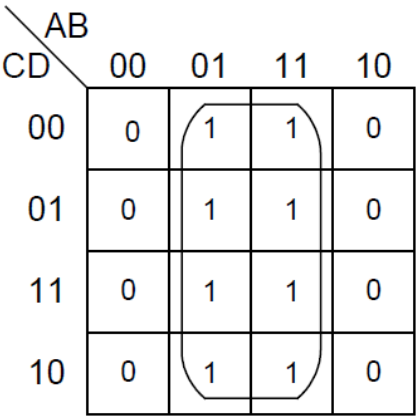
case unique



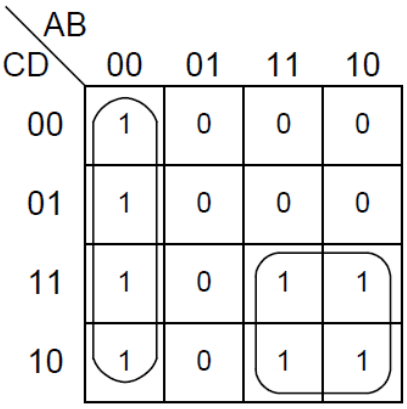
groupement de deux

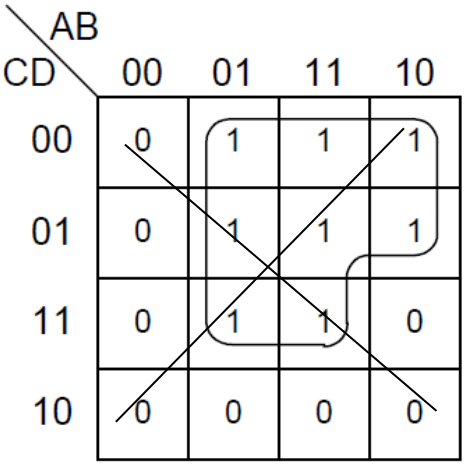


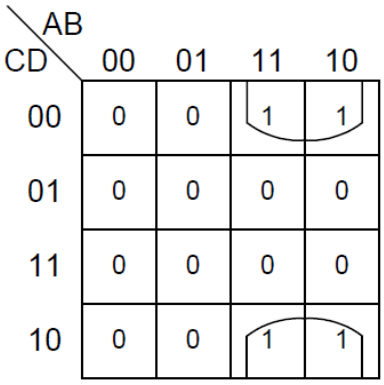
groupement de quatre

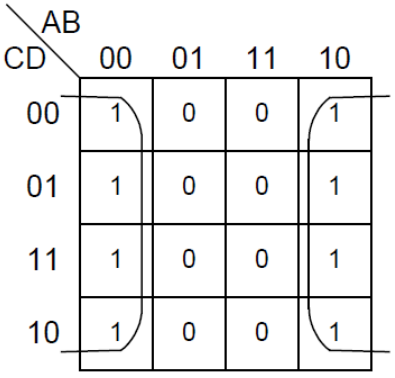


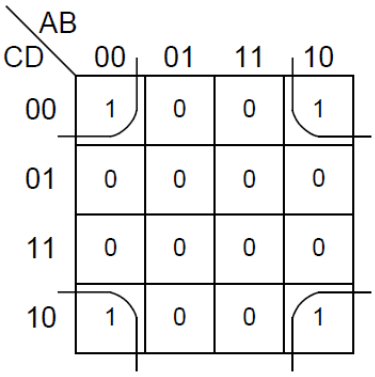
groupement de huit

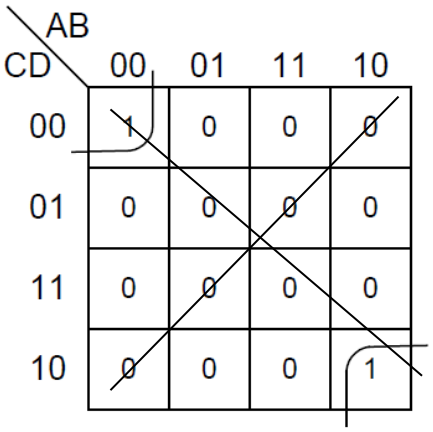




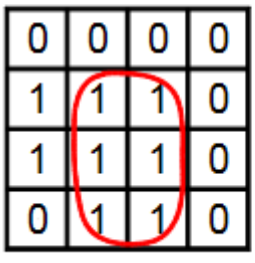


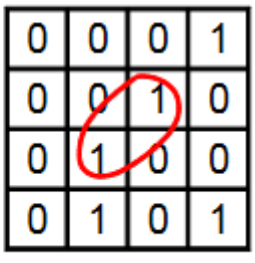






IMPOSSIBLES





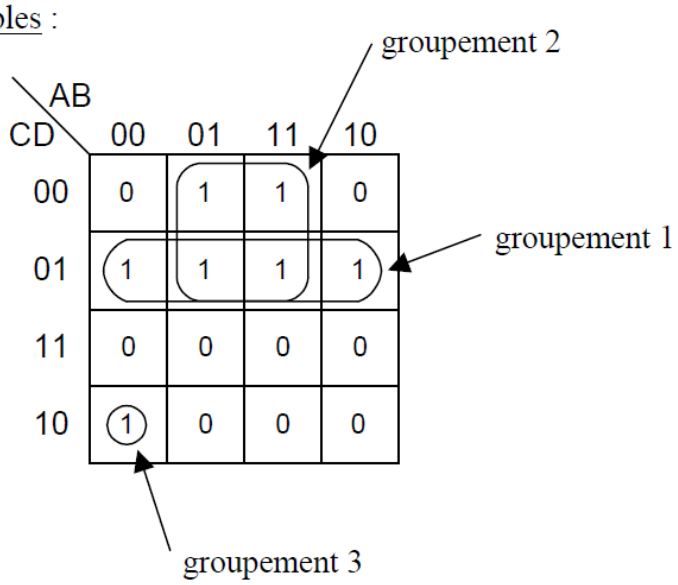
Écriture des équations à partir de regroupement :

Chaque regroupement de 1 donne le produit logique des variables d'entrée qui n'ont pas changé d'état dans ce groupement. L'ensemble de ces regroupements est une somme logique.

Règle : B = 1 ⟹ On la représente par B ;
B = 0 ⟹ On la représente par $\overline{B}$

Exemples :

a)



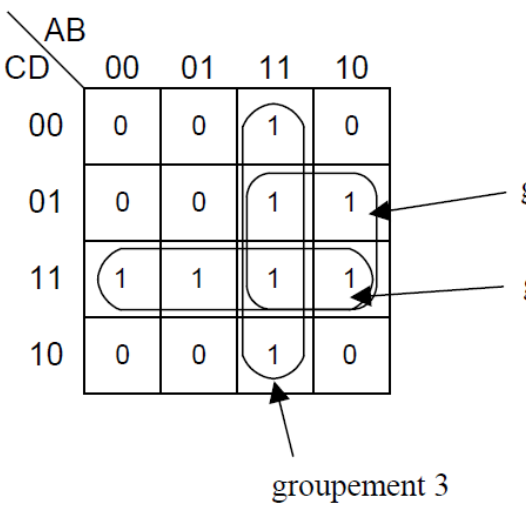
Groupement 1 : A et B changent d'état
C = 0 et D = 1 ne changent pas d'état ⟹ L'équation du groupement : $\overline{C}\bullet D$

