Lycée Technique Mohammedia

Logique Combinatoire

1ère STE Unité ATC

Professeur : **MAHBAB**

1. SYSTÉTÈMES DE NUMÉRISATION :

1.1 Définition :

Le système de numération décrit la façon avec laquelle les nombres sont représentés. Un système de numération est défini par :

- ✓ Un alphabet A : ensemble de symboles ou chiffres,
- ✓ Des règles d'écritures des nombres : Juxtaposition de symboles

1.2 Système décimal :

C'est le système de numération décimal que nous utilisons tous les jours. C'est le système de base 10 qui utilise donc 10 symboles différents : **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8** et **9**. Un nombre N (entier positif) exprimé dans le système de numération décimale est défini par la relation ci-dessous :

$$N = a_n * 10^n + a_{n-1} * 10^{n-1} \ldots\ldots\ldots + a_1 * 10^1 + a_0 * 10^0$$

Ou a_n est un chiffre de rang n.

Dans ce système, le poids est une puissance de 10.

Exemple : $N = (1975)_{10}$

$N = 1 * 10^3 + 9 * 10^2 + 7 * 10^1 + 5 * 10^0$

	Unité	Dizaine	Centaine	Millier	10* Millier	100*Millier
Chiffre	a_0	a_1	a_2	a_3	a_4	a_5
Rang	0	1	2	3	4	5
Poids	10^0	10^1	10^2	10^3	10^4	10^5

1.3 Système binaire :

Le système binaire est le système de base 2, c'est à dire qui utilise deux symboles différents : le 0 et le 1. Chacun d'eux est appelé bit (contraction de binary digit).

Un nombre N (entier positif) exprimé dans le système de numération binaire est défini par la relation ci-dessous :

$$N = b_n * 2^n + b_{n-1} * 2^{n-1} \ldots\ldots\ldots + b_1 * 2^1 + b_0 * 2^0$$

Ou b_n est un bit de rang n.

Dans ce système, le poids est une puissance de 2.

Exemple : $N = (10110)2$

$N = 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 0 * 2^0$

Bit	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
Rang	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Poids	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	2^{13}	2^{14}	2^{15}

Notations des valeurs binaires :

Pour identifier l'écriture en binaire d'un nombre binaire, il peut être précédé du signe % ou suivi de l'indice de base (2) ou d'un B ; Exemple : % 01000110 $(1000110)_2$ 01000110_B

Etendue des valeurs :

En utilisant n bits, on peut former 2^n nombres différents et le plus grand d'entre eux est égal à (2^n-1).Par exemple si n = 8, Nmax = $(2^8 -1) = 255$, on peut former 256 nombres différents de 0 $(00000000)_2$ a 255 $(11111111)_2$.

Définitions :

Quartet : nombre binaire formé de 4 éléments binaires. **0001, 1001 ,1111**

Octet (byte) : nombre binaire formé de 8 éléments binaires. **00000010, 10101111**

Mot (word) : nombre binaire formé de **16**, **32** ou **64** éléments binaires.

L.S.B. : Bit le moins significatif ou bit de poids faible (élément le plus à droite). 0001000**1**

M.S.B. : bit le plus significatif ou bit de poids fort (élément binaire le plus à gauche). **0**0010001

1.4 Système hexadécimale :

Le système hexadécimal est de base 16 et utilise 16 symboles différents :
Les dix premiers chiffres décimaux : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 et les 6 premières lettres de l'alphabet : A, B, C, D, E, F.

La succession des nombres hexadécimaux par ordre croissant est la suivante :

- ✓ 1 chiffre : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, 0, 1, 2, 3.....etc.
- ✓ 2 chiffres : 00, 01, 02 ....., 09, 0A, 0B,....., 0F, 10, 11, 12,....., 19,1A, 1B.....etc.

Les lettres A à F correspondent respectivement aux nombres décimaux 10 à 15.

Un nombre N (entier positif) exprimé dans le système de numération hexadécimale est défini par la relation ci-dessous :

$$N = a_n * 16^n + a_{n-1} * 16^{n-1} \ldots\ldots\ldots + a_1 * 16^1 + a_0 * 16^0$$

Ou a_n est un chiffre de rang n.

Dans ce système, le poids est une puissance de 16.

Exemple :

$N = (AC53)_{16}$

$N = A * 16^3 + C * 16^2 + 5 * 16^1 + 3 * 16^0$

$N = 10 * 16^3 + 12 * 16^2 + 5 * 16^1 + 3 * 16^0$

Chiffre	a_0	a_1	a_2	a_3	a_4	a_5
Rang	0	1	2	3	4	5
Poids	16^0	16^1	16^2	16^3	16^4	16^5

Un nombre hexadécimal peut être précédé du signe $ ou suivi de l'indice de base (16) ou de la lettre H. Exemple : $F6B1 $(F6B1)_{16}$ $F6B1_H$

Tableau de correspondance entre nombre de différentes bases

Décimal (base 10)	Binaire (base 2)	Hexadécimal (base 16)
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

2. Conversion entre systèmes:

2.1 Conversion de la base 10 vers une autre base :

Nous divisons le nombre décimal à convertir par la base b et nous conservons le reste (division entière). Le quotient obtenu est ainsi successivement divisé tant qu'il n'est pas nul. Les restes successifs sont écrits, en commençant par le dernier, de la gauche vers la droite pour former l'expression de N dans le système de base b.

Exemples de conversion de la base 10 vers la base 2 :

205 | 2 1 | 102 | 2 0 | 51 | 2 1 | 25 | 2 1 | 12 | 2 0 | 6 | 2 0 | 3 | 2 1 | 1 | 2 1 | 0	$N = 205_{10}$ $N = 11001101_2$

125 | 2 1 | 62 | 2 0 | 31 | 2 1 | 15 | 2 1 | 7 | 2 1 | 3 | 2 1 | 1 | 2 1 | 0	$N = 125_{10}$ $N = 1111101_2$

Exemples de conversion de la base 10 vers la base 16 :

205 | 16 13 | 12 | 16 12 | 0	$N = 205_{10}$ $N = CD_{16}$
125 | 16 13 | 7 | 16 7 | 0	$N = 125_{10}$ $N = 7D_{16}$
255 | 16 15 | 15 | 16 15 | 0	$N = 255_{10}$ $N = FF_{16}$
200 | 16 8 | 12 | 16 12 | 0	$N = 200_{10}$ $N = C8_{16}$

2.2 Autre conversion :

Conversion Hexa - Binaire :

Chaque symbole du nombre hexadécimal est remplacé par son équivalent écrit dans le système binaire.

Exemple : N = B F 8 $_{16}$

N = **B** **F** **8**$_{16}$

1011 **1111** **1000** **N = 1011.1111.1000** $_2$

Conversion Binaire - Hexa :

C'est l'inverse de la précédente. Il faut donc regrouper les 1 et 0 du nombre par quartet en commençant par la droite, puis chaque groupe est remplacé par le symbole hexadécimal correspondant.

Exemple : N = 100001101111_2

N = **1000** **0110** **1111**$_2$

8 **6** **F** **N = 86F**$_{16}$

3. Le code binaire réfléchi (ou code Gray):

La propriété réside dans le fait qu'un seul bit change d'état entre deux nombres consécutifs.

Comparaison entre le binaire et le binaire réfléchi :

Comparaison entre le binaire et le binaire réfléchi		
Décimal	Binaire pur	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

Le terme réfléchi est du a la symétrie qui apparaît dans le code

4. Le code 'binaire code décimale' (B, C, D):

Ce codage est destiné a l'affichage de valeurs décimales, chaque digit doit être codé en binaire sur 4 bits (unités, dizaines, centaines ...). Il ne permet aucun calcul, il est uniquement destiné a la saisie et a l'affichage de données

Exemple : 236 ≡ [0010] [0011] [0110] (soit un mot de 12 bits)

5. Le code ASCII:

Le code **ASCII** (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange) 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 (b_6 b_5 b_4 b_3 b_2 b_1 b_0), ce qui permet de représenter 128 (2^7) caractères différents. La table suivante montre un tel codage. Par exemple, Le code de la lettre "**A**" (majuscule) est :

- ✓ en binaire : b_6 b_5 b_4 b_3 b_2 b_1 b_0 = **1000001** ;
- ✓ en hexadécimal **41** ;
- ✓ en décimal **65**.

Binaire				Hexadécimal	Décimal	b6 b5 b4 = 000	001	010	011	100	101	110	111
						0	1	2	3	4	5	6	7
b3	b2	b1	b0			0	16	32	48	64	80	96	112
0	0	0	0	0	+0	NUL	TC7 (DEL)	SP	0	@	P	`	p
0	0	0	1	1	+1	TC1 (SOH)	DC1	!	1	A	Q	a	q
0	0	1	0	2	+2	TC2 (STX)	DC2	"	2	B	R	b	r
0	0	1	1	3	+3	TC3 (ETX)	DC3	#	3	C	S	c	s
0	1	0	0	4	+4	TC4 (EOT)	DC4	$	4	D	T	d	t
0	1	0	1	5	+5	TC5 (ENQ)	TC8 (NAK)	%	5	E	U	e	u
0	1	1	0	6	+6	TC6 (ACK)	TC9 (SYN)	&	6	F	V	f	v
0	1	1	1	7	+7	BEL	TC 10 (ETB)	'	7	G	W	g	w
1	0	0	0	8	+8	FE0 (BS)	CAN	(	8	H	X	h	x
1	0	0	1	9	+9	FE1 (HT)	EM	)	9	I	Y	i	y
1	0	1	0	A	+10	FE2 (LF)	SUB	*	:	J	Z	j	z
1	0	1	1	B	+11	FE3 (VT)	ESC	+	;	K	[	k	é
1	1	0	0	C	+12	FE4 (FF)	IS4 (FS)	,	<	L	\	l	ù
1	1	0	1	D	+13	FE5 (CR)	IS3 (GS)	-	=	M	]	m	è
1	1	1	0	E	+14	SO	IS2 (RS)	.	>	N	^	n	¯
1	1	1	1	F	+15	SI	IS1 (US)	/	?	O	_	o	DEL

6. NOTIONS D'ARITHMETIQUE BINAIRE:

6.1. Cas de l'addition et la soustraction :

Quand vous faites une addition en décimal, vous faites la somme des chiffres se trouvant dans une même colonne. Si la somme est inférieure à 10, alors vous posez le résultat obtenu et passez à la colonne suivante.
Si la somme est supérieure à 10, alors vous posez le chiffre des unités et gardez en retenue le chiffre des dizaines.
Si vous faites la somme de 2 nombres, alors la retenue ne pourra être supérieure à 1.

Le principe est exactement le même en binaire. Vous faites la somme, posez le chiffre des unités, et retenez le chiffre de la seconde colonne en retenue (qu'il vaut mieux, évidemment, éviter d'appeler les « dizaines »).

Si vous faites la somme de **2 nombres**, alors il n'y a que **4 cas possibles** :

- ❖ 0 + 0 = 0, on pose 0 et on retient 0
- ❖ 0 + 1 = 1, on pose 1 et on retient 0
- ❖ 1 + 0 = 1, on pose 1 et on retient 0
- ❖ 1 + 1 = 0, on pose 0 et on retient 1

Bit 1	0	0	1	1
Bit 2	0	1	0	1
Résultat	0	1	1	0
Retenu	0	0	0	1

Si vous faites la différence de **2 nombres**, alors il n'y a que **4 cas possibles** :

- ❖ 0 - 0 = 0, on pose 0 et on retient 0
- ❖ 0 - 1 = 1, on pose 1 et on retient 1
- ❖ 1 - 0 = 1, on pose 1 et on retient 0
- ❖ 1 - 1 = 0, on pose 1 et on retient 0

Bit 1	0	0	1	1
Bit 2	0	1	0	1
Résultat	0	1	1	0
Retenu	0	1	0	0

Exemple :

11		1	0	1	1
+ 7			1	1	1
=18	1	0	0	1	0

19	1	0	0	1	1
- 5			1	0	1
=14	0	1	1	1	0

6.2. Représentation des nombres :

Dans les calculs, on manipule des nombres positifs et négatifs ; il faut alors coder le signe algébrique. Plusieurs modes de représentation sont adoptés en fonction des calculs à effectuer et les caractéristiques technologiques des systèmes de traitement.

6.2.1 Représentation par valeur absolue et signe :

Pour le bit de signe, on adopte la convention 0 (+) et 1 (-).
Exemple :

$(+35)_{10}$ = **0 100011** et $(-35)_{10}$ = **1 100011**.

Cette solution a comme inconvénient la complexité de la réalisation technologique due à :
- Un traitement spécifique du signe ;
- Une double représentation du 0.

