

ULSI101.1 : Eléments de logique binaire

Le module ULSI101 vise à présenter aux étudiants nouveaux bacheliers une image la plus attrayante possible (la moins rébarbative ?) de l'enseignement de l'EEA, à travers deux sous-modules : ULSI101.1 et ULSI101.2. L'enseignement se fera sous forme d'enseignement intégré (groupe de 40 au maximum, Cours/TD intégré, Projet d'Initiative Personnelle) avec une série de conférences de présentation des disciplines et métiers de l'EEA. Les deux modules d'enseignement intégré présentent de façon sommaire les bases de l'électronique numérique (ULSI101.1) et des circuits passifs à courant continu (ULSI101.2) à travers la mise en oeuvre d'objets technologiques simples. Les pré-requis de mathématiques et physique sont minimum car l'objectif est d'intéresser les étudiants à nos disciplines sans qu'il soient *a priori* rebutés par les difficultés théoriques. Bien sûr, pour ne pas les tromper, il leur sera expliqué que ces bases théoriques devront être nécessairement acquises dans la suite de leur cursus pour approfondir une compréhension fine de l'électronique impérative pour l'exercice leur futur métier d'ingénieur.

Ce premier sous-module ULSI101.1 sera l'occasion d'introduire les premiers éléments de l'algèbre de Boole qui permet de traiter les signaux binaires ainsi que l'écriture des fonctions logiques et leur simplification.

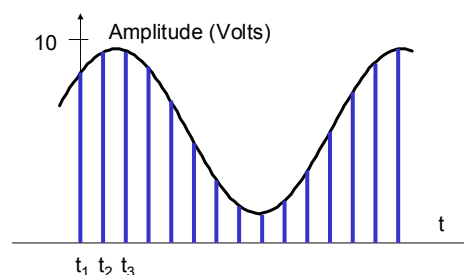
Il permettra de comprendre comment sont conçues les portes logiques utilisées dans le domaine de l'électronique numérique. A cette occasion une présentation sommaire de la constitution des transistors utilisés en technologie CMOS sera proposée ainsi qu'un rapide survol des techniques de fabrication des circuits intégrés. Il est bien entendu que toutes ces notions fondamentales de la microélectronique seront approfondies pour les étudiants qui suivront la filière EEA du cursus universitaire

1 Introduction

On utilise largement les signaux binaires à deux niveaux distincts dans:

- les ordinateurs (informatique),
- les systèmes de commande et de contrôle (automatique, robotique, etc.)
- les systèmes de télécommunications (télévision numérique),
- le multimédia (CD audio, CD numérique, DVD, etc.)
- les systèmes de mesures : microcapteurs, microsystèmes, etc.

Remarque : Bien sûr, les signaux physiques sont essentiellement de type analogique : ondes sonores, ondes lumineuses, signaux électriques issus de capteurs, etc. Pour traiter de façon numérique les signaux analogiques il suffit de les échantillonner (c'est-à-dire, prendre la valeur analogique du signal à des instants discrets donnés : t_1 , t_2 , t_3 , ...) et de convertir ces échantillons analogiques en valeur binaire (conversion analogique/numérique).



instants	amplitude	10 x amplitude	Codage DCB
t_1	9,2	92	1001 0010
t_2	9,6	96	1001 0110
t_3	9,7	97	1001 0111
....

L'exemple ci-dessus illustre le principe de conversion pour un signal sinusoïdal oscillant autour d'une certaine valeur moyenne. Ce principe s'étend bien sûr à tous les types de signaux analogiques quelle que soit leur complexité.

Les éléments de logique binaire dispensés au cours de cet enseignement sont une simple **introduction à l'électronique numérique combinatoire** (c'est-à-dire non séquentielle). Pour illustrer les potentialités de cette discipline, on prendra l'exemple de la réalisation d'un objet technologique simple : un **décodeur pour affichage 7 segments** (figure 1.a).

Le problème qui se pose est le suivant :

On dispose d'un chiffre décimal codé sur quatre bits $D_3 D_2 D_1 D_0$ en code DCB (Décimal Codé Binaire) selon la table de la figure 1.b. :

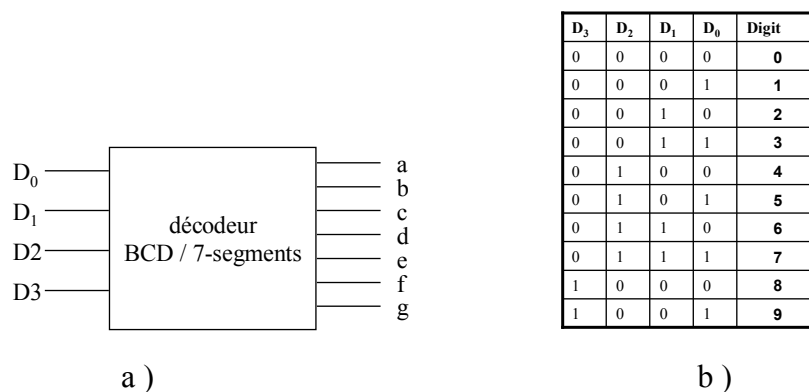


Figure 1 : décodeur BCD/7segments : a) Représentation symbolique, b) Table de vérité.

A partir de ce code numérique en entrée, on doit délivrer les 7 bits (a,b,c,d,e,f,g) qui permettront de représenter le chiffre sur un afficheur "7-segments". Dans ce type d'afficheur, les chiffres sont constitués de diodes électroluminescentes (ou de cristaux liquides) organisées en segments qui, lorsqu'elles sont allumées, dessinent une image du chiffre à représenter (figure 2).

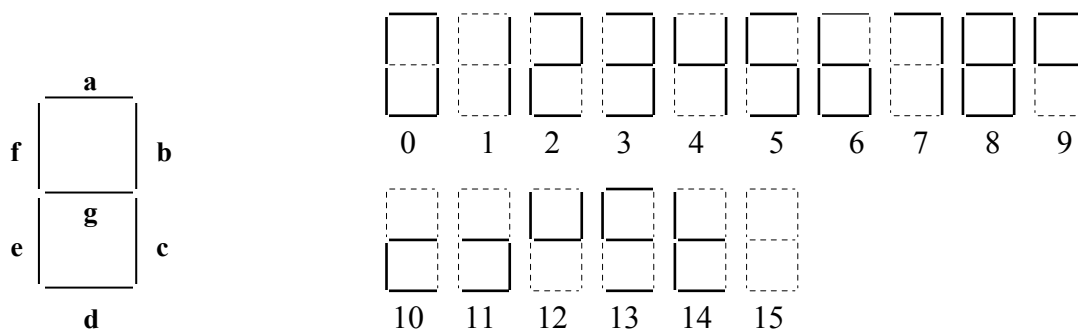


Figure 2 : Afficheur 7 segments : fonctionnement

Pour réaliser ce décodeur, il nous faut d'abord comprendre en quoi consiste l'algèbre binaire et quelles sont les règles qui le régissent (**Algèbre de Boole**). Il nous faudra ensuite comprendre comment sont conçues les **portes logiques** qui nous permettront de réaliser une implantation matérielle de ce décodeur (**technologie CMOS**). Enfin, à travers cet exemple concret nous serons amenés à mettre en œuvre les techniques de **minimisation** des fonctions logiques (**Tables de Karnaugh**) qui permettent d'optimiser la conception de notre décodeur en économisant de plus possible de nombre de portes logiques à utiliser. A la fin de ce cycle d'enseignement ULSI101.1, chaque étudiant remettra un devoir présentant son implantation minimisée du décodeur BCD/7 segments.

