

HLEE205

1^{ère} partie : Electronique Numérique

Contenu du cours

- **Éléments de l'algèbre de Boole** : traitement des signaux binaires, écriture/simplification des fonctions logiques
- **Constitution des portes logiques** utilisées dans le domaine de l'électronique numérique
- **Constitution transistors CMOS** : base des portes MOS
- **Techniques de fabrication des circuits intégrés**

Toutes ces notions fondamentales de la microélectronique seront approfondies pour les étudiants qui suivront la filière EEA.

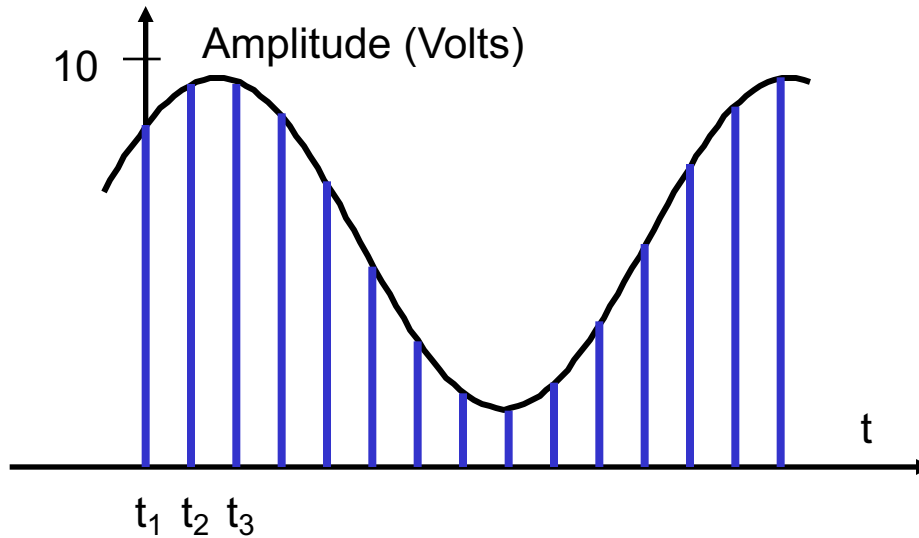
1 / Introduction

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

- ordinateurs (informatique)
- systèmes de commande et de contrôle (automatique, robotique, etc.)
- systèmes de télécommunications (télévision numérique)
- multimédia (CD audio, CD numérique, DVD, lecteur MP3, etc.)
- systèmes de mesures : microcapteurs, microsystèmes, etc.

Traitement des signaux analogiques

- Signaux physiques : type analogique
- Exemples : ondes sonores, ondes lumineuses, signaux électriques issus de capteurs, etc.
- Traitement numérique signaux analogiques : échantillonner et convertir en valeur binaire (CAN)
- Échantillonner : prendre la valeur analogique du signal à des instants discrets donnés : t_1, t_2, t_3, \dots



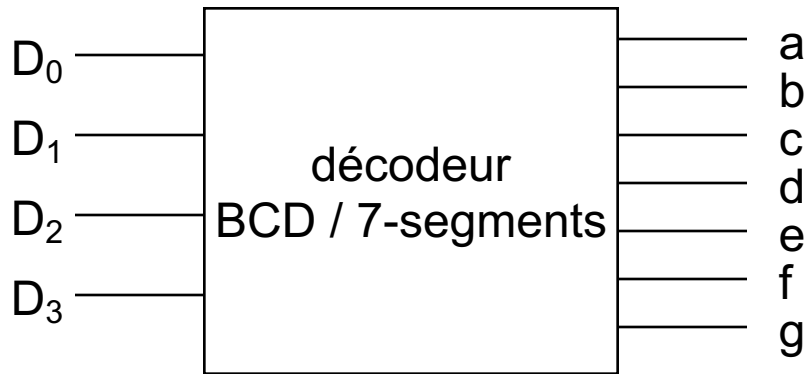
instants	Amplitude (V)	10 x amplitude	Codage DCB
t_1	9,2	92	1001 0010
t_2	9,6	96	1001 0110
t_3	9,7	97	1001 0111
....

Illustration : Décodeur BCD/7segments

Remarque : Les éléments de logique binaire dispensés sont une simple introduction à l'électronique numérique combinatoire (c'est-à-dire non séquentielle).

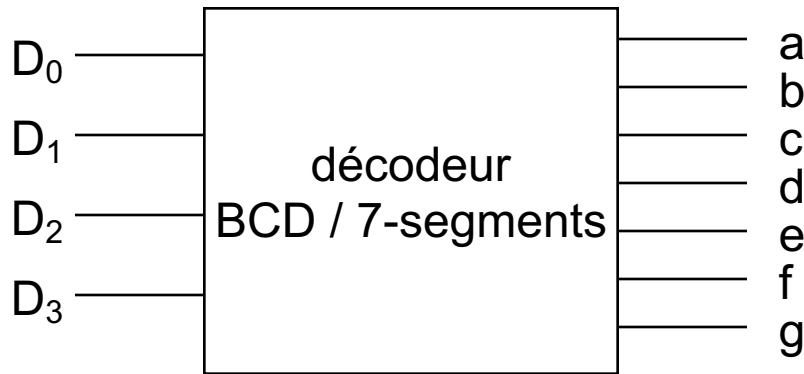
Pour illustrer les potentialités de cette discipline, on prendra l'exemple de la réalisation d'un objet technologique simple : un **décodeur pour affichage 7 segments**

Décodeur BCD/7segments



Représentation symbolique

Décodeur BCD/7segments



D_3	D_2	D_1	D_0	Digit
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

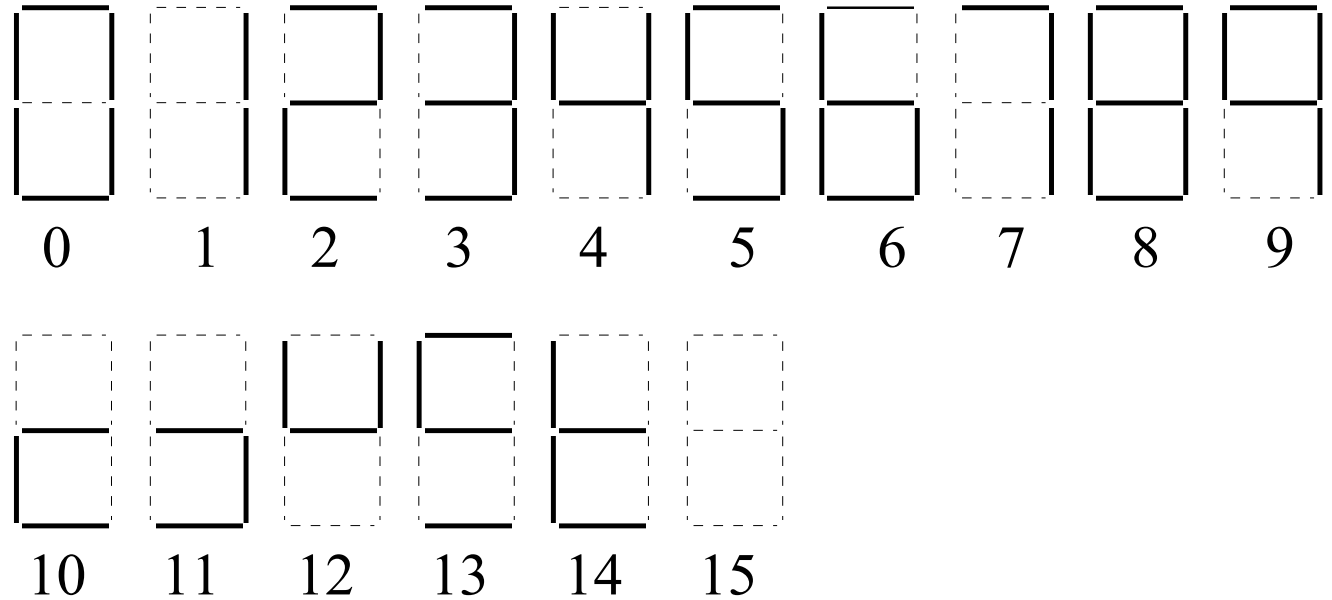
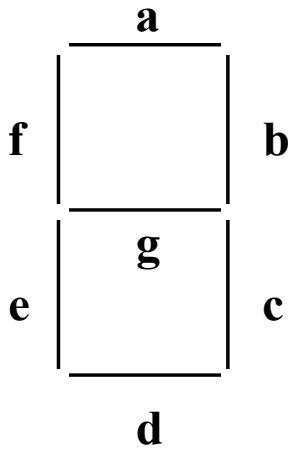
Représentation symbolique

Table de vérité.