6.2.2 Représentation par complément à 2 :

Soit un nombre binaire A sur n bits et son complément (nommé aussi complément à 1 de A), on a : - A = /A + 1 est appelé complément à 2
Exemple : Pour n = 4, on obtient :

$(A)_{10}$	A	/A	(/A + 1)	$(-A)_{10}$
7	0111	1000	1001	-7
6	0110	1001	1010	-6
5	0101	1010	1011	-5
4	0100	1011	1100	-4
3	0011	1100	1101	-3
2	0010	1101	1110	-2
1	0001	1110	1111	-1
0	0000	1111	0000	-0

$(A)_{10}$	7	6	5	4	3	2	1
+ $(-A)_{10}$	-7	-6	-5	-4	-3	-2	-1
= 0	0	0	0	0	0	0	0

A	0111	0110	0101	0100	0011	0010	0001
+ /A +1	1001	1010	1011	1100	1101	1110	1111
= 0	0000	0000	0000	0000	0000	0000	0000

On remarque que :
- Le MSB représente le signe avec 0 (+) et 1 (-).
- Le zéro n'a qu'une seule représentation ;
- Alors pour effectuer une soustraction, il suffit de faire une addition avec le complément à 2.
- Le résultat se lit directement en complément à 2 :
  - Si le signe est + la lecture est direct ;
  - Si le signe est -, on convertit le résultat en recherchant le complément à 2 de celui-ci.

1. OPÉRATIONS BOOLÉENNES ÉLÉMENTAIRES :

Trois opérations élémentaires suffisent pour définir une algèbre de Boole :

- l'inversion : Non (Not) ;
- le produit logique : ET (AND) ;
- la somme logique : OU (OR).booléenne élémentaire

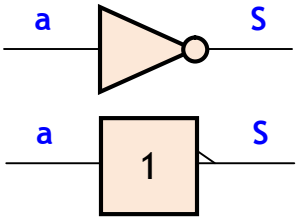
1.1 Opération Inversion (NON) :

C'est une opération définie sur une seule variable. La sortie prend la valeur que n'a pas l'entrée. On dit que la sortie est l'inverse ou le complément de l'entrée.

Table de vérité

a	S
.....	.....
.....	.....

Symbole

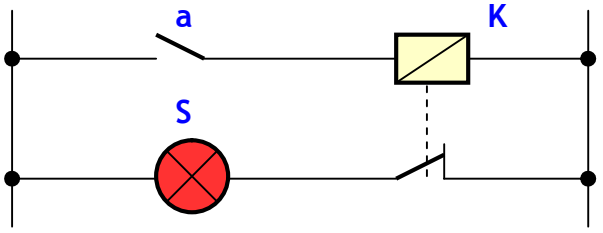


$S = \overline{a}$

(Se lit A barre)

Propriété : $\overline{\overline{S}} = S$

Schéma électrique



- L'interrupteur **a** ouvert (**a = 0**) ; le relais K est non excité et le contact qui lui est associé reste fermé (position de repos) ; la lampe **S** est allumée (**S = 1**) : **a = 0 → S = 1**.
- L'interrupteur **a** fermé (**a = 1**) ; le relais K est excité et le contact qui lui est associé est ouvert; la lampe **S** est éteinte (**S = 0**) : **a = 1 → S = 0.**

1.2 Opération ET (AND) :

C'est une opération sur 2 variables d'entrée au moins. Dans le cas simple de 2 entrées a et b, la sortie est vraie (égale à 1) si **a ET b** sont vraies aussi.

Table de vérité

a	b	S
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....

Symbole

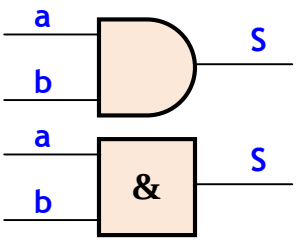
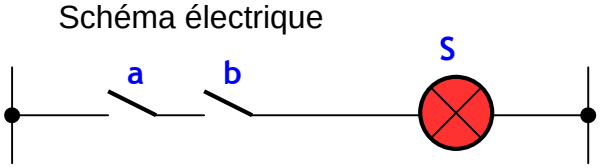


Schéma électrique



La lampe S est allumée (**S = 1**) si l'interrupteur A **ET** l'interrupteur B sont fermés (**a = b = 1**),

Soit **S = a. b**

Propriétés :

- La fonction AND est commutative: **S = a.b = b.a.**
- La fonction AND est associative: **S = a. (b.c) = (a.b).c = a.b.c.**
- La fonction AND est généralisable pour n entrées.
- Identités remarquables : **a.0 = 0 ; a.1 = a ; $a.\overline{a}$ = 0** et **a.a = a.**

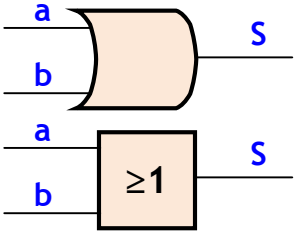
1.3 Opération OU (OR) :

C'est une opération sur 2 variables d'entrée au moins. Dans le cas simple de 2 entrées a et b, la sortie est vraie (égale à 1) si seulement **a OU b** est vraie.

Table de vérité

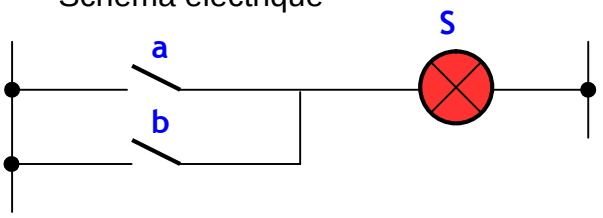
a	b	S
.....	.....	.....
.....	.....	.....
.....	.....	.....
.....	.....	.....

Symbole



$S = a + b$
(Se lit a OU b)

Schéma électrique



La lampe S est allumée (**S = 1**) si l'interrupteur A **OU** l'interrupteur B sont fermés (**a = 1 OU b = 1**),
Soit **S = a + b**

Propriétés :

- La fonction OR est commutative: **S = a + b = b + a.**
- La fonction OR est associative: **S = a + (b + C) = (a + b) + C = a + b + C.**
- La fonction OR est généralisable pour n entrées.
- Identités remarquables : **a + 0 = a ; a + 1 = 1; a + a = a** et **a + $\bar{a}$ = 1.**

1.4 Propriétés et théorèmes remarquables :

Propriétés :

- **(b + c).a = a.b + a.c** (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 + $\bar{a}$.b = b:** b.(a + $\bar{a}$) = b. 1 = b (Factorisation);
- **a + a.b = a** : a (1 + b) = a. 1 = a (Loi d'absorption) ;
- **a + $\bar{a}$.b = a + b:** (a + $\bar{a}$). (a + b) = 1. (a + b) = a + b;

Théorème de Morgan :

a	b	$\bar{a}.\bar{b}$	$\overline{a + b}$	$\overline{a.b}$	$\bar{a} + \bar{b}$
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....
.....	.....	.....	.....	.....	.....

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

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

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

Exemple: $S = a.b.\bar{d} + \bar{a}.d \rightarrow \overline{S} = \overline{a.b.\bar{d} + \bar{a}.d} = \overline{a.b.\bar{d}} \,.\, \overline{\bar{a}.d} = (\bar{a} + \bar{b} + d).(a + \bar{d})$

$S = a.b.d + a.\bar{d} + \bar{a}.b \rightarrow \overline{S} = \overline{a.b.d + a.\bar{d} + \bar{a}.b} = \overline{a.b.d} \,.\, \overline{a.\bar{d}} \,.\, \overline{\bar{a}.b} = (\bar{a} + \bar{b} + \bar{d}).(\bar{a} + d).(a + \bar{b})$

2. AUTRES OPERATIONS :

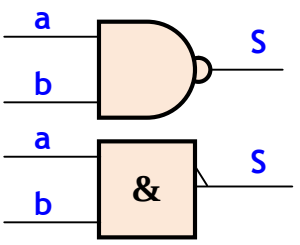
2.1. Opération NAND :

C'est le complément de l'opération AND.

Table de vérité

a	b	S
……	……	……
……	……	……
……	……	……
……	……	……

Symbole



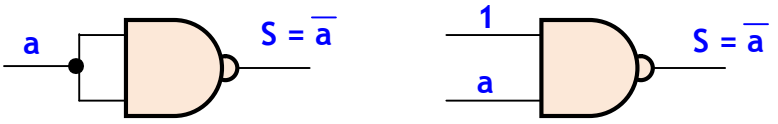
$S = \overline{a.b}$

(Se lit A ET B tout barre)

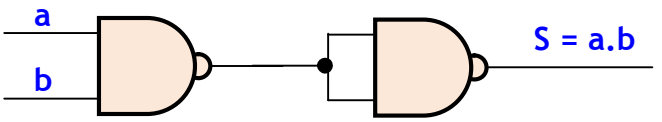
Propriétés :

- La fonction NAND est commutative ;
- La fonction NAND n'est pas associative ;
- La fonction NAND est généralisable pour n entrées ;
- L'opérateur NAND est dit "système logique complet", car il permet de réaliser toutes les opérations de base : Not, AND et OR ; et par conséquent, toute fonction logique :

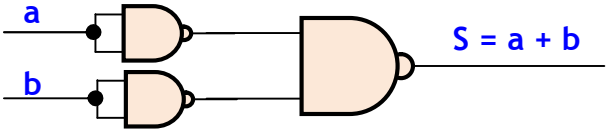
  - Réalisation d'un inverseur :



  - Réalisation d'une AND :



  - Réalisation d'une OR :



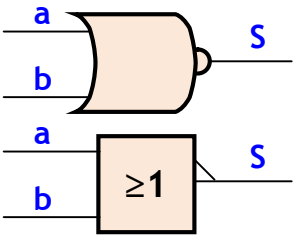
2.2. Opération NOR :

C'est le complément de l'opération OR.

Table de vérité

a	b	S
……	……	……
……	……	……
……	……	……
……	……	……

Symbole



$S = \overline{a + b}$

(Se lit A OU B tout barre)

Propriétés :

- Comme la fonction NAND, la fonction NOR n'est ni combinatoire, ni associative ; elle est aussi généralisable pour n entrées,
- L'opérateur NOR est un système logique complet, comme le NAND.;

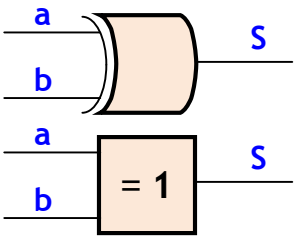
2.3. Opération XOR :

Cette opération diffère du OR classique ou inclusif ; S est égale à 1 si (a=0 ET b=1) OU (a=1 ET b=0) ; formellement, on écrit : **$S = a.\bar{b} + \bar{a}.b$ → $S = a \oplus b$**

Table de vérité

a	b	S
……	……	……
……	……	……
……	……	……
……	……	……

Symbole



$S = a \oplus b$
(Se lit a OU exclusif b)

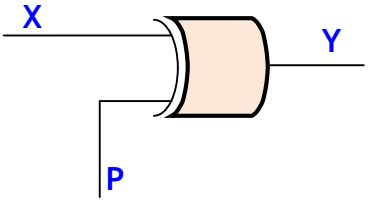
Propretés :

- L'opération XOR est commutative : **$F = a \oplus b = b \oplus a$.**
- L'opération XOR est associative : **$F = a \oplus (b \oplus c) = (a \oplus b) \oplus c = a \oplus b \oplus c$.**
- L'opération XOR n'est pas généralisable pour n entrées.

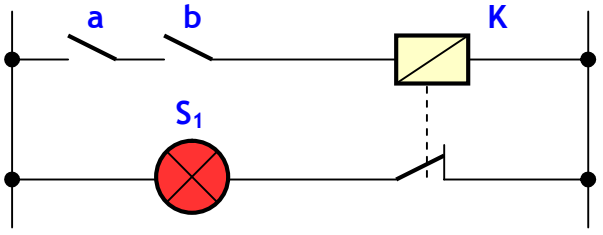
Remarque :

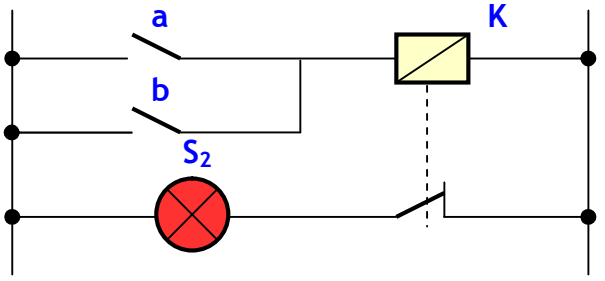
L'opérateur OU Exclusif est considéré comme l'opérateur programmable le plus élémentaire.

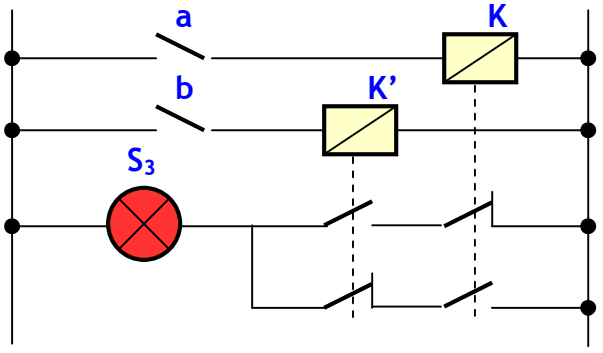
P	X	Y	
……	……	……	SI P = 0 → Y = X → Fonction Identité
……	……	……	
……	……	……	SI P = 1 → Y = /X → Fonction Inversion
……	……	……	



Exercice :







A	B	S1	S2	S3
0	0	……	……	……
0	1	……	……	……
1	0	……	……	……
1	1	……	……	……

S1 = ……
S2 = ……
S3 = ……

1. REPRÉSENTATION DES FONCTIONS LOGIQUES :

Pratiquement, une fonction logique est représentée par :

- son équation logique qui n'est qu'une association de sommes et de produits logiques ;
- sa table de vérité ou son tableau de Karnaugh ;
- Son logigramme qui est une représentation symbolique, sous forme d'un schéma, formé par les différentes liaisons entres les symboles des opérateurs élémentaires.