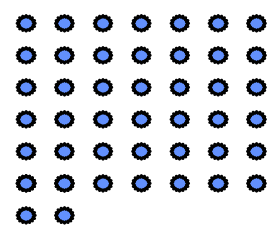
2 Système de numération

Il s'agit juste d'un canevas rapide pour donner le "feeling" aux étudiants, sans toute la rigueur mathématique qui serait nécessaire.

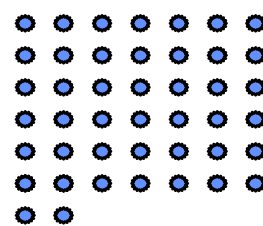
2.1 Exercice d'évaluation des connaissances

Commencer directement par un exercice pour tester les connaissances des étudiants sur les systèmes de numération. On donne une collection d'objets et on demande de les dénombrer dans un système de numération :

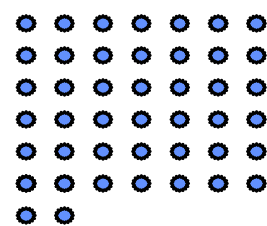
- à base 10
- à base 5
- à base 16
- à base 2.



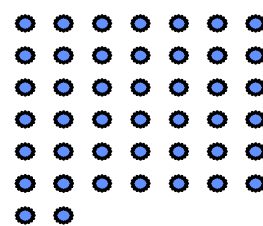
En base 10 :



En base 5 :



En base 16 :

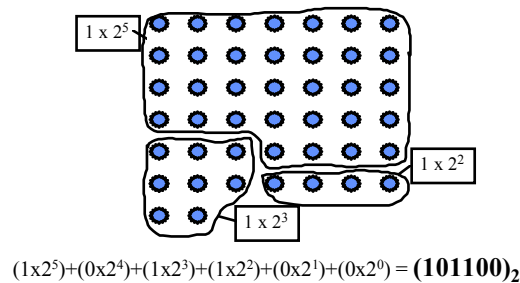
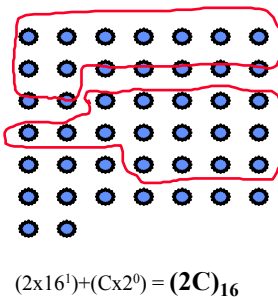
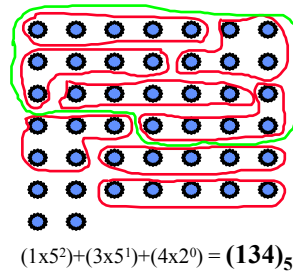
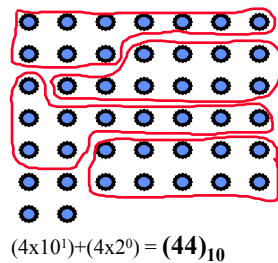


En base 2 :

Les idées à faire passer à la correction de l'exercice :

- système à base "n" : on fait des groupes de $(n)=n^1$, puis de $(n.n)=n^2$, puis de $(n.n.n)=n^3$, etc.
- on dispose de n symboles pour coder le nombre : de 0 à n-1 (0 à 9 en décimal, par exemple)
- on écrit le nombre en alignant les symboles de droite à gauche
 - nombre d'unités, non "regroupables"
 - nombre de groupes de n objets
 - nombre de groupes de n^2 objets

- etc.
- pourquoi le système décimal ? On a 10 doigts
- pour les systèmes de base >10 , on est amené à ajouter d'autres symboles. Pour l'hexadécimal, par exemple : A, B, C, D, E, F.
- Le système binaire permet de n'utiliser que 2 digits (symboles) pour coder n'importe quel nombre (même si la chaîne de digits peut être longue).



2.2 Techniques de conversion décimal/binaire

Le principe : On cherche à écrire le nombre sous la forme : $(bx2^n) + \dots + (bx2^4) + (bx2^3) + (bx2^2) + (bx2^1) + (bx2^0)$, avec $b=0$ ou 1 .

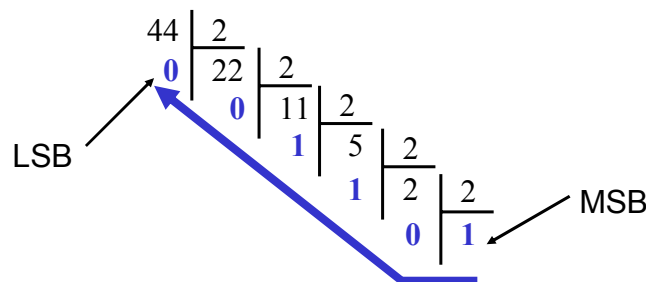
Exemple $(44)_{10}$

$$\begin{aligned} (2^6=64) > 44 > (2^5=32) &\rightarrow 44 = 2^5 + 12 \\ (2^4=16) > 12 > (2^3=8) &\rightarrow 44 = 2^5 + 2^3 + 4 \\ 4 = 2^2 &\rightarrow 44 = 2^5 + 2^3 + 2^2 \end{aligned}$$

D'où l'écriture complète en puissance de 2 :

$$(44)_{10} = (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = (101100)_2$$

La technique des divisions par 2 successives permet de systématiser la méthode.



2.3 Techniques de conversion binaire/décimal :

A chaque digit correspond un "poids" en puissances de 2 à multiplier par 0 ou 1 selon le digit.
Notions de LSB et MSB.

exemple : $(101100)_2$

en décimal : $(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = 32 + 0 + 8 + 4 + 0 + 0 = (44)_{10}$

2.4 Suite des nombres binaires

		MSB			LSB
Hexadécimal	Décimal	2^3	2^2	2^1	2^0
0	0	0	0	0	0
1	1	0	0	0	1
2	2	0	0	1	0
3	3	0	0	1	1
4	4	0	1	0	0
5	5	0	1	0	1
6	6	0	1	1	0
7	7	0	1	1	1
8	8	1	0	0	0
9	9	1	0	0	1
A	10	1	0	1	0
B	11	1	0	1	1
C	12	1	1	0	0
D	13	1	1	0	1
E	14	1	1	1	0
F	15	1	1	1	1

2.5 Notation hexadécimale :

C'est juste une façon plus commode d'écrire "en binaire"

On remarque qu'avec un seul digit hexa on peut représenter sans ambiguïté 4 digits binaires (cf. tableau).