A partir du code numérique $D_3D_2D_1D_0$ en entrée, délivrer les 7 bits (a,b,c,d,e,f,g) pour 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 dessinent une image du chiffre à représenter.

Afficheur 7 segments : fonctionnement

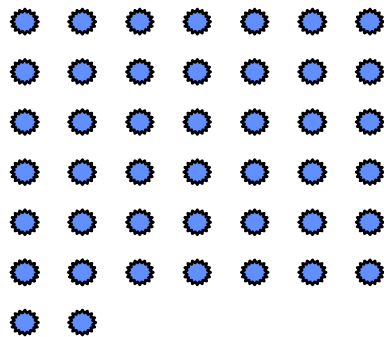


Pour réaliser ce décodeur, il faut :

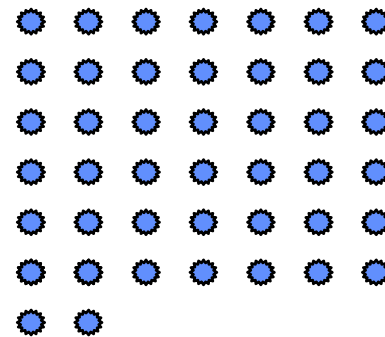
- Comprendre algèbre binaire (algèbre de Boole)
- Comprendre conception portes logiques pour implantation matérielle décodeur (technologie CMOS)
- Savoir minimiser fonctions logiques (Tables de Karnaugh) : économiser de nombre de portes logiques

2 / Système de numération

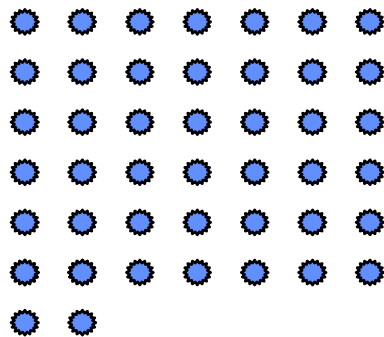
2 . 1 / Exercice d'évaluation des connaissances



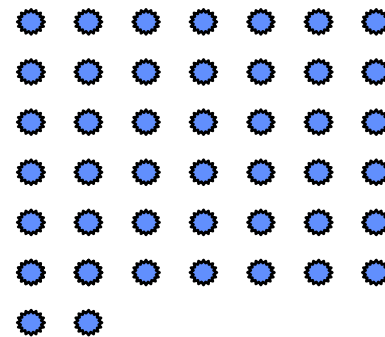
En base 10 :



En base 5 :



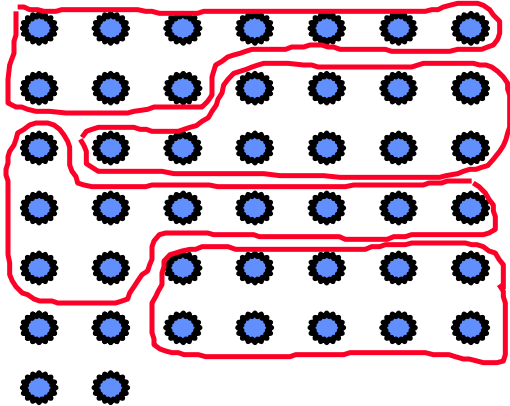
En base 16 :



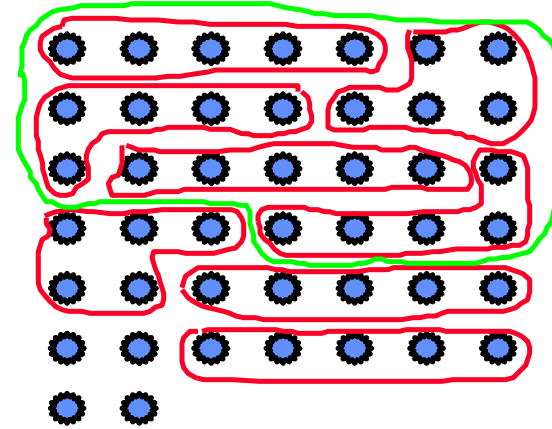
En base 2 :

Eléments de corrections :

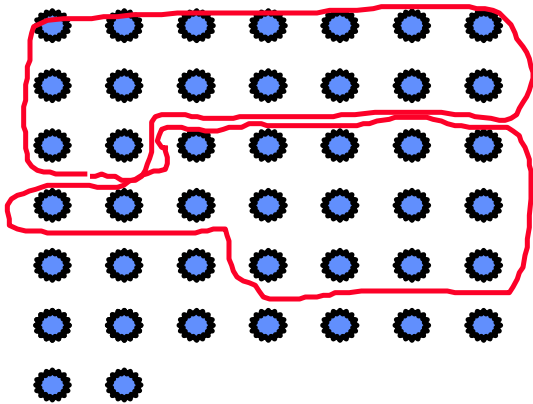
- Système à base "n" : on fait des groupes de $(n)=n^1$, puis de $(n.n)=n^2$, puis de $(n.n.n)=n^3$, etc.
- On dispose de n symboles pour coder le nombre : de 0 à n-1 (0 à 9 en décimal, par exemple)
- On écrit le nombre en alignant les symboles de droite à gauche
 - nombre d'unités, non "regroupables"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



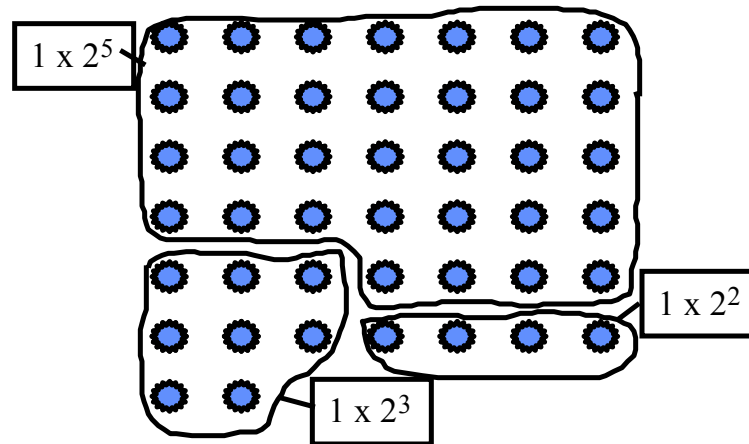
$$(4 \times 10^1) + (4 \times 10^0) = (44)_{10}$$



