

Fascicule 5
CALCUL DIGITAL
CIRCUITS LOGIQUES
formation

adret
ae
electronique®

Fascicule 5
CALCUL DIGITAL
CIRCUITS LOGIQUES
formation

F A S C I C U L E 5

1^{ère} PARTIE

ELEMENTS DE CALCUL DIGITAL

Introduction	page	2
Systèmes de numération		4
Système binaire		5
CONVERSION		
1) Décimal-Binaire		
2) Binaire-Décimal		9
OPERATIONS DANS LE SYSTEME BINAIRE		10
Addition - soustraction		
Résultats négatifs - Digit de signe		11
Complément vrai		12
Complément restreint		13
Multiplication - Division		14
LES CODES		15
Codes pondérés		
Codes non pondérés		
NOTIONS D'ENSEMBLE		20
Egalité de deux ensembles		
Sous ensembles - relation d'inclusion		
Complément d'un ensemble		
Union et intersection d'ensembles		
Ensemble vide - ensemble universel		
Recouvrement d'un ensemble E		
Diagrammes de VENN		
ALGEBRE DE BOOLE		23
Principe - but - conventions		
Axiomatique de l'algèbre de Boole		25
Relations fondamentales		26
Théorèmes de DE MORGAN		28
Résumé des axiomes et théorèmes		29
FONCTIONS BOOLEENNES		30
Définition		
Fonctions logiques fondamentales		31
Fonctions ET OU NON		
Fonctions logiques secondaires		33
Fonctions ET OU		
Tableau des correspondances de fonctions ..		34
Fonction OU EXCLUSIF		35

Fonction identité - somme modulo 2		36
Fonctions particulières		37
Majorité - Inhibition - Mémoire		
FORMES CANONIQUES DES FONCTIONS BOOLEENNES		41
Somme de produits - Produit de sommes		
Simplification des formes canoniques		42
Regroupement des termes et mise en facteurs communs ..		
Méthodes graphiques		44
Diagrammes de KARNAUGH		44
Diagrammes de VEITCH		46
Représentation des formes incomplètes		47
Utilisation du diagramme		48
Règles et types de groupement		
Lecture et minimisation		
Disposition du diagramme		51
Redondances ou cas indifférents		52

2^{ème} PARTIE LES CIRCUITS LOGIQUES

LES CIRCUITS COMBINATOIRES		53
Définition		
Construction d'ensembles de logique combinatoire		
CIRCUITS ARITHMETIQUES		56
Additionneur binaire		
Additionneur série et parallèle		58
Soustracteur binaire		59
Additionneur - soustracteur		61
COMPARATEUR BINAIRE		62
Comparaison de deux poids binaires		
Comparateur d'amplitude binaire à entrées multiples ...		63
LES CONVERTISSEURS DE CODES - LE TRANSCODAGE		65
Codage - décodage		
Codeur décimal/DCB		66
Décodeur DCB/décimal		67
Décodage BCD/7 segments		68
TRANSCODEURS		
Conversion GRAY/BINAIRE		70

Introduction

Il est probable que le premier système de numération fut le système concret de marques, d'encoches ou de représentation par des cailloux ou par des noeuds : par exemple, pour chaque objet à compter, on peut tracer un trait vertical. Pour additionner ou pour soustraire, il suffit d'ajouter ou de barrer des traits.

Ce système est facile à concevoir, mais n'est guère pratique dès qu'il s'agit de représenter des nombres un peu élevés. On peut l'améliorer en groupant les traits, ou encore en plaçant un cinquième trait diagonal barrant chaque groupe de quatre traits, de façon à symboliser une collection de cinq objets.

Comme on le voit en figure 1, de tels groupements facilitent beaucoup la lecture de l'ensemble. Une technique plus évoluée a été imaginée par les Romains et par les Egyptiens, qui remplacent les groupes de traits par des lettres, ce qui leur permet d'écrire de façon raisonnablement claire des nombres très élevés.

Malgré toutes les astuces qui ont pu être proposées, les opérations arithmétiques sont très difficiles avec les chiffres romains, sauf dans les cas les plus élémentaires.

Le progrès le plus important, dans ce domaine, date de l'introduction de ce que l'on appelle les chiffres arabes. Ce système a commencé à être connu en Occident vers le XIIe siècle et a permis un véritable bond en avant de la science européenne.

Malgré leur nom de chiffres arabes, ces chiffres ont été d'abord mis au point par des mathématiciens indiens, aux alentours du Ve ou du VIe siècle de notre ère. Leurs formes actuelles semblent d'ailleurs dériver des initiales des mots hindous correspondants. Les arabes ont appris des Indiens ce système de numération et al-Khawarizmi en a donné une description complète au IXe siècle. Le mot d'algorithme est une corruption du nom de ce mathématicien arabe, ce système ayant été introduit en Occident par des traductions d'ouvrages arabes. Il n'a été bien connu que vers le XIIIe siècle, à la suite des travaux de Leonardo Fibonacci, et, en fait, les chiffres romains sont restés d'un usage courant jusqu'au milieu du XVIe siècle.

Les deux grands mérites de ce système sont, d'une part, qu'il possède un symbole pour zéro et, d'autre part, qu'il utilise la notation par position, c'est à dire que n'importe quel nombre peut être représenté en ordonnant des chiffres, la position de chaque chiffre déterminant sa valeur. Il est intéressant de remarquer que, tout comme ses prédécesseurs, le système hindou utilise comme base un multiple de 5, ce qui provient probablement du fait que nous avons cinq doigts à chaque main, et que l'humanité a compté sur ses doigts avant la mise au point de toute notation.



Fig. 1

Le système de numération le plus couramment utilisé, à l'heure actuelle, dérive des systèmes hindous et arabes, et utilise les symboles 0,1,2,3,4,5,6,7,8,9. On l'appelle le système décimal, parce qu'il utilise dix symboles (en latin, dix se dit decem). Le nombre 10 est la base de ce système, et le rang de chaque chiffre, dans un nombre, correspond à une puissance donnée de 10.

Mis à part le fait qu'il représente n'importe quel nombre de façon claire et simple, le gros avantage d'un système de numération comme le système décimal vient de ce qu'il facilite énormément les opérations mathématiques. De plus, il les rend suffisamment automatiques dans leur principe pour qu'elles puissent être effectuées avec des machines à calculer mécaniques. Les premières machines -et même les premiers ordinateurs électroniques comme l'ENIAC- utilisaient le système décimal.

Le comptage est l'opération arithmétique fondamentale, de laquelle dérivent toutes les autres. Les premières machines utilisaient les mêmes techniques de comptage que nous ; c'est pourquoi il est intéressant de se demander comment nous procédons en réalité.

Dans le système décimal, nous partons de 0 et nous comptons 1,2 jusqu'à 9. Pour continuer, nous revenons à 0, en retenant 1, que nous plaçons à gauche de 0, c'est à dire dans la colonne des dizaines. Après 19, nous retenons à nouveau 1, obtenant ainsi 20. Nous arrivons ainsi à 99. Pour le nombre suivant, on ramène les unités à 0, on retient 1 dans la colonne des dizaines : on revient également ainsi à 0 dans la colonne des dizaines et le chiffre 1 apparaît dans la colonne des centaines. C'est ainsi que l'on obtient le nombre 100. La notion de retenue est ainsi absolument fondamentale dans notre système de numération.