Exemples :

$$(A3F)_{16} = (101000111111)_2$$



$$(6B4)_{16} = (011010110100)_2$$

- Nombre maximum de nombres "codables" avec n bits en base 2 : 2^n .
- Plus grand nombre que l'on peut écrire $N_{\max} = 2^n - 1$
- Par exemple, sur 4 bits binaires, on peut coder $2^4 = 16$ nombres : de 0 à $(2^4 - 1) = 15$ (cf. tableau ci-dessus).
- Généralisation en base B : $N_{\max} = B^n - 1$

3 Notions d'algèbre de Boole

3.1 Etats logiques

Un certain nombre d'éléments technologiques sont conçus pour ne présenter que 2 états de fonctionnement seulement :

Elément	Etat 1	Etat 2
Interrupteur	Fermé 	Ouvert 
Transistor	Passant	Bloqué
Rayon lumineux (lecteur de CD)	Réfléchi	Absorbé
Proposition logique	Vraie	Fausse
Digit binaire	1	0

A chacun des deux états possibles, on associe une valeur logique : 1 ou 0.

L'algèbre qui est définie sur un tel ensemble de 2 valeurs s'appelle algèbre de Boole. Georges Boole (1815-1864) est un mathématicien et logicien anglais, créateur de la logique symbolique. En 1854, Boole publie son traité *An Investigation into the Laws of Thought, on Which Are Founded the Mathematical Theories of Logic and Probabilities*, qui est à la base de l'algèbre de la logique, c'est-à-dire ce qu'on appelle de nos jours l'algèbre de Boole.

3.2 Opérations booléennes

3.2.1 Somme booléenne ("OR", OU logique, Union)

➤ Exemple de proposition logique :

J'ai deux amis : Jean et Paul. Si l'un des deux au moins va au cinéma, j'irai au cinéma avec lui (ou eux).

Proposition a : Jean va au cinéma.

Proposition b : Paul va au cinéma.

Proposition s : Je vais au cinéma.

➤ D'où la formulation logique suivante :

Si	a est Fausse	et	b est Fausse	alors	s est Fausse	(je ne vais pas au cinéma)
Si	a est Fausse	et	b est Vraie	alors	s est Vraie	(je vais au cinéma)
Si	a est Vraie	et	b est Fausse	alors	s est Vraie	(je vais au cinéma)
Si	a est Vraie	et	b est Vraie	alors	s est Vraie	(je vais au cinéma).

- D'où l'établissement de la "table de vérité" (TV) de la fonction "OU logique".

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

Table de Vérité

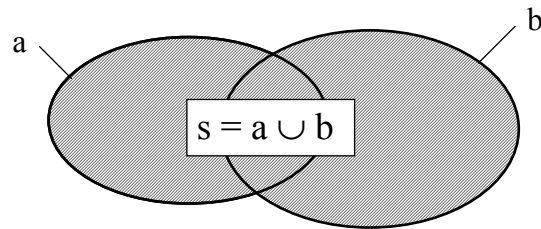


Diagramme de Venn

- Autre manière décrire cette fonction logique :
- $s = 1$ si soit : $a = 1$
 soit : $b = 1$
 soit : $a = 1$ ET $b = 1$.
- ou encore : s n'est égal à "0" que lorsque a ET b valent "0"

- La notation de la fonction "OU logique" est la suivante :

$$s = a + b$$

Autres notations : $s = a \vee b$ ou $s = a \cup b$

3.2.2 Produit booléen ("AND", ET logique, Intersection)

- Exemple de proposition logique :
- J'irai faire du vélo si il fait beau et que je suis en forme.
- Proposition a : Il fait beau.
- Proposition b : Je suis en forme.
- Proposition p : Je fais du vélo.

- D'où la formulation logique suivante :
- Si a est **Fausse** et b est **Fausse** alors p est **Fausse** (je ne fais pas de vélo)
- Si a est **Fausse** et b est **Vraie** alors p est **Fausse** (je ne fais pas du vélo)
- Si a est **Vraie** et b est **Fausse** alors p est **Fausse** (je ne fais pas du vélo)
- Si a est **Vraie** et b est **Vraie** alors p est **Vraie** (je fais du vélo).

- D'où l'établissement de la "table de vérité" (TV) de la fonction "ET logique".

a	b	p
0	0	0
0	1	0
1	0	0
1	1	1

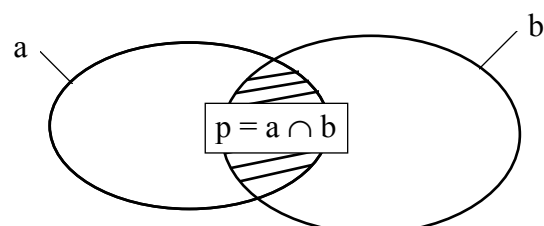


Table de Vérité

- Autre manière de décrire cette fonction logique :

Diagramme de Venn

p n'est égal à "1" que lorsque a ET b valent "1"

- La notation de la fonction "ET logique" est la suivante :

$$p = a \cdot b$$

Autres notations : $p = a \wedge b$ ou $p = a \cap b$

3.2.3 Négation ("INV", Inversion, Complémentation)

- Table de vérité" (TV) de la fonction "Inversion".

a	x
0	1
1	0

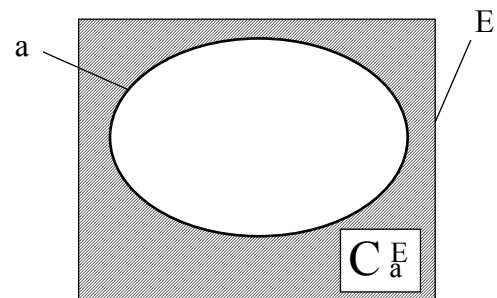


Table de Vérité

Diagramme de Venn

- La notation de la fonction "Inversion" est la suivante :

$$\overline{x} = a = C_a^E$$

En notation ensembliste : " x " est le complément de " a " dans " E ".

3.2.4 Somme exclusive ("XOR", OU exclusif)

- Table de vérité de la fonction "OU exclusif".

a	b	s'
0	0	0
0	1	1
1	0	1
1	1	0

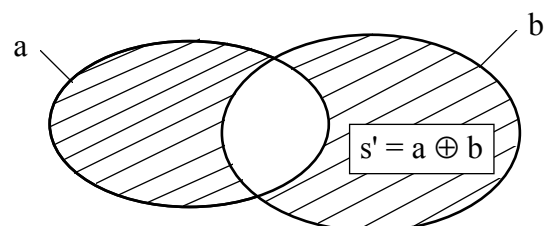


Table de Vérité

Diagramme de Venn

- La notation de la fonction "OU exclusif" est la suivante :

$$s' = a \oplus b$$

- Cette fonction est utilisée pour la comparaison :

$$a \oplus b = 0 \quad \Rightarrow \quad a = b$$

$$a \oplus b = 1 \quad \Rightarrow \quad a \neq b$$

Remarque 1 : $a \oplus b = \bar{a}b + a\bar{b}$. On peut le vérifier à l'aide des TV :

a	b	\bar{a}	\bar{b}	$\bar{a}b$	$a\bar{b}$	$\bar{a}b + a\bar{b}$	$a \oplus b$
0	0	1	1	0	0	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	1	1	1
1	1	0	0	0	0	0	0

Remarque 2 : La fonction XOR permet d'implanter la fonction parité : P=somme algébrique (modulo 2) des digits. Elle permet donc de réaliser des additionneurs.