$$(1 \times 5^2) + (3 \times 5^1) + (4 \times 5^0) = (134)_5$$

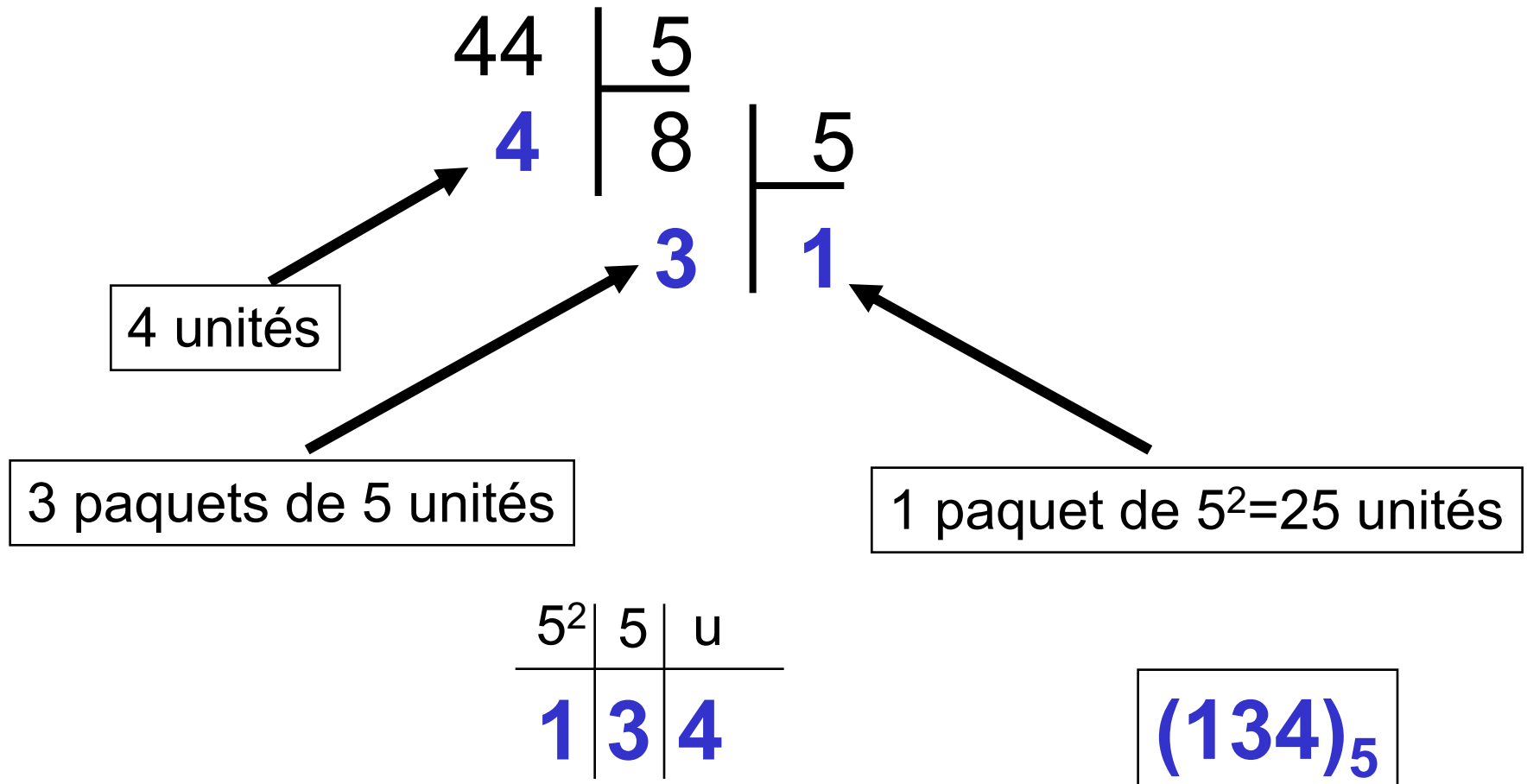


$$(2 \times 16^1) + (C \times 16^0) = (2C)_{16}$$



$$(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) = (101100)_2$$

Exemple : numération à base 5



Commentaires :

- Pourquoi le système décimal ? On a 10 doigts
- Pour les systèmes de base >10 , on ajoute d'autres symboles. Pour l'hexadécimal, par exemple :
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 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 $(44)_{10}$

$$(2^6=64) > 44 > (2^5=32) \quad \rightarrow \quad 44 = 2^5 + 12$$

$$(2^4=16) > 12 > (2^3=8) \quad \rightarrow \quad 44 = 2^5 + 2^3 + 4$$

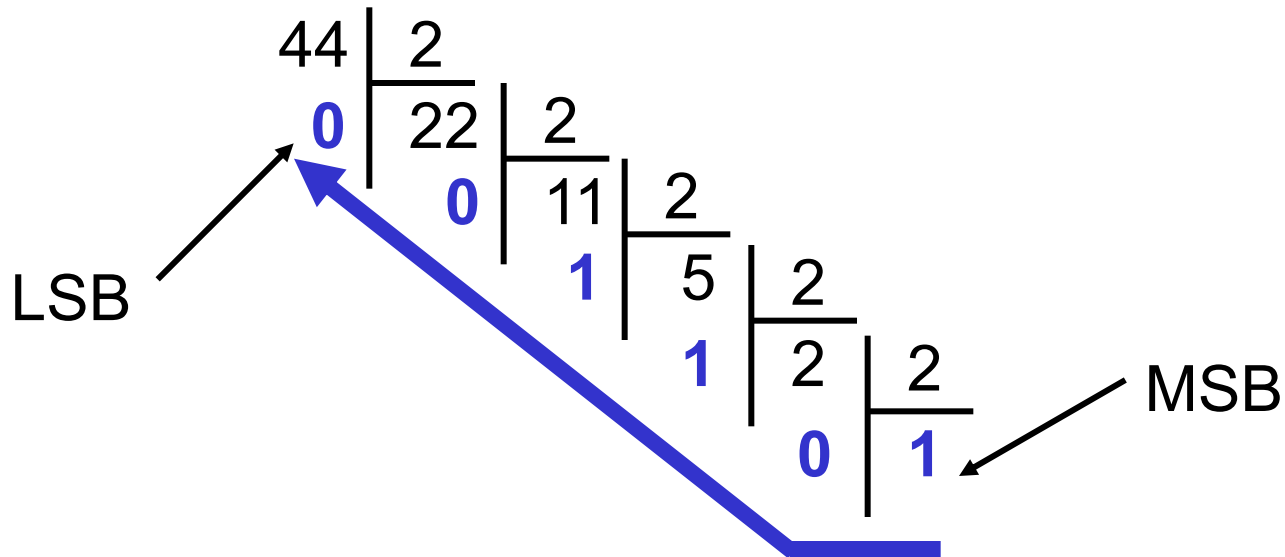
$$4 = 2^2 \quad \rightarrow \quad 44 = 2^5 + 2^3 + 2^2$$

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

$$(44)_{10} = (1x2^5) + (0x2^4) + (1x2^3) + (1x2^2) + (0x2^1) + (0x2^0)$$

$$= (101100)_2$$

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 :

$$\begin{aligned} & (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) \\ & = 32 + 0 + 8 + 4 + 0 + 0 \\ & = (44)_{10} \end{aligned}$$

2.4 / Suite de 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

Façon plus commode d'écrire "en binaire"

Avec un seul digit hexa on peut représenter 4 digits binaires

Exemples :

$$(A3F)_{16} = (\underbrace{1010}_{A} \underbrace{0011}_{3} \underbrace{1111}_{F})_2$$

$$(6B4)_{16} = (\underbrace{0110}_{6} \underbrace{1011}_{B} \underbrace{0100}_{4})_2$$

Pour résumer

- Nombre max. de nombres "codables" sur n bits en base 2 : 2^n
 - Exemple : sur 3 bits, on code $2^3 = 8$ nombres : de 000 à 111
 - Sur 4 bits, on code $2^4 = 16$ nombres : de 0000 à 1111
- Plus grand nombre que l'on peut écrire : $N_{\max} = 2^n - 1$
 - Exemple : sur 3 bits : plus grand nombre $2^3 - 1 = 7$
 - sur 4 bits : plus grand nombre $N_{\max} = 2^4 - 1 = 15$
- Généralisation en base B : $N_{\max} = B^n - 1$
 - Par exemple, en base 5, sur 2 digits, on peut coder $5^2 = 25$ nombres :
de 0 à $(5^2 - 1) = 0$ à 24
de $(00)_5$ à $(44)_5$: 00, 01, 02, 03, 04, 10, 11, 12, 13, 14, 20, 21, ..., 43, 44

Exercice : conversion en binaire d'un grand nombre

324689

Technique des paquets de puissance de 2

$$\begin{aligned}324689 &= 2^{18} + 62545 \\ &= 2^{18} + 2^{15} + 29777 \\ &= 2^{18} + 2^{15} + 2^{14} + 13393 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 5201 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 1105 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 81 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^6 + 17 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^6 + 2^4 + 1 \\ &= 2^{18} + 2^{15} + 2^{14} + 2^{13} + 2^{12} + 2^{10} + 2^6 + 2^4 + 2^0\end{aligned}$$

Technique divisions par 2 :
nécessiterait 20 divisions
successives → risque erreur

$$324689 = (0100\ 1111\ 0100\ 0101\ 0001)_2$$

Exercice : conversion en binaire d'un grand nombre

324689

Passage par la conversion hexadécimale

$$\begin{aligned} 324689 &= 4 \cdot 16^4 + 62545 \\ &= 4 \cdot 16^4 + 15 \cdot 16^3 + 1105 \\ &= 4 \cdot 16^4 + 15 \cdot 16^3 + 4 \cdot 16^2 + 81 \\ &= 4 \cdot 16^4 + 15 \cdot 16^3 + 4 \cdot 16^2 + 5 \cdot 16^1 + 1 \cdot 16^0 \\ &= (4F451)_{16} \end{aligned}$$

C'est la méthode la plus efficace !


