

#### ELEMENTS DE LOGIQUE COMBINATOIRE

- 1. Codage des nombres entiers
- 2. Algèbre de BOOLE
- 3. Composants électroniques
- 4. Fonctions complexes

## 1. Codage des nombres entiers



Ecriture polynomiale
Codage Binaire
Codage Hexadécimal
Codage BCD
Codage des nombres signés

### Ecriture polynomiale



Base 10: - 10 symboles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- Expression polynomiale associée à un nombre codé sur n digits :

$$N_{10} = \sum_{i=0}^{n-1} S(i) \times 10^{i}$$
 avec  $S(i) \in \{0,1,2,...,9\}$ 



**Base B**: - B symboles (digits):  $S_0, S_1, ..., S_{B-1}$ 

- Expression polynomiale associée à un nombre codé sur n digits :

$$N_B = \sum_{i=0}^{n-1} S(i) \times B^i \quad \text{ avec } \quad S(i) \in \left\{S_0, S_1, ..., S_{B-1}\right\}$$

### **Codage Binaire**

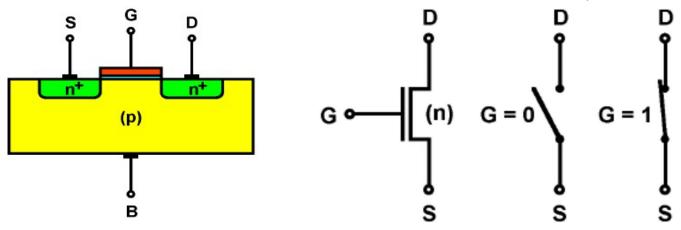


Base 2: - 2 symboles (bits): 0 et 1

- Expression polynomiale associée à un nombre codé sur n bits :

$$N_2 = \sum_{i=0}^{n-1} S_i \times 2^i$$
 avec  $S_i \in \{0,1\}$ 

**Intérêt**: Utilisation du comportement électronique du transistor (tout ou rien)



S = Source G = Grille D= Drain

## Codage Binaire



décimal	<b>b</b> <sub>3</sub>	b <sub>2</sub>	b <sub>1</sub>	b <sub>0</sub>
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

Codage binaire sur 4 bits :

$$N_2 = S_3 2^3 + S_2 2^2 + S_1 2^1 + S_0 2^0$$

$$MSB$$
LSB



Convertir en binaire (38)<sub>10</sub>

## Codage Binaire



décimal	puissances de 2	binaire
0	0	0000 0000 0000
1	20	0000 0000 0001
2	21	0000 0000 0010
4	22	0000 0000 0100
8	23	0000 0000 1000
16	24	0000 0001 0000
32	2 <sup>5</sup>	0000 0010 0000
64	2 <sup>6</sup>	0000 0100 0000
128	27	0000 1000 0000
256	28	0001 0000 0000
512	2 <sup>9</sup>	0010 0000 0000
1024	2 <sup>10</sup> = 1 k	0100 0000 0000
2048	2 <sup>11</sup> = 2 k	1000 0000 0000

## Codage Hexadécimal



Base 16: - 16 symboles: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

- Expression polynomiale associée à un nombre codé sur n digits :

$$N_{16} = \sum_{i=0}^{n-1} S_i \times 16^i$$
 avec  $S_i \in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$ 

décimal	puissances de 16	hexadécimal
0		00000 <sub>H</sub>
1	16 <sup>0</sup>	00001 <sub>H</sub>
16	16 <sup>1</sup>	00010 <sub>H</sub>
256	16 <sup>2</sup>	00100 <sub>H</sub>
4096	$16^3 = 2^{12} = 4 \text{ k}$	01000 <sub>H</sub>
65636	$16^4 = 2^{16} = 64 \text{ k}$	10000 <sub>H</sub>

$$N_{16} = N_H (= $N)$$

Convertir en décimal (2D3)<sub>H</sub>



## Codage Hexadécimal



décimal	binaire	hexadécimal
0	0000 0000	00 <sub>H</sub>
1	0000 0001	01 <sub>H</sub>
2	0000 0010	02 <sub>H</sub>
4	0000 0100	04 <sub>H</sub>
8	0000 1000	08 <sub>H</sub>
16	0001 0000	10 <sub>H</sub>
32	0010 0000	20 <sub>H</sub>
64	0100 0000	40 <sub>H</sub>
128	1000 0000	80 <sub>H</sub>



Intérêt: représentation plus compacte que le Binaire



Convertir en hexadécimal  $(100110)_2$ 

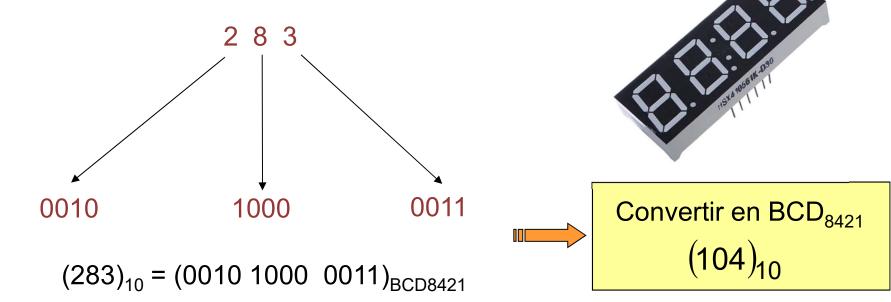
## Codage BCD (Binary Coded Decimal)



#### **Binary Coded Decimal = Décimal Codé en Binaire**

Principe : Chaque chiffre d'un nombre décimal est considéré séparément

puis codé en binaire.



## Codage en Gray



Principe: Pour passer d'une valeur à la suivante, on ne change qu'un

seul bit.

décimal	<b>g</b> <sub>2</sub>	<b>g</b> <sub>1</sub>	<b>g</b> <sub>0</sub>
0	0	0	0
1	0	0	1
2	0	1	1
3	0	1	0
4	1	1	0
5	1	1	1
6	1	0	1
7	1	0	0

15 1001 1110 1011 0010 1101 0011 12 1100 01004 0111 5 10,111 8 1000

Code Gray sur 3 bits

3IRC 18-19

#### Conversion Décimal vers Binaire



#### **Méthode:**

Divisions et multiplications successives.

**Exemple**:  $(34,625)_{10} \rightarrow (100010,101)_2$ 

#### **Partie entière**

$34 \div 2$	= 17	reste C
17 ÷ 2	8 =	reste 1
8 ÷ 2	= 4	reste C
4 ÷ 2	= 2	reste C
2 ÷ 2	= 1	reste C
1 ÷ 2	=0	reste 1

#### **Partie fractionnaire**

0,625	x 2	= 1,25
0,25	x 2	= 0.5
0,5	x 2	=1

#### Code ASCII



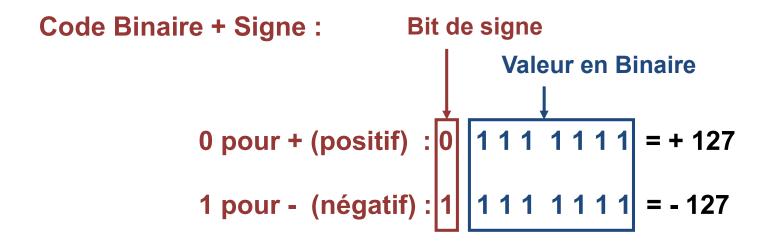
Le code ASCII est un code alphanumérique. Il associe à chaque caractère (alphabétique ou numérique) un code exprimé par un nombre représenté en binaire ou en hexadécimal.

