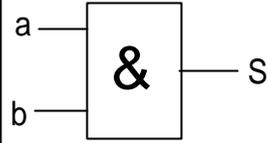
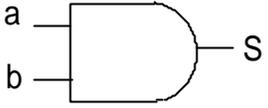
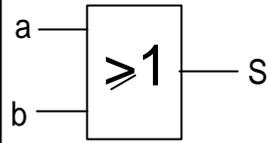
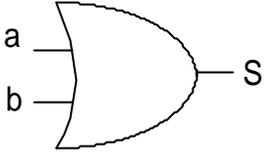
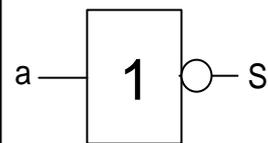
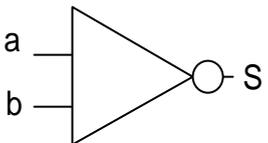
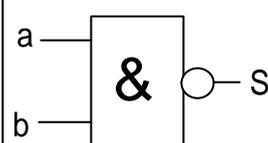
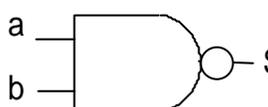
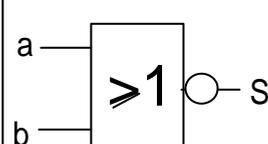
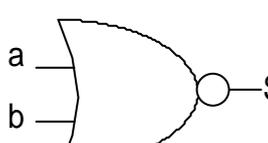
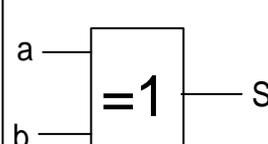
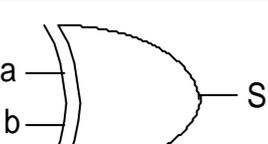


1. Rappels sur les fonctions logiques de base.

Chaque fonction logique est caractérisée par une ou des variables d'entrées qui ne peuvent prendre que deux états logiques 0 et 1, et elles ne possèdent qu'une seule sortie. Il faut surtout retenir les symboles, les équations et les tables de vérité. Ce sont ces trois représentations qui illustrent le mieux leur fonctionnement.

Fonction	Equation	Table de vérité	Normes CEI	Normes Ansi															
ET (AND)	$S = a.b$	<table border="1"> <tr><td>a</td><td>b</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	S	0	0	0	0	1	0	1	0	0	1	1	1		
a	b	S																	
0	0	0																	
0	1	0																	
1	0	0																	
1	1	1																	
OU (OR)	$S = a+b$	<table border="1"> <tr><td>a</td><td>b</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	a	b	S	0	0	0	0	1	1	1	0	1	1	1	1		
a	b	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	1																	
PAS (NOT)	$S = \bar{a}$	<table border="1"> <tr><td>a</td><td>S</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	a	S	0	1	1	0											
a	S																		
0	1																		
1	0																		
ET NON (NAND)	$S = \overline{a.b}$	<table border="1"> <tr><td>a</td><td>b</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	S	0	0	1	0	1	1	1	0	1	1	1	0		
a	b	S																	
0	0	1																	
0	1	1																	
1	0	1																	
1	1	0																	
OU NON (NOR)	$S = \overline{a+b}$	<table border="1"> <tr><td>a</td><td>b</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	S	0	0	1	0	1	0	1	0	0	1	1	0		
a	b	S																	
0	0	1																	
0	1	0																	
1	0	0																	
1	1	0																	
OU exclusif	$S = \bar{a}.b + a.\bar{b}$	<table border="1"> <tr><td>a</td><td>b</td><td>S</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	a	b	S	0	0	0	0	1	1	1	0	1	1	1	0		
a	b	S																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	

Par convention la fonction **ET** se représente avec un point " ." et la fonction **OU** avec un plus " + "

La fonction pas se représente avec un trait au dessus (dit barre) ou un slash /
 $S = a + b$ se dit : S égal a OU b $T = a . \bar{b}$ se dit T égal a ET b barre

Théorèmes de l'algèbre de Boole*Théorèmes à une variable*

$$x \cdot 0 = 0$$

$$x \cdot 1 = x$$

$$x \cdot x = x$$

$$x \cdot \overline{x} = 0$$

$$x + 0 = x$$

$$x + 1 = 1$$

$$x + x = x$$

$$x + \overline{x} = 1$$

Théorèmes à plusieurs variables

$$x + y = y + x$$

$$x \cdot y = y \cdot x$$

$$x + (y + z) = (x + y) + z = x + y + z$$

$$x(yz) = (xy)z = xyz$$

$$x(y + z) = xy + xz$$

$$(w + x)(y + z) = wy + xy + wz + xz$$

$$x + xy = x$$

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

Théorèmes de De Morgan

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

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

Simplification d'une expression logique

La réduction, pour une même expression, du nombre d'opérateurs et/ou du nombre de variables exprimées, conduit à une écriture simplifiée de cette expression.