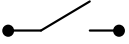
$$324689 = (4F451)_{16}$$

$$\text{Vérification : } (0100\ 1111\ 0100\ 0101\ 0001)_2 = (4F451)_{16}$$

3 / Notions d'algèbre de Boole

3.1 / États logiques

Plusieurs éléments technologiques sont conçus pour ne présenter que **2 états de fonctionnement seulement** :

Élément	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

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 avec 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 :

a **Fausse** et b **Fausse** alors s **Fausse** (je ne vais pas au cinéma)
 a **Fausse** et b **Vraie** alors s **Vraie** (je vais au cinéma)
 a **Vraie** et b **Fausse** alors s **Vraie** (je vais au cinéma)
 a **Vraie** et b **Vraie** alors s **Vraie** (je vais au cinéma)

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

Table de Vérité (TV)

OU logique

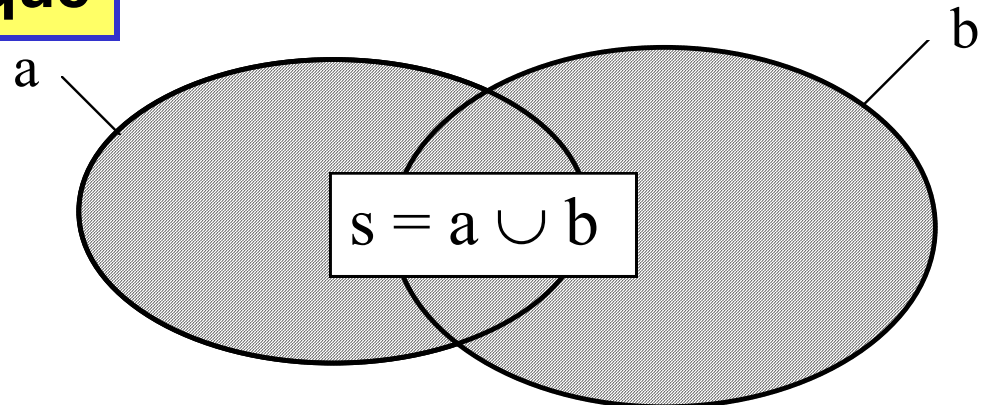


Diagramme de Venn

➤ Autre manière d'écrire 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 si 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 :

a est **Fausse** et b **Fausse** alors p **Fausse** (je ne fais pas de vélo)

a est **Fausse** et b **Vraie** alors p **Fausse** (je ne fais pas du vélo)

a est **Vraie** et b **Fausse** alors p **Fausse** (je ne fais pas du vélo)

a est **Vraie** et b **Vraie** alors p **Vraie** (je fais du vélo)

ET logique

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

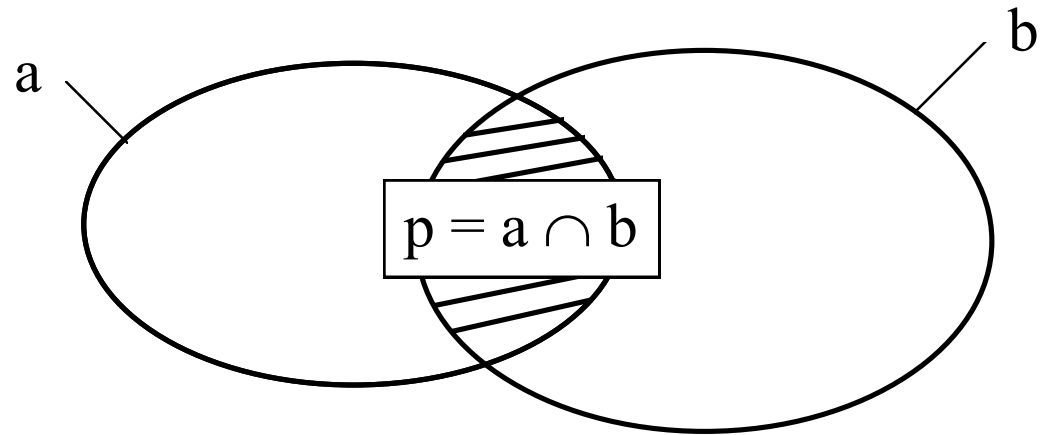


Table de Vérité (TV)

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 :

$$p = a . b = a b$$

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

3.2.3 / Négation ("INV", Inversion, Complémentation)

a	x
0	1
1	0

Table de Vérité (TV)

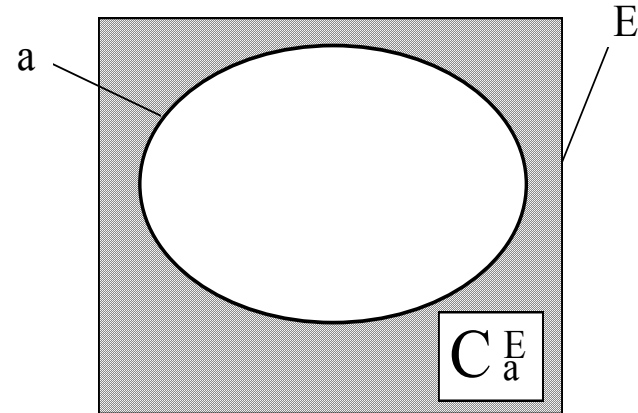


Diagramme de Venn

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

$$x = \bar{a}$$

- Autres notations : $x = C_a^E$

3.2.4 / Somme exclusive ("XOR", OU exclusif)

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

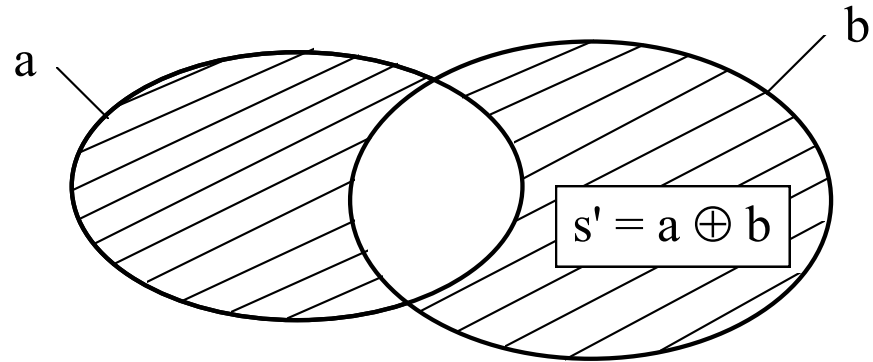
XOR

Table de Vérité (TV)

Diagramme de Venn

s' n'est égal à "1" que lorsque a ET b sont différents

➤ La notation de la fonction "Ou Exclusif" est la suivante :

$$s' = a \oplus b$$

➤ La fonction XOR 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 $a \oplus b = \bar{a}b + a\bar{b}$ (Vérifier à l'aide d'une TV)

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 : Écrire 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.

Vérification par TV de : $a \oplus b = \bar{a}b + a\bar{b}$

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

3.3 / Propriétés des fonctions booléennes

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

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	$x + 1 = 1$	$x \cdot 0 = 0$

Elles découlent directement de la définition des fonctions booléennes. On peut les retrouver aisément à partir des TV ou des diagrammes de Venn correspondants.

3.3.2 / Propriétés générales

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

Ces propriétés découlent directement de la définition des fonctions booléennes. On peut les retrouver 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.