Groupement 2 : A et D changent d'état
B = 1 et C = 0 ne changent pas d'état ⟹ L'équation du groupement : $B\bullet\overline{C}$

Groupement 3 : A = 0, B = 0, C = 1 et D = 0 ne change pas d'état ⟹ L'équation du groupement : $\overline{A}\bullet\overline{B}\bullet C\bullet\overline{D}$

D'où l'équation finale : $S=\overline{C}\bullet D+B\bullet\overline{C}+\overline{A}\bullet\overline{B}\bullet C\bullet\overline{D}$

b)



Groupement 1 : $A\bullet D$

Groupement 2 : $C\bullet D$

Groupement 3 : $A\bullet B$

⟹ $S=A\bullet D+C\bullet D+A\bullet B$

Exercices :

Ex 1 : Élaborer des diagrammes de Karnaugh à la droite des tables de vérité qui suivent :

A	B	S
0	0	1
0	1	1
1	0	0
1	1	1

A	B	C	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

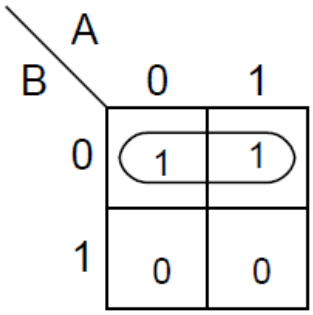
A	B	C	S
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Ex 2 : Établir le diagramme de Karnaugh correspondant à l'équation suivant :

$$S = A \bullet \overline{C} + \overline{A} \bullet B + \overline{A} \bullet \overline{B} \bullet \overline{C}$$

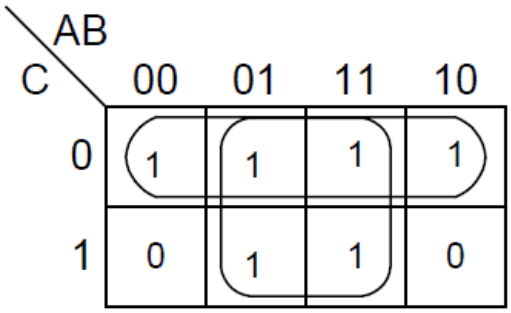
$$F_1 = a.b.\overline{c} + \overline{a}.\overline{b}.c + a.\overline{b}.\overline{c} + a.\overline{b}.c$$

Ex 3 : Écrire les équations correspondantes aux regroupements de cases des diagrammes de Karnaugh suivants :



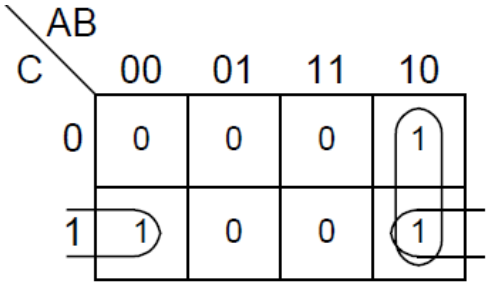
B \ A	0	1
0	1	1
1	0	0

S=……



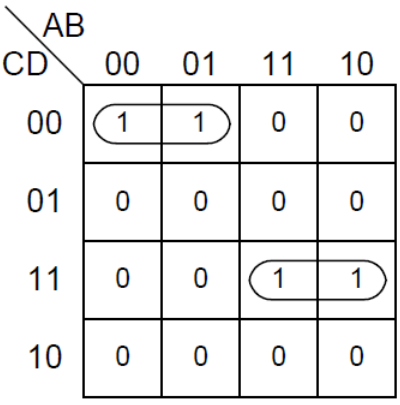
C \ AB	00	01	11	10
0	1	1	1	1
1	0	1	1	0

S=……



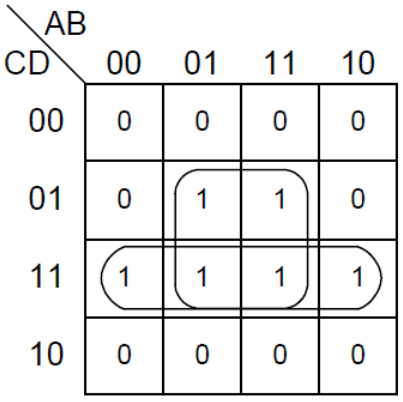
C \ AB	00	01	11	10
0	0	0	0	1
1	1	0	0	1

S=……



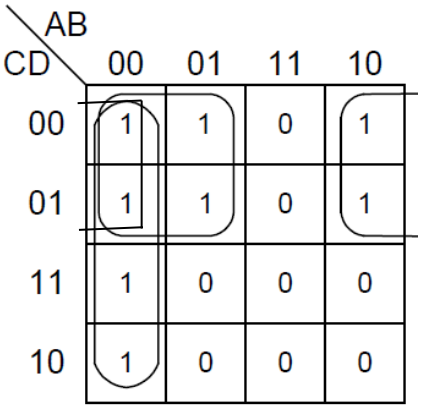
CD \ AB	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	0	0	1	1
10	0	0	0	0

S=……



CD \ AB	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	1	1
10	0	0	0	0

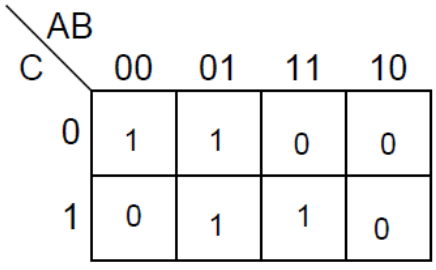
S=……



CD \ AB	00	01	11	10
00	1	1	0	1
01	1	1	0	1
11	1	0	0	0
10	1	0	0	0

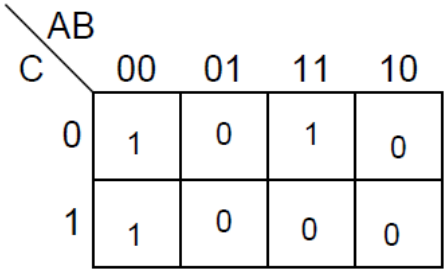
S=……

Ex 4 : A partir des diagrammes de Karnaugh suivant, effectuer les regroupements et écrire l'équation logique correspondante :



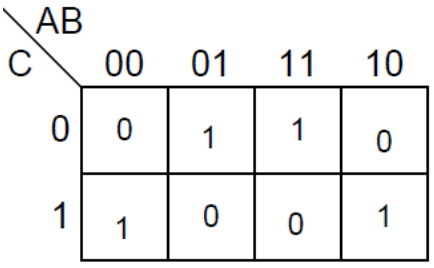
C \ AB	00	01	11	10
0	1	1	0	0
1	0	1	1	0

S=……



C \ AB	00	01	11	10
0	1	0	1	0
1	1	0	0	0

S=……



C \ AB	00	01	11	10
0	0	1	1	0
1	1	0	0	1

S=……

CD \ AB	00	01	11	10
00	1	1	0	0
01	1	1	1	1
11	1	1	1	1
10	0	0	0	0

S=.....

CD \ AB	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	0	1	1	1
10	1	1	1	1

S=.....