Exercice de cours (à faire faire par l'étudiant): Ecrire les tables de vérité des fonctions logiques AND, OR et XOR de trois variables logiques a, b et c. Trouver une formulation simple pour exprimer chacune d'elles.

3.3 Propriétés des fonctions booléennes

3.3.1 Propriétés des "OU logique" et "ET logique"

Les propriétés données dans le tableau ci-dessous découlent directement de la définition des fonctions booléennes. On peut les retrouver aisément à partir des TV ou des diagrammes de Venn correspondants.

Propriétés	OU Logique	ET Logique
Idempotence	$x + x = x$	$x \cdot x = x$
Commutativité	$x + y = y + x$	$x \cdot y = y \cdot x$
Associativité	$x + (y + z) = (x + y) + z$	$x \cdot (y \cdot z) = (x \cdot y) \cdot z$
Élément neutre	$x + 0 = x$	$x \cdot 1 = x$
Complémentarité	$x + \bar{x} = 1$	$x \cdot \bar{x} = 0$
Élément absorbant	$x + 1 = 1$	$x \cdot 0 = 0$

3.3.2 Propriétés générales

Les propriétés données dans le tableau ci-dessous découlent directement de la définition des fonctions booléennes. On peut les retrouver aisément à partir des TV ou en utilisant les propriétés précédentes. On remarquera, qu'en algèbre de Boole, il faut parfois rajouter des termes pour mettre en évidence une simplification ultérieure (cf. démonstration donnée ci-après).

Propriétés	
Involution	$\overline{\overline{x}} = x$
Distributivité	$x \cdot (y + z) = x \cdot y + x \cdot z$ $x + (y \cdot z) = (x + y) \cdot (x + z)$
Absorption	$x + x \cdot y = x$ $x \cdot (x + y) = x$ $x + \overline{x} \cdot y = x + y$

Démonstration de la première propriété d'absorption :

$$\begin{aligned}
 x + xy &= x \cdot 1 + xy = x(y + \overline{y}) + xy && \text{; élément neutre ET et complémentarité OU} \\
 &= xy + x\overline{y} + xy = xy + x\overline{y} && \text{; distributivité et idempotence OU} \\
 &= x(y + \overline{y}) = x \cdot 1 = x && \text{; complémentarité OU et élément neutre ET}
 \end{aligned}$$

Démonstration de la dernière propriété d'absorption :

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

3.3.3 Théorème de "de Morgan"

C'est un des théorèmes les plus utilisés. Il comprend deux parties :

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

et

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

3.3.4 Représentation symbolique des opérateurs logiques

Les fonctions logiques sont représentées (figure 3) par des symboles appelés "**portes logiques**" ou plus simplement "portes" ("gates" en anglais). En électronique une porte représente un circuit à diodes et transistors capable d'implanter la fonction donnée. La

structure électronique des diverses portes logiques en technologie CMOS, sera donnée dans la partie suivante du cours (§2).

Les fonctions NON-OU ("NOR") et NON-ET ("NAND") qui correspondent simplement à l'inversion des fonctions OU et ET sont très utilisées en électronique numérique compte-tenu de leur plus grande facilité d'implantation. Leur représentation est aussi donnée dans la figure 3. Cette figure donne à la fois la représentation courante issue de la normalisation américaine et la représentation préconisée par la norme internationale.


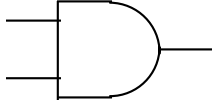
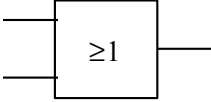
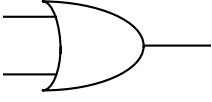
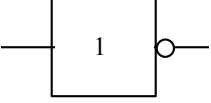
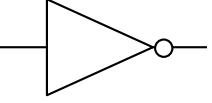
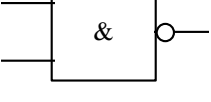
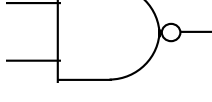
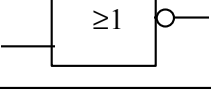
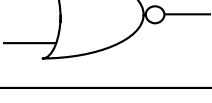
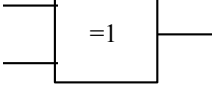

Fonction	Norme internationale	Représentation courante
ET AND		
OU OR		
NON NOT, INV		
NON-ET NAND		
NON-OU NOR		
Ou Exclusif XOR		

Figure 3 : Représentation des portes logiques

4 Logique CMOS

Les circuits numériques exploitent le comportement en interrupteur de certains dispositifs électroniques comme la diode et le transistor. A l'heure actuelle, il existe principalement quatre technologies courantes pour la fabrication des circuits logiques, à savoir :

- **TTL** (Transistor-Transistor Logic) : technologie à transistors bipolaires, la plus ancienne
- **ECL** (Emitter-Coupled Logique) : technologie à transistors bipolaires, la plus rapide mais délicate à mettre en œuvre,
- **NMOS** : technologie n'utilisant que des transistors de type NMOS
- **CMOS** : technologie utilisant à la fois des transistors de type NMOS et PMOS

De nos jours, la technologie CMOS est très nettement prépondérante. C'est la raison pour laquelle nous nous limiterons à l'étude de l'implantation des portes logiques dans cette dernière technologie. Bien sûr, il n'est pas question, dans le cadre d'un cours d'initiation sur la logique, de développer l'étude complète de la technologie de fabrication des circuits CMOS et encore moins d'étudier le fonctionnement du transistor MOS de façon détaillée. Ces points seront traités plus tard dans le cursus EEA, au niveau du DEUG STPI-GS et surtout en licence/maîtrise EEA, IUP électronique ou filière MEA de l'ISIM.

Comme l'élément de base de la logique est l'interrupteur, nous allons commencer par étudier le transistor en tant qu'interrupteur.

4.1 Transistor MOS en commutation

4.1.1 Transistor NMOS

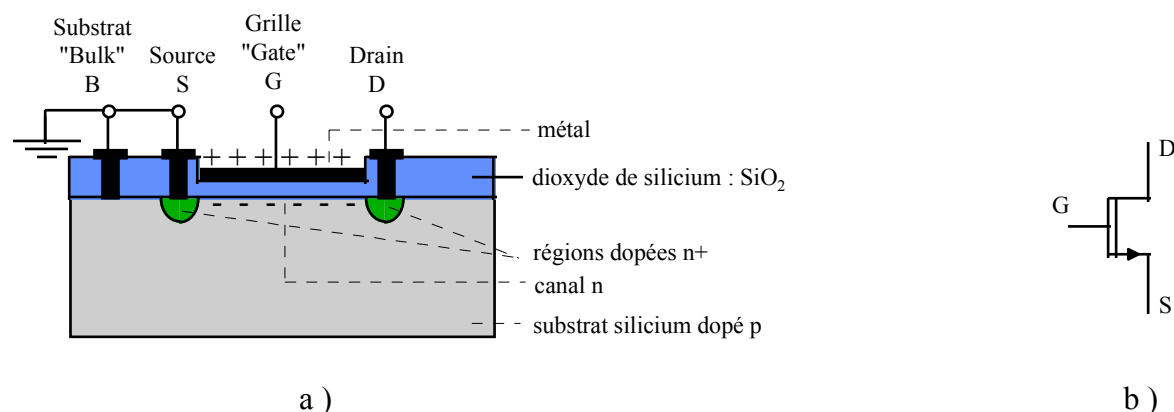


Figure 4 : Transistor NMOS à enrichissement

a) Vue en coupe. Ici, le transistor est polarisé : Source et Substrat sont reliés à la masse et la grille portée à un potentiel positif induit une charge négative dans le canal