Exemple :

Voilà les 3 représentations d'une certaine fonction S à 3 variables a, b et c :

- L'équation logique donnée est : S (a, b, c) = a.b + a.c ;
- La table de vérité, déduite à partir de l'équation, est : On à 3 variables d'entrées, donc on a 2^3 combinaisons possibles (2^3 lignes de la table). D'une façon générale, on a 2^n combinaisons pour n variables d'entrée. On déduit l'équation logique de la fonction S, à partir de la table de vérité suivant le raisonnement suivant :

  - On cherche les lignes où la fonction S est égale à 1 ;
  - On note la combinaison des entrées pour chacune de ces lignes ;
  - On somme logiquement ces combinaisons.

Table de vérité

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**

Ainsi, la fonction S est égale à 1 si on a : $\bar{a}.b.\bar{c}$ OU $\bar{a}.b.c$ OU $a.\bar{b}.c$ OU $a.b.c$, ce qui donne :

S = ...
S = ...
S = ...

- Le logigramme déduit de l'équation est :

a	b	c

Remarque :

On remarque que cette petite fonction emploie différents types de portes logiques : inverseur, AND et OR. Il est évident qu'il serait rentable de réaliser cette fonction logique avec le minimum de matériel (circuits logiques), ce qui demande une bonne analyse du problème pour simplifier la fonction en question.

2. SIMPLIFICATION DES FONCTIONS LOGIQUES :

2.1 Mise en situation :

Soit à déterminer une équation simplifiée de la sortie S dont la table de vérité et la suivante :

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

On obtient ainsi l'équation :
S = ...

Problème : On doit simplifier cette équation.

Pour simplifier une équation logique on utilise deux méthodes :
- Une méthode **algébrique**
- Une méthode **graphique** dite par **tableau de Karnaugh**

2.2 Méthode algébrique :

Propriétés de la fonction OU	
a + 0 = ...	0 : élément neutre
a + 1 = ...	1 : élément absorbant
a + a = ...	Idempotence
a + $\overline{a}$ = ...	Complémentation
a + b = ...	Commutativité
a + b + c = a + (b +c) = ...	Associativité

Propriétés de la fonction ET	
a. 0 = ...	0 : élément neutre
a. 1 = ...	1 : élément neutre
a. a = ...	Idempotence
a . $\overline{a}$ = ...	Complémentation
a. b = ...	Commutativité
a. b. c = a. (b. c) = ...	Associativité

Applications :

Simplifier les équations logiques suivantes

1°- H1 = a + ab = ...
2°-H2 = a + a. $\overline{b}$ = ...
3°-H3 = (a + b). (a + c) = ...

Simplification de S par la méthode algébrique :

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

Conclusion :

La méthode de simplification algébrique peut nous conduire à des calculs relativement longs. Pour éviter ces calculs, on emploie une deuxième méthode qui utilise le **tableau de Karnaugh**

Relations fondamentales
a + a. b = ...
a + $\overline{a}$.b = ...
(a + b).(a + c) = ...
(a + b). (a + c).(a + d) ...= .....

2.3 Méthode graphique :

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

$S = \bar{a}.\,b.\,\bar{c} + \bar{a}.\,\bar{b}.\,c + \bar{a}.\,b.\,\bar{c} + a.\,\bar{b}.\,c + a.\,b.\,\bar{c} + a.\,b.\,c$

Affecter **d'indice «1»** les cases correspondantes aux termes de l'équation à simplifier et l'indice **«0»** aux autres.

C \ A.B	$\bar{A}.\bar{B}$	$\bar{A}.B$	$A.B$	$A.\bar{B}$
$\bar{C}$	1	1	1	0
C	1	1	1	0

Ces cases forment la surface **S1**

C \ A.B	$\bar{A}.\bar{B}$	$\bar{A}.B$
$\bar{C}$	1	1
C	1	1

C \ A.B	$\bar{A}.\bar{B} + \bar{A}.B$
$\bar{C} + C$	1 1
	1 1

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

Donc $S1 = \bar{A}$

Ces cases forment la surface **S2**

C \ A.B	$\bar{A}.B$	$A.B$
$\bar{C}$	1	1
C	1	1

C \ A.B	$\bar{A}.B + A.B$
$\bar{C} + C$	1 1
	1 1

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

Donc $S2 = B$

$S = S1 + S2$ d'où $S = A + B$

Exemple 1 : Déterminer graphiquement, l'équation de la sortie S par tableau de Karnaugh

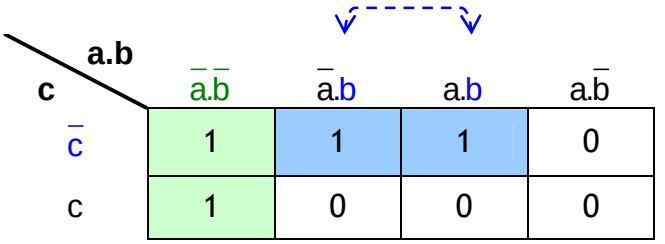
a	b	c	S
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

c \ a.b	$\bar{a}.\bar{b}$	$\bar{a}.b$	$a.b$	$a.\bar{b}$
$\bar{c}$	0	0	0	0
c	1	1	0	0

D'où S = ...

Exemple 2 : Déterminer l'équation de la sortie S par tableau de Karnaugh

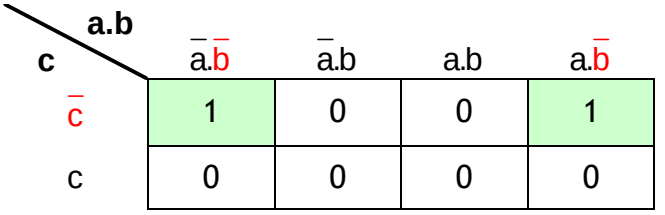
a	b	c	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	0	0



D'où S = ...

Exemple 3 : Déterminer l'équation de la sortie S par tableau de Karnaugh

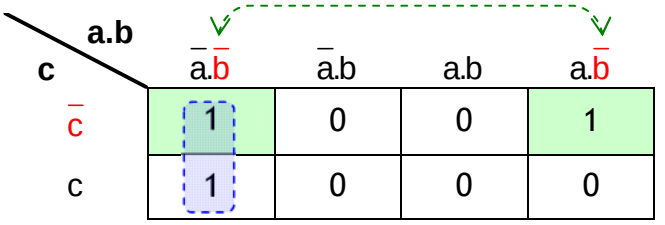
a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



D'où S = ...

Exemple 4 : Déterminer l'équation de la sortie S par tableau de Karnaugh

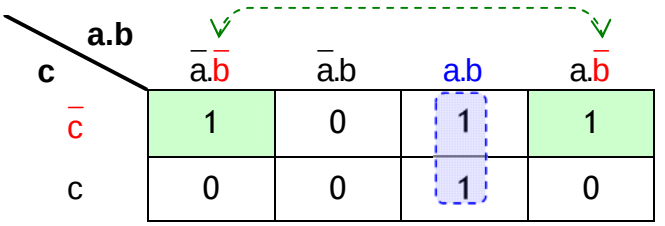
a	b	c	S
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



D'où S = ...

Exemple 5 : Déterminer l'équation de la sortie S par tableau de Karnaugh

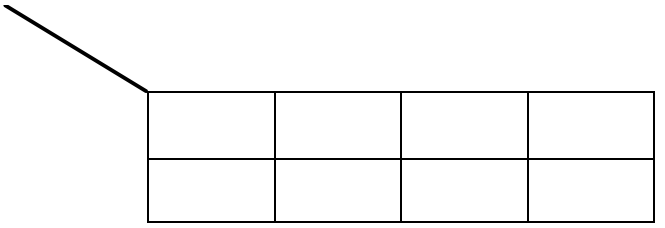
a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



D'où S = ...

Exemple 6 : Déterminer l'équation de la sortie S par tableau de Karnaugh

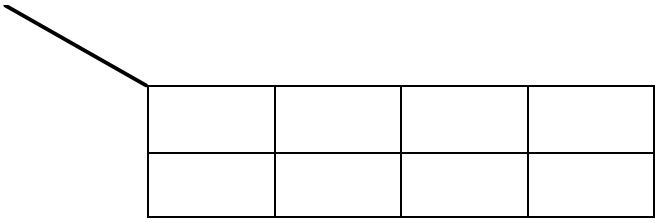
a	b	c	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



D'où S = ...

Exemple 7 : Déterminer l'équation de la sortie S par tableau de Karnaugh

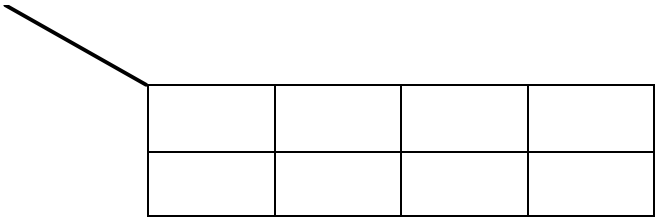
a	b	c	S
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1



D'où S = ...

Exemple 8 : Déterminer l'équation de la sortie S par tableau de Karnaugh

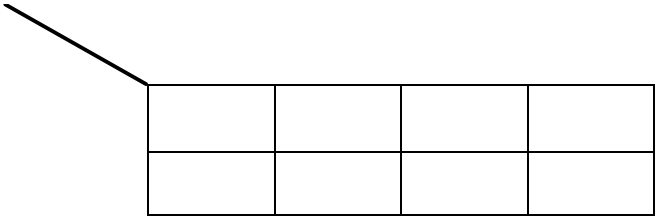
a	b	c	S
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



D'où S = ...

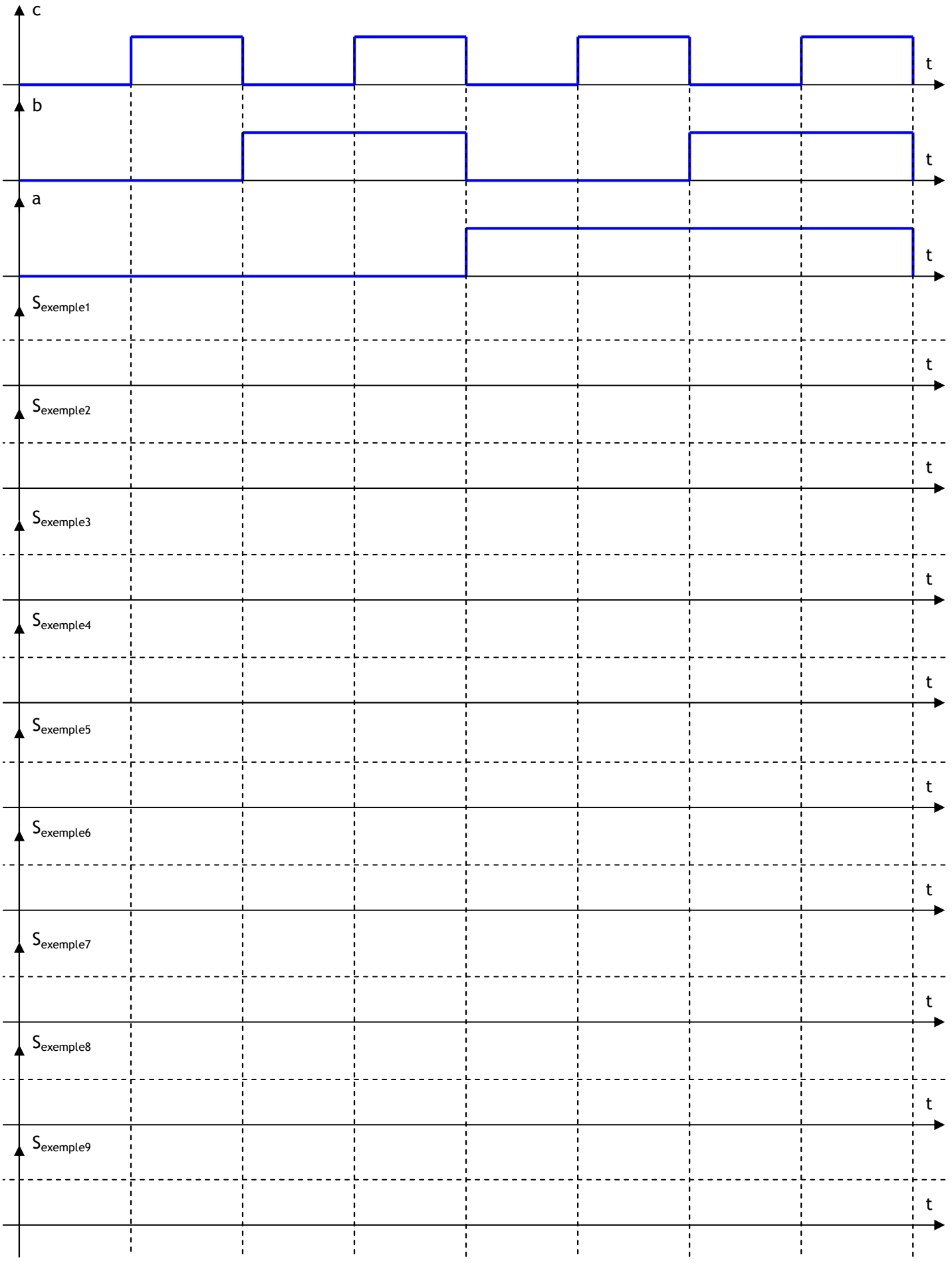
Exemple 9 : Déterminer l'équation de la sortie S par tableau de Karnaugh

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



D'où S = ...