Cette technique de comptage est très facile à transposer mécaniquement : on obtient ainsi la première machine à calculer, qui est le boulier ou abaque. La figure 2 représente un boulier décimal simple. On représente un chiffre en déplaçant des boules le long d'une tige horizontale, jusqu'à obtenir dans la partie droite un nombre de boules égal au chiffre en question. Chaque tige (rangée) correspond à une puissance de 10 ou à une position en notation arabe. La tige du haut correspond aux unités, la tige suivante aux dizaines, et ainsi de suite. On s'est donc contenté de faire tourner de 90° la notation décimale écrite (pour contrebalancer les forces de gravité). Le boulier de la figure 2 affiche le nombre 4103. Pour compter ou pour additionner sur un boulier, il suffit de déplacer les boules de la première rangée de la gauche vers la droite jusqu'à ce que toutes les boules aient été utilisées. On repousse alors toutes les boules vers la gauche et on déplace vers la droite la première boule de la rangée suivante : ces opérations sont exactement équivalentes à notre système de retenue.

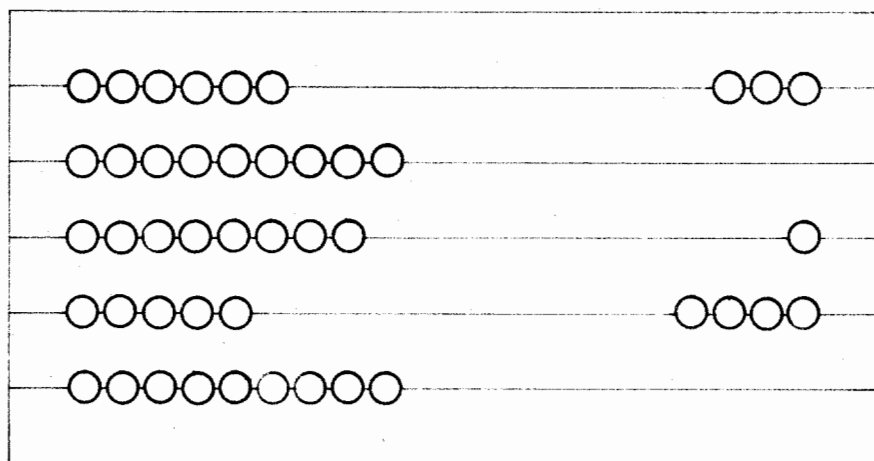


Fig. 2

Bien que le système décimal permette de représenter n'importe quel nombre, les calculateurs mécaniques ont une capacité finie déterminée par leurs dimensions verticales (nombre de rangées). La capacité maximale d'un compteur s'appelle son "module", comme d'ailleurs la capacité en mots machine d'un ordinateur. Par exemple, le boulier de la figure 2 peut représenter n'importe quel nombre compris entre 0 et 99 999, c'est un compteur modulo 100 000.

Les différentes bases.

Nous avons considéré jusqu'à présent le système de numération de base 10, auquel nous sommes tellement habitués qu'il nous semble difficile d'en imaginer un autre. Il n'y a néanmoins aucune raison théorique pour ne pas utiliser des systèmes de numération ayant une autre base, ou même plusieurs bases. Dans l'histoire, la plupart des systèmes ont été décimaux, mais de nombreux autres systèmes ont été utilisés. Certains systèmes anciens ont utilisé les bases 5 et 6, la civilisation maya utilisait le système de base 20 et les Chinois du XI^e siècle ont mis au point un système à base double, 2 et 5, destiné à être utilisé avec le boulier. Les ordinateurs modernes utilisent le système de base 2 (système binaire).

SYSTEMES DE NUMERATION

Un nombre est représenté par une suite de chiffres, ces chiffres peuvent prendre un certain nombre de valeurs différentes $0, 1, 2, \dots, b-1$ dans un système de numération à base b . Aucun des chiffres qui composent le nombre ne peut être supérieur à la base.

$$a_i < b$$

On appelle ces chiffres des "DIGITS".

La position de chaque chiffre représente une puissance entière de la base.

Un nombre N pourra par exemple s'écrire en base b de la façon suivante : *

$$N_{(b)} = a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_0 b^0 + a_{-1} \cdot b^{-1} + \dots + a_{-m} \cdot b^{-m}$$

Dans le système décimal ($b = 10$) nous aurons par exemple :

$$N_{(10)} = 137,42$$

$$\text{ou } N_{(10)} = 1 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0 + 4 \cdot 10^{-1} + 2 \cdot 10^{-2}$$

On appelle poids d'un digit (a_i) son terme multiplicatif b^i . On parlera ainsi de poids faibles suivant le rang occupé par le chiffre considéré dans le nombre. Dans l'exemple

précédent,	le poids de	1	est	10^2
	"	3	"	10^1
	"	7	"	10^0
	"	4	"	10^{-1}
	"	2	"	10^{-2}

- les poids peuvent être positifs ou négatifs.

Outre le système décimal (base 10) et le système binaire (base 2), il est possible de compter dans une infinité de bases différentes ;

exemples : système octal de base 8 (intéressant, l'octet = 8 étant une puissance de 2)
duodécimal de base 12
hexadécimal de base 16 (lui aussi puissance de 2)
sexagésimal de base 60.

Il est intéressant de remarquer que quelque soit la base employée, la somme du produit des chiffres par leurs poids respectifs donne le nombre en base 10.

* Pour une base b donnée il n'existe qu'une seule décomposition possible du nombre N .

SYSTEME BINAIRE

La nature nous fournit un grand nombre d'éléments physiques présentant deux états d'équilibre stable (donc distincts).

Ce sont tous ceux qui possèdent, soit une permanence d'état, telle que la perforation en modulation mécanique, soit un cycle d'hystérésis, telle que l'induction magnétique des matériaux ferromagnétiques. Mais la structure à deux positions est bien plus générale puisque, par construction, il suffira de fixer, conventionnellement, deux valeurs distinctes du domaine d'évolution d'une quelconque grandeur physique ou mécanique. C'est ainsi que nous emploierons, le plus souvent, l'absence et la présence d'un signal électrique, lumineux, etc.... C'est la forme caractéristique des systèmes impulsionnels. On utilisera de tels dispositifs pour représenter l'information. L'élément technique ayant 2 positions, le langage sera du type binaire.

L'utilisation de la base 2 se déduit d'études mathématiques :

étant donné un nombre N à n chiffres, il est possible de former b^n nombres. En terme informatique ce nombre b^n définit la "puissance lexicographique" d'un langage considéré ou nombre de mots possibles de format n en base b .

Exemple : $b = 10$ $n = 3$, on peut obtenir 1000 nombres ou mots différents.

On montra alors en théorie de l'information que le nombre de circuits nécessaires au traitement d'un signal quantifié, c'est à dire discontinu suivant les amplitudes, est minimum si le produit nb est minimum, à quantité d'information $Q = n \log b = Cte$.