Le 8<sup>ième</sup> bit peut servir à définir les caractères spécifiques à une langue (é, è, ...).

	ASCII 7	Hex
Α	100 0001	\$41
В	100 0010	\$42
С	100 0011	\$43
	• • •	
Ζ	101 1010	\$5A
0	011 0000	\$30
1	011 0001	\$31
	• • •	
9	011 1001	\$39
blanc	010 0000	\$20
\$	010 0100	\$24
,	010 1100	\$2C







8 bits permettent de couvrir l'intervalle [-127,+127]



```
4 bits permettent de couvrir l'intervalle [ - ,+ ]
16 bits permettent de couvrir l'intervalle [- ,+ ]
```

### Codage des nombres signés



Codage en Complément à 1 : Un nombre X positif compris entre 0 et (2<sup>n</sup>-1) est représenté sur (n+1) bits par son code binaire naturel et un MSB égale à 0 :

$$X > 0 \rightarrow (X)_{C1} = X_2 \tag{S = 0}$$

Un nombre X négatif de valeur absolue comprise entre 0 et (2<sup>n</sup>-1) est représenté sur (n+1) bits par :

$$X < 0 \rightarrow (X)_{C1} = (2^{n+1}-1)_2 - |x|_2$$
 (S = 1)

$$n = 3$$

Pour obtenir le code complément à 1 d'un nombre négatif, il suffit d'inverser bit à bit le code de sa valeur absolue.

### Codage des nombres signés



#### Codage en Complément à 2 :

Un nombre positif compris entre 0 et (2<sup>n</sup>-1) est représenté (n+1) bits par son code binaire naturel et un MSB égale à 0 :

$$X > 0 \rightarrow (X)_{C2} = X_2 \tag{S = 0}$$

Un nombre X négatif de valeur absolue comprise entre 0 et (2<sup>n</sup>-1) est représenté sur (n+1) bits par :

$$X < 0 \rightarrow (X)_{C2} = (2^{n+1})_2 - |x|_2$$
 (S = 1)

$$n = 3$$

Pour obtenir le code complément à 2 d'un nombre négatif, il suffit d'ajouter 1 au code complément à 1 de ce nombre : C2 (pour x<0) = C1 (pour x<0) + 1

## Codage binaire des nombres fractionnaires C



décimal		binaire
0	0	0,0000 0000
0,5	2-1	0,1000 0000
0,25	2-2	0,0100 0000
0,125	2-3	0,0010 0000
0,0625	2-4	0,0001 0000
0,03125	<b>2</b> -5	0,0000 1000

#### **Codage binaire sur 4 bits:**

$$N_2 = S_{-1}2^{-1} + S_{-2}2^{-2} + S_{-3}2^{-3} + S_{-4}2^{-4}$$



Convertir en binaire :  $(0,75)_{10}$ 





$$(N)_{B(i,j)} = \underbrace{S(i-1).B^{i-1} + S(i-2).B^{i-2} + \dots + S(1).B^{1} + S(0).B^{0}}_{+} + \underbrace{S(-1).B^{-1} + \dots + S(-m).B^{-m}}_{+}$$

#### Partie entière

#### Partie fractionnaire

Exemple: 
$$N_2 = (0001101,0110)_{2(7,4)} = (13,375)_{10}$$



# Exemples : nombres à virgule fixe (mots de 3 bits)

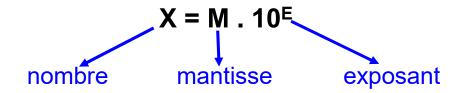
	k=3	k=2	k=1	k=0
	m=0	m=1	m=2	m=3
000	0	0	0	0
001	1	0,5	0,25	0,125
010	2	1	0,5	0,25
011	3	1,5	0,75	0,375
100	4	2	1	0,5
101	5	2,5	1,25	0,625
110	6	3	1,5	0,75
111	7	3,5	1,75	0,875

### Codage des nombres à virgule flottante



#### **Principe**:

Le codage utilisé est basé sur la notation scientifique des nombres :



Le nombre est dit à virgule flottante car il possède de nombreuses représentations qui dépendent de la position de la virgule :

$$0,3141592.10^{1} = 3,141592.10^{0} = 31,41592.10^{-1} = 314,1592.10^{-2} ... !$$

Forme normalisée (un unique digit non nul à la gauche de la virgule du nombre)

### 2. Algèbre de BOOLE



#### **Définitions**

Fonctions logiques combinatoires élémentaires Fonctions logiques combinatoires remarquables Expressions algébriques d'une fonction logique Simplification d'une fonction logique

#### **Définitions**



## Fonction logique



E<sub>i</sub> et S<sub>i</sub> : variables logiques

L'algèbre de BOOLE permet d'écrire les expressions des fonctions logiques à partir des variables logiques.

#### **Définitions**



#### Variable logique:

Une variable logique ne peut prendre que deux valeurs '0' ou '1' qui s'excluent mutuellement.

L'ensemble {0;1} est le domaine de définition des variables logiques.

$$x = 0 <=> x \ne 1$$

$$x = 1 <=> x \neq 0$$

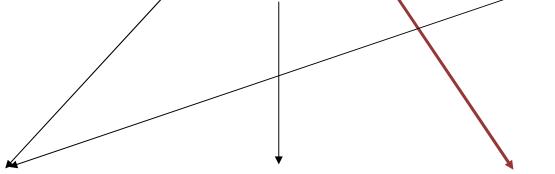
#### Fonction logique:

Une fonction logique de n variables logiques,  $F(x_1, x_2, ..., x_n)$  est une fonction qui ne prend que deux valeurs '0' ou '1'.

Cette valeur est obtenue par combinaison des valeurs des variables.



	•			
X	F1(x)	F2(x)	F3(x)	F4(x)
0	0	0	1	1
1	0	1	0	1

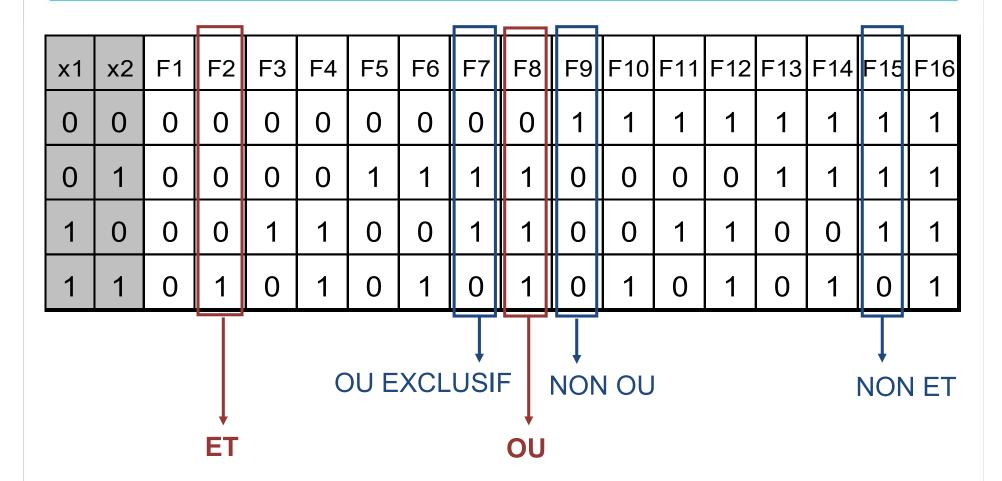


Fonctions constantes ...