3. CHRONOGRAMMES :



1. INTRODUCTION :

Dans les systèmes numériques, on utilise souvent des fonctions qui on justifié leurs réalisations en circuits intégrés. On note en particulier les décodeurs, les multiplexeurs, les démultiplexeurs et les circuits arithmétiques. Bien qu'ils soient plus ou moins remplacés actuellement par les systèmes programmables (circuits logiques programmables et microprocesseur), ils sont encore utilisés.

2. LE DECODEUR 1 PARMI N :

2.1 Mise en situation :

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

2.2 Décodeur 1 parmi n :

Ce type de décodeur permet de faire correspondre à un code présent en entrée sur m lignes une sortie et une seule active parmi les n = 2^m sorties possibles.

On le désigne aussi par décodeur m lignes vers n lignes.

Si E_{m-1}... E_0 =i alors S_i = 1 avec i [0...n-1] et n = 2^m

2.3 Décodeur 1 parmi 4 :

C'est un décodeur **2** lignes vers **4**.

Table de vérité

E_1	E_0	S_0	S_1	S_2	S_3
0	0				
0	1				
1	0				
1	1				

Sortie active sur niveau haut :

Equations des sorties :

S_0 = ...
S_1 = ...
S_2 = ...
S_3 = ...

Logigramme :

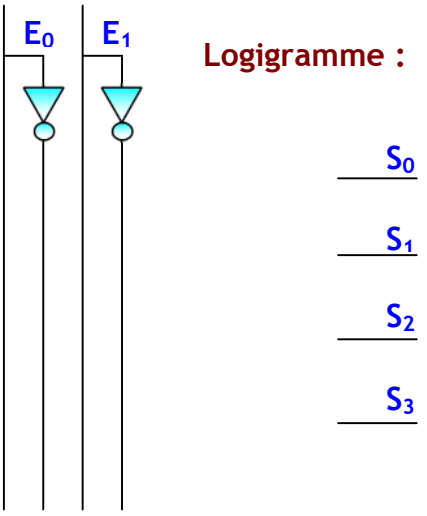


Table de vérité

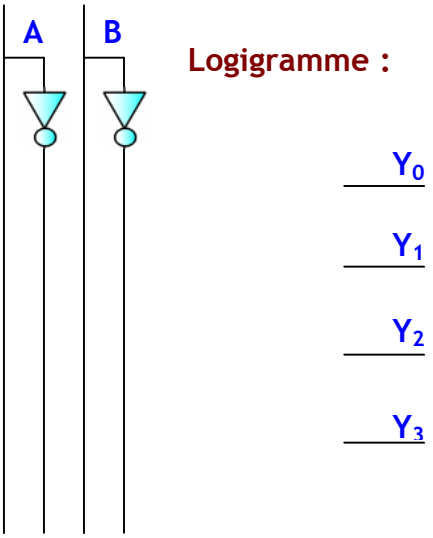
B	A	Y_0	Y_1	Y_2	Y_3
0	0				
0	1				
1	0				
1	1				

Sortie active sur niveau bas :

Equations des sorties :

$\overline{Y_0}$ = ...
$\overline{Y_1}$ = ...
$\overline{Y_2}$ = ...
$\overline{Y_3}$ = ...

Logigramme :



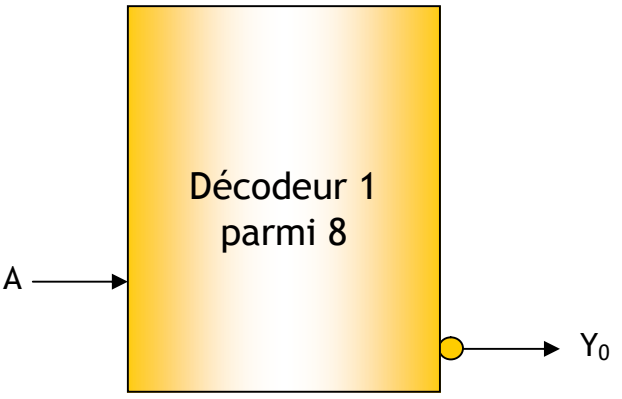
2.4 *Décodeur 1 parmi 8 :*

C'est un décodeur **3** lignes vers **8**.

Table de vérité

		A	Y_0						

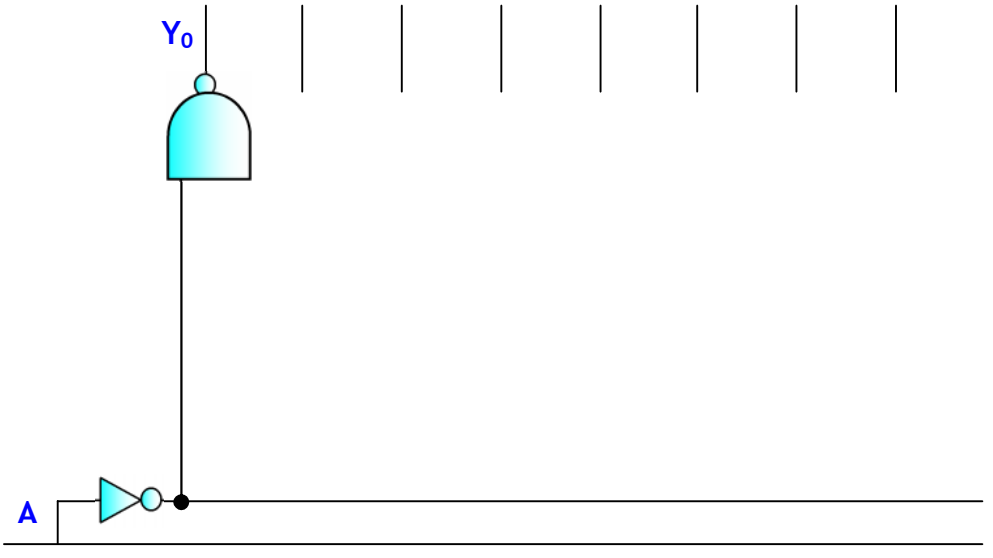
Sortie active sur niveau bas :



Equations des sorties :

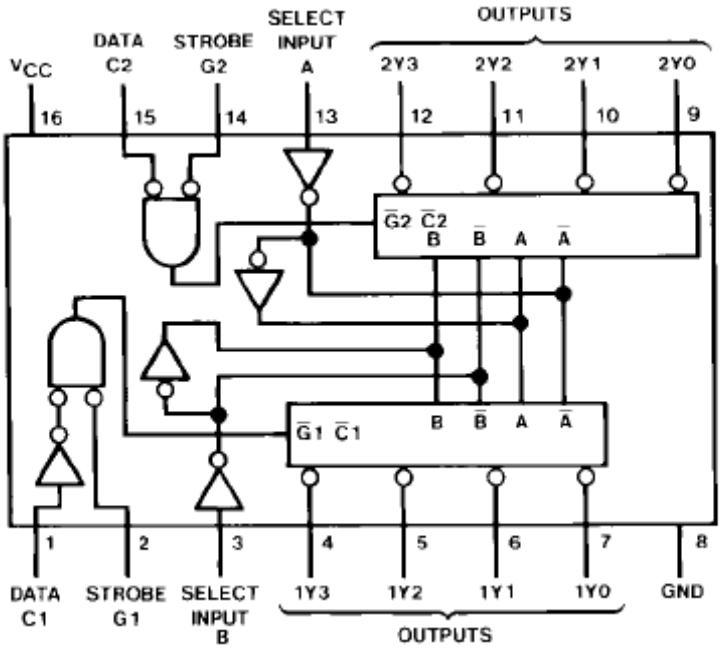
$\overline{Y_0}$ = ...
$\overline{Y_1}$ = ...
$\overline{Y_2}$ = ...
$\overline{Y_3}$ = ...
$\overline{Y_4}$ = ...
$\overline{Y_5}$ = ...
$\overline{Y_6}$ = ...
$\overline{Y_7}$ = ...

Logigramme :



2.5 Circuit intégré 74LS156 :

Connection Diagram

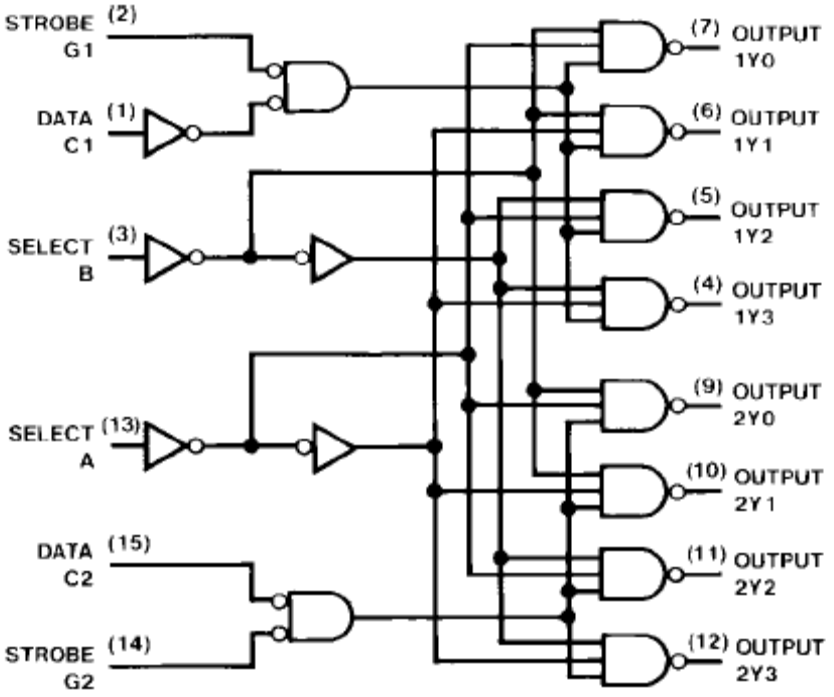


Function Tables

Inputs				Outputs			
Select		Strobe	Data	1Y0	1Y1	1Y2	1Y3
B	A	G1	C1				
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

Inputs				Outputs			
Select		Strobe	Data	2Y0	2Y1	2Y2	2Y3
B	A	G2	C2				
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H

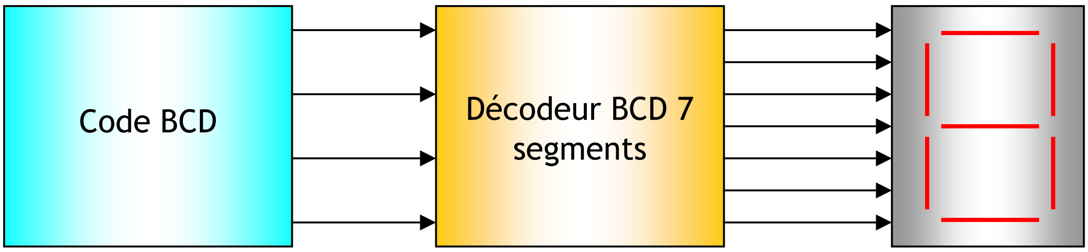
Logic Diagram



3. LE DECODEUR BCD 7 SEGMENTS :

3.1 Définition :

Ce type de décodeur permet de convertir le code BCD 4bits à l'entrée pour obtenir à la sortie un code 7 segments permettant de commander un afficheur 7 segments permettant l'écriture de tous les chiffres.



3.2 Etude d'un décodeur BCD 7 segments :

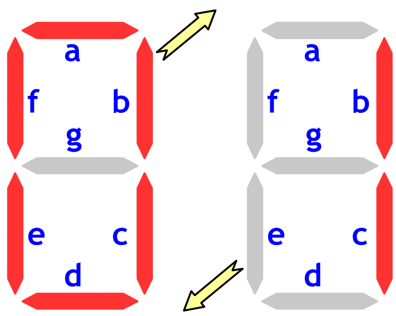
Trouve le code binaire correspondant à l'affichage des chiffres ci-dessous en plaçant :

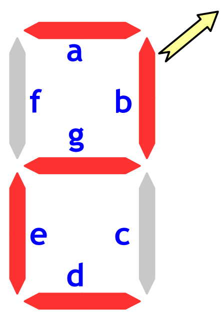
- un « **0** » pour les segments devant être éteint.
- un « **1** » pour les segments devant être allumé.