b) représentation schématique simplifiée lorsque Source et Substrat sont reliés

Un transistor MOS est un dispositif microélectronique qui comporte 4 broches **Drain (D)**, **Grille [Gate] (G)**, **Source (S)** et Substrat ["Bulk"] (B). Il est constitué d'un sandwich "Métal-Oxyde-Semiconducteur" qui donne son nom à la structure : MOS. Le schéma de la figure 4

représente un transistor MOS à canal N, ou NMOS. L'étude complète du fonctionnement de ce transistor pour son utilisation dans les circuits analogiques (amplificateurs différentiels, amplificateurs opérationnels, etc.) nécessite des connaissances en physique des semiconducteurs qui sont pas encore acquises par les étudiants à bac+1. Elle ne sera donc pas entreprise dans le cadre de ce cours. Par contre, pour l'utilisation de ce transistor dans les circuits logiques, on peut se contenter d'une première approche simple (et même simpliste) pour comprendre le fonctionnement en "interrupteur" de ce type de circuit.

Plaçons nous dans la configuration où la source S et le substrat B sont reliés à la masse ($V_S = V_B = 0$).

a) Lorsque la grille G est portée au potentiel nul ($V_{GS} = 0$), le drain et la source du transistor sont isolés l'un de l'autre par la zone dopée p (charges positives). Le transistor ne peut pas conduire. On dit que le transistor est dans son état **bloqué** ["OFF"].

b) Lorsque l'on polarise la grille au-dessus d'un potentiel positif minimum, appelé tension de seuil ["threshold"] ($V_{GS} > V_T$ et $V_T \approx 1V$), des charges négatives sont induites dans la zone située sous l'oxyde de grille, créant ainsi un "canal n". Ce canal correspond à une zone conductrice entre la source et le drain et le transistor est dans son état **passant** ["ON"]. L'état passant du transistor comporte plusieurs régimes de fonctionnement (linéaire, saturé, etc.) selon les valeurs de V_{GS} et de V_{DS} . Pour ce qui nous concerne (circuits logiques), nous considérerons que les tensions appliquées sont telles que lorsque le transistor est dans son état passant il se comporte pratiquement comme un court-circuit, c'est-à-dire que l'on a : $V_D = V_S$.

D'une façon générale, dans une logique positive à deux niveaux, on choisit de représenter le niveau logique "0" par le potentiel nul $V = 0$ et le niveau logique "1" par la tension d'alimentation du circuit $V = V_{DD}$. [Pour la plupart des circuits $V_{DD} = 5V$ mais depuis le développement des dispositifs portables (calculatrice, ordinateur, téléphone, etc.) on cherche à réduire au maximum la puissance dissipée et en conséquence à diminuer les tensions d'alimentation jusqu'à 3V, voire même 1,3V].

4.1.2 Transistor PMOS

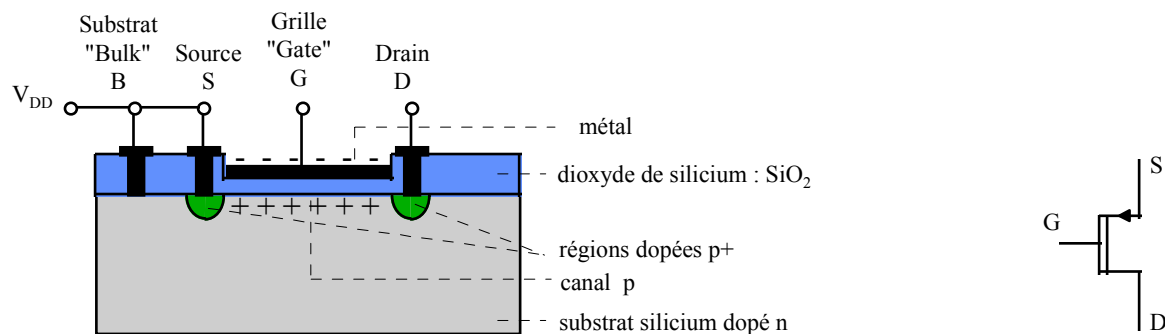


Figure 5 : Transistor PMOS à enrichissement

a) Vue en coupe. Ici, le transistor est polarisé : Source et Substrat sont reliés à la tension d'alimentation V_{DD} et la grille portée à un potentiel négatif par rapport à la source induit une charge positive dans le canal

b) représentation schématisée simplifiée lorsque Source et Substrat sont reliés

Le schéma de la figure 5 représente un transistor MOS à canal P, ou PMOS. Plaçons nous dans la configuration où la source S et le substrat B sont reliés à la tension d'alimentation ($V_S = V_B = V_{DD}$).

a) Lorsque la grille G est portée au potentiel d'alimentation V_{DD} ($V_{GS}=0$), le drain et la source du transistor sont isolés l'un de l'autre par la zone dopée n (charges négatives). Le transistor ne peut pas conduire. On dit que le transistor est dans son état **bloqué** ["OFF"].

b) Lorsque l'on polarise la grille en-dessous du potentiel de la source de telle façon que $V_{GS} < 0$ et $|V_{GS}| > |V_T|$, avec $V_T \approx -1V$, tension de seuil négative, des charges positives sont induites dans la zone située sous l'oxyde de grille, créant ainsi un "canal p ". Ce canal correspond à une zone conductrice entre la source et le drain et le transistor est dans son état **passant** ["ON"]. L'état passant du transistor comporte plusieurs régimes de fonctionnement (linéaire, saturé, etc.) selon les valeurs de V_{GS} et de V_{DS} . Pour ce qui nous concerne (circuits logiques), nous considérerons que les tensions appliquées sont telles que lorsque le transistor est dans son état passant il se comporte pratiquement comme un court-circuit, c'est-à-dire que l'on a : $V_D = V_S$.

4.2 Inverseur CMOS

Les fonctionnements des transistors NMOS et PMOS sont donc complémentaires comme le résume le schéma de la figure 6, en appliquant les niveaux logiques 0 (0V) et 1 (V_{DD}).



Figure 6 : Etat de conduction des transistors NMOS (a) et PMOS (b)

En associant en cascade un transistor PMOS et un transistor NMOS on obtient la fonction d'inversion logique comme indiquée sur la figure 7.