CD \ AB	00	01	11	10
00	0	0	1	1
01	1	0	0	1
11	1	0	0	1
10	0	0	1	1

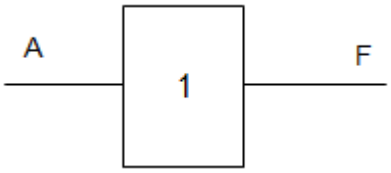
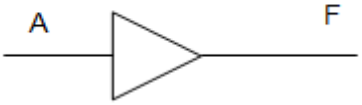
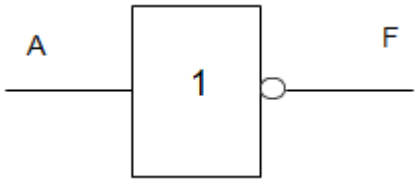
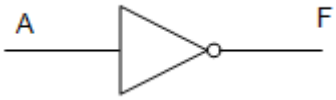
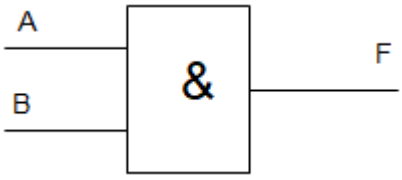
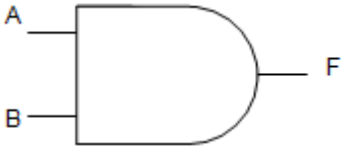
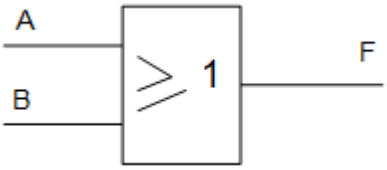
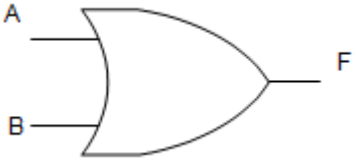
S=.....

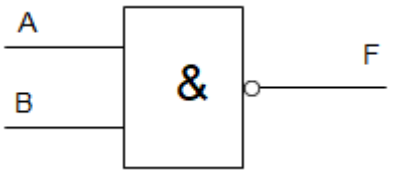
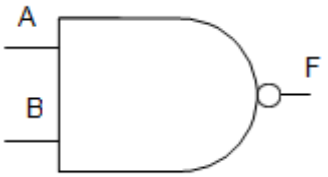
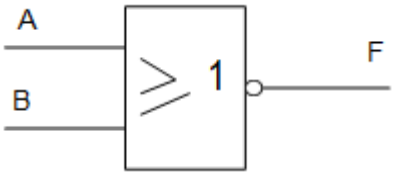
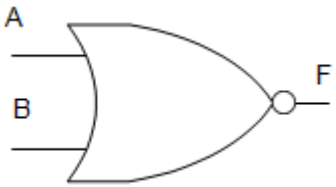
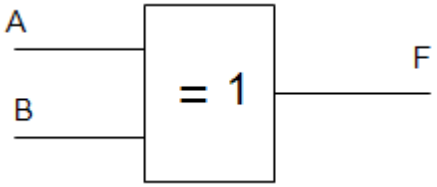
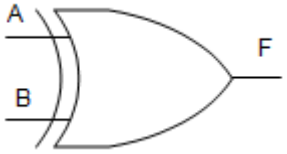
Traduire des équations en schémas logiques.

Généralités :

Un schéma logique est la représentation graphique de l'équation d'une ou plusieurs variables de sortie grâce aux opérateurs ou fonctions logiques de base vus précédemment

Symbole des fonctions logiques ou portes logiques :

	Symboles	
Fonction	**Norme IEEE**	**Norme IEC**
OUI : $\mathbf{F = A}$		
NON : $\mathbf{F = \overline{A}}$		
ET (AND) : $\mathbf{F = A \cdot B}$		
OU (OR): $\mathbf{F = A + B}$		

NON ET (NAND): $\mathbf{F} = \overline{\mathbf{A} \cdot \mathbf{B}}$		
NON OU (NOR): $\mathbf{F} = \overline{\mathbf{A} + \mathbf{B}}$		
OU exclusif : $\mathbf{F} = \mathbf{A} \oplus \mathbf{B}$		

NB : il existe des portes logiques à 3 entrées et plus

Les circuits intégrés (CI) en électronique numérique appartiennent soit à la famille TTL (transistor transistoc logic « GND, +5V ») ou la famille CMOS (Comptementary Métal Oxyde Semiconductor « GND,+ 3V à +12 V(A) ou 18V(B) »)

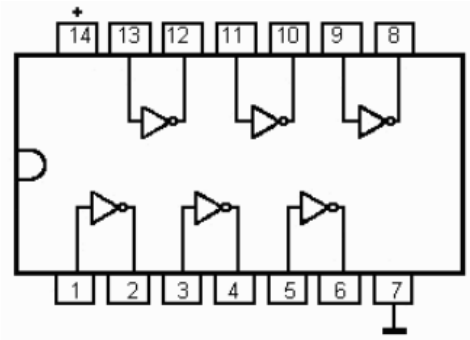
Suivant la gamme de température d'utilisation, on distingue deux séries des C.I TTL :

- *La série 5400 : gamme de température d'utilisation militaire indiquée par 5 (-55°, +125°C);*
- *La série 7400 : gamme de température d'utilisation générale indiquée par 7 (0°,+70°C).*

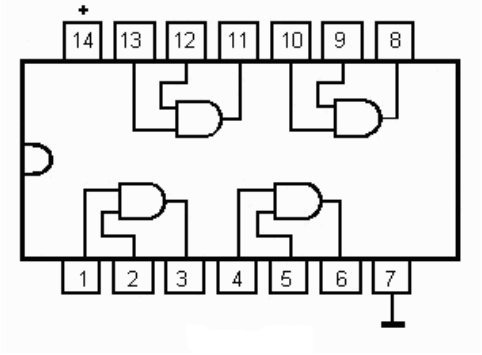
On distingue 3 types de schémas logiques :

1- Schémas avec des opérateurs NON, ET, OU

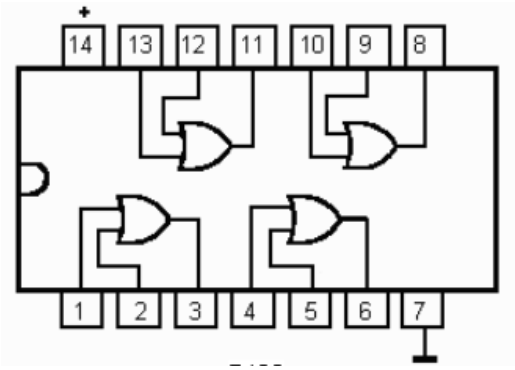
Fonction Non (74-04)



Fonction ET (74-08)

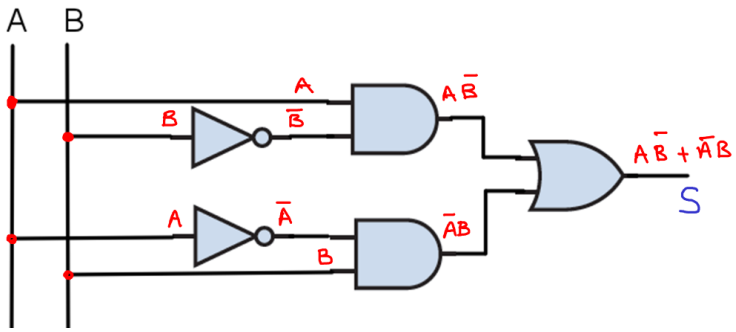


Fonction OU (74-32)