Fonction identité OUI Fonction négation (INVERSION)

NON

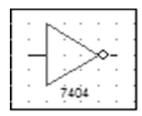






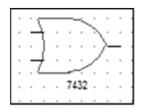
**INVERSION**: opérateur NON (NOT)

$$X \rightarrow X$$



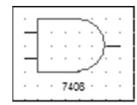
**SOMME LOGIQUE**: opérateur OU (OR) (noté '+')

$$(x,y) \rightarrow x + y$$



PRODUIT LOGIQUE: opérateur ET (AND) (noté '.')

$$(x,y) \rightarrow x.y$$





#### **Propriétés**

Involution : x = x

Idempotence : X.X = X x + x = x

Commutativité : x.y = y.x x + y = y + x

Associativité : (x.y).z = x.(y.z) = x.y.z (x + y) + z = x + (y + z) = x + y + z

Complémentarité : x.x = 0 x + x = 1

Eléments neutres : x.1 = x x + 0 = x

Eléments absorbants : x.0 = 0 x + 1 = 1

Distributivité : x.(y+z) = x.y + x.z x + (y.z) = (x + y).(x + z)



Démontrer les propriétés ci-dessous :



Absorbtion : x + (x.y) = x x.(x + y) = x

Simplification: (x + y).y = x.y (x.y) + y = x + y

Consensus :  $(x \cdot y) + (x \cdot z) + (y \cdot z) = (x \cdot y) + (x \cdot z)$ 

## Fonctions logiques combinatoires remarquables

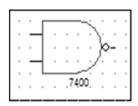


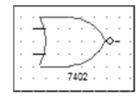
#### NON ET (NAND):

$$(x,y) \rightarrow \overline{x \cdot y}$$

#### NON OU (NOR):

$$(x,y) \rightarrow \overline{x+y}$$

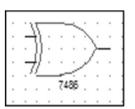




Opérateurs complets

#### **OU EXCLUSIF (XOR):**

$$(x,y) \rightarrow x \oplus y$$



## Fonctions logiques combinatoires remarquables



### **Propriétés**

#### NON ET / NON OU

Commutativité

Pas d'associativité!

#### **OU EXCLUSIF**

Commutativité

Associativité

Elément neutre '0'

## Fonctions logiques combinatoires remarquables



#### Théorème de DE MORGAN

$$\overline{x.y.z....} = \overline{x} + \overline{y} + \overline{z}....$$



$$\leftrightarrow \boxed{x + y + z + \dots = x.y.z.\dots}$$



$$F(x,y,z,\cdot,+) = F(x,y,z,+,\cdot)$$



Développer l'expression :

$$(a \cdot c) + b \cdot c$$

## Fonctions logiques combinatoires remarquables



#### **NON ET : opérateur complet**

NON: 
$$X = X \cdot X$$

ET: 
$$x \cdot y = (\overline{x \cdot y}) \cdot (\overline{x \cdot y})$$

OU: 
$$x + y = (\overline{x \cdot x}) \cdot (\overline{y \cdot y})$$



Démontrer que NOR est un opérateur complet

## Expressions algébriques d'une fonction logique



#### **Applications du OU EXCLUSIF**

x	у	F(x,y)
0	0	0
0	1	1
1	0	1
1	1	0



Inverseur programmable :  $y = 1 \rightarrow x \oplus y = x$ 

Test d'égalité :  $y = x \rightarrow x \oplus y = 0$ 

## Expressions algébriques d'une fonction logique



а	b	С	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$$F_1(a,b,c) = (\overline{a} \cdot \overline{b} \cdot c) + (\overline{a} \cdot b \cdot c) + (\overline{a} \cdot \overline{b} \cdot \overline{c}) + (\overline{a} \cdot \overline{b} \cdot c) + (\overline{a} \cdot b \cdot c)$$

$$F_2(a,b,c) = (a+b+c)\cdot(a+\overline{b}+\overline{c})\cdot(\overline{a}+\overline{b}+c)$$

## Expressions algébriques d'une fonction logique



Donner les expressions algébriques de x⊕y :



 $1^{\text{ère}}$  forme :  $F_1$  =  $2^{\text{ème}}$  forme :  $F_2$  =

## Simplification d'une fonction logique



Tableau de KARNAUGH à 2 entrées :

F(a,b)

x\y	0	1
0		
1		

Tableau de KARNAUGH à 4 entrées :

F(a,b,c,d)

ab\cd	00	01	11	10
00				
01				
11		0	0	
10		1001-	→1011	

Une seule variable change d'état en passant d'une case à l'autre!

### Simplification d'une fonction logique



$$F(a,b,c,d) = \overline{a \cdot b \cdot c \cdot d} + \overline{a \cdot b \cdot c \cdot d}$$

ab\cd	00	01	11	10
00				
01				
11				
10				

- Dresser un tableau de Karnaugh pour les variables suivant l'ordre du code Gray.
- 2. Remplir le tableau exprimant la fonction (placer les 1).
- Procéder aux regroupements des termes adjacents (en puissance de 2).
- 4. Exprimer chacun des termes.

# Simplification d'une fonction logique



$$F(a,b,c,d) = \overline{a \cdot b \cdot c \cdot d} + \overline{a \cdot b \cdot c \cdot d}$$

ab\cd	00	01	11	10
00	\ <u></u>	1		
01		1		
11			1	1
10			1	1

$$F(a,b,c,d) = a \cdot c + a \cdot \overline{b} \cdot c + a \cdot \overline{c} \cdot d + a \cdot \overline{b} \cdot d$$

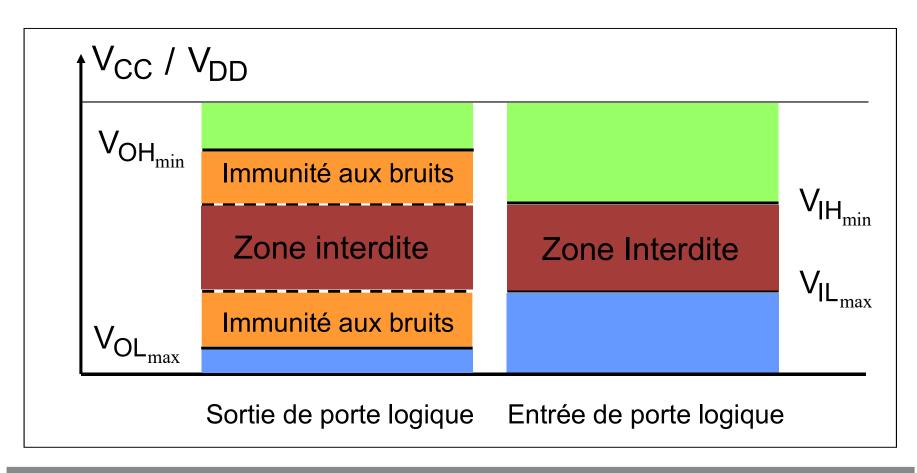




Familles logiques
Alimentation
Caractéristiques d'entrée / sortie
Exemple de documentation technique
Logique 3 états