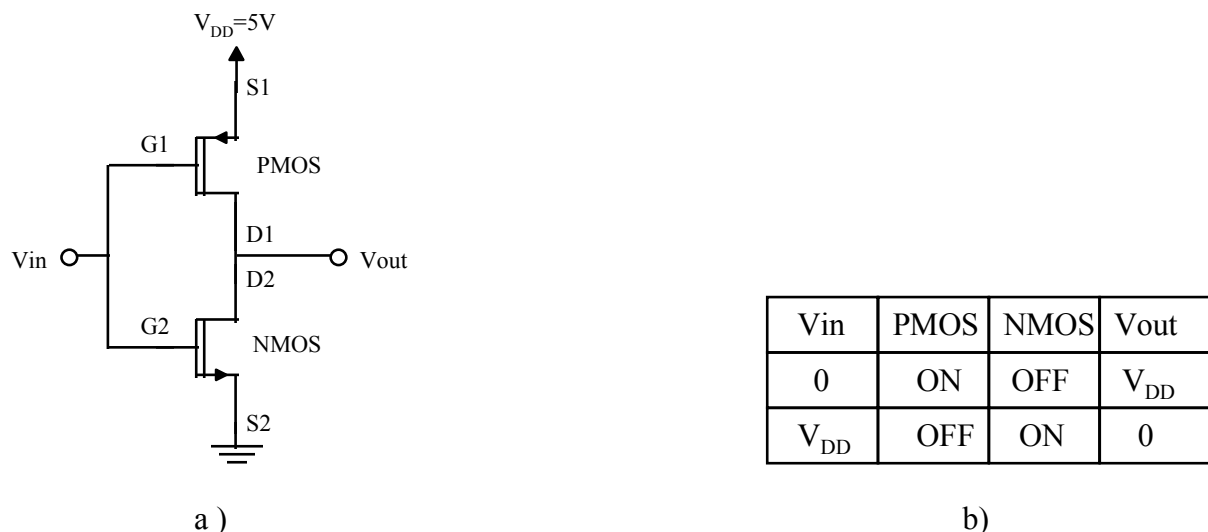


Figure 7 : Inverseur CMOS : schéma (a) et table de vérité (b)

4.3 Fabrication des circuits intégrés CMOS

Il ne sera donné ici qu'un bref aperçu de la technique de fabrication des circuits intégrés ["**Integrated Circuits : IC**"]. Le matériau semiconducteur utilisé est le Silicium (Si) obtenu à partir de la silice (oxyde de silicium SiO_2) qui est le constituant essentiel du sable. On part (figure 8) d'un énorme **lingot monocristallin** de silicium dans lequel on découpe des tranches ["**Wafers**"]. Sur ces tranches on implante par photolithogravure, implantation ionique et diffusion les transistors qui constituent les circuits électroniques (figure 9). Le procédé de photolithogravure est un procédé assez complexe qui permet de "graver" les transistors dans le silicium. Il ne sera pas détaillé ici. A ce niveau, il suffit de savoir que ce procédé de fabrication se fait à partir d'un jeu de **masques**. C'est ce jeu de masques que produisent les ingénieurs de conception lorsqu'ils conçoivent un circuit intégré à partir d'un outil de **Conception Assistée par Ordinateur (CAO)**. Pour la fabrication des circuits, ce jeu de masques (codé sous forme numérique) est envoyé à un fabricant de circuits (**fondeur**) qui renvoie les circuits fabriqués (nus ou encapsulés) quelques semaines plus tard.

A l'heure actuelle (2000), les principales caractéristiques de la technologie pour un circuit numérique de complexité moyenne sont les suivants :

- dimension du canal transistor : $0,6\ \mu\text{m}$
- dimension d'un circuit : $0,5\text{cm} \times 0,5\text{cm}$
- diamètre des wafers : 6" et 8" (15,25 cm et 20,35 cm)
- épaisseur des wafers : $100\ \mu\text{m}$
- nombre de circuits ["die"] par wafer : 1000
- nombre de transistors par circuit : 100 000 (plus de 10 millions pour les microprocesseurs avancés).

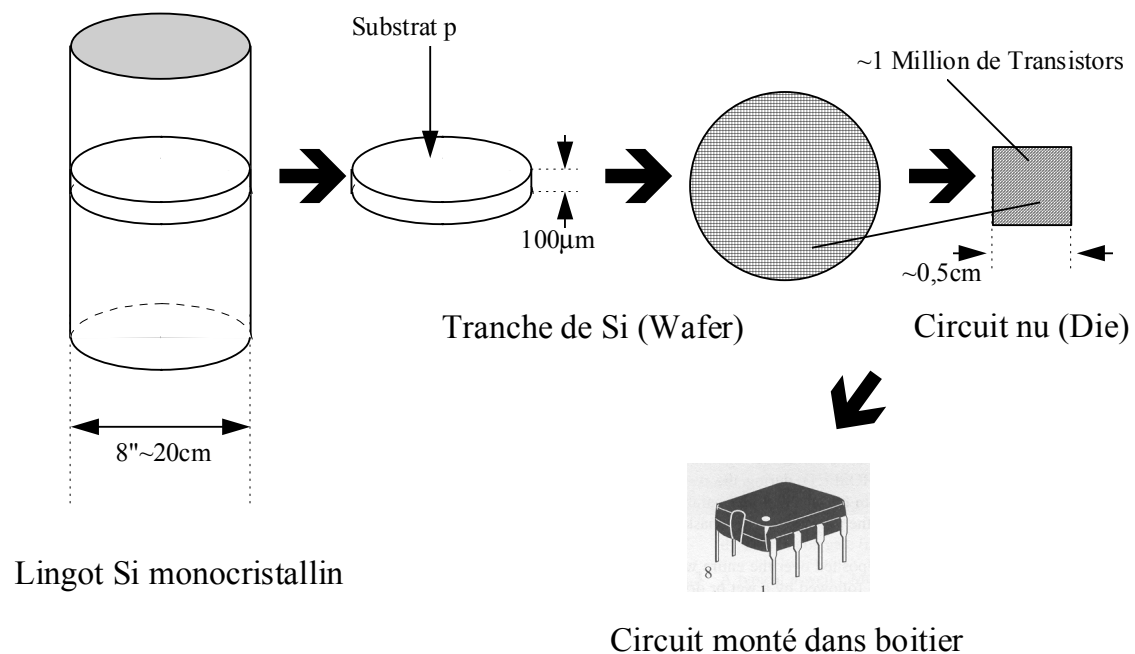


Figure 8 : Fabrication d'un circuit intégré.