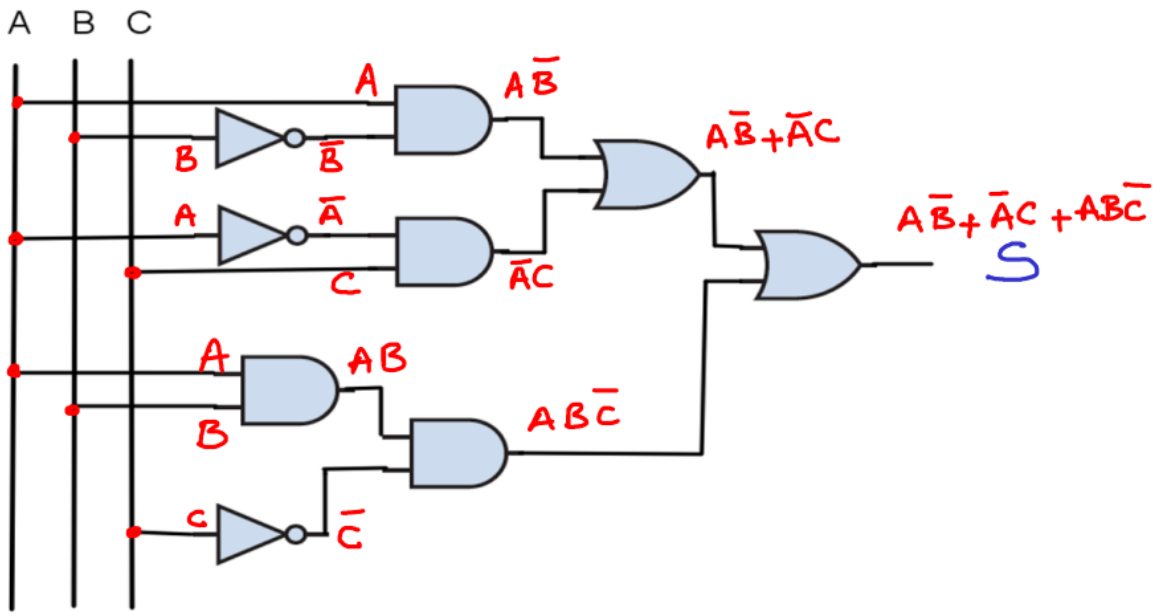
Exemple 1

$S = A\overline{B} + \overline{A}B$



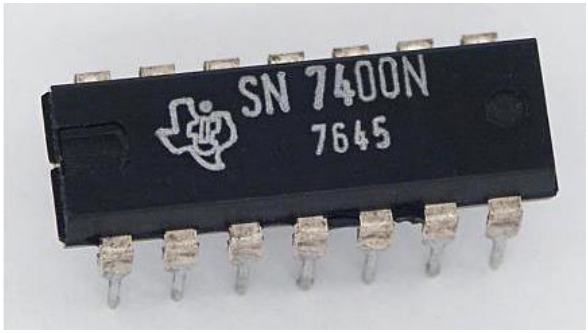
Exemple 2

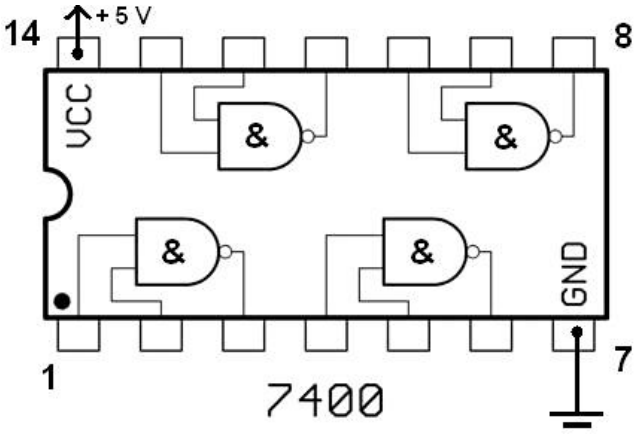
$$S = A\overline{B} + \overline{A}C + AB\overline{C}$$



2- Schémas avec des opérateurs NON ET, NAND

Fonction NON ET (NAND) (74-00)





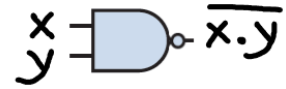
Exemple :

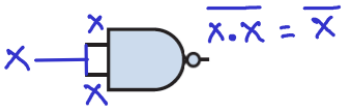
$$S = A\bar{B} + \bar{A}B$$

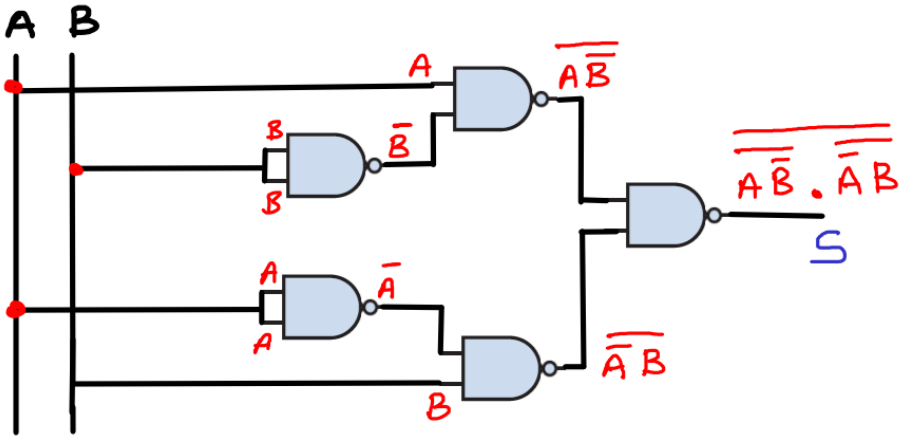
$$S = A\bar{B} + \bar{A}B$$

$$S = A.\bar{B} \cdot \bar{A}.B$$

NAND $S = \overline{X.Y}$

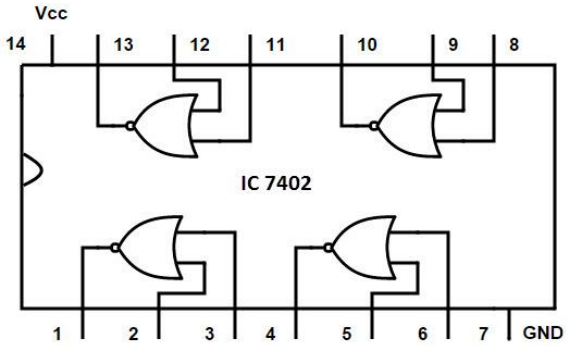






3. ***Schéma avec des opérateurs NON OU (NOR).***

Fonction Non OU (NOR) 74-02



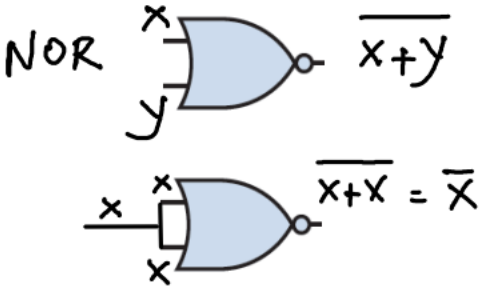
Exemple :