(de $Q = n \log b$ on tire $n = \frac{Q}{\log b}$ remplaçant n dans le second membre de (1)).

Or $nb = \frac{Q}{\log b} b$ (1) est minimum pour $b = e \approx 2,718$, e étant la constante d'Euler

Ainsi les systèmes binaires et ternaires sont les plus aptes au traitement de l'information. Pour des raisons pratiques, c'est le système binaire qui est le plus employé.

Ce système n'est utilisé pratiquement que dans les calculateurs numériques.

Il est en effet beaucoup plus simple et plus sûr de n'utiliser que 2 états stables, correspondants à 2 valeurs 0 et 1. Par combinaison de tels éléments, on pourra représenter n'importe quelle grandeur.

La base étant de valeur faible, un grand nombre de digits ou bits (contraction de Binary digits) sera nécessaire à l'expression des données.

La longueur du nombre ou "mot" étant fixé à n , on déduit la puissance lexicographique en base 2, soit : 2^n .

CONVERSION

1) DECIMAL - BINAIRE

Il existe plusieurs méthodes de conversion.

- La méthode du tableau.
- La méthode par soustraction.
- La méthode par division et multiplication.
- La méthode globale.

Nous étudierons successivement chacune de ces méthodes.

Nota : Pour une meilleure compréhension de cet exposé, la notation des nombres binaires sera faite comme en notation décimale, de gauche à droite ; les poids les plus forts à gauche ;

A) METHODE DU TABLEAU

C'est la méthode la plus simple, elle consiste à établir un tableau de correspondance entre le nombre décimal et son équivalent binaire. Cette méthode donne directement la correspondance bilatérale mais ne peut s'appliquer qu'à des nombres de faible valeur.

Exemple : pour les nombres de 0 à 11

DECIMAL		BINAIRE			
$a_1 10^1$	$a_0 10^0$	2^3	2^2	2^1	2^0
	0	0	0	0	0
$1 \times 1 = 1$	1	0	0	0	$1 = 2^0 = 1$
$2 \times 1 = 2$	2	0	0	1	$0 = 2^1 = 2$
$3 \times 1 = 3$	3	0	0	1	$1 = 2^0 + 2^1 = 1 + 2 = 3$
$4 \times 1 = 4$	4	0	1	0	$0 = 2^2 = 4$
$5 \times 1 = 5$	5	0	1	0	$1 = 2^2 + 2^0 = 4 + 1 = 5$
$6 \times 1 = 6$	6	0	1	1	$0 = 2^2 + 2^1 = 4 + 2 = 6$
$7 \times 1 = 7$	7	0	1	1	$1 = 2^2 + 2^1 + 2^0 = 4 + 2 = 7$
$8 \times 1 = 8$	8	1	0	0	$0 = 2^3 = 8$
$9 \times 1 = 9$	9	1	0	0	$1 = 2^3 + 2^0 = 8 + 1 = 9$
$(1 \times 10) + 0 = 10$	1 0	1	0	1	$0 = 2^3 + 2^1 = 8 + 2 = 10$
$(1 \times 10) + (1 \times 1) = 11$	1 1	1	0	1	$1 = 2^3 + 2^1 + 2^0 = 8 + 2 + 1 = 11$

B) METHODE PAR SOUSTRACTION

D'un nombre décimal N on peut obtenir l'équivalent binaire par soustractions successives, au nombre décimal, des puissances de 2 de valeurs les plus approchantes de N puis N' , N'' , etc....

Un nombre décimal $N_{(10)}$ étant toujours compris entre deux puissances de 2 consécutives soit :

$$2^p \geq N_{(10)} \geq 2^{p-1}$$

on effectue $N_{(10)} - 2^{p-1} = N'_{(10)}$

puis $N'_{(10)} - 2^{q-1} = N''_{(10)}$

$N''_{(10)} - \text{etc} \dots$

d'où $N_{(10)} = 2^{p-1} + 2^{q-1} + \dots + 2^j$

Nous voyons donc qu'en divisant par les puissances croissantes successives de 2 nous obtenons les différents digits comme restes des divisions et dans l'ordre poids faibles $\rightarrow$ poids forts.

Remarquons au passage que 2^0 , le bit le moins significatif, donne le caractère de parité du nombre soit :

$$2^0 = 1, N \text{ est impair}$$

$$2^0 = 0, N \text{ est pair.}$$

Partie fractionnaire : Chaque digit du nombre en base 2 est donnée par les parties entières successives des produits successifs du nombre décimal par la base 2.

Exemple : $N_{(10)} = 0,372$

$0,372 \times 2 = 0,744$	$2^{-1} = 0$	(parties entières de 0,744)
$0,744 \times 2 = 1,488$	$2^{-2} = 1$	(partie entière de 1,488)
$0,488 \times 2 = 0,976$	$2^{-3} = 0$	(partie entière de 0,976)
$0,976 \times 2 = 1,952$	$2^{-4} = 1$	(partie entière de 1,952)
$0,952 \times 2 = 1,904$	$2^{-5} = 1$	" " etc....

d'où $N_{(2)} = 0,01011$

Nous voyons donc qu'en multipliant par les puissances croissantes successives de 2 nous obtenons les différents digits comme parties entières des multiplications et dans l'ordre poids forts $\rightarrow$ poids faibles.

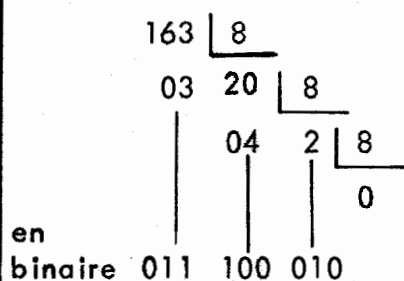
Remarque : un nombre fractionnaire fini dans une certaine base, peut ne pas s'exprimer exactement dans une autre base, c'est le cas du nombre 0,372 de l'exemple précédent.

D) METHODE GLOBALE

Cette méthode pourtant rapide est peu utilisée, elle regroupe 2 méthodes déjà décrites, la méthode du tableau et la méthode par division et multiplication. La procédure reste identique à la méthode division/multiplication.

Partie entière : On procède par divisions successives de $N_{(10)}$ par une puissance quelconque de 2. Le reste étant exprimé par $n - 1$ si la puissance de 2 utilisée comme diviseur est n .
Il suffit ensuite de traduire les restes successifs des divisions (en base 10) en binaire à l'aide du tableau de correspondance. Il sera d'autant plus incommode de traduire ces restes si n est grand. Généralement, on prend comme diviseur $2^3 = 8$ car jusqu'à ce chiffre, la correspondance binaire-décimale est facile à retenir. Si $n = 3$, on aura des restes successifs de 3 bits.