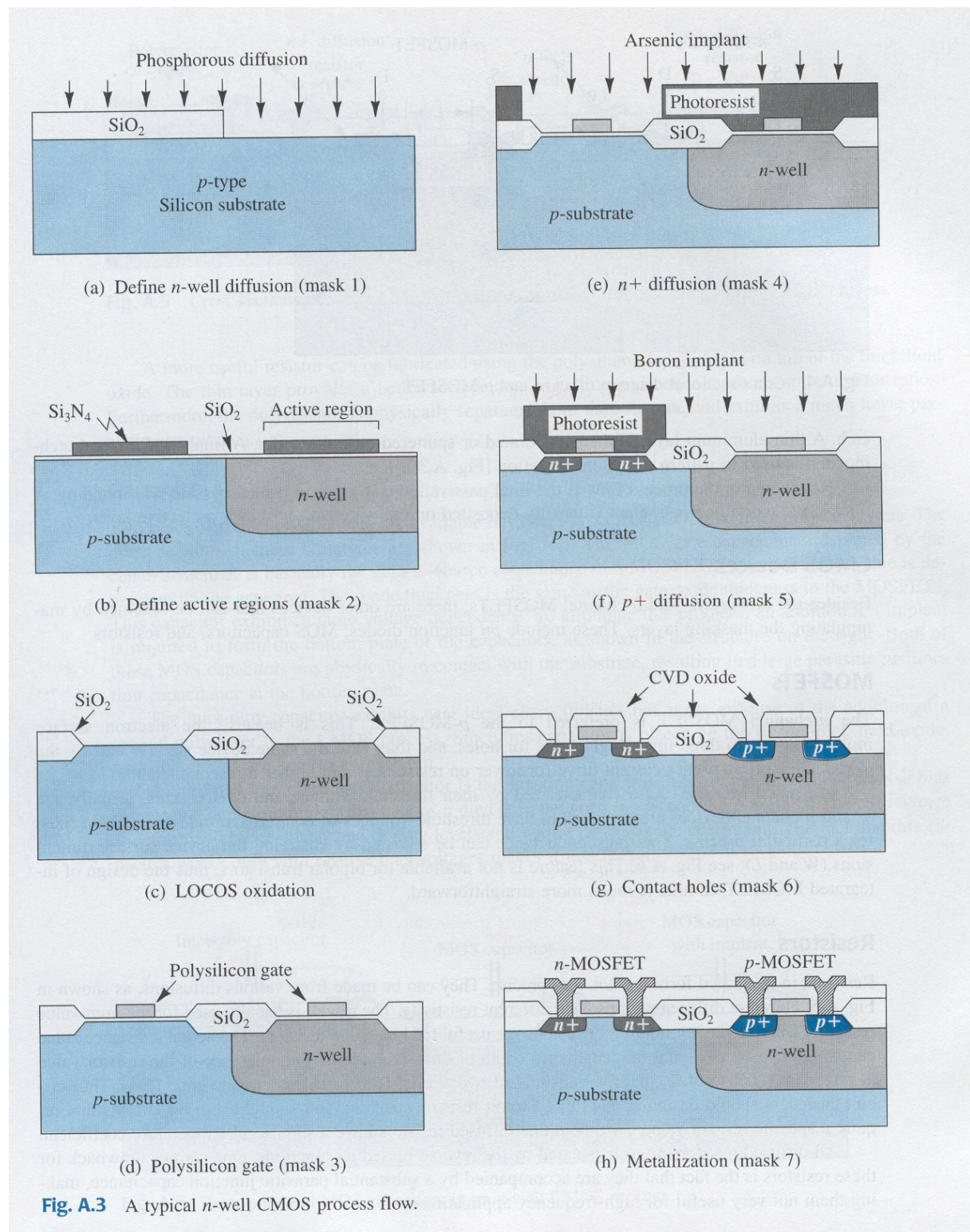
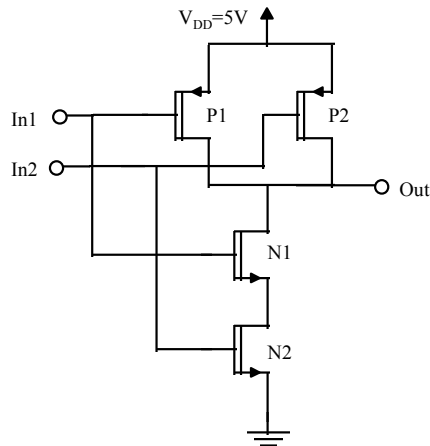


Figure 9 : Process de fabrication CMOS (d'après Sedra & Smith).

4.4 Portes logiques CMOS

4.4.1 Porte NAND à deux entrées



In1	In2	P1	P2	N1	N2	Out
0	0	ON	ON	OFF	OFF	V_{DD}
0	V_{DD}	ON	OFF	OFF	ON	V_{DD}
V_{DD}	0	OFF	ON	ON	OFF	V_{DD}
V_{DD}	V_{DD}	OFF	OFF	ON	ON	0

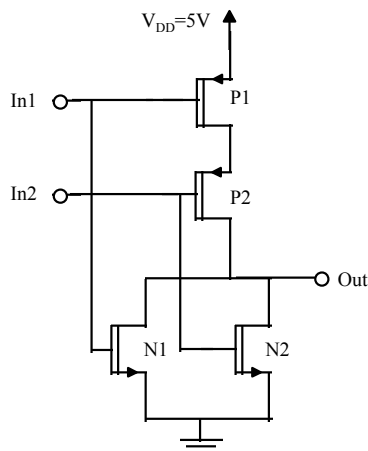
Figure 10 : structure électronique et table de vérité d'une porte NAND en technologie CMOS

Dans le "plan p", les transistors P1 et P2 sont en parallèle : il suffit que l'un des deux au moins soit passant (état "ON") pour que le niveau de sortie soit égal à la tension d'alimentation V_{DD} (à quelques dixièmes de Volts près). Dans le "plan n", les transistors N1 et N2 sont en série : il suffit que l'un des deux soit bloqué ("OFF") pour que la sortie soit isolée de la masse. Ceci explique les trois premières lignes de la TV : Out = V_{DD} = "1".

Pour la dernière ligne de la TV, on constate que P1 et P2 bloqués isolent la sortie de la tension d'alimentation et que N1 et N2 passants assure la liaison entre la sortie et la masse (Out = 0V = "0").

Exercice d'application de cours : concevoir une porte NAND à trois entrées.

4.4.2 Porte NOR à deux entrées



In1	In2	P1	P2	N1	N2	Out
0	0	ON	ON	OFF	OFF	V_{DD}
0	V_{DD}	ON	OFF	OFF	ON	0
V_{DD}	0	OFF	ON	ON	OFF	0
V_{DD}	V_{DD}	OFF	OFF	ON	ON	0

Figure 11 : structure électronique et table de vérité d'une porte NOR en technologie CMOS

5 Représentation des fonctions logiques

5.1 Forme algébrique

Une fonction logique peut se représenter sous deux formes algébriques :

- une somme de produits : **fonction normale disjonctive**
- un produit de sommes : **fonction normale conjonctive**

Si toutes les variables sont explicitement exprimées, on parle de forme canonique, sinon de **forme simplifiée**.

5.1.1 Forme normale disjonctive ($\Sigma \Pi_i$)

Écriture de la fonction :

A partir de la Table de Vérité (TV), on "réalise" les "1".

- La somme aura donc autant de termes qu'il y a de "1" dans la TV. Pour que la fonction résultat soit vraie il suffit qu'un seul terme de la somme soit égal à "1" :
($x + y + \dots + 1 + w + z = 1$).
- Chaque terme de la somme est un produit des variables ou de leur complément. Pour qu'un produit soit égal à "1" il faudra que chacun de ses facteurs soit égal à "1"
=> valeur à "1" => facteur non complémenté
valeur à "0" => facteur complémenté

Exemple :

On se propose d'écrire la forme normale disjonctive de la fonction "parité de trois variables".