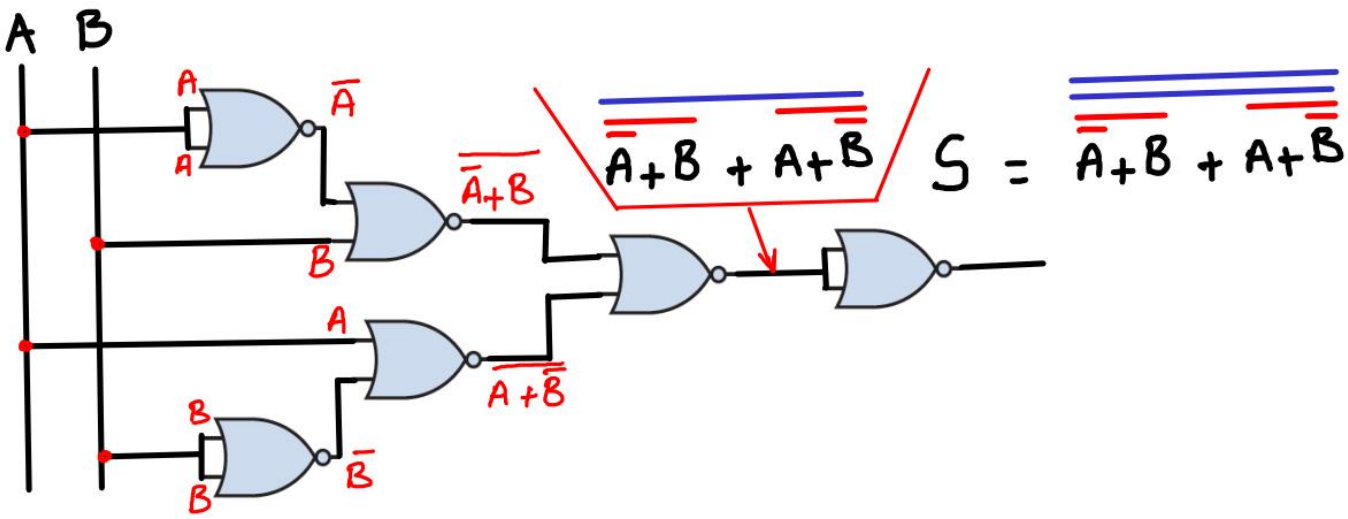
$$S = A\overline{B} + \overline{A}B$$

$$S = \overline{\overline{A\overline{B}} + \overline{\overline{A}B}}$$

$$S = \overline{\overline{A}+B} + \overline{A+\overline{B}}$$

$$S = A+\overline{B} + \overline{A}+B$$





Exercices :

Dessinez les logigrammes des équations suivantes avec des portes logiques à deux entrées en utilisant les 3 types de logigramme (NON, ET, OU), NAND et NOR :

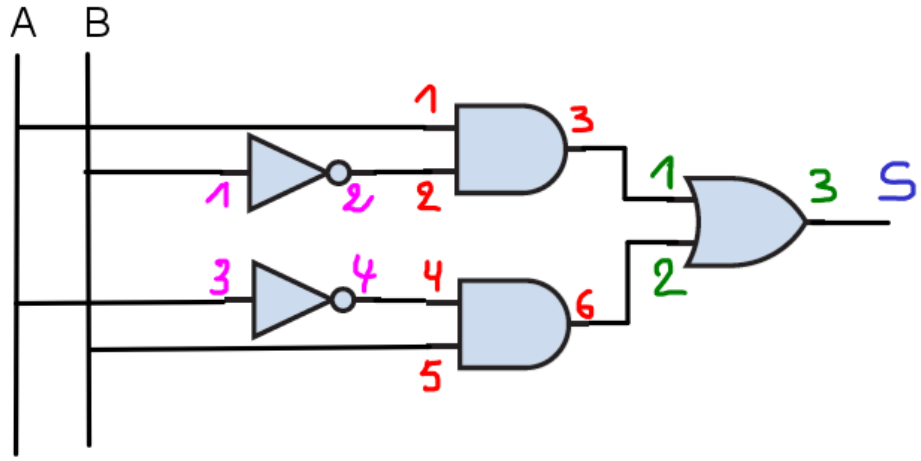
S1= A. (BC+$\overline{C}$)

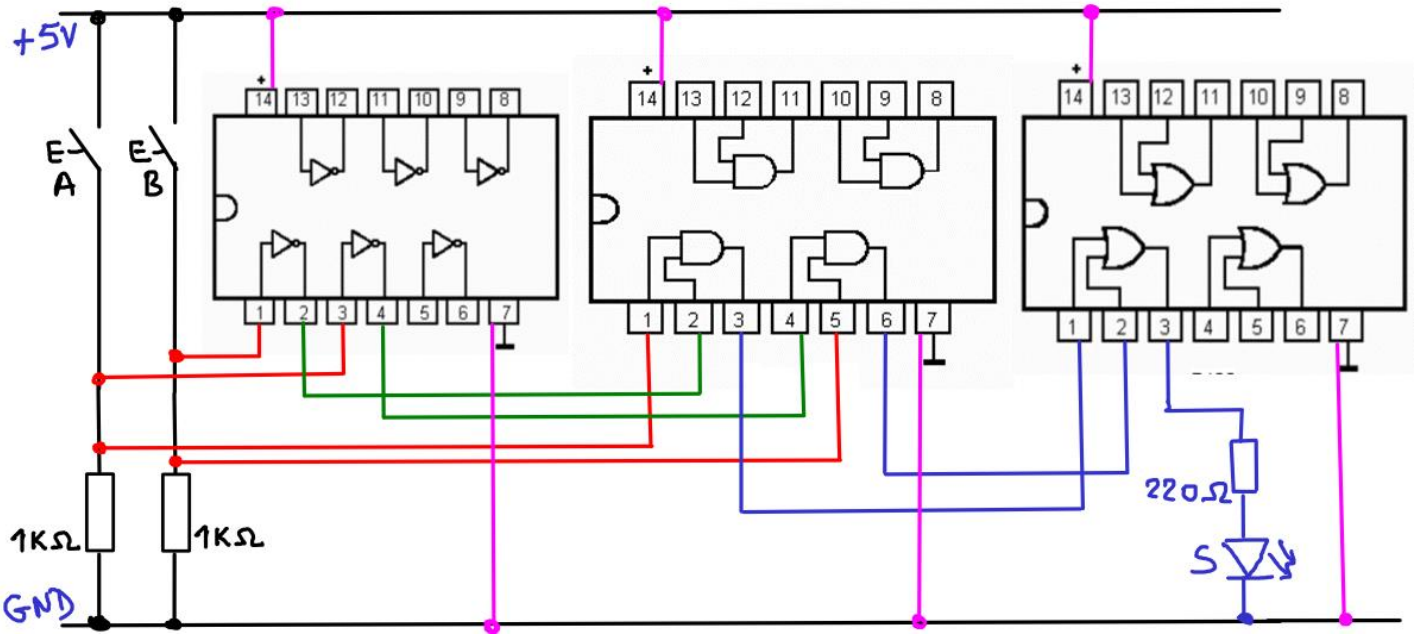
S2= (AB+C).D

Câblage d'un logigramme ou un schéma logique

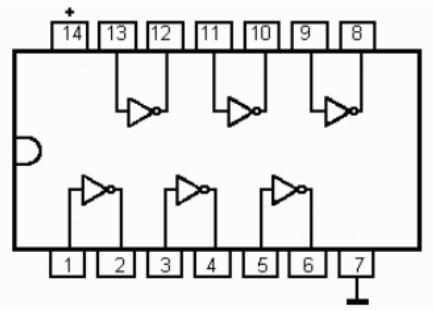
Exemple 1 :