Démonstration de la première propriété d'absorption :

$$x + xy = x.1 + x.y = x.(1 + y) = x.1 = x$$

Élément
neutre du ET

Distributivité
ET/ OU

Élément
absorbant
du OU

Élément
neutre du ET

Démonstration de la dernière propriété d'absorption :

$$x + \bar{x}y = x + x.y + \bar{x}.y = x + y(x + \bar{x}) = x + y.1 = x + y$$

Absorption 1
(ci-dessus)

Commutativité ET
+
Distributivité ET/ OU

Complémentarité
du OU

Élément
neutre
du ET

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



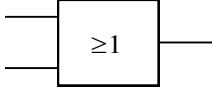
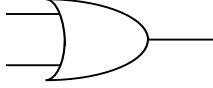

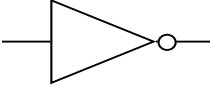
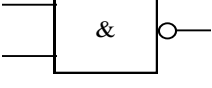

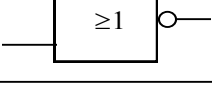

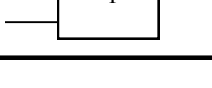

$$\overline{x + y} = \overline{x} \cdot \overline{y}$$

3.3.4 / Représentation symbolique des opérateurs logiques

Les fonctions logiques sont représentées par des symboles appelés "**portes logiques**" ou plus simplement "portes" ("gates" en anglais).

En électronique une porte représente un circuit à transistors capable d'implanter la fonction donnée.

La structure électronique des diverses portes logiques en technologie CMOS, sera donnée dans la partie suivante.

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

Les fonctions NOR et NAND sont très utilisées en électronique numérique compte tenu de leur plus grande facilité d'implantation.

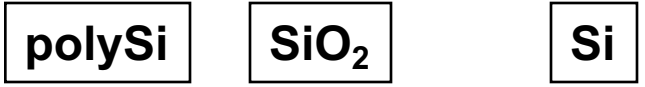
4 / Logique CMOS

A base de diodes et le transistors. Il existe principalement 4 technologies courantes pour la fabrication des circuits logiques :

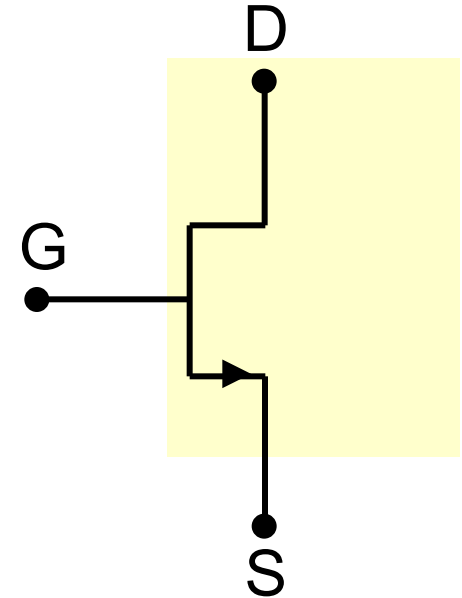
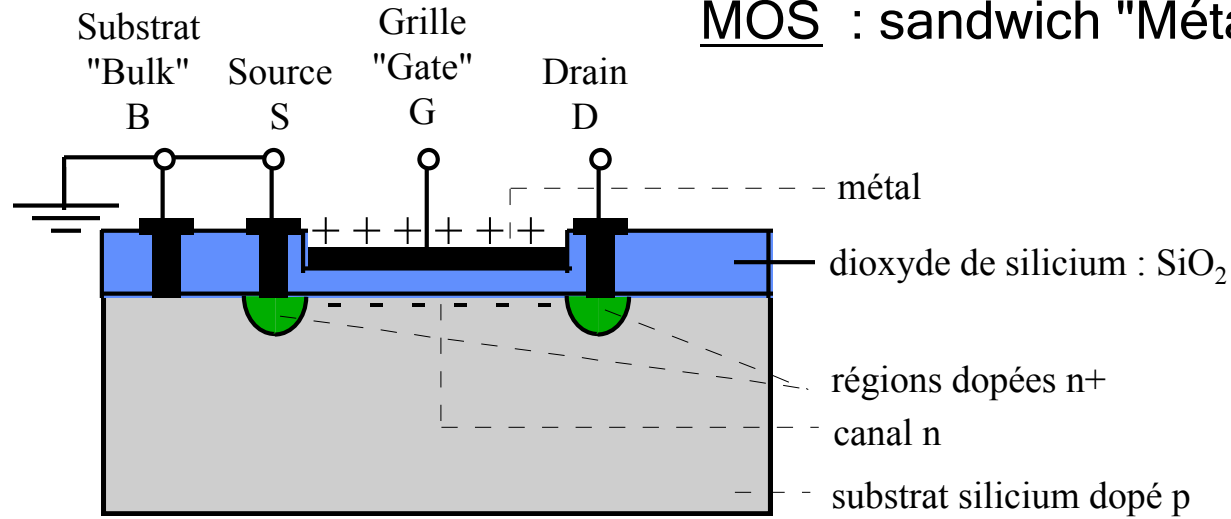
- **TTL** (Transistor-Transistor Logic) : technologie à transistors bipolaires, la plus ancienne
- **ECL** (Emitter-Coupled Logic) : technologie à transistors bipolaires, la plus rapide mais délicate à mettre en œuvre,
- **NMOS** : technologie n'utilisant que des transistors de type NMOS
- **CMOS** :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

4.1 / Transistor MOS en commutation

4.1.1 / Transistor NMOS



MOS : sandwich "Métal-Oxyde-Semiconducteur"



Dispositif microélectronique qui comporte 4 broches

- Drain (D)
- Grille [Gate] (G)
- Source (S)
- Substrat ["Bulk"] (B), ici B et S toujours reliés

Symbole NMOS
(quand S et B reliés)

Configuration : S et B reliés à la masse : $V_S = V_B = 0$

a) Grille G au potentiel nul ($V_{GS} = 0$), D et la S 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) $V_{GS} > V_T$ (ex : $V_T \approx 1V$) (V_T , tension de seuil ["threshold"]) :

- apport de charges (+) sur la grille,

- des charges (-) sont induites dans zone sous oxyde grille (effet condensateur), créant un "canal n" conducteur entre S et D,

- le transistor est **passant ["ON"]**.

Remarque 1 :

État passant : \exists 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 considèrerons 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 : $V_D = V_S$.

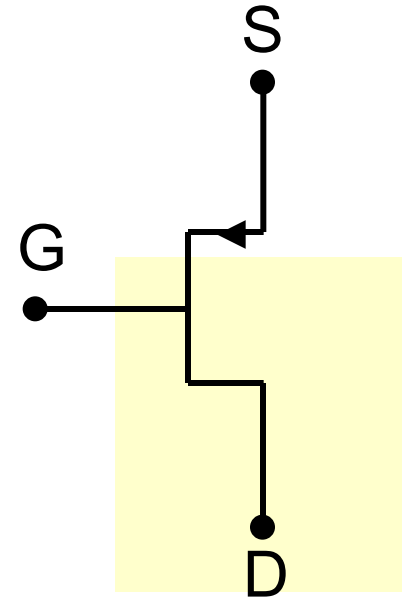
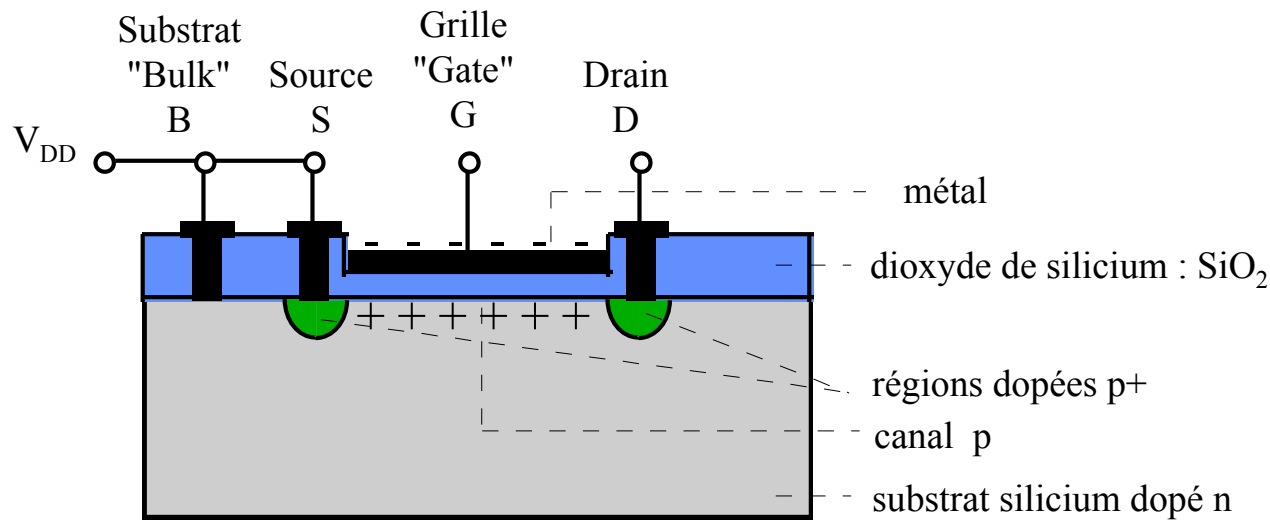
Remarque 2 :

Logique positive à deux niveaux, on choisit :

- Niveau logique "0" \equiv potentiel nul $V=0$
- Niveau logique "1" \equiv tension d'alimentation circuit $V=V_{DD}$

[En général, $V_{DD}=5V$ 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 en dessous du Volt].