Il existe un grand nombre de méthodes de simplification d'expression booléennes, parmi lesquelles on peut distinguer:

- Simplification algébrique

Exemple: $F = \overline{\overline{a}} \overline{\overline{b}} c + \overline{\overline{a}} b \overline{\overline{c}} + \overline{\overline{a}} b c + a b c$

Le monôme $\overline{\overline{a}} b c$ peut être regroupé avec chacun des trois autres monômes. On peut donc le rajouter deux fois puisque:

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

$$F = \overline{\overline{a}} b c + \overline{\overline{a}} b c + \overline{\overline{a}} b \overline{\overline{c}} + \overline{\overline{a}} b c + a b c + \overline{\overline{a}} b c$$

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

Dans la pratique, on constate qu'il n'est pas toujours simple de trouver les regroupements possibles et cette méthode s'avère assez hasardeuse au-delà de trois variables d'entrée. Aussi a-t-on cherché d'autres méthodes de simplification que l'on va étudier dans les chapitres suivants.

■ Simplification par tableau de Karnaugh

Le tableau de **Karnaugh**:

Une fonction logique qui a un nombre de variables d'entrée limité peut facilement être représentée dans un tableau de Karnaugh.

Pour n variables d'entrées, le tableau rectangulaire ou carré possède 2^n cases dans lesquelles sont notées les valeurs de la fonction logique.

Chaque case correspond à une combinaison des variables d'entrée.

Exemple:

tableau de vérité

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

tableau de Karnaugh

		a	
		0	1
b	0	1	0
b	1	0	1

(S)

		a	
		0	1
b	0	1	0
b	1	0	1

les 2 cases grisées correspondent à :
 $a = 0$

		a	
		0	1
b	0	1	0
b	1	0	1

les 2 cases grisées correspondent à :
 $a = 1$

		a	
		0	1
b	0	1	0
b	1	0	1

les 2 cases grisées correspondent à :
 $b = 0$

		a	
		0	1
b	0	1	0
b	1	0	1

les 2 cases grisées correspondent à :
 $b = 1$

		a	
		0	1
b	0	1	0
b	1	0	1

donc la case grisée ci-contre correspond à :
 $a = 0$ et $b = 0$
je met donc un 1 dans cette case si le tableau de vérité indique :
quand $a = 0$ et $b = 0$ alors $S = 1$

table de vérité

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

exercice : compléter le tableau de Karnaugh à partir de la table de vérité

		a	
		0	1
b	0		
b	1		

(S)

Le passage d'une case à une case adjacente ne doit se faire qu'avec le changement d'état d'une seule variable d'entrée.

table de vérité

	c	b	a	S
1ère L	0	0	0	1
2ème L	0	0	1	1
3ème L	0	1	0	1
4ème L	0	1	1	0
5ème L	1	0	0	0
6ème L	1	0	1	0
7ème L	1	1	0	1
8ème L	1	1	1	0

		ab			
		00	01	11	10
c	0				
	1				

la case grisée correspond à : $a = 1$ et $b = 1$ et $c = 0$ sur la table de vérité, on constate que dans ce cas là, $S = 0$ (4 ème ligne) donc on place un 0 dans cette case

		ab			
		00	01	11	10
c	0				
	1				

compléter le tableau ci-dessus

Pour la simplification, on recherche les cases adjacentes qui ont pour valeur 1 et on les regroupe, par multiple de 2, en paquets les plus gros possibles (par 2, 4, 8, 16 etc ...) La fonction simplifiée est l'union des paquets qui représente l'ensemble des valeurs 1. (on ne garde dans un regroupement que ce qui est commun à toutes les cases