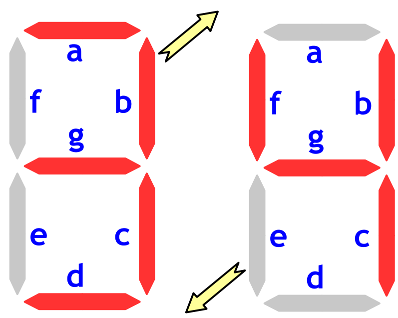
a	b	c	d	e	f	g

a	b	c	d	e	f	g

a	b	c	d	e	f	g







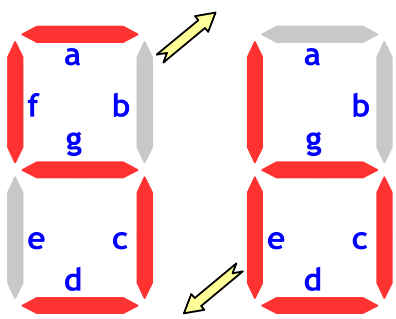
a	b	c	d	e	f	g

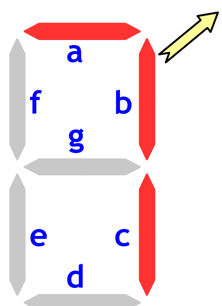
a	b	c	d	e	f	g

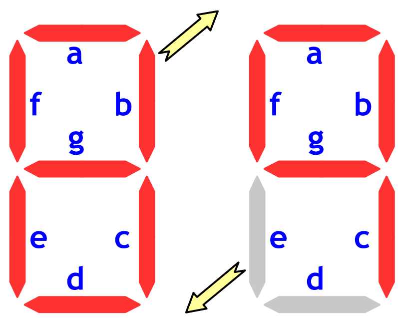
a	b	c	d	e	f	g

a	b	c	d	e	f	g

a	b	c	d	e	f	g







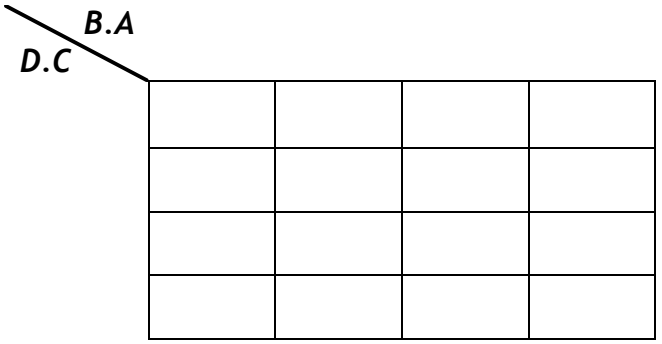
a	b	c	d	e	f	g

a	b	c	d	e	f	g

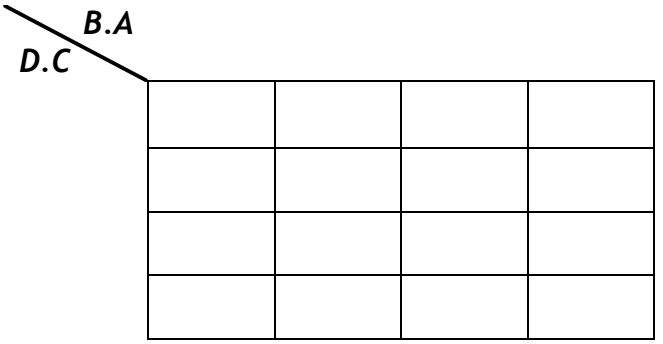
3.2.1 Table de vérité :

N	D	C	B	A		a	b	c	d	e	f	g
0												
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												
11												
12												
13												
14												
15												

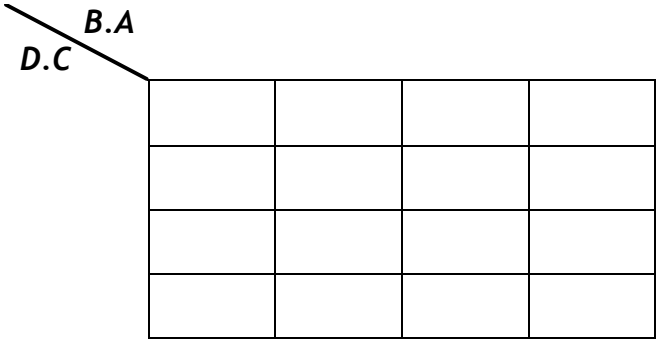
3.2.2 Equations logiques des sorties :



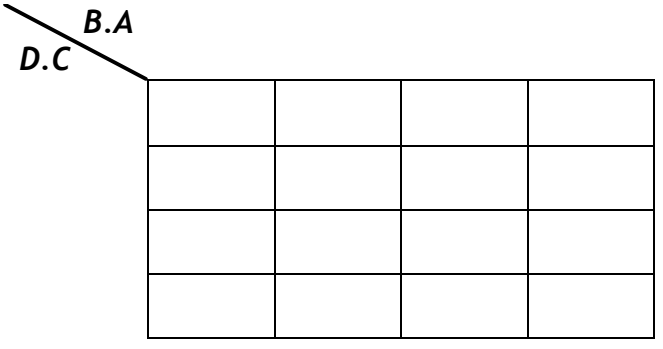
a = ...



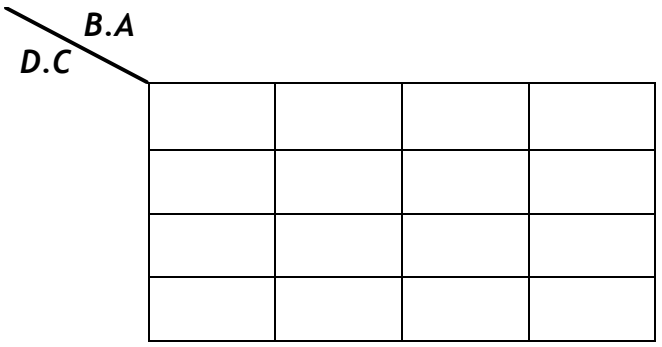
b = ...



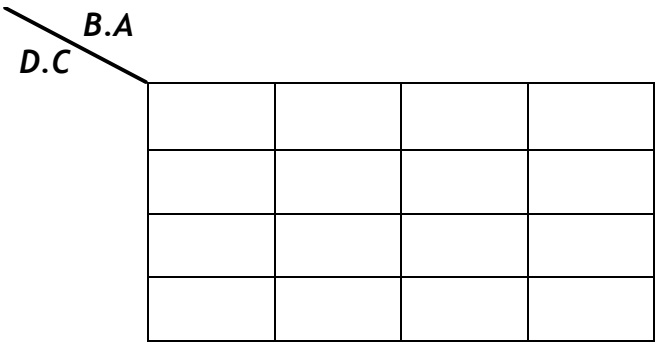
c = ...



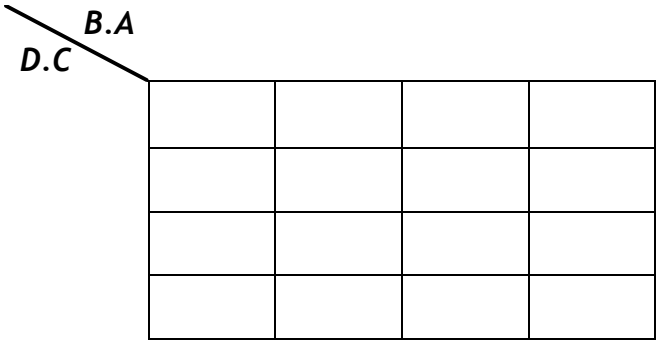
d = ...



e = ...

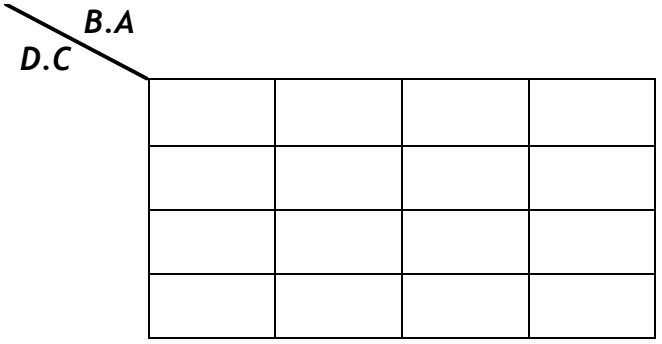


f = ...

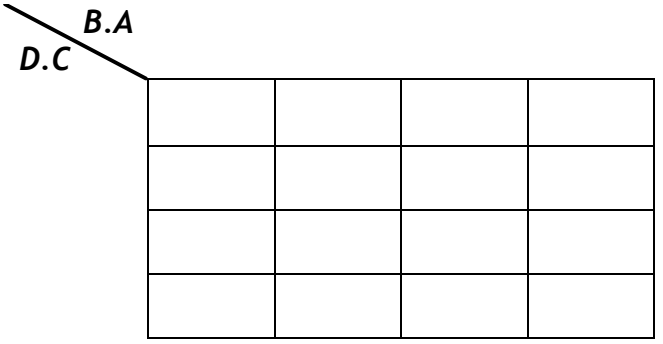


g = ...

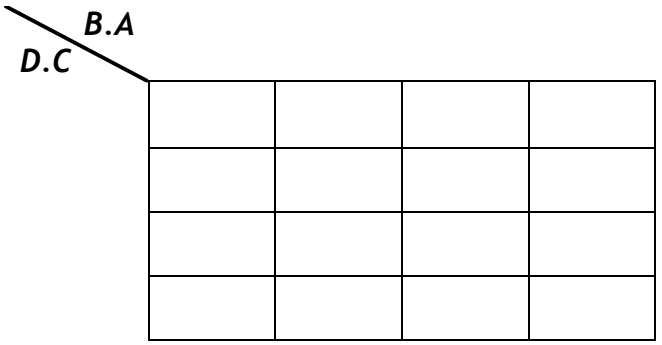
3.2.3 Equations logiques des sorties - Autre solution -:



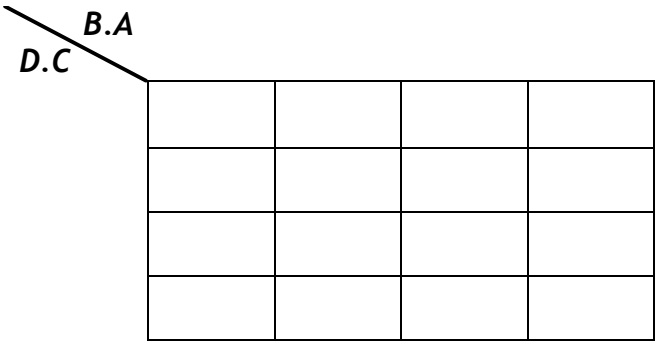
/a = ...



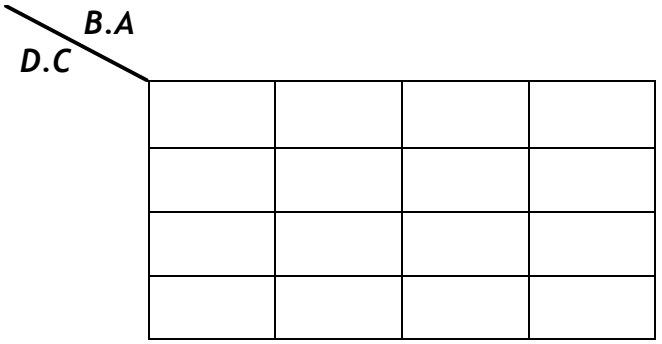
/b = ...



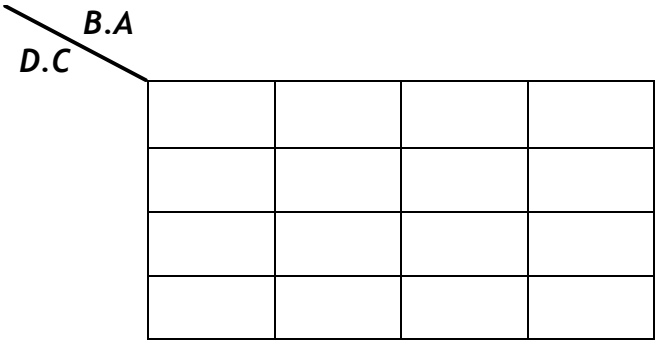
/c = ...



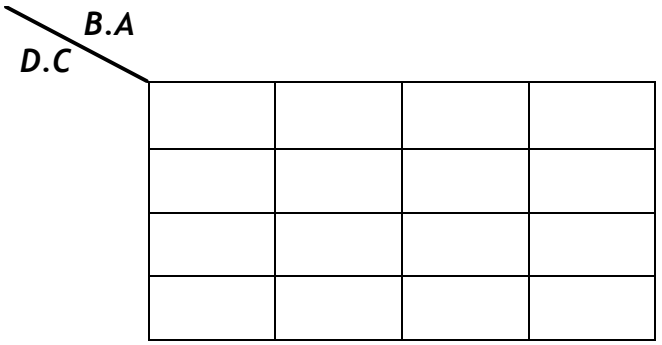
/d = ...



/e = ...