4.1.2 / Transistor PMOS



Symbole PMOS
(quand S et B reliés)

Configuration : S et B reliés à la tension alim. : $V_S = V_B = V_{DD}$

a) Grille G au potentiel V_{DD} ($V_{GS} = 0$), D et la S sont isolés l'un de l'autre par la zone dopée n (charges négatives).

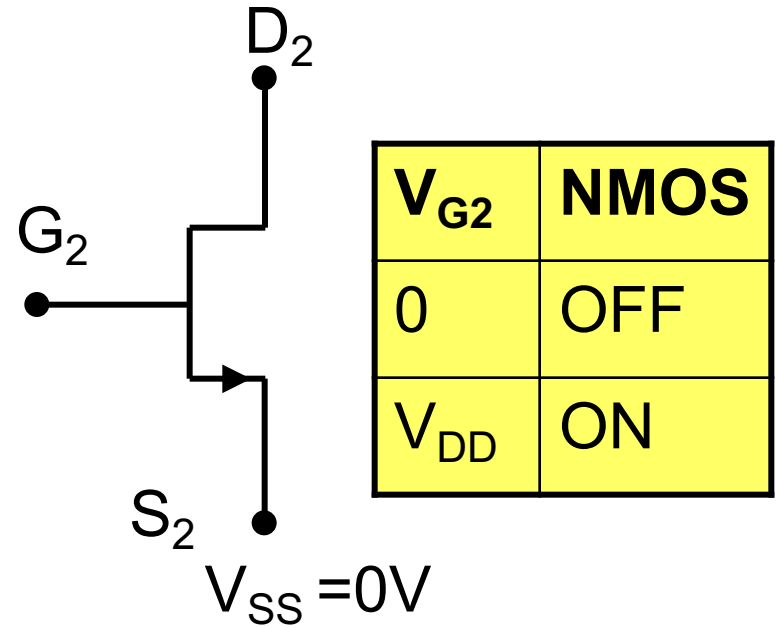
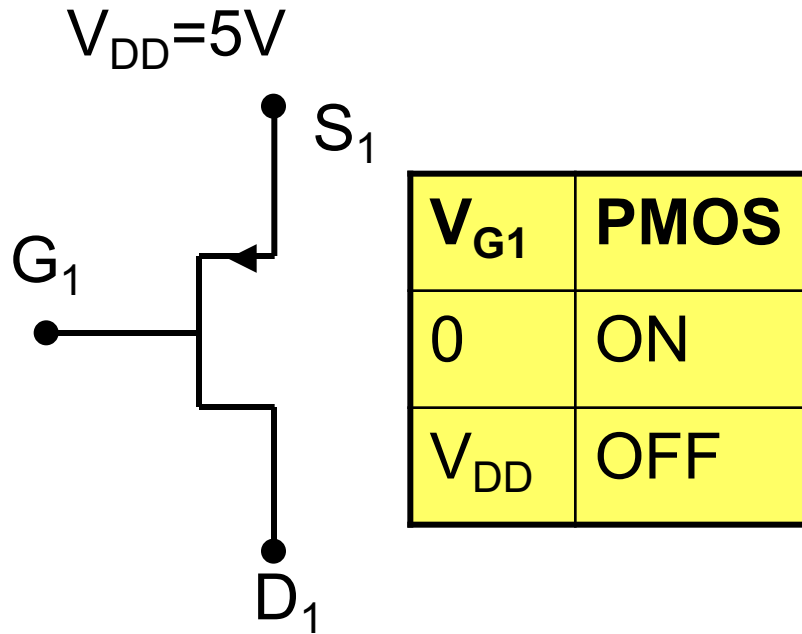
Le transistor ne peut pas conduire.

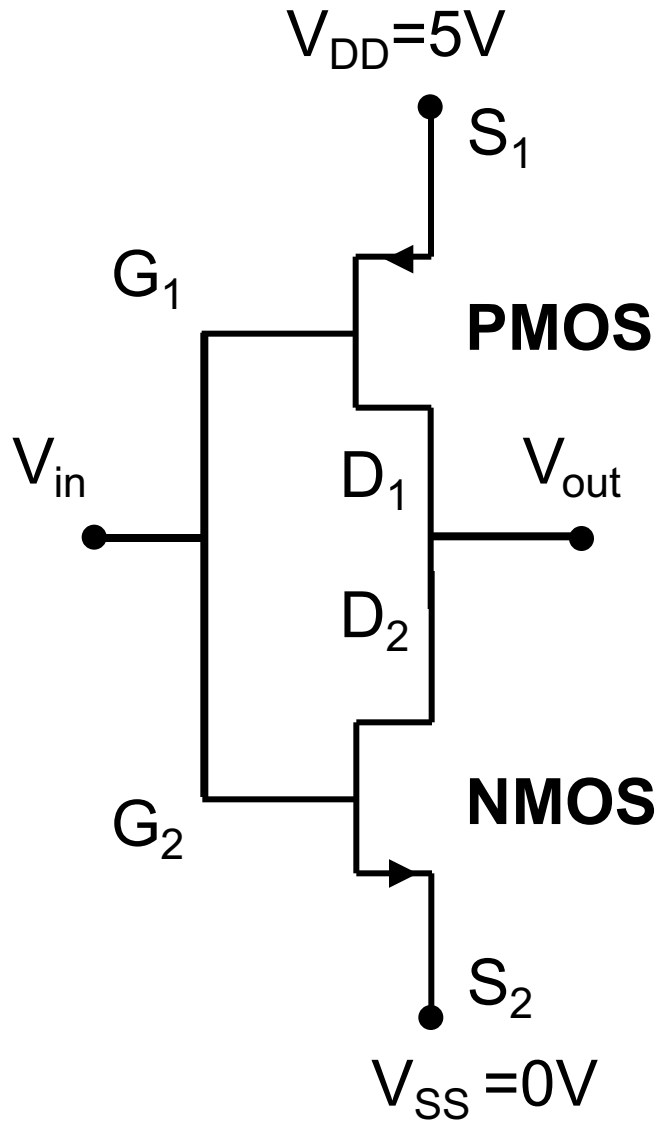
On dit que le transistor est dans son état **bloqué ["OFF"]**.

b) $V_{GS} < 0$ et $|V_{GS}| > |V_T|$, avec $V_T \approx -1V$, (V_T , tension de seuil ["threshold"]) :

- apport de charges (-) sur la grille,
- des charges (+) sont induites dans zone sous oxyde grille (effet condensateur), créant un "canal p" conducteur entre S et D,
- le transistor est **passant ["ON"]**.