# Réalisation d'une fonction logique

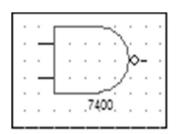




# Familles logiques



#### Les familles : LS, F, 4000, HC, HCT, AC, ACT, LV, ...



74LS00

74HC00

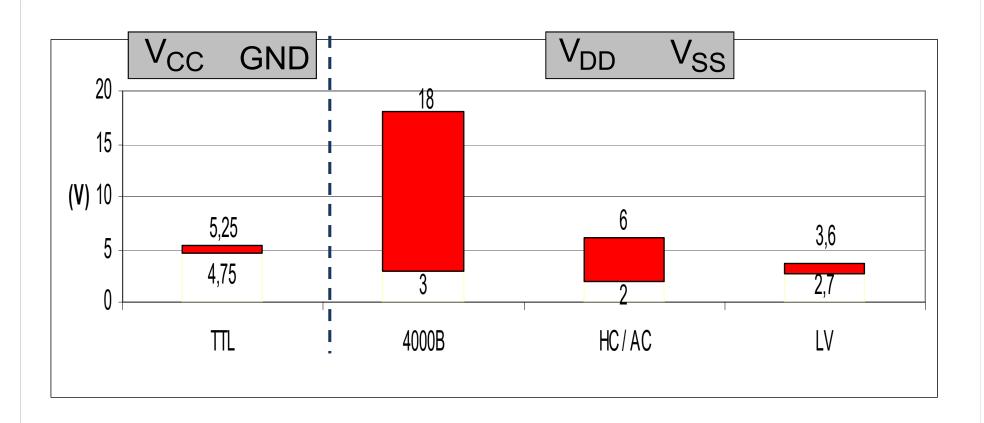
etc ...

	TTL	CMOS
Vitesse de fonctionnement	+	_
Immunité au bruit	-	+
Sortance	-	+
Intégration	_	+
Consommation	-	+

+ : avantage - : inconvénient

#### Alimentation





#### Alimentation



Type NAND

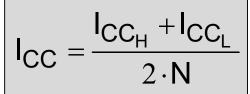
LS: 0.4 mA

F: 4 mA

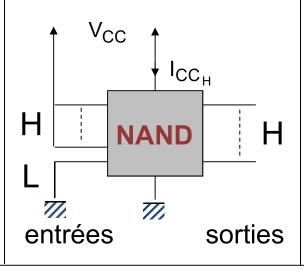
4000B : 1  $\mu$ A

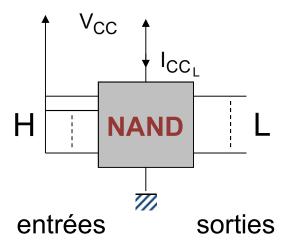
 $HC/AC:1\mu A$ 

LV : 20 μA



N : nombre de portes par boîtier

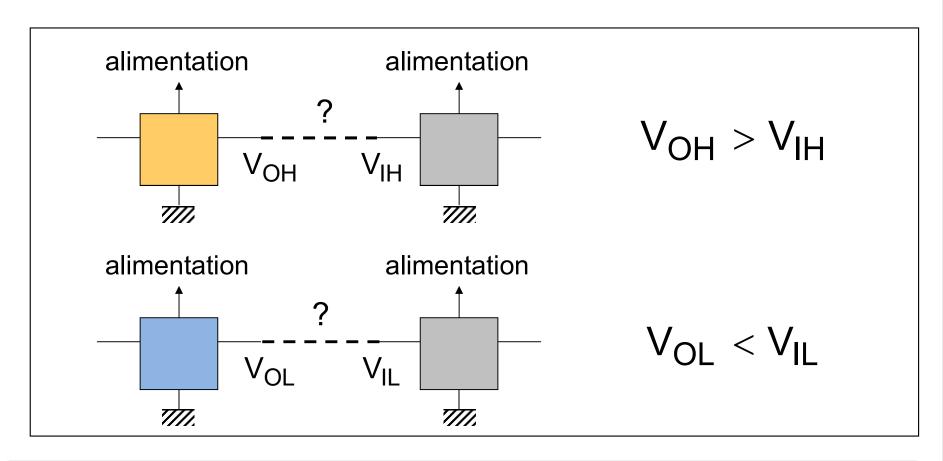






Etat logique Niveau logique	Potentiel en entrée	Potentiel en sortie		
H ou 1	V <sub>IH</sub>	V <sub>OH</sub>		
L ou 0	V <sub>IL</sub>	V <sub>OL</sub>		

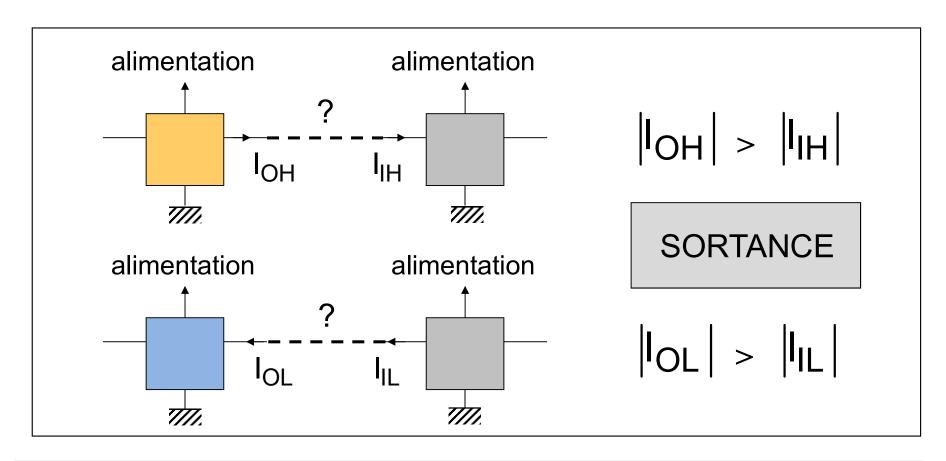






Etat logique Niveau logique	Courant en entrée	Courant en sortie
H ou 1	I <sub>IH</sub>	I <sub>OH</sub>
L ou 0	I <sub>IL</sub>	I <sub>OL</sub>

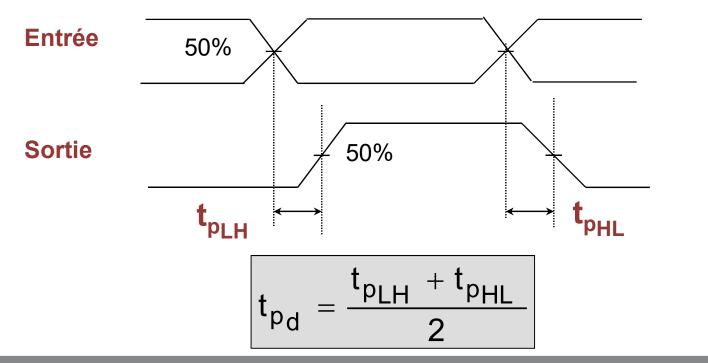




# Temps de propagation



La logique utilise 2 états électroniques d'un transistors : bloqué et saturé Un transistor ne passe pas instantanément d'un état à un autre. Dans un circuit la somme des temps de passage est regroupée sous le terme temps de propagation



#### Exemple de documentation technique

#### DM74LS00 Quad 2-Input NAND Gate

# NAND en technologie TTL



#### **General Description**

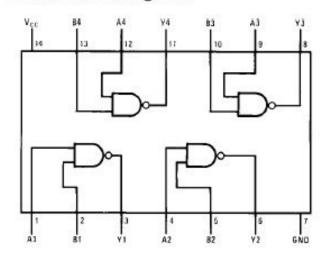
This device contains four independent gates each of which performs the logic NAND function.