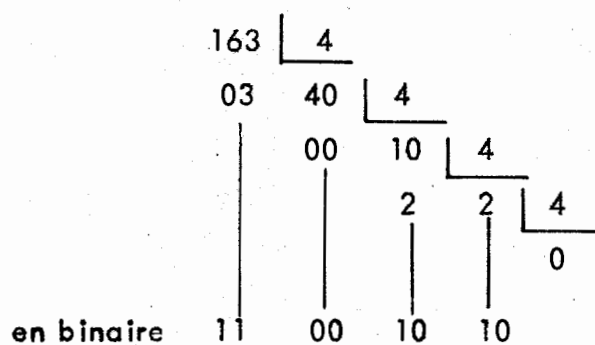
1er exemple : $N_{(10)} = 163$



Les restes des divisions successives étant donnés, là aussi dans le sens poids faibles $\rightarrow$ poids forts, il convient pour exprimer $N_{(2)}$ d'inverser les sens des groupes de digits sans modifier les groupes eux-mêmes.

d'où $N_{(2)} = (010)(100)(011)$

2è. exemple : $n = 3$ les groupes de digits auront $n-1 = 2$ digits.



soit $N_{(2)} = (10)(10)(00)(11)$

Partie fractionnaire : La procédure est identique à celle du paragraphe précédent.

Exemple : $N_{(10)} = 0,372$

$$0,372 \times 8 = 2,976$$

$$2 = 010$$

$$0,976 \times 8 = 7,808$$

$$7 = 111 \text{ etc...}$$

d'où $N_{(2)} = 0,010.111$

2) BINAIRE - DECIMAL

La conversion Binaire-décimale est aisée et ne nécessite pas de méthodes complexes.

Citons :

- Méthode du tableau, déjà citée dans le paragraphe précédent.
- Méthode des poids.

On affecte à chaque digit son poids

Exemple : $N_{(2)} = 101101,01$

$$2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0, \ 2^{-1} \ 2^{-2}$$

$$1 \ 0 \ 1 \ 1 \ 0 \ 1, \ 0 \ 1$$

$$N_{(10)} = 1.2^5 + 0.2^4 + 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 + 0.2^{-1} + 1.2^{-2}$$

$$N_{(10)} = 32 + 0 + 8 + 4 + 0 + 1 + 0 + 0,25$$

$$N_{(10)} = 45,25$$

OPERATIONS DANS LE SYSTEME BINAIRE

1) ADDITION

Il suffit d'établir un tableau comparatif binaire-décimal, la table d'addition s'en déduit immédiatement.

Opérations	Résultats	
	Décimal	Binaire
0 + 0	0	0
0 + 1	1	1
1 + 0	1	1
1 + 1	2	10

D'où on tire
la table

+	0	1
0	0	1
1	1	10

digit de report
(ou retenue)

Exemples :

$$\begin{array}{r} 6 \\ + 9 \\ \hline 15 \end{array} \quad \begin{array}{l} \text{donne en} \\ \text{binaire} \end{array} \quad \begin{array}{r} 0110 \\ + 1001 \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 3 \\ + 10 \\ \hline 13 \end{array} \quad \begin{array}{l} \text{donne en} \\ \text{binaire} \end{array} \quad \begin{array}{r} 1 \text{ report} \\ 0011 \\ + 1010 \\ \hline 1101 \end{array}$$

2) SOUSTRACTION

Nous procédons de façon similaire à une soustraction en décimal. Lorsque dans la soustraction, le chiffre à soustraire se trouve supérieur, on porte en retenue le complément à la base (2). Cette retenue s'ajoutera au chiffre suivant (*).

Opérations	Résultats			
	Décimal R		Binaire R'	
0 - 0		0		0
1 - 0		1		1
0 - 1	1	9	1	1
1 - 1		0		0

D'où on tire
la table

-	0	1
0	0	1
1	1	0

et retenue de 1

R' = retenue ou complément à 2

$$\begin{array}{r} 11 \\ - 5 \\ \hline 6 \end{array} \quad \begin{array}{r} R' \\ 1011 \\ - 0101 \\ \hline 0110 \end{array}$$

$$\begin{array}{r} 9 \\ - 6 \\ \hline 3 \end{array} \quad \begin{array}{r} R' \quad R' \\ 1001 \\ - 0111 \\ \hline 0011 \end{array}$$

$$\begin{array}{r} 10 \\ - 3 \\ \hline 07 \end{array} \quad \begin{array}{r} R' \quad R' \quad R' \\ 1010 \\ - 0101 \\ \hline 0111 \end{array}$$

$$\begin{array}{r} 8 \\ - 7 \\ \hline 1 \end{array} \quad \begin{array}{r} R' \quad R' \quad R' \\ 1000 \\ - 0111 \\ \hline 0001 \end{array}$$

(*) Le résultat sera égal à 1 sachant qu'en binaire $10-1 = 2-1 = 1 + \text{retenue de } 1$.

Résultats négatifs

Un problème se pose quand la soustraction donne un résultat négatif. Ce problème revient à celui de l'expression des nombres négatifs dans un calculateur.

En appliquant les règles précédentes, nous aurions par exemple :

$$\begin{array}{r}
 \begin{array}{cccc} & R & & \\ 11 & 1 & 0 & 1 & 1 \\ -14 & -\textcircled{1}1\textcircled{4} & 1 & 1 & 0 \\ \hline -3 & \textcircled{4}1 & 1 & 0 & 1 \end{array} & \begin{array}{r} 8 \\ -15 \\ \hline -7 \end{array} & \begin{array}{cccc} 1 & 0 & 0 & 0 \\ -\textcircled{1}1\textcircled{4} & 1\textcircled{4} & 1\textcircled{4} & 1\textcircled{4} & 1 \\ \hline \textcircled{4}1 & 0 & 0 & 1 \end{array}
 \end{array}$$

Cette dernière retenue indique le signe négatif du résultat, celui-ci étant alors le complément (à 10000 en base 2) du nombre à obtenir. Il en est de même en décimal car en appliquant les règles de l'arithmétique nous aurions par exemple :

$$\begin{array}{r}
 11 \\
 \textcircled{4}1\textcircled{4} \\
 \hline
 \textcircled{1}97
 \end{array}$$

Ce qui veut dire : moins le complément (à 100 en base 10) de 97 soit $(-) 100 - 97 = -3$

Une autre soustraction sera donc nécessaire, celle du complément, trouvé comme résultat à la base en conservant la dernière retenue comme "Digit de signe" (voir

$$\begin{array}{r}
 10000 \\
 -\textcircled{1}1\textcircled{4}1\textcircled{4}0\textcircled{4}1 \\
 \hline
 \textcircled{1}00011 = -3
 \end{array}$$

$$\begin{array}{r}
 10000 \\
 -\textcircled{1}1\textcircled{4}1\textcircled{4}0\textcircled{4}1 \\
 \hline
 \textcircled{1}00111 = -7
 \end{array}$$

DIGIT DE SIGNE

Le signe ne pouvant avoir que 2 états différents + ou -, il constitue une variable binaire. On peut donc le représenter par exemple :

+ par 0 (en binaire)
 " 0 (en décimal)
 - par 1 (en binaire)
 " 9 (en décimal)

0.58 signifiant + 58
 9.58 signifiant - 58

1.111 signifiant - 7
 0.111 signifiant + 7

Cette méthode de soustraction en binaire est la méthode la plus proche de celle utilisée en décimal. En pratique, en particulier dans les calculateurs, on utilise une autre représentation des nombres à soustraire de façon à ramener la soustraction à une addition. On utilise les valeurs complémentées des nombres.