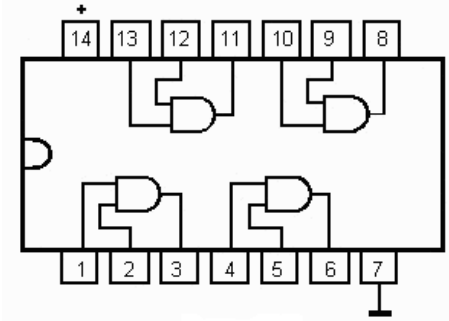
$S = A\overline{B} + \overline{A}B$

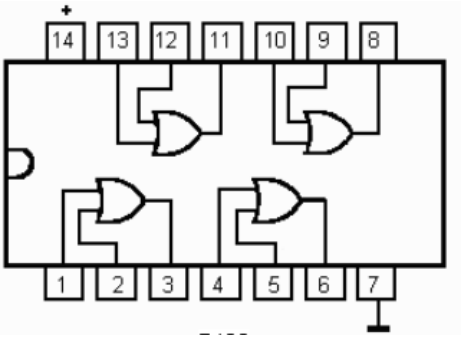


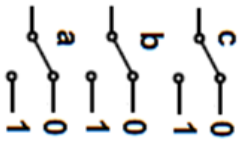


Exemple 2 : $F = (\bar{a} + b).c + a.c + (\overline{a + b})$





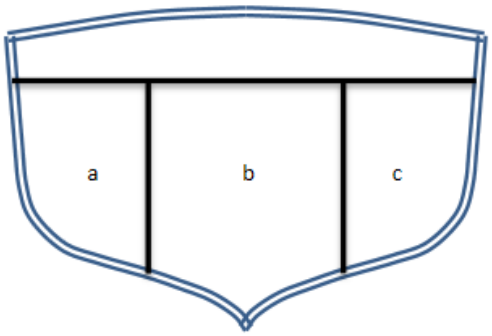




Exercices

Exercice 1 :

Afin d'équilibrer la charge d'un pétrolier comportant trois réservoirs, comportant respectivement chacun les capteurs (a, b, c,).



L'équilibre est correct et une lampe verte (V) s'allume dans les cas suivants :

Quand on charge ou décharge le réservoir b ;

Quand on charge ou décharge les réservoirs a et c ;

Quand on charge ou décharge les réservoirs a, b et c

Quand on n'en charge aucun

Exprimer V en fonction de a, b, c et dessiner le logigramme.

Exercice 2 :

Une société à 4 actionnaires ayant le nombre suivant d'actions :

A : 60 ; B : 100 ; C : 160 ; D : 180.

On désire construire une machine permettant le vote automatique lors des réunions. Chaque actionnaire dont le poids de vote est proportionnel au nombre d'actions appuie sur un bouton qui porte son nom (A, B, C ou D).

Si un actionnaire vote OUI, sa variable (par exemple A) vaut 1, s'il vote NON, elle vaut 0.

Une résolution sera votée (V=1) si la somme des actions correspondant aux vote OUI représente au moins la moitié des actions plus 1.

Exprimer V en fonction de A , B , C et D et dessiner le logigramme.

Exercice 3 :

Le comité directeur d'une entreprise est constitué de quatre membres :

le directeur D et ses trois adjoints A, B, C. Lors des réunions, les décisions sont prises à la majorité. Chaque personne dispose d'un interrupteur pour voter sur lequel elle appuie en cas d'accord avec le projet soumis au vote. En cas d'égalité du nombre de voix, celle du directeur compte double.

On vous demande de réaliser un dispositif logique permettant l'affichage du résultat du vote sur lampe R.

1. Donner l'équation logique de R

2. Réaliser le schéma logique de la sortie R

Exercice 4 :

On désire réaliser la logique de commande d'un distributeur de boissons chaudes capable de délivrer du thé (électrovanne "T"), du café ("C") et du sucre ("S").

Trois boutons «t", "c" et "s" permettent d'obtenir :

-du café, sucré ou non ;

-du thé, sucré ou non ;

-du sucre seul (gratuit).

Une pièce "p" doit être introduite après avoir choisi une boisson. La pièce est rendue en cas de fausse manœuvre ; c'est la fonction "P" de restitution.

1) Etablir la table de vérité,

2) Trouver les équations de T , C , S et P,

3) Proposer un logigramme des fonctions T , C , S et P.

Exercice 5 :

En utilisant les diagrammes de Karnaugh, simplifier les fonctions suivantes :

$$F_1 = a.b.\bar{c} + \bar{a}.\bar{b}.c + a.\bar{b}.\bar{c} + a.\bar{b}.c$$

$$F_2 = \bar{a}.\bar{b}.c.\bar{d} + \bar{a}.b.\bar{c}.d + a.\bar{b}.c.\bar{d} + \bar{a}.\bar{b}.\bar{c}.\bar{d} + a.b.\bar{c}.d + \bar{a}.b.c.d + a.\bar{b}.\bar{c}.\bar{d}$$

$$F_{3(abcd)} = \Sigma(0;1;2;3;9;10;11;13;15)$$

Exercice 5 :

Une installation industrielle comporte 4 interrupteurs **a, b, c,** et **d**, permettant d'allumer les lampes **E**, **F**, **G** et **H** en fonction des équations et des conditions suivantes :

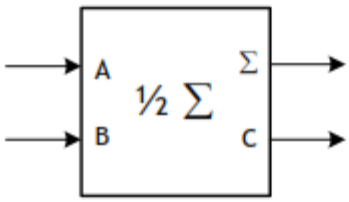
- $E = \bar{a}.\bar{b}.\bar{c}.\bar{d} + \bar{a}.b.\bar{c}.\bar{d} + \bar{a}.b.\bar{c}.d + \bar{a}.b.c.d + \bar{a}.b.c.\bar{d} + \bar{a}.\bar{b}.c.\bar{d}$
- $F = \bar{a}.\bar{b}.\bar{c}.\bar{d} + a.\bar{b}.\bar{c}.\bar{d} + \bar{a}.\bar{b}.c.\bar{d} + a.\bar{b}.c.\bar{d} + \bar{a}.b.\bar{c}.d + a.b.\bar{c}.d + \bar{a}.b.c.d + a.b.c.d$
- ***G*** s'allume quand **E=F**. et - ***H*** s'allume quand **E≠ F**.

1. A l'aide de la table de Karnaught simplifier les équations **E** et **F**.
2. Déterminer les équations de **G** et **H** en fonction de **E** et **F**.
3. Tracer le logigramme du circuit de cette installation (vous pouvez utiliser les portes logiques **XOR**).