#### Ordering Code:

Order Number	Package Number	Package Description
DM74LS00M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS00SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS00N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

#### Connection Diagram



#### **Function Table**

Inp	uts	Output
Α	В	Y
L	L	н
L	н	н
н	L	н
н	н	L

Source Texas Instrument

#### Exemple de documentation technique



#### Absolute Maximum Ratings(Note 1)

Supply Voltage 7V
Input Voltage 7V
Operating Free Air Temperature Range 0°C to +70°C
Storage Temperature Range -65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

#### Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V <sub>cc</sub>	Supply Voltage	4.75	5	5.25	V
V <sub>IH</sub>	HIGH Level Input Voltage	2			V
V <sub>IL</sub>	LOW Level Input Voltage			8.0	V
loh	HIGH Level Output Current			-0.4	mA
lor	LOW Level Output Current			8	mA
TA	Free Air Operating Temperature	0		70	7

Source Texas Instrument

## Exemple de documentation technique (3)

#### OLE SUPÉRIEURE : CHIMIE PHYSIQUE ÉLECTRONIQUE

#### **Electrical Characteristics**

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
VI	Input Clamp Voltage	V <sub>CC</sub> = Min, I <sub>I</sub> = -18 mA	55 63		-1.5	V
VOH	HIGH Level Output Voltage	V <sub>CC</sub> = Min, l <sub>OH</sub> = Max, V <sub>IL</sub> = Max	2.7	3.4		V
VaL	LOW Level Output Voltage	V <sub>CC</sub> = Min, l <sub>OL</sub> = Max, V <sub>IH</sub> = Min		0.35	0.5	v
	RE 5550	I <sub>OL</sub> = 4 mA, V <sub>CC</sub> = Min		0.25	0.4	
l <sub>l</sub>	Input Current @ Max Input Voltage	V <sub>CC</sub> = Max, V <sub>I</sub> = 7V			0.1	mΑ
liH .	HIGH Level Input Current	$V_{OC} = Max$ , $V_1 = 2.7V$			20	μΑ
I <sub>IL</sub>	LOW Level Input Current	V <sub>CC</sub> = Max, V <sub>I</sub> = 0.4V	55 63		-0.36	mA
los	Short Circuit Output Current	V <sub>CC</sub> = Max (Note 3)	-20		-100	mA
Госи	Supply Current with Outputs HIGH	V <sub>CC</sub> = Max		0.8	1.6	mA
loca.	Supply Current with Outputs LOW	V <sub>CC</sub> = Max		2.4	4.4	mA

Note 2: All typicals are at V<sub>CC</sub> = 5V, T<sub>A</sub> = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

#### **Switching Characteristics**

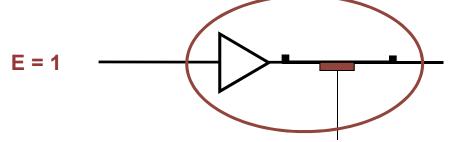
at V<sub>CC</sub> = 5V and T<sub>A</sub> = 25°C

		Ī	$R_L = 2 k\Omega$					
Symbol	Parameter			C <sub>2</sub> 15 pr		C <sub>L</sub> = 50 pF		
		_	Min	Max	Min	Max		
<sup>t</sup> PLH	Propagation Delay Time LOW-to-HIGH Level Output		3	10	4	15	ns	
<sup>t</sup> PHL	Propagation Delay Time HIGH-to-LOW Level Output		3	10	4	15	200	