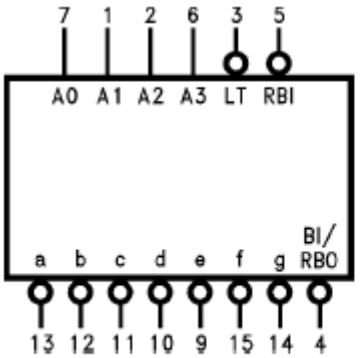


/f = ...

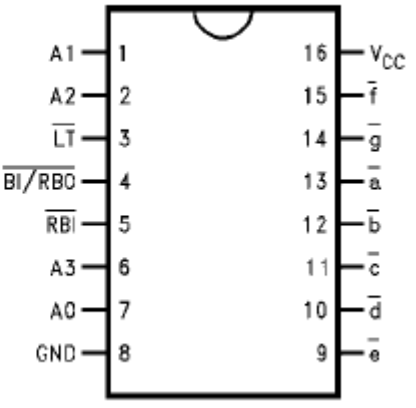


/g = ...

3.3 Diagramme de brochage et table de fonctionnement du 74LS47 :



Connection Diagram

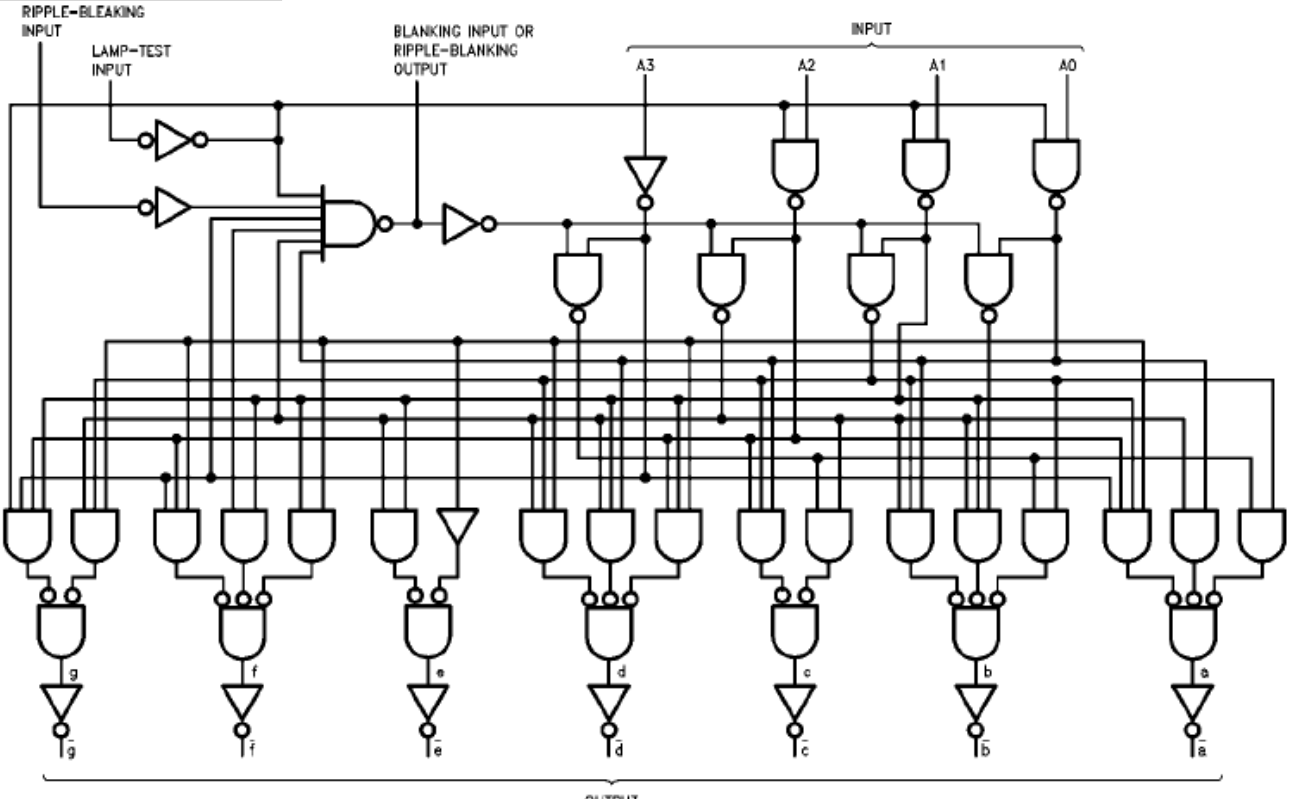


Pin Names	Description
A0–A3	BCD Inputs
$\overline{RBI}$	Ripple Blanking Input (Active LOW)
$\overline{LT}$	Lamp Test Input (Active LOW)
$\overline{BI}/\overline{RBO}$	Blanking Input (Active LOW) or Ripple Blanking Output (Active LOW)
$\overline{a}$ –$\overline{g}$	Segment Outputs (Active LOW) (Note 1)

Function Tables

Decimal or Function	Inputs							Outputs						
	$\overline{LT}$	$\overline{RBI}$	A3	A2	A1	A0	$\overline{BI}/\overline{RBO}$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H
$\overline{BI}$	X	X	X	X	X	X	L	H	H	H	H	H	H	H
$\overline{RBI}$	H	L	L	L	L	L	L	H	H	H	H	H	H	H
$\overline{LT}$	L	X	X	X	X	X	H	L	L	L	L	L	L	L

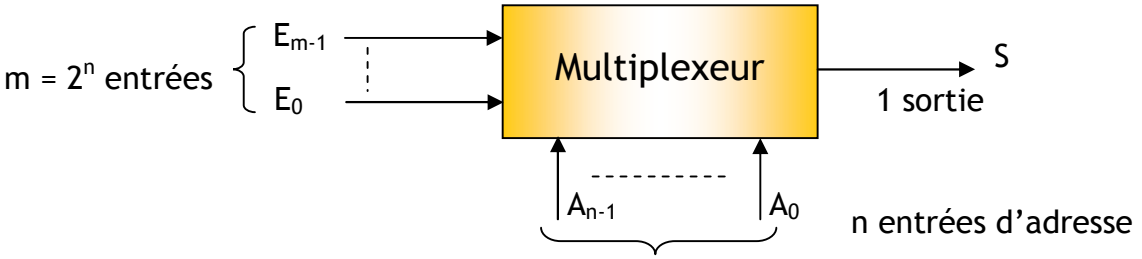
Logic Diagram



4. LE MULTIPLEXEUR :

4.1 Définition :

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.



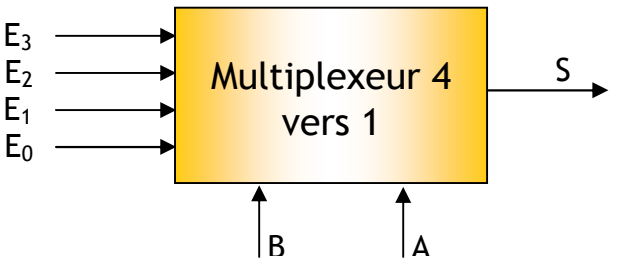
Si A_{n-1}... A_0 =i alors $S = E_i$ avec i [0...m-1] et $m = 2^n$

4.2 Multiplexeur 4 vers 1 :

C'est un multiplexeur **4** entrées et **2** lignes d'adresse.

Table de vérité		
B	A	S

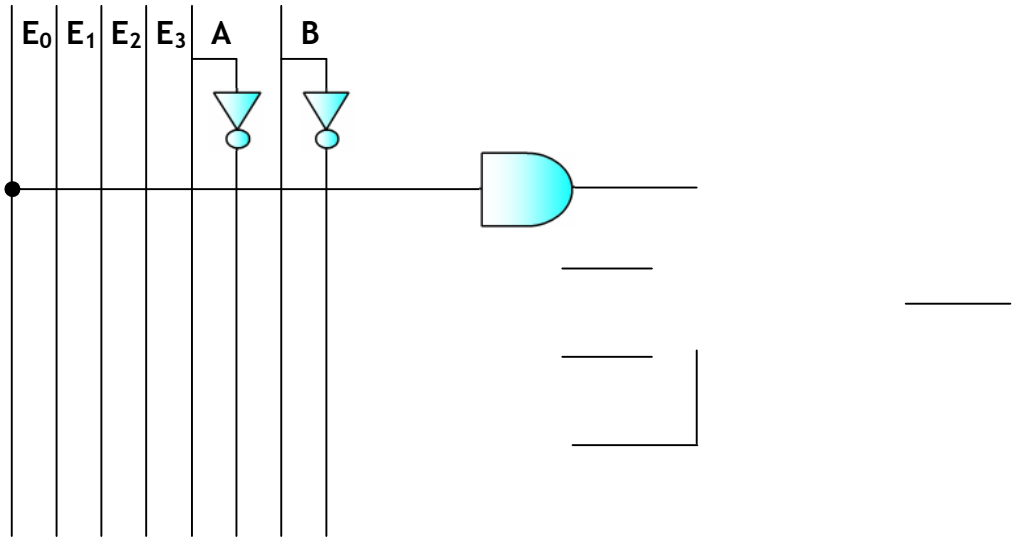
Schéma synoptique :



Equations de la sortie :

S = ...

Logigramme :



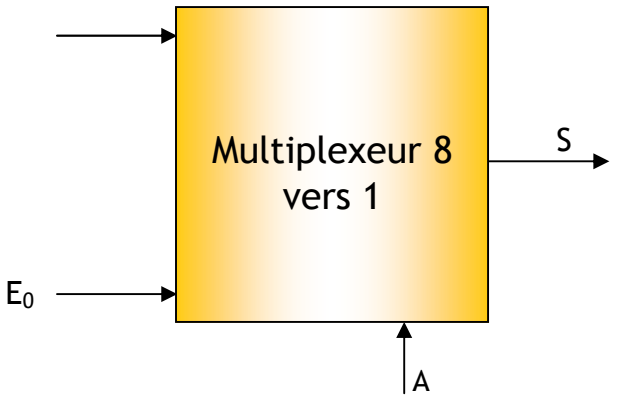
4.3 Multiplexeur 8 vers 1 :

C'est un multiplexeur **8** entrées et **3** lignes d'adresse.

Table de vérité

		A	S

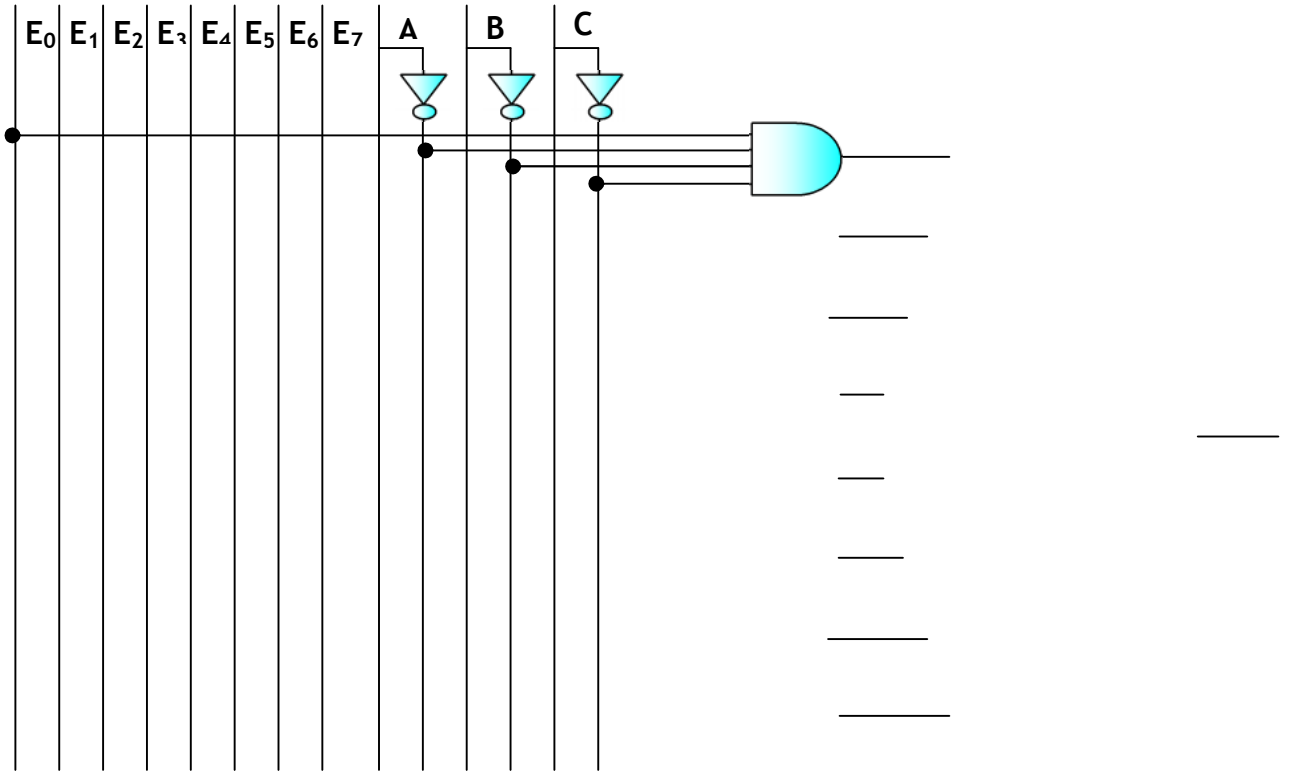
Schéma synoptique :



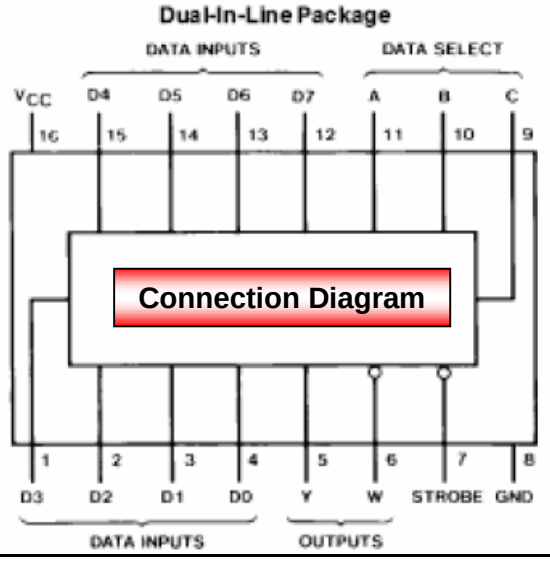
Equations de la sortie :

S = ...