4.2 / Inverseur CMOS

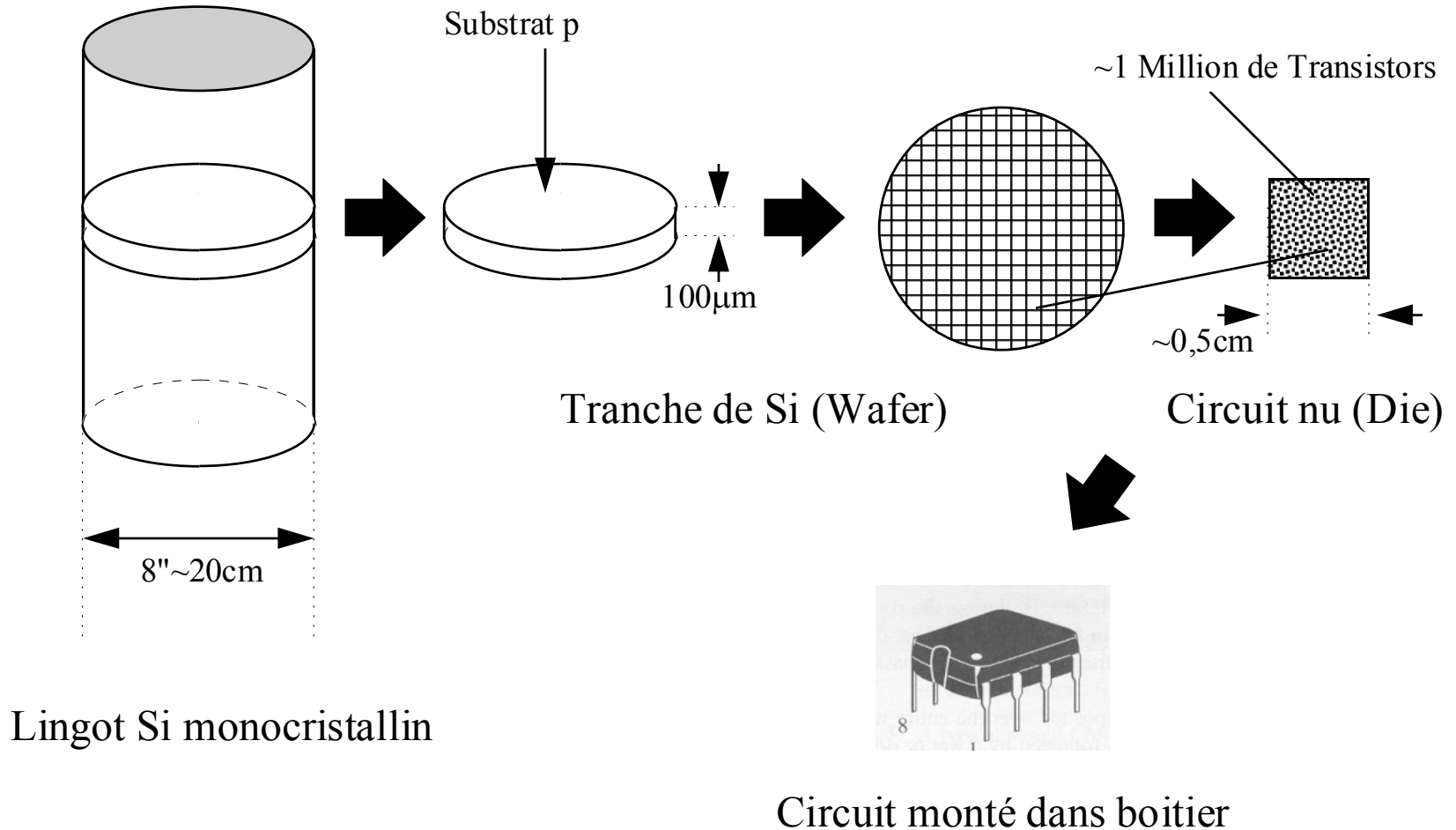




En associant en cascade un transistor PMOS et un transistor NMOS on obtient la fonction d'inversion logique

V_{in}	PMOS	NMOS	V_{out}
0	ON	OFF	V_{DD}
V_{DD}	OFF	ON	0

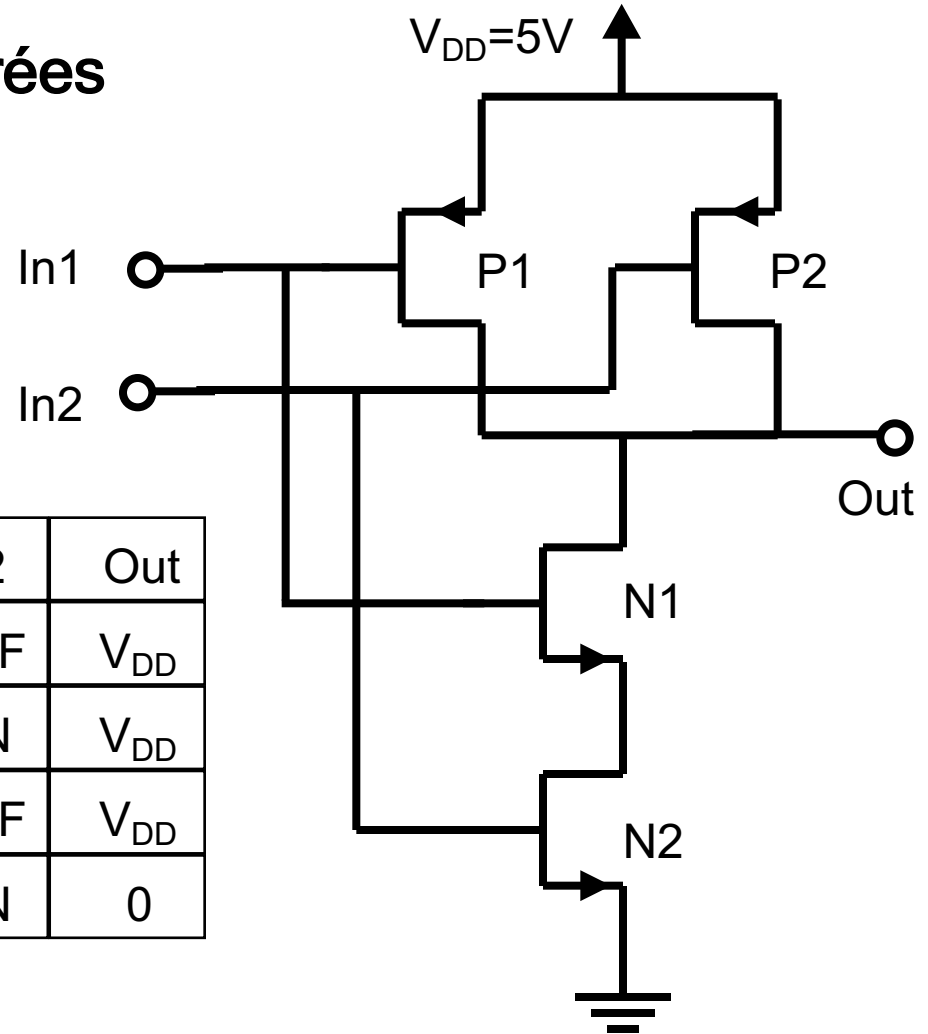
4.3 / Fabrication des circuits intégrés CMOS