Source Texas Instrument

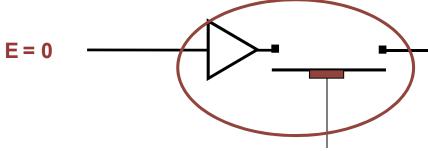


#### Niveau Haut (H) ou Bas (L)



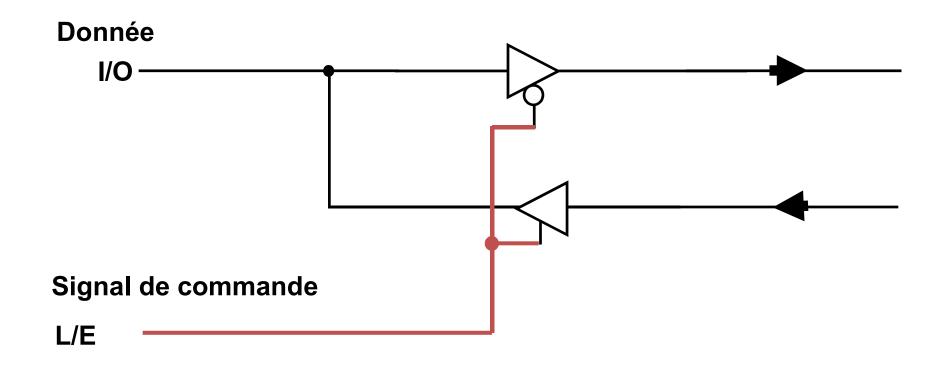


#### Haute impédance (Z)



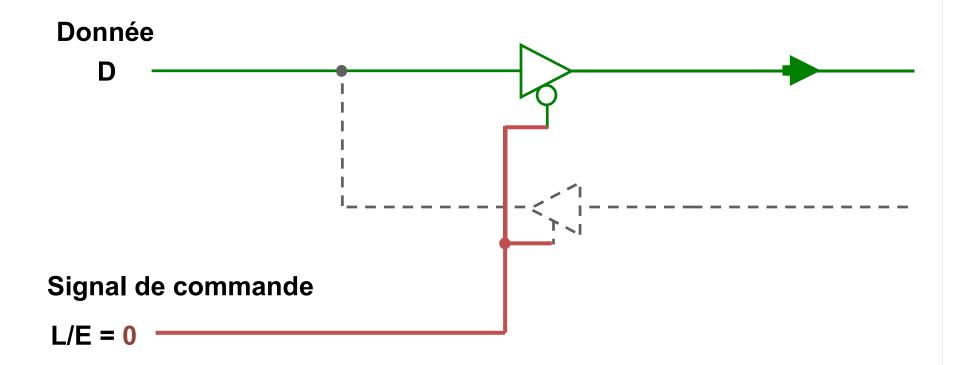


#### Ligne de données bidirectionnelle



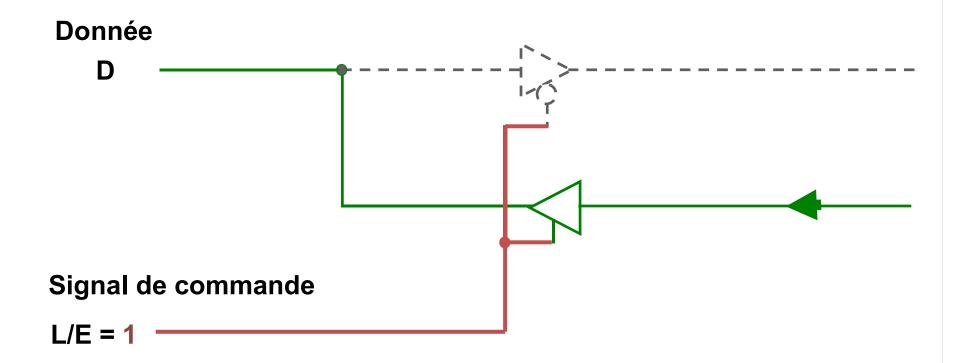


#### Ligne de données bidirectionnelle



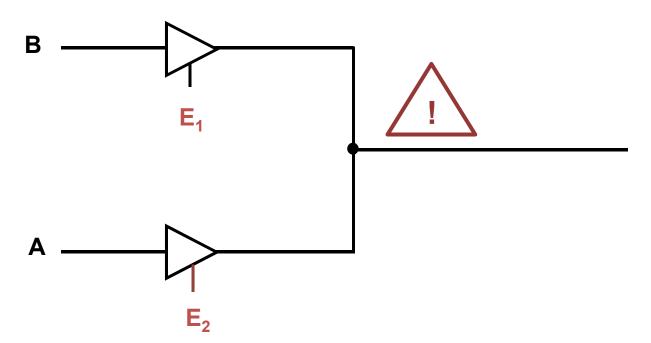


#### Ligne de données bidirectionnelle





#### Ligne de bus de données



#### E<sub>1</sub> et E<sub>2</sub> ne doivent <u>pas</u> être <u>actifs en même temps</u> :

$$E_1 = \overline{E_2}$$

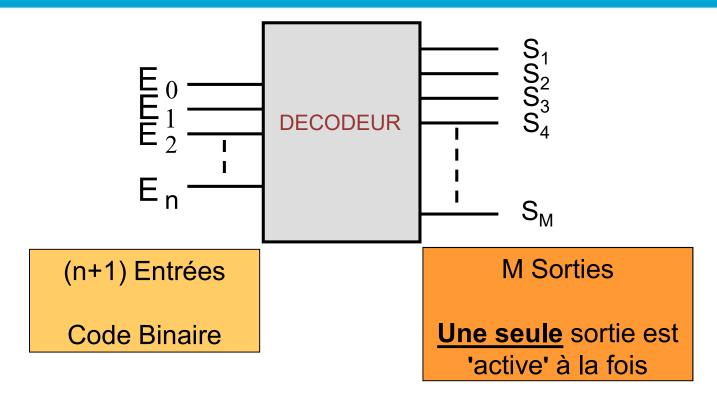
# 4. Fonctions complexes



# Décodeur Multiplexeur Additionneur binaire

# Décodeur





Les entrées E<sub>i</sub> permettent de 'décoder' les M sorties : en fonction de la combinaison des entrées, une seule des M sorties est activée.





Entrées : E<sub>0</sub>, E<sub>1</sub>

Sorties : S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>, S<sub>3</sub>

Sortie  $S_i$  active :  $S_i = 1$ 

Sortie  $S_i$  non active :  $S_j = 0$ 

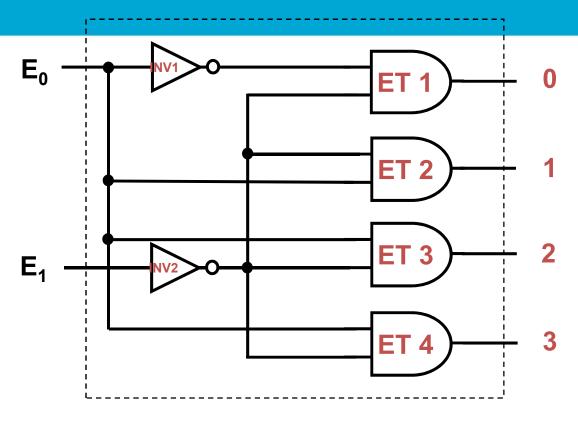
 Dresser la table de vérité des valeurs des sorties S<sub>j</sub> du décodeur en fonction des valeurs des entrées E<sub>i</sub>.

- 2. Donner les expressions des fonctions associées aux sorties en utilisant les fonctions logiques élémentaires.
- 3. Donner le schéma de réalisation électronique.



# EPE ÉCOLE SUPERIEURE DE CHIMIE PHYSIOUE ÉLECTRONIQUE DE LYON

#### Décodeur



Les entrées  $E_0$  et  $E_1$  permettent de « décoder » 4 sorties. En fonction de la combinaison des entrées, une seule des 4 sorties est active à la fois.

#### Décodeur



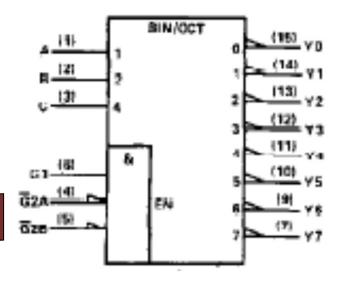
#### Décodeur type 138

- 1. 3 entrées (A, B, C) 1 sortie activée parmi 8 (Y0, Y1, ..., Y7)
  - entrées codées en Binaire
  - sortie active Yj = 0
- 2. 3 signaux d'autorisation d'utilisation du boîtier (G1,  $\overline{G}2A$ ,  $\overline{G}2B$ )

#### Décodeur



#### Décodeur type 138



Utilisation du boîtier autorisée pour :

$$EN = G1 \cdot \overline{G2A} \cdot \overline{G2B} = H$$

#### 'LS138, SN54138, SN74S138A FUNCTION TABLE

	13	#PUT	s	_						-	_	
ENA	BLE	s	FLFC	Τ			_ `	ידטכ	-01	-		
GI	Ğ2°	C	В	A	YO	Υ1	YZ	Y3	Y4	Y5	Y6	٧7
×	н	х	x	×	H	Н	н	н	Н	Н	Н	н
L	x	×	×	×	н	Н	н	н	н	н	н	н
н	L	L	L	L	(L)	H	н	н	н	н	н	н
н	L	L	L	н	н	L	Н	н	н	н	н	н
н	L	L	н	L	н	H	L	Н	н	н	н	н
H	L	t.	н	н	н	н	H	(L)	Н	н	н	н
н	L	н	L	L	н	н	н	H	(L)	Н	Н	н
н	L	Н	L	н	н	Н	н	н	Н	L	Н	н
н	Ł	н	H	L	н	н	н	н	н	Н	(L)	Н
H	L	н	н	н	н	н	н	н	н	н	Н	L

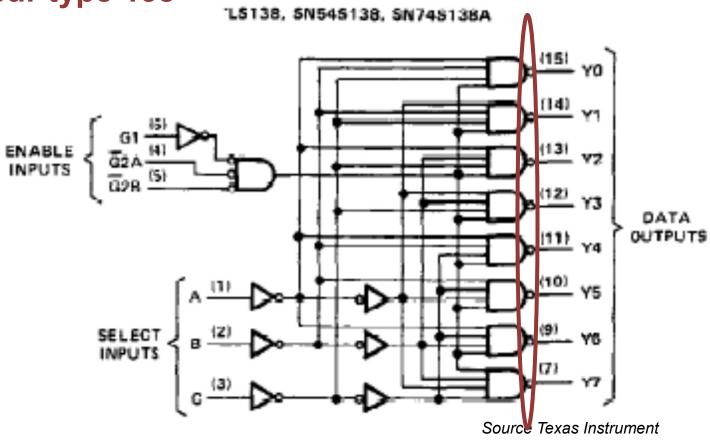
\*G2 = G2A + G28 H = high level, L = low level, X = irrelevant