- complément vrai
- complément restreint.

Sans faire d'opérations sur les compléments (inutiles dans cet exposé) il est bon de les définir.

COMPLEMENT VRAI (OU COMPLEMENT ARITHMETIQUE)

Soit un nombre composé de n chiffres, la plus grande valeur que l'on pourra représenter sera :

$$10^n - 1 = \underbrace{999 \dots 9}_{n \text{ fois}} \quad \text{en décimal}$$

$$2^n - 1 = \underbrace{111 \dots 1}_{n \text{ fois}} \quad \text{en binaire}$$

Si on limite les nombres à n chiffres, 10^n , ne sera représenté que par n zéros consécutifs.

On appelle complément à 10 d'un nombre décimal $N_{(10)}$ exprimé avec n digits le nombre $N_{(10)}^*$ tel que :

$$N_{(10)}^* = 10^n - N_{(10)}$$

par exemple, le complément à 10 de 0.68 (+ 68) est

$$N_{(10)}^* = 1000 - 0.68 = 932$$

On appelle complément à 2 d'un nombre binaire $N_{(2)}$ exprimé avec n bits, le nombre

$$N_{(2)}^* = 2^n - N_{(2)}$$

par exemple, le complément à 2 de 0.1101 est

$$N_{(2)}^* = 10000 - 0.1101 = 0.0011$$

On représente ainsi les nombres négatifs par leur complément en faisant précéder le "Mot" par son digit de signe ; exemple :

$$\begin{array}{rcl} \text{en binaire} & + & 13 \quad 0.1101 = N \\ & - & 13 \quad 1.0011 = N^* \end{array}$$

en effet :

$$10000 - 0.1101 = (1) 0011 = N^*$$

└─ digit de signe

On utilise alors N^* pour effectuer les opérations de soustractions par le calculateur de façon techniquement plus simple (la soustraction se ramenant à une addition). Il existe une autre méthode encore plus simple sur le plan de l'utilisation technique employée couramment dans les calculateurs, il s'agit du complément restreint.

COMPLEMENT RESTREINT

Particulièrement intéressant en binaire car il suffit de remplacer les zéros par des 1 et les 1 par des zéros.

Le complément restreint d'un nombre N à n digits est exprimé par :

$$N_{(9)}^{**} = (10^n - 1) - N_{(10)} \quad \text{ou complément à 9 en décimal.}$$

$$N_{(1)}^{**} = (2^n - 1) - N_{(2)} \quad \text{ou complément à 1 en binaire.}$$

On peut remarquer immédiatement que :

$$N^* = N^{**} + 1 \quad (1) \quad \text{car} \quad \left[(2^n - 1) - N_{(2)} \right] + 1 = 2^n - N_{(2)} = N^*$$

Exemple : en décimal, le complément restreint de 68 sera :

$$N_{(9)}^{**} = (1000 - 1) - 068 = 999 - 068 = 931$$

en binaire, le complément restreint de $\begin{array}{cccc} 1 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \end{array}$ sera :

$$N_{(1)}^{**} = (10000 - 1) - 1101 = 1111 - 1101 = \begin{array}{cccc} 0 & 0 & 1 & 0 \end{array}$$

Voyons qu'elles sont les modifications dans l'opération soustraction.

1^a) $a > |J|$ b étant la base de numération

$$\text{on forme } J^{**} = (b^n - 1) - J \quad (3)$$

$$\text{puis } a + J^{**} = a + (b^n - 1) - J$$

$$a + J^{**} = (a - J) + b^n - 1 \geq b^n \quad (2)$$

Nous obtiendrons donc un nombre de $n+1$ digits, alors que nos nombres sont limités à n digits. Il y aura dépassement. Ce dépassement s'écrit toujours à l'aide d'un 1 que l'on ajoute au résultat de l'opération ;

On obtient :

$$a + J^{**} + 1 = (a - J) + b^n \quad \text{de (2)}$$

$$J^* = J^{**} + 1 \quad \text{d'où } a + (J^{**} + 1) = (J^*) + a \quad \text{de (1)}$$

$$= a - J \quad \text{car } a > |J| \quad (\text{complément vrai})$$

Exemple : en décimal :

$$a = 17 \quad J = 5$$

$$\begin{array}{r} a \rightarrow 017 \\ J^{**} \rightarrow + 994 \\ \hline \textcircled{1} 011 \\ \hline \textcircled{1} + \textcircled{1} \\ \hline 012 \end{array}$$

$$\text{d'où } a - J = +12$$

en binaire :

$$\begin{array}{l} a = 17 = 10001 \\ J = 5 = 00101 \end{array}$$

$$\begin{array}{r} a = 0.10001 \\ J^{**} = +1.11010 \\ \hline \textcircled{1} 001011 \\ \hline \textcircled{1} + \textcircled{1} \\ \hline 0.01100 = +12 = a + J^{**} \end{array}$$

Digit de signe

2^a) $a < |J|$ b étant la base

On forme $J^{**} = (b^n - 1) - J$

puis $a + J^{**} = a + (b^n - 1) - J$

on opère alors sur la valeur absolue de $a - J$

soit $a + J^{**} = b^n - 1 - |a - J|$ or nous savons (3) que $|a - J|^{**} = b^n - 1 - |a - J|$

d'où $a + J^{**} = |a - J|^{**}$

On obtient le complément restreint de la valeur absolue de la différence. En binaire, il suffira d'inverser les zéros et les 1 pour obtenir le résultat.

On voit ainsi que l'utilisation du complément restreint dans les opérations de soustraction, simplifie considérablement les opérations techniques dans le calculateur, puisqu'elles se réduisent à 2 opérations élémentaires, inversion des bits et addition.

MULTIPLICATION

En binaire, la multiplication s'effectue de la même façon qu'en décimal.

opérations	résultats	
	Décimal	Binaire
0×0	0	0
0×1	0	0
1×0	0	0
1×1	1	1

d'où on tire
la table

x	0	1
0	0	0
1	0	1

Exemple :

$$\begin{array}{r}
 7 \\
 \times 5 \\
 \hline
 35
 \end{array}
 \qquad
 \begin{array}{r}
 111 \\
 \times 101 \\
 \hline
 111 \\
 111 \\
 \hline
 100011 = 35
 \end{array}$$

Dans les calculateurs, la multiplication se ramène à une suite d'additions et de décalages successifs.

DIVISION

De même que la multiplication, en binaire, la division s'effectue de la même façon qu'en décimal et revient à une suite de soustractions et de décalages successifs.

Exemple :

$$\begin{array}{r}
 12 \overline{) 4} \\
 \underline{0 3} \\
 00
 \end{array}
 \qquad
 \begin{array}{r}
 1100 \overline{) 100} \\
 \underline{0100} \\
 00
 \end{array}$$

LA MESURE ET LE CODAGE

La mesure d'une grandeur peut être obtenue selon deux méthodes.

- La méthode analogique.

Cette méthode fournit le résultat sous une forme continue transformant la grandeur en une autre plus directement perceptible à nos sens et qui lui est liée d'une manière simple.