Logigramme :



4.4 Brochage et table de fonctionnement du multiplexeur 74LS151 :



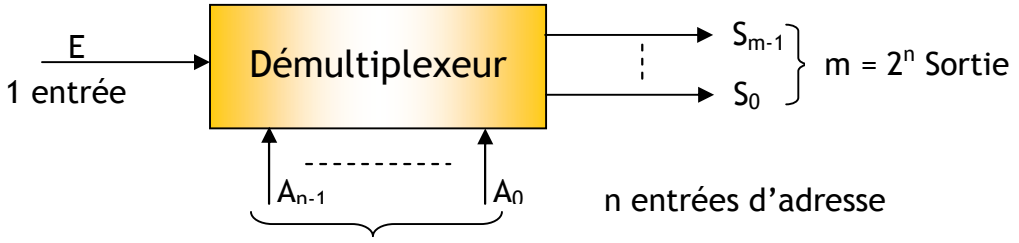
Inputs				Outputs	
Select			Strobe		
C	B	A	S	Y	W
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

Function Tables

5. LE DEMULTIPLEXEUR :

5.1 Définition :

Le démultiplexeur effectue l'opération inverse d'un multiplexeur à savoir il permet de distribuer l'information présente à l'entrée vers l'une des 2^n sorties. La sélection de la sortie se fait à l'aide de n lignes d'adressage.



Si A_{n-1}... A_0 =i alors S_i = E avec i [0...m-1] et m = 2^n

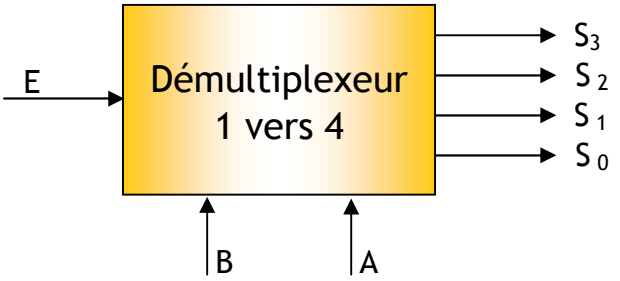
5.2 Démultiplexeur 1 vers 4 :

C'est un démultiplexeur **4** sorties et **2** lignes d'adresse.

Table de vérité

B	A	S_0	S_1	S_2	S_3

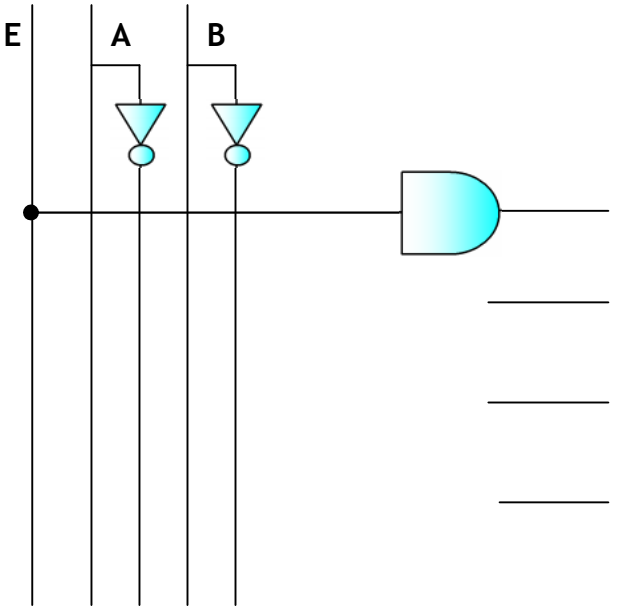
Schéma synoptique :



Equations des sorties :

S_0 = ...
S_1 = ...
S_2 = ...
S_3 = ...

Logigramme :

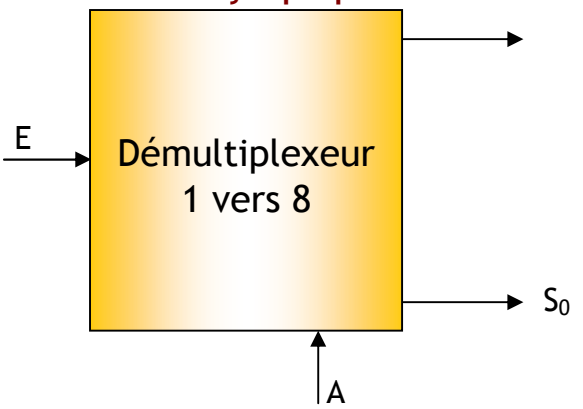


5.3 Démultiplexeur 1 vers 8 :

C'est un démultiplexeur **8** sorties et **3** lignes d'adresse.

Table de vérité										
			S_0							

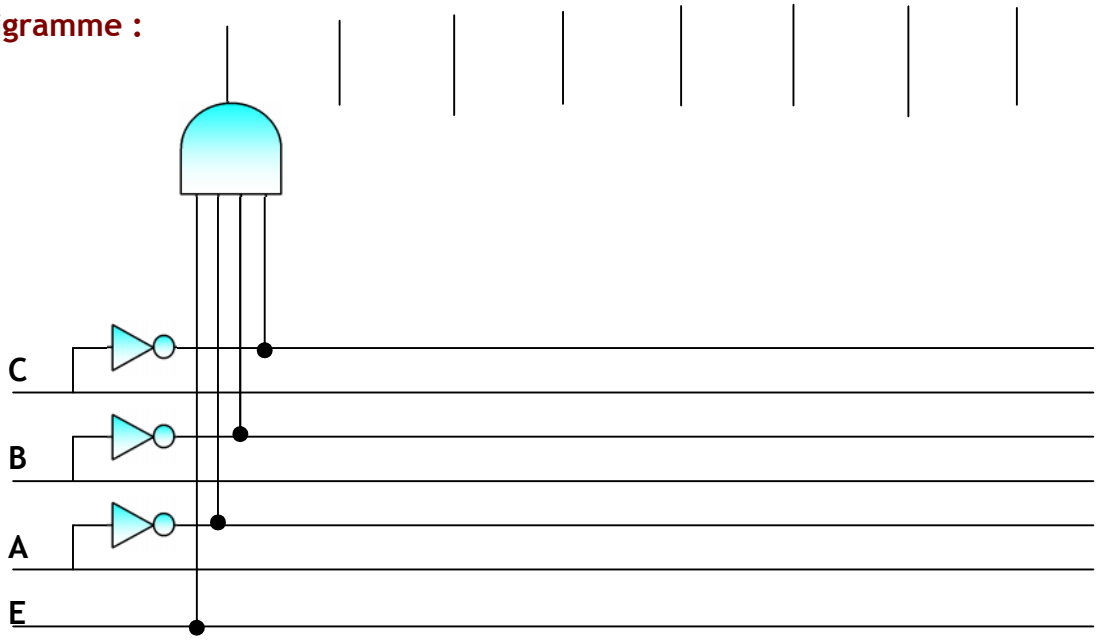
Schéma synoptique :



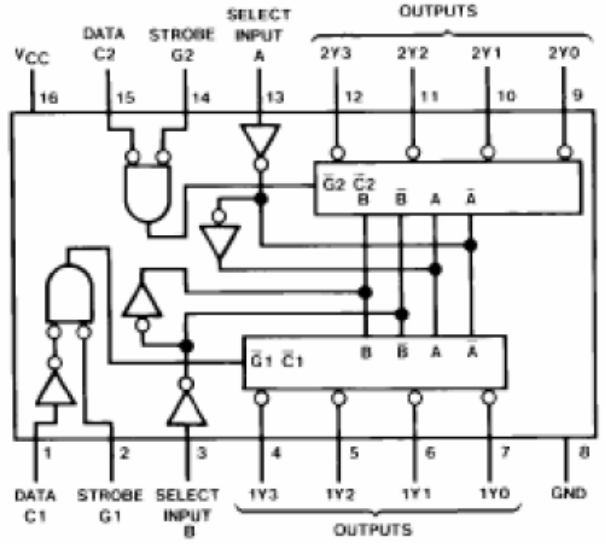
Equations des sorties:

S_0 = ... S_4 = ...
S_1 = ... S_5 = ...
S_2 = ... S_6 = ...
S_3 = ... S_7 = ...

Logigramme :



5.4 *Diagramme de brochage et table de vérité du démultiplexeur 74LS155 :*



Connection Diagram

2-Line-to-4-Line Decoder or
1-Line-to-4-Line Demultiplexer

Inputs				Outputs			
Select		Strobe	Data				
B	A	G1	C1	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

Function Tables

6. L'ADDITIONNEUR :

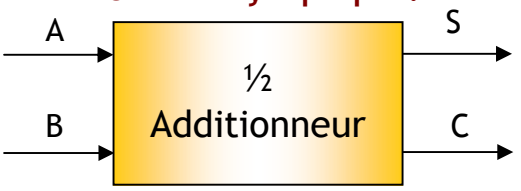
6.1 Le demi-additionneur :

C'est un circuit permettant d'effectuer l'addition de deux bits A et B pour générer leur somme S et leur retenue C (Carry).

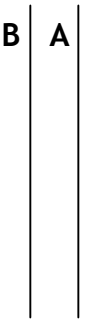
Table de vérité

Entrées		Sorties	
B	A	S	C
0	0		
0	1		
1	0		
1	1		

Schéma synoptique :



Logigramme :



Equations :

S = ...

C = ...

6.2 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 :

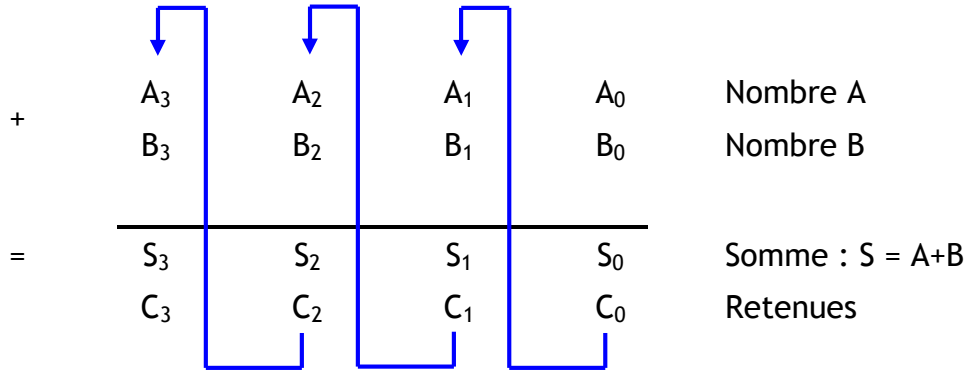
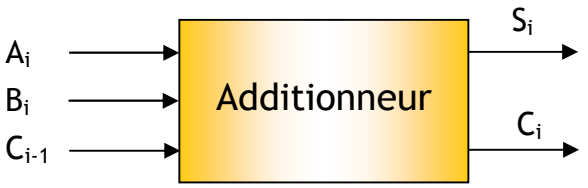


Table de vérité

Entrées			Sorties	
A_i	B_i	C_{i-1}	S_i	C_i

Schéma synoptique :



Equations :

S_i = ...

S_i = ...

S_i = ...

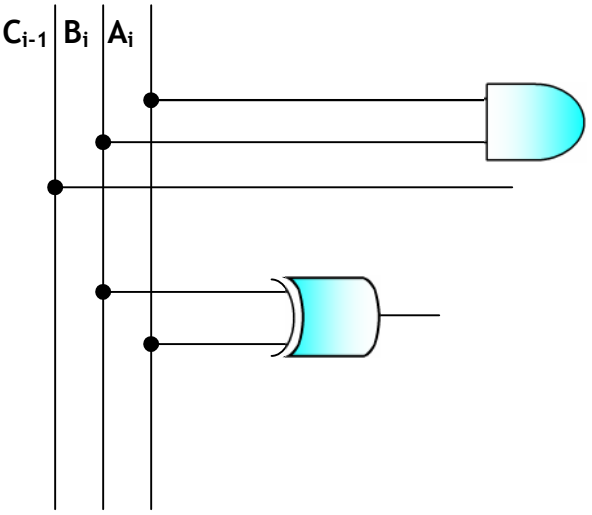
S_i = ...