Source Texas Instrument

# ECOLE SUPÉRIEURE DE CHIMIE PHYSIQUE ÉLECTRONIQUE DE LYON

#### Décodeur

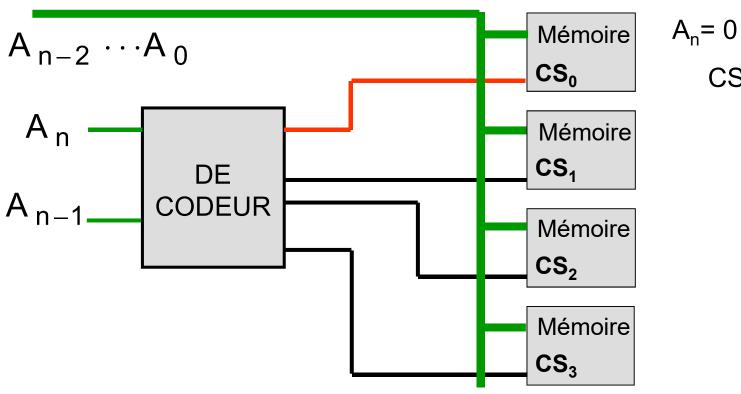
#### Décodeur type 138





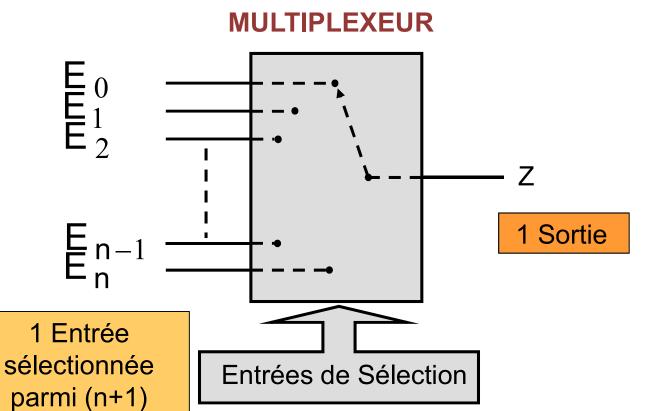
#### Décodeur

#### Adressage de la mémoire d'un micro-processeur



$$A_n = 0 \text{ à } A_{n-1} = 0$$
 $CS_0 \text{ actif}$ 





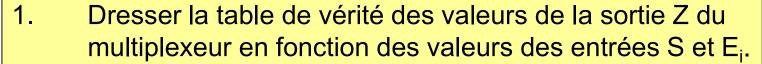
Les 2 informations  $E_0$  à  $E_n$  sont 'multiplexées' (dans le temps) sur une même sortie Z en fonction des valeurs des 'entrées de sélection'.

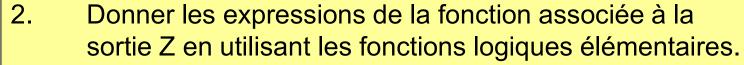


Entrées : E<sub>0</sub>, E<sub>1</sub>

Sortie: Z

Signal de sélection S

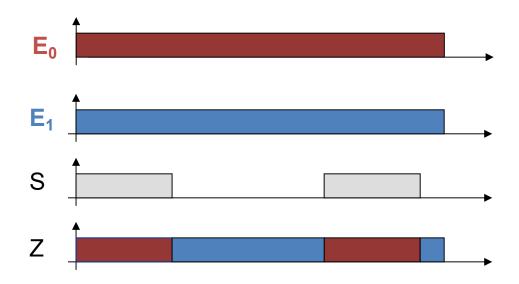




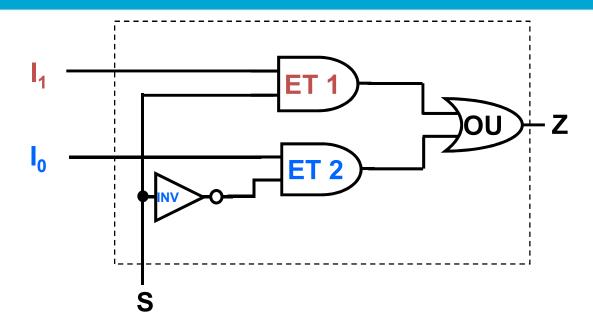
3. Donner le schéma de réalisation électronique.











Les 2 informations  $I_0$  et  $I_1$  sont « multiplexées » (dans le temps) sur une même ligne Z.

S est le signal de sélection de l'information. Ce signal doit également être disponible à l'arrivée pour démultiplexer.



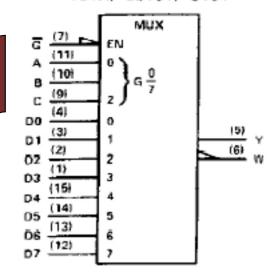
#### **Multiplexeur type 151**

- 1. 8 entrées (D0, D1, ..., D7) 2 sorties Y et Y
- 2. 3 signaux de sélection d'une des entrées (A, B, C)
- 3. 1 signal d'autorisation d'utilisation du boîtier  $\overline{G}$



#### **Multiplexeur type 151**

'151A, 'LS151, 'S151



Utilisation du boîtier autorisée pour :

$$EN = \overline{G} = L$$

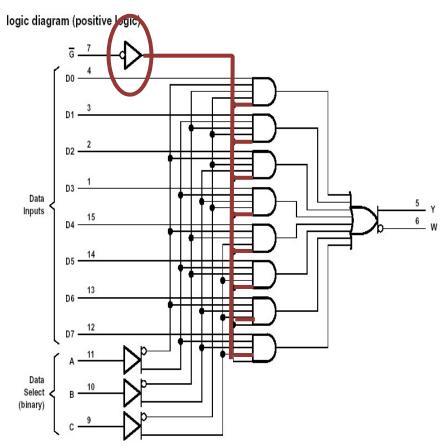
'151A, 'LS151, 'S151 FUNCTION TABLE

		OUT	PUTS		
S	SELECT		STROBE		, Ar
C	8	A	Ğ	. *	w
х	×	×	Н	L	н
L	L	L	L	DO	$\overline{00}$
L	L	н	L	DI	D1
L	н	L	L	D2	02
L	H	н	L	D3	<b>D</b> 3
н	£	L	L	D4	<u>D4</u>
н	L	Н	L L	D5	D5
Н	н	L	L	06	$\overline{D6}$
н	н	н	L	D7	D7

Source Texas Instrument



#### **Multiplexeur type 151**



Source Texas Instrument

### Additionneur binaire



					_
	$R_n$	$R_{n-1}\cdots$	$R_{i-1}$	$\cdots R_0$	
		$A_n \cdots$	Ai	$\cdots$ $A_1$ $\cdots$ $B_1$	$\cdots A_0$
$\langle + \rangle$		$B_n \cdots$	B <sub>i</sub>	B₁	$\cdots B_0$
	$R_n$	$S_n \cdots$	Si	$\cdots$ $s_1$	$\cdots$ S <sub>0</sub>