Reprenons l'exemple de la fonction: $F = a \bar{b} c + \bar{a} b \bar{c} + \bar{a} b c + a b c$

		ab			
		00	01	11	10
c	0	0	1	0	0
	1	1	1	1	0

premier regroupement possible de 2 cases : les variables qui sont communes : \bar{a} et b

		ab			
		00	01	11	10
c	0	0	1	0	0
	1	1	1	1	0

deuxième regroupement possible de 2 cases : les variables qui sont communes : \bar{a} et c

		ab			
		00	01	11	10
c	0	0	1	0	0
	1	1	1	1	0

troisième regroupement possible de 2 cases : les variables qui sont communes : b et c

Donc la simplification de l'équation F est : $F = \bar{a} b + \bar{a} c + b c$

2. Circuits intégrés logiques

La matérialisation des fonctions logiques a d'abord été réalisée avec des composants discrets puis elle s'est transformée en intégrant plusieurs composants sur un seul circuit. Actuellement les circuits intégrés logiques sont étudiés en fonction de la technologie d'intégration maximale.

Les circuits intégrés se divisent en deux grandes catégories :

- les circuits intégrés linéaires comprenant les circuits électroniques d'amplification ;
- les circuits intégrés numériques qui fonctionnent en tout ou rien, c'est la base de la commutation.

2.1 le transistor en commutation

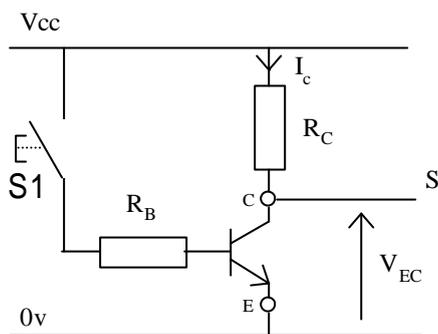


Table de vérité

S_1	S
0	1
1	0

$$S = \overline{S_1}$$

Lorsque S1 est ouvert, il n'y a pas de courant de base, le transistor est bloqué, la résistance entre émetteur et collecteur est très grande devant la résistance RC. La sortie S est sensiblement au potentiel Vcc par exemple +5v. Lorsque S1 est fermé, un courant de base provoque la saturation du transistor, seule la résistance RC limite le courant, le potentiel de S est à 0v

2.2 Notions de famille technologique de circuits intégrés



Une famille technologique de circuits intégrés est un ensemble d'opérateurs réalisés à partir de composants regroupés dans une structure appelée **circuit intégré**.

Les circuits intégrés d'une même famille ont comme point commun la technologie de fabrication employée. Les circuits intégrés d'une famille peuvent être interconnectés sans intermédiaire de manière à mettre à la disposition de l'utilisateur un jeu de construction aussi complet que possible pour réaliser des structures plus complexes.

Un circuit intégré désigne un bloc constitué par un monocristal de silicium de quelques millimètres carrés en forme de parallélépipède rectangle aplati, à l'intérieur duquel se trouve inscrit en nombre variable des composants électroniques élémentaires (transistors, diodes, résistances et, plus rarement, des condensateurs).

Ces différents composants sont associés pour former des opérateurs électroniques intégrés.

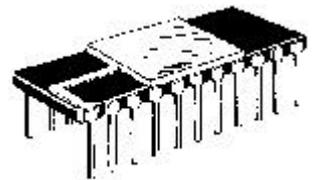
Le monocristal encore appelé puce à cause de sa très petite taille, est englobé dans un boîtier plastique ou céramique comportant un certain nombre de broches en liaisons avec les entrées et sorties des opérateurs intégrés dans la puce.

Au sein d'une même famille technologique, un élément se distingue par :

- la fonction logique qu'il réalise ;
- le nombre d'opérateurs contenus dans un même boîtier.

Il existe deux types fondamentaux de circuits intégrés :

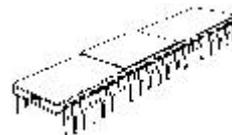
- les circuits intégrés linéaires (amplificateurs, régulateurs, comparateurs, etc.) ;
- les circuits intégrés logiques (opérateurs logiques élémentaires, opérateurs logiques complexes, opérateurs microprogrammés).