Les fonctions combinatoires avancées

Le demi-additionneur :

C'est un circuit permettant d'effectuer l'addition de deux bits A et B pour générer leur somme $\sum$ *et leur retenue C (Carry) comme le montre le schéma et la table de vérité de la figure suivante :*

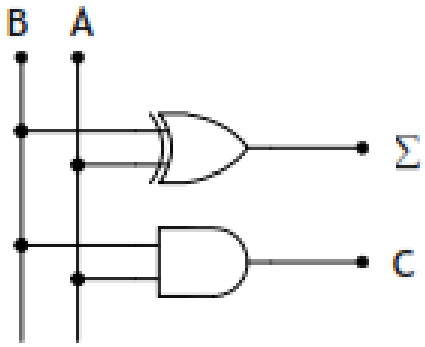


ENTREES		SORTIES	
B	A	Σ	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A partir de la table de vérité, on peut écrire les deux fonctions sous la forme suivante :

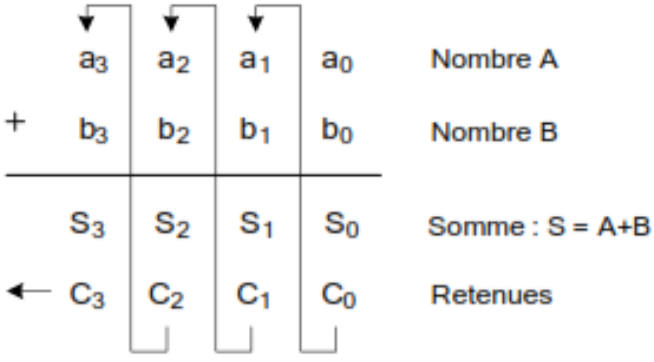
$$\sum = A.\bar{B} + \bar{A}.B = A \oplus B, \quad C = A.B$$

Ce qui peut être réalisé par le circuit schématisé sur le logigramme de la figure ci-dessous.

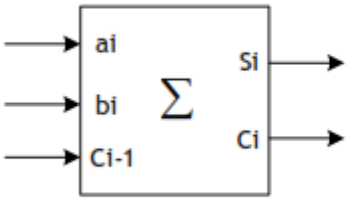


L'additionneur complet :

Pour effectuer une addition de deux nombres binaires de n bits, on additionne successivement les bits du même poids en tenant compte de la retenue de l'addition précédente comme le montre l'exemple suivant :



Il faut donc concevoir une cellule élémentaire appelée additionneur complet et qui permet de réaliser l'addition des bits ai et bi en plus de la retenue Ci-1 de l'addition précédente. Un tel circuit est définit par le schéma et la table de vérité de la figure suivante :

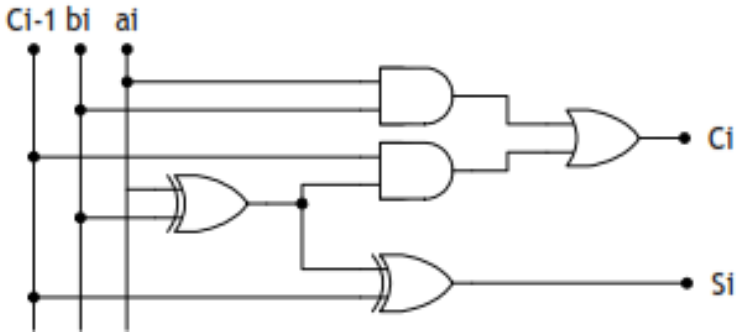


ENTREES			SORTIES	
ai	bi	Ci-1	Si	Ci
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

A l'aide de la table de Karnaugh, on détermine les équations de sorties suivantes :

$$S_i = a_i.\bar{b}_i.\bar{C}_{i-1} + \bar{a}_i.b_i.\bar{C}_{i-1} + \bar{a}_i.\bar{b}_i.C_{i-1} + a_i.b_i.C_{i-1} = a_i \oplus b_i \oplus C_{i-1}$$

$$C_i = a_i.b_i + a_i.\bar{b}_i.C_{i-1} + \bar{a}_i.b_i.C_{i-1} = a_i.b_i + (a_i \oplus b_i).C_{i-1}$$



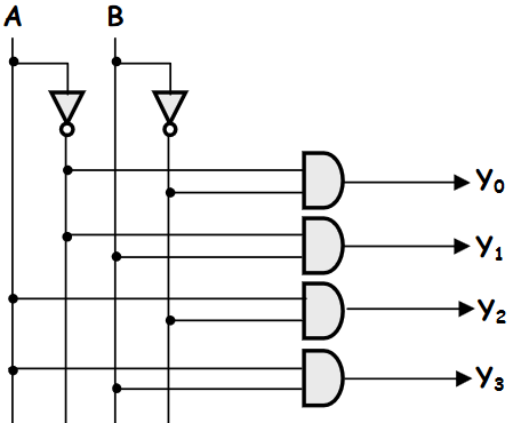
Les décodeurs :

La fonction de décodage consiste à faire correspondre à un code présent en entrée sur m lignes, un autre code en sortie sur n lignes avec en général m ≠ n.

Décodeur 1 parmi 4

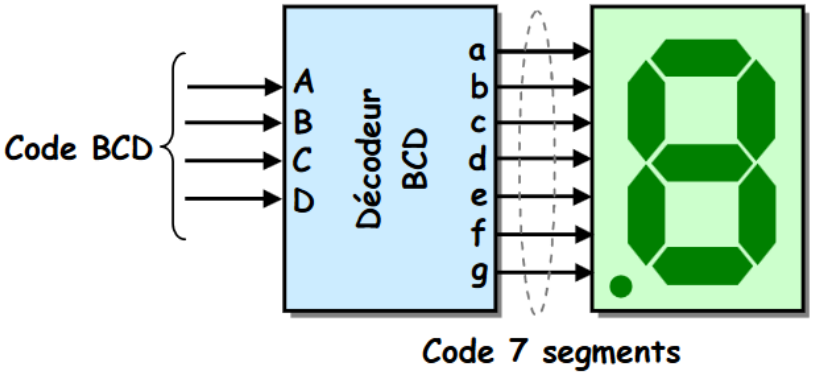
Table de vérité					
Entrées		Sorties			
A	B	Y_0	Y_1	Y_2	Y_3
0	0	**1**	0	0	0
0	1	0	**1**	0	0
1	0	0	0	**1**	0
1	1	0	0	0	**1**

Equations logiques
$Y_0 = \bar{A}\bar{B}$
$Y_1 = \bar{A}B$
$Y_2 = A\bar{B}$
$Y_3 = AB$

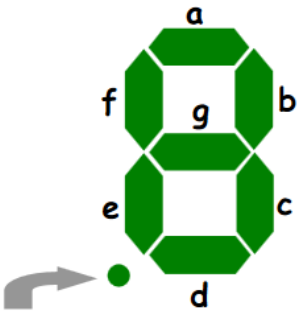


Décodeur BCD – 7 segments :

Ce décodeur permet de convertir le code BCD, présent à son entrée sous 4 bits, en un code 7 segments disponible à sa sortie. Il est utilisé pour commander un afficheur 7 segments afin d'écrire des chiffres de 0 à 9, certaines lettres et aussi quelques symboles.



Code 7 segments



PD : point décimal

On souhaite concevoir un décodeur BCD pour piloter un afficheur 7 segments à cathode commune.