C_i = ...

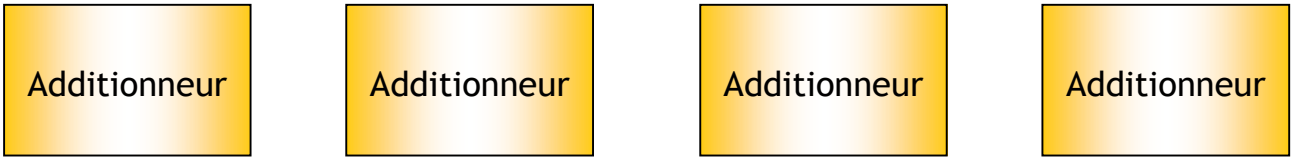
C_i = ...

C_i = ...

Logigramme :

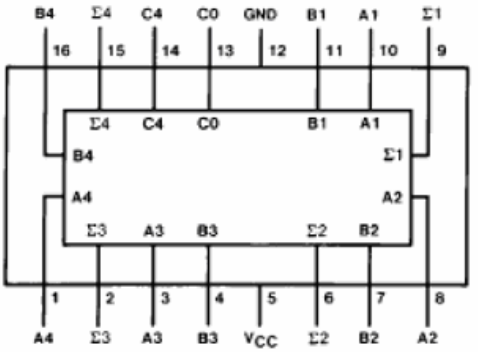


6.3 Additionneur 4 bits :



6.4 Additionneur 4 bits 74LS83 :

Connection Diagram



Function Tables

Inputs				Outputs					
				When C0 = L			When C0 = H		
						When C2 = L			When C2 = H
A1 / A3	B1 / B3	A2 / A4	B2 / B4	Σ1 / Σ3	Σ2 / Σ4	C2 / C4	Σ1 / Σ3	Σ2 / Σ4	C2 / C4
L	L	L	L	L	L	L	H	L	L
H	L	L	L	H	L	L	L	H	L
L	H	L	L	H	L	L	L	H	L
H	H	L	L	L	H	L	H	H	L
L	L	H	L	L	H	L	H	H	L
H	L	H	L	H	H	L	L	L	H
L	H	H	L	H	H	L	L	L	H
H	H	H	L	L	L	H	H	L	H
L	L	L	H	L	H	L	H	H	L
H	L	L	H	H	H	L	L	L	H
L	H	L	H	H	H	L	L	L	H
H	H	L	H	L	L	H	H	L	H
L	L	H	H	L	L	H	H	L	H
H	L	H	H	H	L	H	L	H	H
L	H	H	H	H	L	H	L	H	H
H	H	H	H	L	H	H	H	H	H

7. LE COMPARATEUR :

7.1 Le comparateur :

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

Pour comprendre le principe, on va réaliser un comparateur simple permettant de comparer deux mots de 1 bit.

Table de vérité

Entrées		Sorties		
B	A	S_1	S_2	S_3

Schéma synoptique :

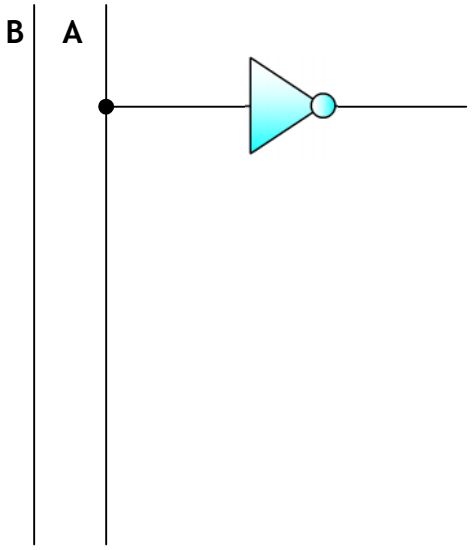
Equations :

S_1 = ...

S_2 = ...

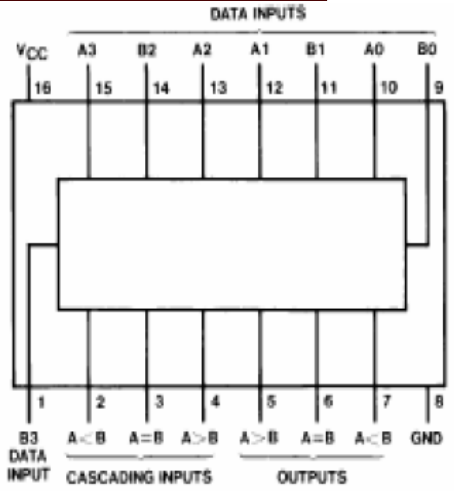
S_3 = ...

Logigramme :



7.2 Le Comparateur 4 bits 74LS85 :

Connection Diagram



Function Tables

Comparing Inputs				Cascading Inputs			Outputs		
A3, B3	A2, B2	A1, B1	A0, B0	A > B	A < B	A = B	A > B	A < B	A = B
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

1. FONCTIONNEMENT :

Afin de la rendre mobile, la parabole est équipée d'un appareil positionneur et un vérin moteur (**M**) à double sens (Il existe alors deux relais électromagnétiques **X** et **Y** et deux boutons poussoirs **e** et **w**) :

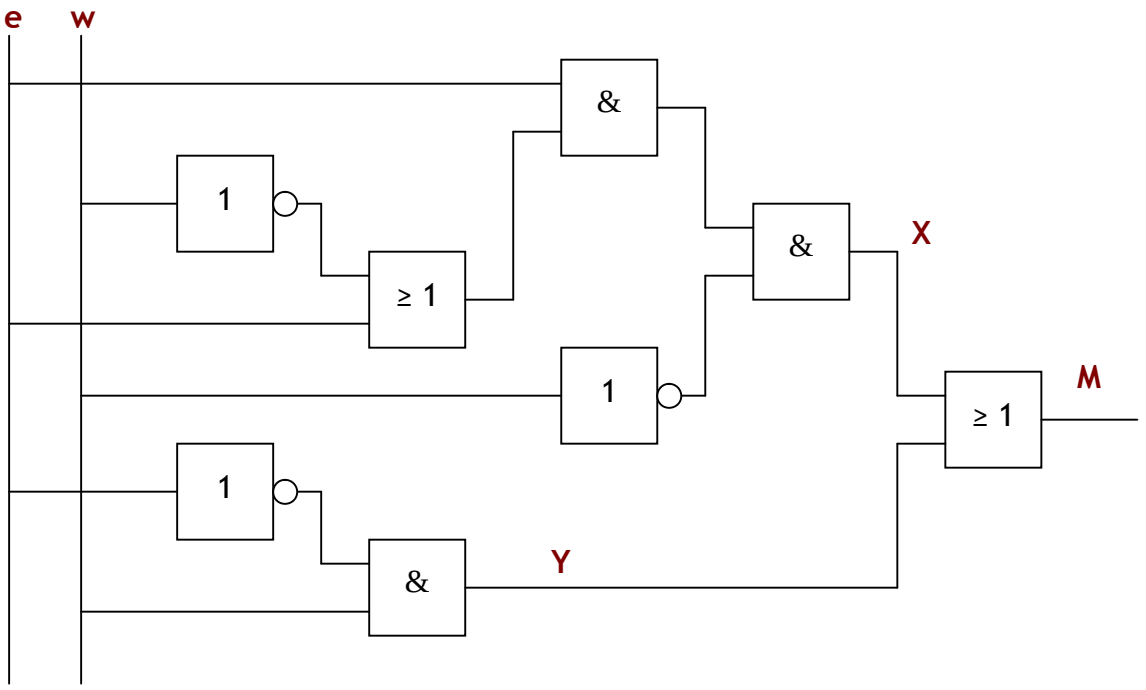
- ✓ Pour tourner la parabole vers l'Est, on appui sur le bouton poussoir : **e**
- ✓ Pour tourner la parabole vers l'Ouest on appui sur le bouton poussoir : **w**

Remarque : **X** = 1 signifie que la parabole tourne dans le sens Est.
Y = 1 signifie que la parabole tourne dans le sens Ouest.

2. TRAVAIL DEMANDÉ :

A. Fonctionnement du positionneur :

On donne le logigramme du positionneur :



1- Colorer en rouge les portes Non,en vert les portes Ou et bleu les portes Et.

2- Déduire l'équation de **M** en fonction de **x** et **y**.

M = ...

3- Déterminer l'équation logique de la sortie **Y** en fonction des entrées **e** et **w**.

Y = ...

4- Déterminer l'équation logique de la sortie **X** en fonction des entrées **e** et **w**.

X = ...

5- Simplifier l'équation logique de **X**.

X = ...

6- Déterminer l'équation logique simplifiée de la sortie **M** en fonction des entrées **e** et **w**.

M = ...

7- Compléter la table de vérité de la sortie **M.**

e	w	$\overline{e}$	$\overline{w}$	$\overline{e}$.w	e.$\overline{w}$	M
....	....	....	....	....	....	....
....	....	....	....	....	....	....
....	....	....	....	....	....	....
....	....	....	....	....	....	....

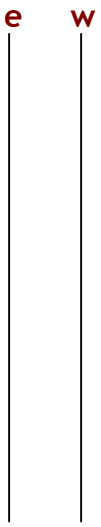
8- Compléter le schéma électrique à contactes du moteur **M** en fonction des variables **e** et **w.**

9- Exprimer l'équation de **M** avec NOR (NON OU) seulement à deux entrées :

M = ...

.....

10- Tracer le logigramme de **M** a l'aide des fonctions NOR (NON OU) seulement.

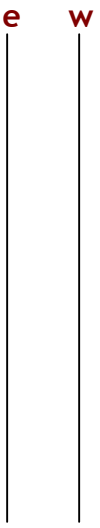


11- Exprimer l'équation de **M** avec NAND (NON ET) seulement à deux entrées :

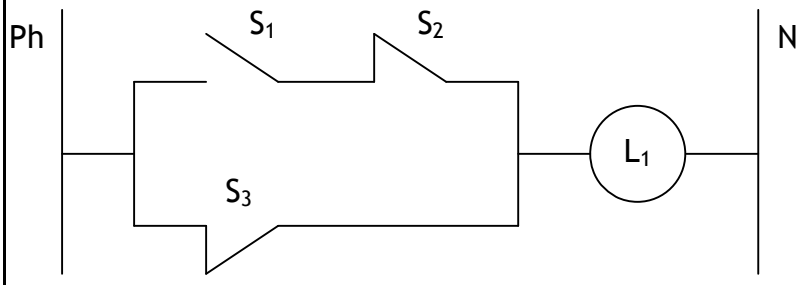
M = ...

.....

12- Tracer le logigramme de **M** a l'aide des fonctions NOR (NON OU) seulement.



B. Soit les schémas à contact suivants :

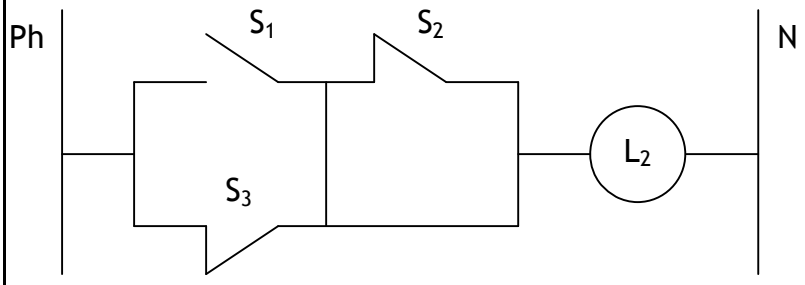


1- Donner les équations des sorties L_1, L_2 et L_3.

L_1 =.....

L_2 =.....

L_3 =.....



2- Exprimer les équations des sorties L_1, L_2 et L_3 avec NAND (NON ET) seulement à deux entrées.

L_1 =.....

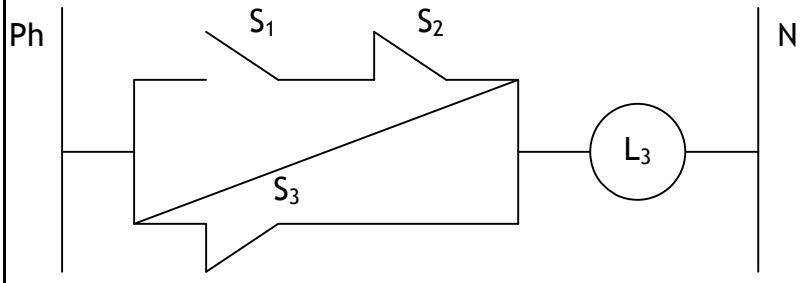
.....

L_2 =.....

.....

L_3 =.....

.....



3- Exprimer les équations des sorties L_1, L_2 et L_3 avec NOR (NON OU) seulement à deux entrées.

L_1 =.....

.....

L_2 =.....

.....

L_3 =.....

.....