2.2.1 Classes d'intégration des circuits intégrés

Dans l'ordre chronologique, on distingue :

- les microcircuits SSI (Single Size Integration). Ce sont des circuits intégrés à faible intégration donc peut coûteux, environ 100 transistors par cm² ;
- les circuits intégrés MSI (Medium Size Integration). Ces circuits sont à intégration moyenne, environ 1000 transistor par cm² ;
- les circuits LSI (Large Size Integration). Circuits à très haute intégration, environ 10000 à 100000 transistors par cm² ;
- les circuits VLSI (Very Large Size Integration). Ces circuits, de dernière génération, regroupent sur une puce de 1 cm², de 100000 à 1000000 de transistors !



2.3 Famille technologique TTL (Tout Transistor Logique)

Cette famille technologique est conçue essentiellement à partir de transistors bipolaires et de résistances, ces deux composants étant réalisés à partir de jonction PN.

Valeurs caractéristiques de la famille TTL

Tension d'alimentation : elle est fixe et égale à +5v avec une tolérance de 5%

Niveaux de tension en entrée : $V_{ILmax} = 0,8v$; tension d'entrée à l'état BAS maximal ;
 $V_{ILmin} = 0v$; tension d'entrée à l'état BAS minimal ;
 $V_{IHmax} = 5v$; tension d'entrée à l'état HAUT maximal ;
 $V_{IHmin} = 2,4v$; tension d'entrée à l'état HAUT minimal.

Niveaux de tension en sortie : $V_{OLmax} = 0,4v$; tension de sortie à l'état BAS maximal ;
 $V_{OHmin} = 2,4v$; tension de sortie à l'état HAUT minimal ;

Sortance(1) : la sortance est de 10 pour la famille TTL, elle peut être de 20 pour certains dérivés de cette famille.

Courants sur une entrée : $I_{IHmax} = + 40 A$; courant demandé par une entrée à l'état HAUT (courant entrant).
 $I_{ILmax} = - 1,6 mA$; courant fourni par l'entrée à l'état BAS (courant sortant).

Courant sur une sortie : $I_{OHmax} = + 400 A$; courant fourni par une sortie à l'état HAUT (courant sortant).
 $I_{OLmax} = - 16 mA$; courant maximal absorbé pour une sortie à l'état BAS (courant entrant).

(1) La sortance d'une porte logique correspond au nombre maximal d'entrées unitaires quelle peut alimenter simultanément, sans que les niveaux de sortie sortent des spécifications.

Dérivé de la famille TTL : la famille TTL-LS

Les évolutions technologiques sont orientés vers l'amélioration de la puissance consommée et de la sortance, tout en conservant un temps de propagation. C'est le cas de la famille TTL-LS.

2.3 Famille technologique C-MOS (transistor MOS complémentaire)

Cette famille technologique permet d'atteindre un double objectif par rapport à la famille TTL. Le premier est de minimiser l'encombrement des opérateurs élémentaires intégrés. Cela est réalisé par l'emploi de transistors MOS (Métal Oxide Semiconductor) et par la suppression des résistances non utiles pour ce type de transistor. C'est la génération des circuits LSI et VLSI. Le second est de diminuer la dissipation de chaleur en vue d'augmenter l'intégration. A ce titre, les transistors MOS Complémentaires (C-MOS) offrent des performances intéressantes.

Valeurs caractéristiques de la famille C-MOS.

Tension d'alimentation : le fonctionnement des opérateurs C-MOS est garanti pour une plage de tension d'alimentation variant de + 3 v à + 15 v.

Consommation : Une faible consommation fait de la C-MOS la famille idéale pour les systèmes autonomes en énergie (alimentation par piles ou accumulateurs). Les courants en entrée ou en sortie d'une fonction logique sont très faibles (quelques μA).

Temps de propagation Le temps de propagation est directement lié à la tension d'alimentation. Pour les applications à faible vitesse, la tension d'alimentation de + 5 v est optimale. (la puissance dissipée étant la plus faible).

Niveaux de tensions en entrée :
 $V_{ILmax} = 45\%$ de V_{cc} ; tension d'entrée à l'état BAS maximal ;
 $V_{ILmin} = 0v$; tension d'entrée à l'état BAS minimal ;
 $V_{IHmax} = V_{cc}$; tension d'entrée à l'état HAUT maximal ;
 $V_{IHmin} = 55\%$ de V_{cc} ; tension d'entrée à l'état HAUT minimal.

Niveaux de tensions en sortie :
 $V_{OLmax} = 0v$; tension de sortie à l'état BAS ;
 $V_{OHmin} = V_{cc}$; tension de sortie à l'état HAUT minimal.