- La méthode numérique.

Permettant d'obtenir directement le résultat sous forme quantifiée ; c'est une méthode discontinue qui fournit des résultats immédiatement exploitables pour les calculs, mais présente un inconvénient car la quantification conduit à une perte de l'information (d'autant plus faible que la résolution est grande).

Le passage de la représentation analogique à la représentation numérique, s'effectue par l'intermédiaire de convertisseurs analogiques-numériques.

Les appareils utilisés à cet effet sont appelés des "codeurs".

Les transformations inverses s'effectuent par des décodeurs. Enfin, le passage d'un code à un autre code est fait à l'aide de transcodeurs.

LES CODES

Définition dans le système numérique.

Un code est une combinaison permettant, dans un système de numération donné, de représenter par un nombre : un phénomène physique, l'état d'un système, une instruction machine, etc....

L'intérêt du codage binaire réside dans le fait qu'il permet l'utilisation des systèmes du type "tout ou rien".

Il existe deux grandes catégories de codes, suivant qu'il est possible ou non d'attribuer à chacun des digits du code une signification :

- Lorsque les digits ont une signification, "un poids", on dit que le code est "Pondéré".
- A l'inverse, chaque digit du code n'a pas de poids significatif, le code est dit "Non Pondéré", ou code sans poids.

CODES PONDERES

a) Code binaire naturel (CBN)

Les poids sont représentés par les puissances successives de 2.

b) Code décimal codé binaire (DCB)

Les états utilisés sont les 10 premiers états du code binaire naturel (0 à 9). Les combinaisons 10 à 15 étant interdites. Il faudra donc 4 bits pour représenter ce code puisque $2^3 < 9 < 2^4$. On l'appelle aussi code 8.4.2.1.

c) Code 2421 - Voir tableau -

d) Code Auto-complémenté.

Un code est auto-complémenté lorsqu'on peut obtenir à partir d'un chiffre décimal A le chiffre 9-A en inversant systématiquement les bits constituant le nombre A.

Les codes auto-complémentaires sont précieux dans les calculateurs pour effectuer les soustractions par addition de la forme complémentaire.

C B N					D C B				2 4 2 1				2 4 2 1 auto-complémenté ou code Aiken			
2^3	2^2	2^1	2^0		8	4	2	1	2	4	2	1	2	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0
5	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1
6	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0	0
7	0	1	1	1	0	1	1	1	0	1	1	1	1	1	0	1
8	1	0	0	0	1	0	0	0	1	1	1	0	1	1	1	0
9	1	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0												
etc																

Exemple :
 $A = 2$
 $9 - A = 7$
 7 dans ce code est le complément restreint de 2.
 axe de symétrie

CODES NON PONDERES (OU CODES SANS POIDS)

Dans ces codes, chaque digit considéré individuellement, n'a aucune signification particulière, c'est le groupement des digits qui est significatif.

a) CODE LIPPEL

Ce code est intéressant pour coder des déplacements.

Sa particularité est d'avoir 5 digits "1" consécutifs dans 3 colonnes.

b) CODE "exès de 3"

On l'appelle ainsi parce que son état zéro correspond au chiffre 3 du CBN. C'est un code non pondéré auto-complémenté. Il a l'avantage de toujours avoir au moins un digit "1" par état.

c) CODES REFLECHIS

Ils constituent la majorité des codes sans poids.

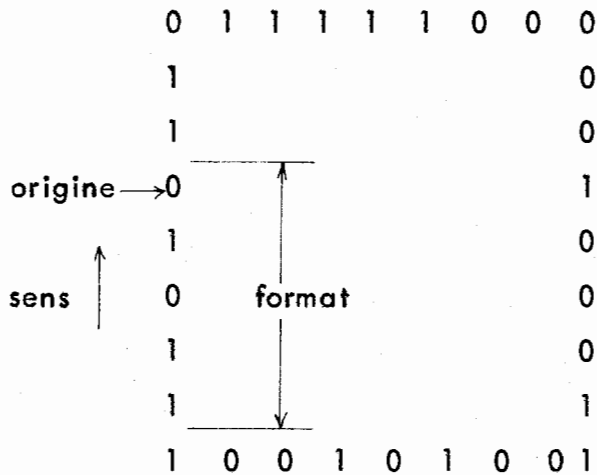
Ils sont caractérisés par le fait que deux niveaux consécutifs ne diffèrent que par un seul digit. Deux niveaux situés à égale distance d'un certain axe de symétrie sont représentés par des combinaisons identiques des digits des ordres inférieurs.

	LIPPEL				EXCES DE 3				CODE GRAY				
0	0	0	0	0	0	0	1	1	0	0	0	0	A
1	0	0	0	1	0	1	0	0	0	0	0	1	
2	0	0	1	0	0	1	0	1	0	0	1	1	
3	0	0	1	1	0	1	1	0	0	0	1	0	
4	0	1	1	0	0	1	1	1	0	1	1	0	axe de symétrie A/B
5	1	1	1	1	1	0	0	0	0	1	1	1	B
6	1	1	1	0	1	0	0	1	0	1	0	1	
7	1	1	0	1	1	0	1	0	0	1	0	0	
8	1	1	0	0	1	0	1	1	1	1	0	0	axe de symétrie AB/CD
9	1	0	0	1	1	1	0	0	1	1	0	1	C
10									1	1	1	1	
11									1	1	1	0	
12									1	0	1	0	axe de symétrie C/D
13									1	0	1	1	D
14									1	0	0	1	
15									1	0	0	0	

d) CODES RECURRENENTS

Ils sont obtenus en écrivant une suite de 2^n digits 0 ou 1 et en considérant les 2^n groupements de n digits successifs qu'il est possible de former lorsqu'on avance d'un pas dans un sens ou dans l'autre.

Exemple : Suite récurrente à 32 termes (2^5).



On peut écrire par exemple les 32 combinaisons codées à 5 bits comme suit :

0 = 0 1 0 1 1

1 = 1 0 1 1 1

2 = 0 1 1 1 0

3 = 1 1 1 0 0

4 = 1 1 0 0 1

5 = 0 0 1 0 1

etc...

L'intérêt des codes récurrents est la possibilité d'assurer la transmission des informations avec tout le secret désirable grâce à des changements d'origine de la suite ou du sens du parcours.

e) CODES AUTODETECTEURS D'ERREURS

Ces codes permettent de détecter certaines erreurs affectant leur transmission.

Exemples :

Code 2 parmi 5 : Chaque combinaison de 5 Digits comprend 2 Digits "1"

Code					
0	0	0	0	1	1
1	0	0	1	0	1
2	0	0	1	1	0
3	0	1	0	0	1
4	0	1	0	1	0
5	0	1	1	0	0
6	1	0	0	0	1
7	1	0	0	1	0
8	1	0	1	0	0
9	1	1	0	0	0

Contrôle de parité : On ajoute au message "information" un digit de parité tel que le nombre total de "1" du message complet soit, par exemple, toujours impair.