4.4 / Portes logiques CMOS

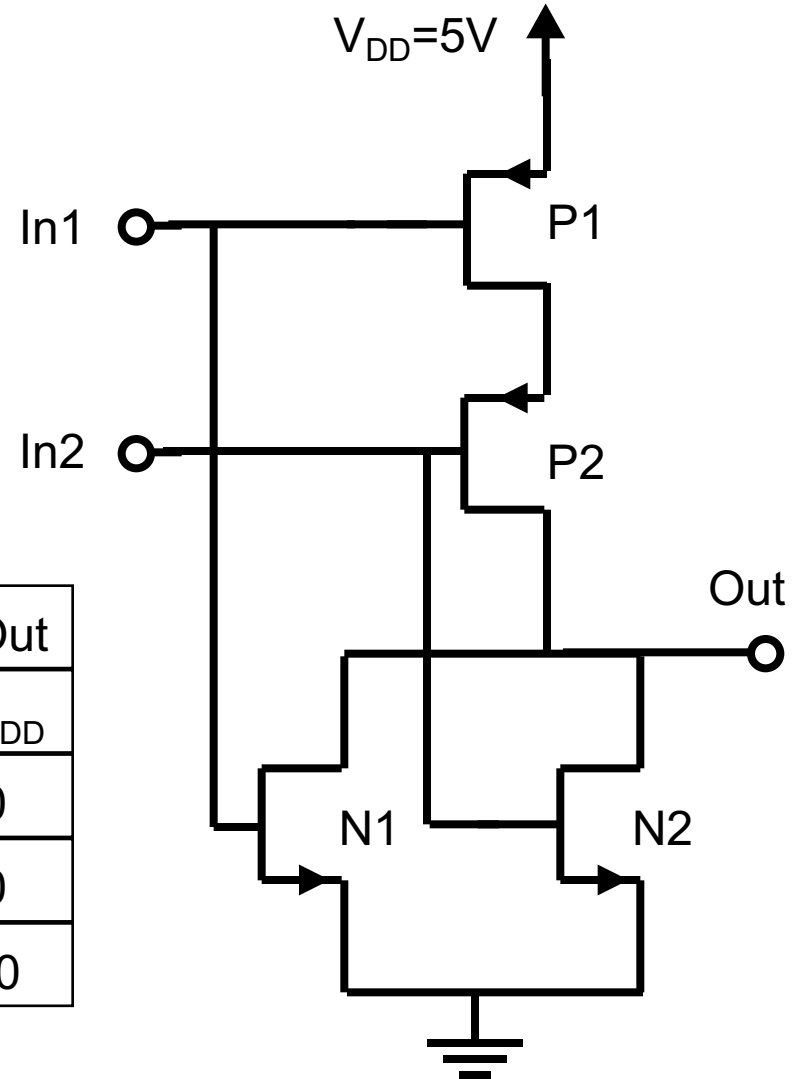
4.4.1 /Porte NAND à deux entrées

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



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



Exercice d'application de cours :

- concevoir une porte NAND à trois entrées.
- concevoir une porte NOR à trois entrées.

5 / Représentation des fonctions logiques

5.1 / Forme algébrique

∃ deux formes algébriques :

- **forme normale disjonctive** : somme de produits
- **forme normale conjonctive** : produit de sommes

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

5.1.2 / Forme normale disjonctive ou « somme de produits » ($\Sigma \Pi_i$)

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

■ La somme aura autant de termes que de "1" dans TV.

Pour que la somme soit égale à "1", il suffit qu'un seul terme de cette 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 faut que **chacun** de ses facteurs soit égal à "1"*

d'où la technique :

→ si variable à "1" => on la garde telle quelle
si variable à "0" => on la complémente

Exemple :

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

Définition de la parité P d'un ensemble de bits :

■ si $\sum \text{"bits"} = 1 \pmod{2}$, alors $P = 1$.

La fonction est impaire (il y a un nombre impair de "1")

■ si $\sum \text{"bits"} = 0 \pmod{2}$, $P = 0$.

La fonction est paire (il y a un nombre pair de "1").

Parité P d'un ensemble de 3 bits

	a	b	c	P	/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 :

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

5.1.3 / Forme normale conjonctive ou « produit de sommes » ($\Pi \Sigma_i$)

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

- Le produit aura autant de facteurs qu'il y a de "0" dans TV :

$$P_0 = S_0 . S_3 . S_5 . S_6$$

- 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 :

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

5.1.4 / Choix d'une forme

On choisit celle qui conduit au plus petit nombre de termes (dans l'exemple parité, 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)

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

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

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

5.2 / Tables de Karnaugh (2^n cases)

Table de Karnaugh = 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 doit changer

Si n variables $\rightarrow 2^n$ cases

5.2.1 / Table de Karnaugh à 2 variables

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

a\b	0	1
0	0	1
1	1	0

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

5.2.2 / Table de Karnaugh à 3 variables

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

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

Dans cet exemple, la fonction est irréductible (cf. ci-après).

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

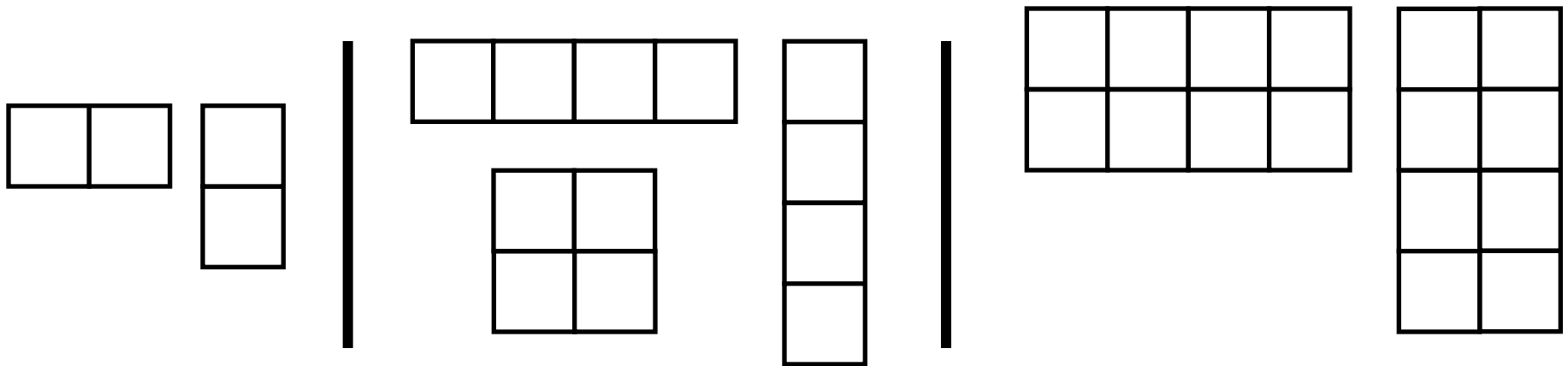
5.2.3 / Table de Karnaugh à 4 variables

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

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

5.3 / Simplification de 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 regroupement d'ordre supérieur, on élimine la variable qui **change dans l'adjacence**.

Exemple 1 : blocs d'ordre 2

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

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

$$\bar{a}\bar{b}\bar{c}$$

$$abd$$

$$S = abd + \bar{a}\bar{b}\bar{c}$$

Exemple 2 : blocs d'ordre 4

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

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

$\overline{a}\overline{b}$

$$S = \overline{a}\overline{b}$$

Exemple de la table de Karnaugh à 4 variables précédente

$$f = \bar{a}\bar{b}\bar{c}d + \bar{a}bcd + \bar{a}bc\bar{d} + abcd + abc\bar{d} + a\bar{b}\bar{c}d$$

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

$\bar{b}\bar{c}d$
 bc

$f = bc + \bar{b}\bar{c}d$

5 / Application des fonctions logiques

5.1 / Exemple 1 : comparaison entre deux mots logiques

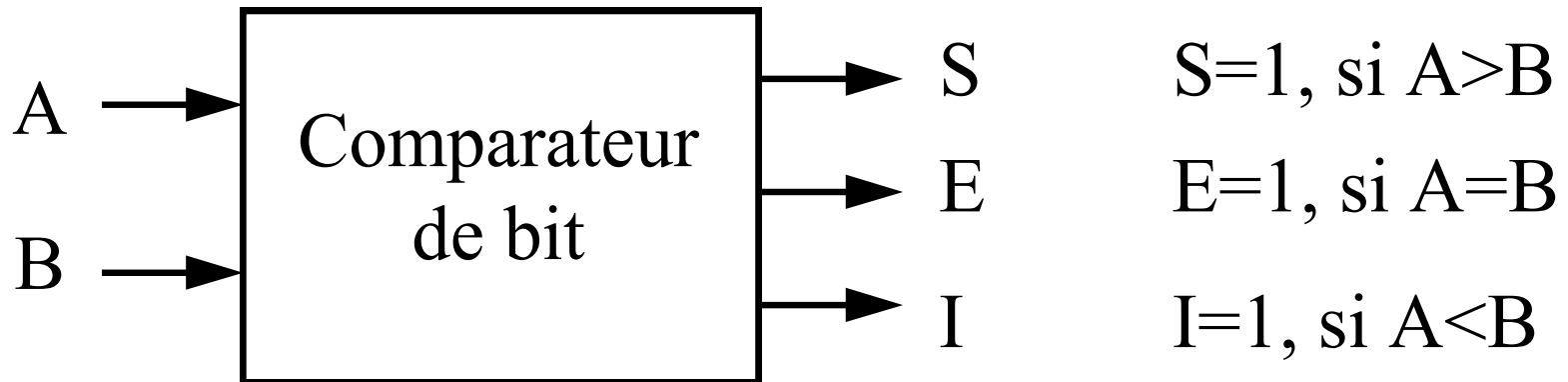


Table de Vérité pour A et B sur 1 bit :

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

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