#### Additionneur binaire



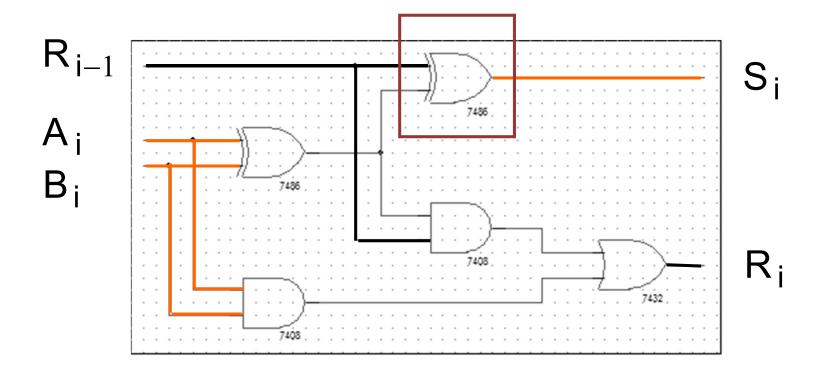
$R_{i-1}$	Ai	B <sub>i</sub>	R <sub>i</sub>	Si
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$S_i = A_i \oplus B_i \oplus R_{i-1}$$

$$R_{i} = A_{i} \cdot B_{i} + R_{i-1} \cdot (A_{i} \oplus B_{i})$$

### Additionneur binaire



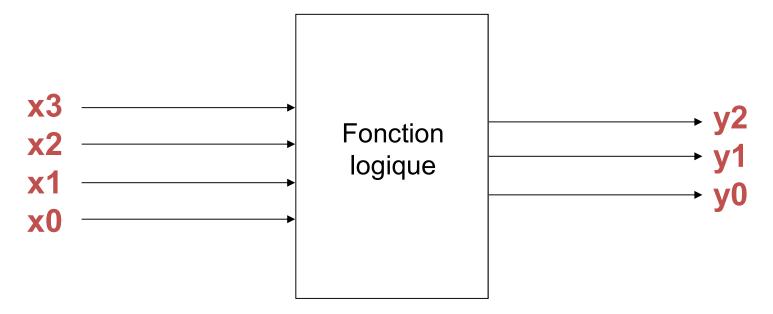


#### Réalisation d'une fonction logique



#### Objectif:

Assembler des circuits électroniques pour réaliser des fonctions logiques.



#### Réalisation d'une fonction logique



#### **Transcodage Binaire Gray sur 4 bits**

Binaire :  $B_3$ ,  $B_2$ ,  $B_1$ ,  $B_0$ Gray :  $G_3$ ,  $G_2$ ,  $G_1$ ,  $G_0$ 



- Dresser la table de vérité des valeurs des sorties G<sub>j</sub> du transcodeur en fonctions des valeurs de ses entrées B<sub>i</sub>
- 2. Donner les expressions des fonctions associées aux sorties en utilisant les fonctions logiques élémentaires

## Simplification d'une fonction logique



$$F(a,b,c,d) = \overline{a \cdot b \cdot c \cdot d} + \overline{a \cdot b \cdot c \cdot d}$$

ab\cd	00	01	11	10
00	1	1		
01		1		
11			1	1
10		1	1	1

- Dresser un tableau de Karnaugh pour les variables suivant l'ordre du code Gray,
- 2. Remplir le tableau exprimant la fonction (placer les 1),
- Procéder aux regroupements des termes adjacents (en puissance de 2),
- 4. Exprimer chacun des termes.

#### Simplification d'une fonction logique



$$F(a,b,c,d) = \overline{a \cdot b \cdot c \cdot d} + \overline{a \cdot b \cdot c \cdot d}$$

ab\cd	00	01	11	10
00	1	1		
01		1		
11			1	1
10			F	1

- Dresser un tableau de Karnaugh pour les variables suivant l'ordre du code Gray,
- 2. Remplir le tableau exprimant la fonction (placer les 1),
- 3. Procéder aux regroupements des termes adjacents (en puissance de 2),
- 4. Exprimer chacun des termes.

# Développement d'une fonction à l'aide de NAND



#### Principe:

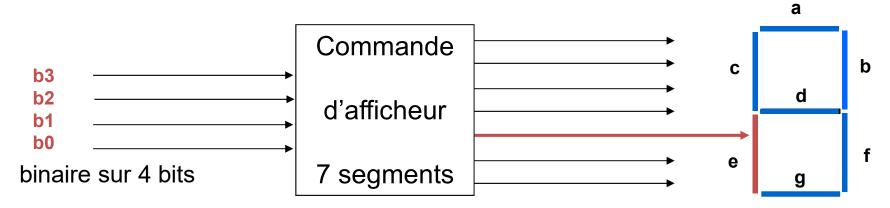
A l'aide du théorème de De Morgan toute fonction logique peut s'écrire à partir de fonctions NAND.

Somme de produits + 2 complémentations

$$a \cdot b \cdot c + a \cdot b \cdot d + e = a \cdot b \cdot c + a \cdot b \cdot d + e = a \cdot b \cdot c \cdot a \cdot b \cdot d \cdot e$$

#### Réalisation d'une commande d'afficheur





Le segment e est " allumé " pour : 0, 2, 6, 8

- Dresser la table de vérité de la fonction associée au segment " e "
- Donner l'expression de la fonction associée à " e " en utilisant les fonctions logiques élémentaires
- Donner le schéma de réalisation électronique

# Construire un A.L.U

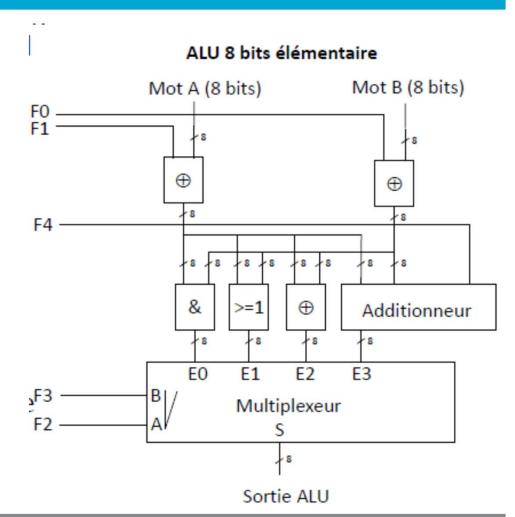


- Une UAL existe dans le cœur de tous les microprocesseurs.
- Elle permet de réaliser différents types d'opérations arithmétiques (+, -, \*, :) ou logiques (And, Or, ....) plus ou moins complexes.
- L'organisation, la structure et la complexité des UAL (structures RISC, CISC, ...), les opérations réalisées (Opérations MAC, barrel shifter, Sinus,...), l'optimisation des temps de calculs (Mips, ...) permettent de différencier les processeurs (microcontrôleur basique, Pentium, ARM, ...)

# Construire un A.L.U



- Explication du fonctionnement de l'ALU 8 bits:
- rôle du XOR,... 🛚
- Quelles combinaisons faut –il mettre sur les entrées F4,F3,F2,F1,F0
- pour réaliser: les fonctions logiques:
- PAANDBPAXORB
- PANAND B PANOR B
- 2 /A OR B •
- les fonctions arithmétiques:
- ? A + B ? A + B +1 ? A B ? B A ?
- Quelle est la fonction réalisée par l'UAL si F4,F3,F2,F1,F0 = 01011?



•