Par exemple, le code DCB donnera :

DCB					Parité	Message complet
0	0	0	0	0	1	0 0 0 0 1
1	0	0	0	1	0	0 0 0 1 0
2	0	0	1	0	0	0 0 1 0 0
3	0	0	1	1	1	0 0 1 1 1
4	0	1	0	0	0	0 1 0 0 0
5	0	1	0	1	1	
6	0	1	1	0	1	
7	0	1	1	1	0	etc
8	1	0	0	0	0	
9	1	0	0	1	1	
10	1	0	1	0	1	

A la réception, la détection d'un nombre pair de 1 indiquera une erreur.

NOTION D'ENSEMBLE

La notion d'ensemble est une notion première que l'on peut seulement suggérer par des exemples : ainsi on parlera de l'ensemble des Français, de l'ensemble des nombres entiers, de l'ensemble des polynômes, de l'ensemble des séquences binaires de longueur n , etc... Les êtres qui font partie de l'ensemble sont appelés éléments, membres, ou points. Pour écrire qu'un élément x appartient à un ensemble A , on utilise la notation :

$$x \in A$$

(lire : "x appartient à A") et, plus généralement, pour n éléments :

$$x_1, x_2, \dots, x_n \in A$$

On dit aussi que A contient x (ou $x_1, x_2, \dots, x_n$). Inversement, étant donné un élément x n'appartenant pas à A , on écrira :

$$x \notin A$$

(lire : "x n'appartient pas à A").

Par exemple, si Z désigne l'ensemble des nombres entiers, on peut écrire :

$$\begin{array}{l} 3 \in Z \\ 2/5 \notin Z \end{array}$$

Un ensemble est formé de points ayant au moins une propriété commune.

EGALITE DE DEUX ENSEMBLES

On dit que 2 ensembles A et B sont égaux (ou identiques) s'ils sont formés des mêmes éléments, et on écrit alors :

$$A = B$$

En d'autres termes, tout élément x appartenant à A appartient à B , et vice versa.

SOUS-ENSEMBLES. RELATION D'INCLUSION

On dit qu'un ensemble A est un sous-ensemble d'un ensemble B si tout élément de A appartient à B . On dit encore que A est inclus dans B , ou que A est une partie de B ou enfin que B contient A et on utilise les notations :

$$A \subset B$$

ou :

$$B \supset A$$

Exemples :

- L'ensemble des entiers pairs est un sous-ensemble de l'ensemble des entiers.
- L'ensemble des carrés, en géométrie, est un sous-ensemble de l'ensemble des rectangles.
- L'ensemble des ellipses est un sous-ensemble de l'ensemble des coniques (courbes du second ordre).

N.B. La définition donnée ci-dessus n'exclue pas le cas où l'on a $A = B$. Lorsqu'il y a des éléments qui appartiennent à B mais n'appartiennent pas à A , on dit parfois que A est un vrai sous-ensemble de B .

COMPLEMENT D'UN ENSEMBLE

Etant donné un ensemble A qui est un sous-ensemble d'un ensemble U , on appelle complément de A par rapport à U , l'ensemble des points de U qui n'appartiennent pas à A . On le note :

$$U - A \text{ ou } C_U(A)$$

Exemple : si U est l'ensemble des nombres réels et A celui des nombres rationnels, $U-A$ est l'ensemble des nombres irrationnels.

L'ensemble $U-A$ est un complément relatif. On peut, lorsqu'il n'y a pas d'ambiguïté sur l'ensemble U à considérer, définir un complément absolu, comme l'ensemble des points qui n'appartiennent pas à A . On le note fréquemment $\bar{A}$.

Notons que l'opération par laquelle à partir de A , on obtient $\bar{A}$, ou complémentation, est une opération à un terme (A), ou unaire. Nous allons maintenant définir des opérations à 2 termes (opérations binaires).

UNION ET INTERSECTION D'ENSEMBLES

Union

L'union (ou réunion) de deux ensembles A et B est l'ensemble des points qui appartiennent à A ou B ou aux deux à la fois.

L'union de A et B est notée : $A \cup B$ (se lit : "A union B").

Ainsi l'union de l'ensemble des points du segment $0 \leq x \leq 1$ et du segment $0,5 \leq x \leq 1,5$ est l'ensemble des points du segment $0 \leq x \leq 1,5$.

Intersection

L'intersection de deux ensembles A et B est l'ensemble des points qui appartiennent à la fois à A et à B . On la note $A \cap B$ (se lit : "A intersection B").

ENSEMBLE VIDE. ENSEMBLE UNIVERSEL

Il est commode d'introduire un ensemble ne possédant aucun élément, appelé ensemble vide, et noté $\emptyset$. Dans ces conditions, l'égalité

$$A \cap B = \emptyset$$

signifie que les ensembles A et B n'ont aucun point commun. On dit encore qu'ils sont disjoints.

De manière analogue, il est commode d'introduire un ensemble contenant tous les éléments possibles et appelé ensemble universel, ou univers (que l'on notera U , I , etc.).

RECOUVREMENT D'UN ENSEMBLE

Etant donné un ensemble E (appartenant à un ensemble universel U), on appelle recouvrement de E un ensemble de parties de E tel que E soit égal à leur réunion.

Exemple : l'ensemble des nombres entiers, des nombres rationnels, des nombres irrationnels et des nombres transcendants constitue un recouvrement de l'ensemble des nombres réels.

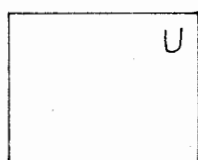
PARTITION D'UN ENSEMBLE E

C'est un recouvrement de E tels que deux ensembles quelconques de celui-ci soient disjoints. Tout point de E se trouve alors dans un ensemble de la partition et un seul.

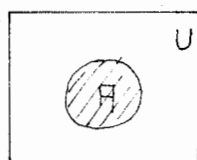
Ainsi, l'ensemble des nombres rationnels et des nombres irrationnels constitue une partition de l'ensemble des réels.

Diagrammes de VENN

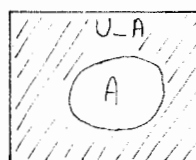
Ces différentes définitions peuvent recevoir une interprétation graphique au moyen de ce que l'on appelle des diagrammes de Venn. L'ensemble universel y est représenté par un rectangle et les ensembles A, B , etc... évoqués plus haut sont représentés par des cercles. Les figures suivantes donnent l'interprétation des différentes opérations définies plus haut (sur chaque figure, l'ensemble hachuré correspond à celui qui est donné en légende).