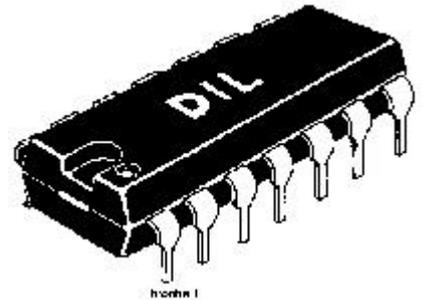
Sortance La sortance est pratiquement illimitée si l'on considère les courants demandés en entrée d'une porte (de l'ordre du μA). Cependant on limite la sortance à 50 pour des considérations sur les temps de propagation.

Interfaçage C-MOS / TTL : L'interfaçage avec la famille TTL est réalisé à l'aide d'opérateurs spécifiques lorsque la famille C-MOS fonctionne avec des tensions d'alimentations supérieures à + 5 v.

Dérivés de la famille C-MOS : la famille HC-MOS

L'évolution de cette famille est orientée vers une compatibilité totale avec la famille TTL.

La technologie HC-MOS (MOS Complémentaires à grande vitesse) possède des temps de propagation similaires à ceux de la famille TTL-LS, tout en conservant les performances de la famille CMOS en ce qui concerne la puissance consommée et la sortance.

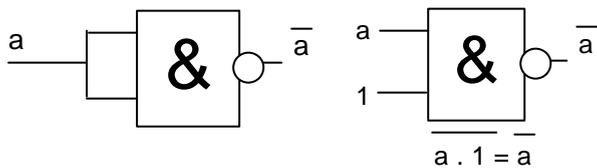


2.4 Caractéristiques comparées des différentes familles TTL et C-MOS.

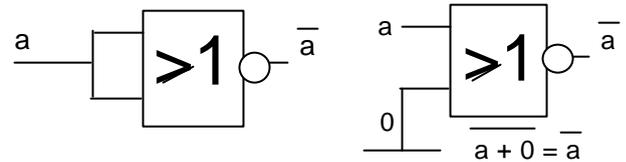
Famille	Temps de propagation t_{pd} (ns)	Puissance consommée P_C (mW)	Facteur de qualité Q_f (pJ)	Fréquence maximale (MHz)	Tension alimentation V_n (v)	Sortance
TTL	10	10	100	30	+ 5v à $\pm 5\%$	10
TTL-LS	10	2	20	30	+ 5v à $\pm 5\%$	20
C-MOS	40	0,3	12	10	+ 3v à + 15v	50 mini
HC-MOS	10	0,3	3	30	+ 3v à + 15v	50 mini

3. L'association des portes : universalité des opérateurs NAND et NOR.

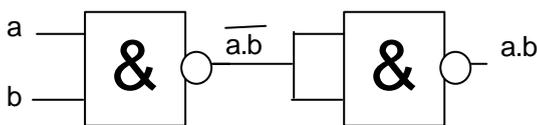
Comment faire une fonction NON avec une porte ET-NON (NAND) :



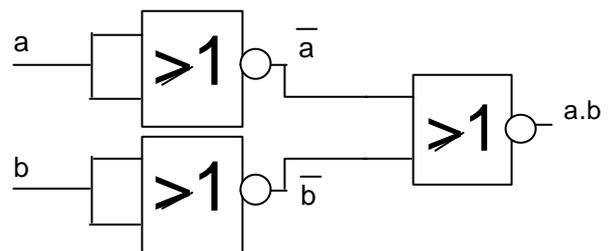
Comment faire une fonction NON avec une porte OU-NON (NOR) :



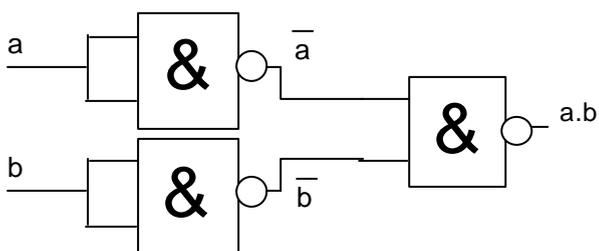
Comment faire une fonction ET avec des portes ET-NON (NAND) :



Comment faire une fonction ET avec des portes OU-NON (NOR) :



Comment faire une fonction OU avec des portes ET-NON (NAND) :



Comment faire une fonction OU avec des portes OU-NON (NOR) :