Fonction parité P :

si $\Sigma "1" = 1 \pmod{2}$, $P = 1$. La fonction est impaire (il y a un nombre impair de "1")

si $\Sigma "1" = 0 \pmod{2}$, $P = 0$. La fonction est paire (il y a un nombre pair de "1").

	a	b	c	P	\bar{P}
S0	0	0	0	0	1
S1	0	0	1	1	0
S2	0	1	0	1	0
S3	0	1	1	0	1
S4	1	0	0	1	0
S5	1	0	1	0	1
S6	1	1	0	0	1
S7	1	1	1	1	0

Lire : P sera égal à "1" si :

($a=0$ ET $b=0$ ET $c=1$) OU ($a=0$ ET $b=1$ ET $c=0$) OU ($a=1$ ET $b=0$ ET $c=0$) OU
($a=1$ ET $b=1$ ET $c=1$)

d'où l'écriture de P sous sa forme normale disjonctive :

$$P = \bar{a}\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c + ab\bar{c}$$

5.1.2 Forme normale conjonctive ($\prod \Sigma_i$)

Ecriture de la fonction :

A partir de la Table de Vérité (TV), on "réalise" les "0".

- Le produit aura donc autant de facteurs qu'il y a de "0" dans la TV : P_0
 $= S_0 \cdot S_3 \cdot S_5 \cdot S_6$
- Chaque facteur S_i du produit est une somme des variables ou de leur complément. Un facteur est égal à "0" si chacun des termes qui le composent est égal à "0"

Exemple :

Pour la fonction parité de trois variables :

la forme normale conjonctive de P s'écrit :

$$P = (a+b+c) \cdot (a+\bar{b}+\bar{c}) \cdot (\bar{a}+b+\bar{c}) \cdot (\bar{a}+\bar{b}+c)$$

$$S_0=0 \text{ ssi } a+b+c=0 \text{ soit } a=0 \text{ et } b=0 \text{ et } c=0$$

$$S_3=0 \text{ ssi } a+\bar{b}+\bar{c}=0 \text{ soit } a=0 \text{ et } b=1 \text{ et } c=1$$

$$S_5=0 \text{ ssi } \bar{a}+b+\bar{c}=0 \text{ soit } a=1 \text{ et } b=0 \text{ et } c=1$$

$$S_6=0 \text{ ssi } \bar{a}+\bar{b}+c=0 \text{ soit } a=1 \text{ et } b=1 \text{ et } c=0$$

5.1.3 Choix d'une forme

On choisit celle qui conduit au plus petit nombre de termes (dans l'exemple, il y a 4 termes pour chaque forme).

On peut passer facilement de la forme disjonctive à la forme conjonctive :

a) On réalise les "1" de \bar{P} sous forme disjonctive (ΣP)

$$\bar{P} = \bar{a} \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot c + a \cdot \bar{b} \cdot c + a \cdot b \cdot \bar{c}$$

b) On complémente \bar{P} pour obtenir P et on applique le théorème de de Morgan (2 fois)

$$P = (a+b+c) \cdot (a+\bar{b}+\bar{c}) \cdot (\bar{a}+b+\bar{c}) \cdot (\bar{a}+\bar{b}+c)$$

Remarque 1 : Ici, on utilisera systématiquement la forme disjonctive (somme de produits).

Remarque 2 : On peut montrer que les deux formes sont équivalentes en développant la forme conjonctive et en simplifiant.

5.2 Tables de Karnaugh (2ⁿ cases)

n variables : 2ⁿ cases

Par définition, c'est une TV ordonnée spécialement de façon à mettre en évidence les simplifications. L'ordre de rangement est tel que si l'on passe d'une case à une case adjacente, une seule variable change

5.2.1 Table de Karnaugh à 2 variables

a\b	0	1
0	0	1
1	1	0

Exemple : ou exclusif : $x = a \oplus b$

La table est très simple. Elle n'est pas vraiment utile.

5.2.2 Table de Karnaugh à 3 variables

a\bc	00	01	11	10
0	(000) 0	(001) 1	(011) 0	(010) 1
1	(100) 1	(101) 0	(111) 1	(110) 0

Exemple : fonction parité $P = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$. Elle est irréductible.

Remarque : Les cellules extrêmes, elles-aussi, sont adjacentes (imaginer la table roulée sur un cylindre vertical, puis sur un cylindre horizontal)

(110)	(100)	(101)	(111)	(110)	(100)
(010)	(000)	(001)	(011)	(010)	(000)
(110)	(100)	(101)	(111)	(110)	(100)
(010)	(000)	(001)	(011)	(010)	(000)

5.2.3 Table de Karnaugh à 4 variables

ab\cd	00	01	11	10
00	(0000) 0	(0001) 1	(0011) 0	(0010) 0
01	(0100) 0	(0101) 0	(0111) 1	(0110) 1
11	(1100) 0	(1101) 0	(1111) 1	(1110) 1
10	(1000) 0	(1001) 1	(1011) 0	(1010) 0

Exemple : fonction $f = \bar{a}\bar{b}cd + \bar{a}bcd + \bar{a}b\bar{c}d + abcd + abc\bar{d} + ab\bar{c}d$

5.3 Simplification des fonctions logiques

La méthode de simplification algébrique se traduit, sur une table de Karnaugh, par un regroupement de cases adjacentes pour lesquelles on a un « 1 », en blocs d'ordre 2, 4, 8, 16, ..., 2^n . A chaque fois, on élimine la variable qui change dans l'adjacence.

Exemple 1 : blocs d'ordre 2

$$S = abcd + ab\bar{c}d + \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}c\bar{d} = abd(c + \bar{c}) + \bar{a}\bar{b}\bar{c}(d + \bar{d}) = abd + \bar{a}\bar{b}\bar{c}$$

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

Exemple 2 : blocs d'ordre 4

$$S = \bar{a}\bar{b}\bar{c}\bar{d} + \bar{a}\bar{b}\bar{c}d + \bar{a}\bar{b}c\bar{d} + \bar{a}\bar{b}cd = \bar{a}\bar{b}\bar{c} + \bar{a}\bar{b}c = \bar{a}\bar{b}$$

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

6 Applications des fonctions logiques

Exemple 1 : comparaison entre deux mots binaires

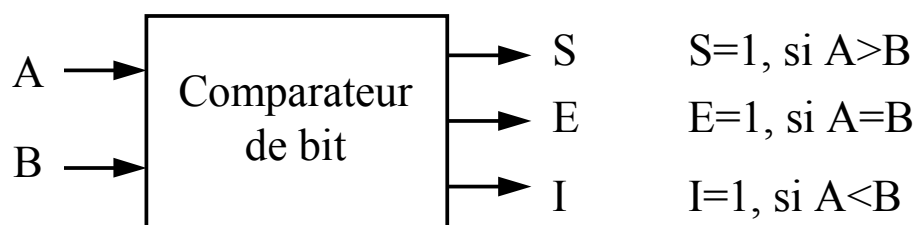


Table de Vérité pour A et B sur 1 bit :

A	B	S	E	I	/A	/B	A./B	/A.B	A.B	/A./B	A.B+/A./B
0	0	0	1	0	1	1	0	0	0	1	1
0	1	0	0	1	1	0	0	1	0	0	0
1	0	1	0	0	0	1	1	0	0	0	0
1	1	0	1	0	0	0	0	0	1	0	1

$$S = A.\bar{B} \quad I = \bar{A}.B \quad E = A.B + \bar{A}.\bar{B}$$

D'où l'implantation :