Table de vérité												
Nombre BCD	Entrées				Sorties							
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

a

AB \ CD	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	1	X	X	1
10	0	1	X	1

b

AB \ CD	00	01	11	10
00	1	1	X	1
01	1	X	X	0
11	1	X	X	1
10	1	1	X	0

c

AB \ CD	00	01	11	10
00	1	1	X	1
01	0	X	X	1
11	1	X	X	1
10	1	1	X	1

d

AB \ CD	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	1	X	X	0
10	0	1	X	1

e

AB \ CD	00	01	11	10
00	1	1	X	0
01	1	X	X	1
11	0	X	X	0
10	0	0	X	0

f

AB \ CD	00	01	11	10
00	1	1	X	1
01	0	X	X	1
11	0	X	X	0
10	0	1	X	1

g

AB \ CD	00	01	11	10
00	0	1	X	1
01	1	X	X	1
11	1	X	X	0
10	0	1	X	1

Equations logiques	
$a=B+D+AC+\bar{A}\bar{C}$	$e=\bar{A}B+\bar{A}\bar{C}$
$b=\bar{C}+AB+\bar{A}\bar{B}$	$f=D+\bar{A}\bar{B}+\bar{B}C+C\bar{A}$
$c=A+C+\bar{B}$	$g=D+\bar{A}C+C\bar{B}+B\bar{C}$
$d=D+\bar{A}B+B\bar{C}+\bar{C}\bar{A}+A\bar{B}C$	

Multiplexeur

Un multiplexeur permet de sélectionner une entrée parmi 2^n pour transmettre l'information portée par cette ligne à un seul canal de sortie. La sélection de l'entrée se fait alors à l'aide de n lignes d'adressage.

Exemple : Mux 4 vers 1

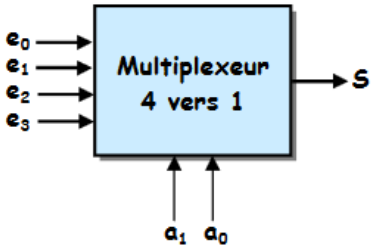
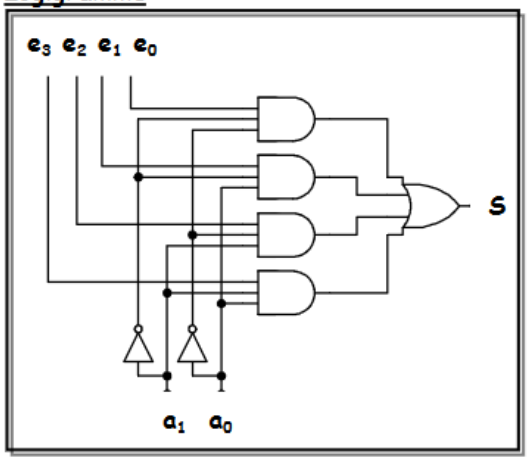


Table de vérité		
Adresses		Sortie
a_1	a_0	S
0	0	e_0
0	1	e_1
1	0	e_2
1	1	e_3

Equation logique
$S=e_0\bar{a}_1\bar{a}_0+e_1\bar{a}_1a_0+e_2a_1\bar{a}_0+e_3a_1a_0$

Logigramme



Démultiplexeur

Le démultiplexeur effectue l'opération inverse d'un multiplexeur, c'est-à-dire qu'il permet d'aiguiller l'information présente à l'entrée unique vers l'une des 2 n sorties. La sélection de la sortie se fait à l'aide de n lignes d'adressage.

Exemple : Démux 1 vers 4

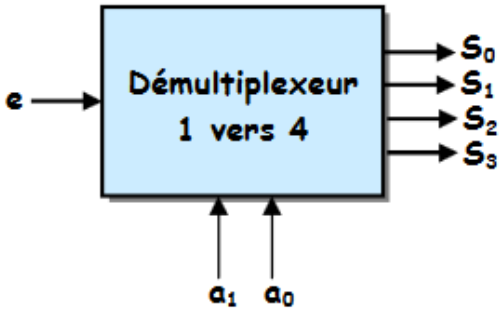
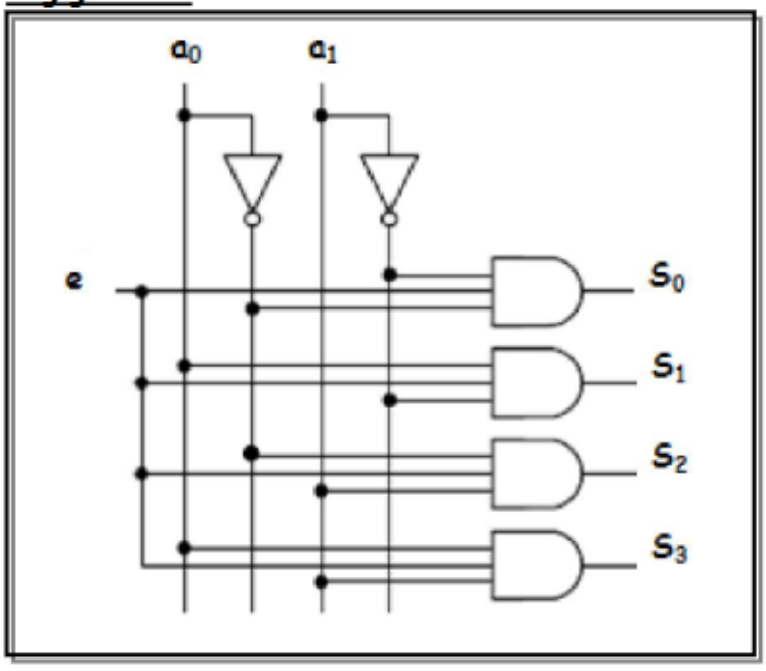


Table de vérité					
Adresses		Sorties			
a_1	a_0	S_0	S_1	S_2	S_3
0	0	e	0	0	0
0	1	0	e	0	0
1	0	0	0	e	0
1	1	0	0	0	e

Equations logiques
$S_0=e.\bar{a}_1\bar{a}_0$
$S_1=e.\bar{a}_1a_0$
$S_2=e.a_1\bar{a}_0$
$S_3=e.a_1a_0$

Logigramme

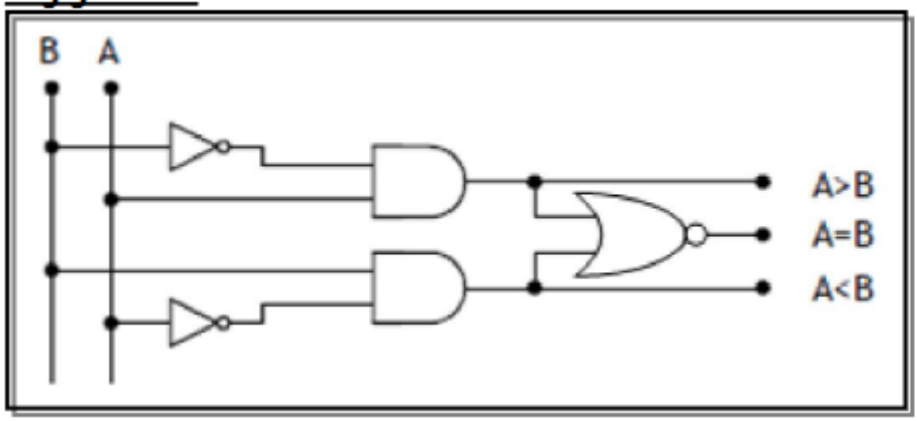


Comparateur

Le comparateur est un circuit permettant de détecter l'égalité de deux nombres binaires et éventuellement d'indiquer le nombre le plus grand ou le plus petit.

Pour comprendre le principe, on va concevoir un comparateur élémentaire de deux mots A et B de 1 bit.

Equations logiques
$(A>B)=A\overline{B}$
$(A=B)=\overline{A \oplus B}$
$(A<B)=\overline{A}B$

Table de vérité				
Entrées		Sorties		
B	A	A>B	A=B	A<B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Logigramme