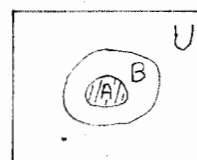
ensemble
universel



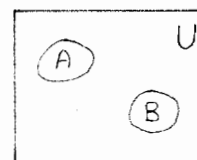
$A \in U$



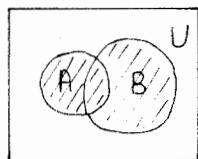
$U - A = C_U(A)$



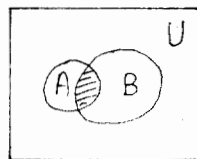
$A \subset B$
 $B \supset A$
 $B \in U$



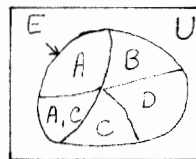
$A \in U$
 $B \in U$
 $B \notin A$



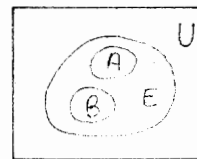
$A \cup B$
Union



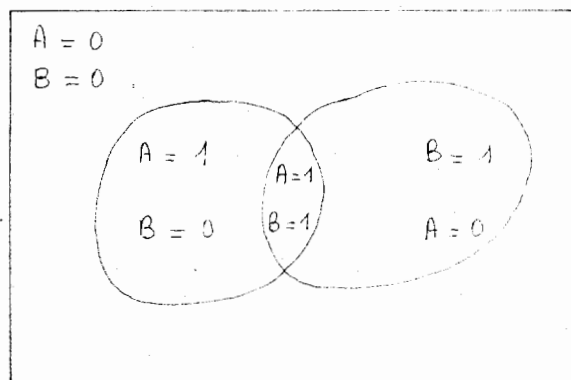
$A \cap B$
Intersection



$E \in U$
 $A, B, C, D \in E$
E est
recouvrement
de A, B, C, D .



$A \cap B = \emptyset$
A est disjoint de B
A et B sont partitions de E



Principe du diagramme de Venn.

ALGEBRE DE BOOLE OU ALGEBRE LOGIQUE

Introduction

L'algèbre logique, ou algèbre de Boole, doit son nom à un philosophe mathématicien anglais, Georges Boole, qui en 1854, fit une théorie mathématique de la logique d'Aristote. Cette théorie n'eut que des applications philosophiques jusqu'en 1938 environ où Shannon l'introduisit pour l'étude des réseaux électriques.

Principe

Une proposition est vraie ou fausse, mais ne peut pas être les deux à la fois. De même, un circuit est ouvert ou fermé, une lampe est allumée ou éteinte.

En d'autres termes, des grandeurs sont susceptibles de prendre deux états distincts dans des circonstances caractérisées par la position d'un certain nombre de variables susceptibles de prendre, chacune, également deux états. Par convention, on représente les états par les nombres binaires 0 et 1.

On voit immédiatement l'intérêt d'appliquer l'algèbre de Boole à l'étude logique des circuits de commutation qui n'ont que deux états stables 0 ou 1. On obtient une algèbre binaire. Remarquons en outre qu'il s'agit d'une algèbre qualitative et non quantitative car elle permet de connaître uniquement l'état final d'un système.

But de l'algèbre logique

La correspondance très étroite existant entre les fonctions logiques fondamentales et les opérations sur les ensembles, permet, à partir de l'expression des conditions à réaliser d'écrire la fonction de transmission et de tracer le schéma correspondant ce qui permet la "synthèse" des schémas.

Inversement, les fonctions de commutation permettent de faire une "analyse" complète des conditions de fonctionnement. Ces fonctions peuvent s'écrire facilement à partir du schéma s'il a été obtenu par ailleurs.

Une bonne connaissance des règles opératoires permet de simplifier au maximum les fonctions de commutation.

Relations de l'algèbre logique avec les ensembles

	Ensemble	Algèbre logique	Se dit
Union	$A \cup B$ ou $A \vee B$	$A + B$	A ou B
Intersection	$A \cap B$ ou $A \wedge B$	$A \cdot B$ ou AB	A et B
Complément	$U-A$ ou $C_U(A)$	$\bar{A}$ ou A' ou $\sim A$	complément de A ou non A ou A barre

Conventions

Il existe deux types de logique suivant des conventions symboliques opposées s'attachant à ce que nous appelons : logique positive et logique négative.

- 1/ Logique positive. On est en logique positive lorsqu'à une affirmation logique correspond un niveau haut symbolisé par le "1" logique. Une infirmation logique sera exprimée par le niveau bas symbolisé par le "0" logique.
- 2/ Logique négative. On est en logique négative lorsque les conventions précédentes sont inversées. Ainsi, une affirmation logique correspond à un niveau bas et une infirmation logique à un niveau haut.

Il n'y a pas de règles imposant le choix d'une logique, il existe même des ensembles complexes dont la conception a été facilitée par l'emploi simultané des deux types de logique. Certains systèmes même travaillent en logique positive avec des niveaux négatifs (ce qui n'est pas, bien sûr, incompatible), l'affirmation logique "1" étant exprimée par le zéro électrique, l'infirmité logique "0" par un niveau négatif.

Disons tout de suite que la coutume est maintenant acquise d'utiliser la logique positive, ceci pour des raisons purement mnémotechniques, car il est plus commode d'utiliser un niveau haut électrique pour symboliser le "1" logique et le niveau zéro électrique pour le "0" logique.

Dans cet exposé, nous serons toujours en logique positive.

Le niveau logique "1" correspondra au niveau électrique + ou à un circuit fermé (passage du courant).

Le niveau logique "0" correspondra au niveau électrique 0 ou à un circuit ouvert (coupure du courant).

Les opérateurs seront représentés par :

ET : un point ou rien.

OU : +.

NON : par une barre sur la variable. Ex. $\bar{A}$.

Notons ici qu'une dualité existe entre le mode de logique choisi par convention pour l'établissement du schéma et la fonction réalisée. Ainsi, un opérateur " ET " en logique positive constituera un opérateur " ou " en logique négative et inversement (voir 3ème partie " éléments binaires fondamentaux ").

Axiomatique de l'algèbre de BOOLE

De même que l'algèbre classique, l'algèbre booléenne possède des propriétés de :

Symétrie : $\forall A \forall B \text{ si } A = B \leftrightarrow B = A$

Reflexivité : $\forall x \quad x = x$

Transitivité : $\forall A \forall B \forall C \text{ si } A = B \text{ et } B = C \rightarrow A = C$

Commutativité : (Axiome 1).



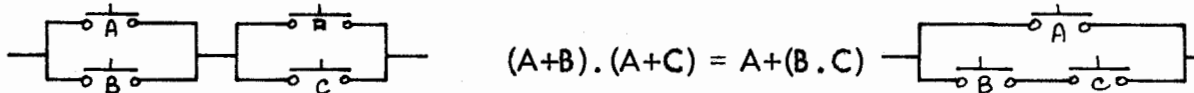
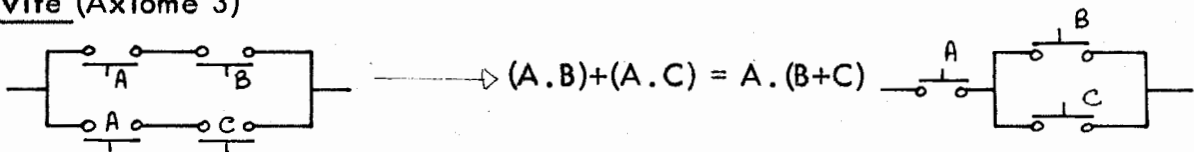
Associativité (Axiome 2)

$$(A.B).C = A.(B.C) = A.B.C$$

$$(A+B)+C = A+(B+C) = A+B+C$$

On ne change pas le résultat en décomposant les opérations différemment.

Distributivité (Axiome 3)



Absorption (Axiome 4)

Une des propriétés les plus importantes est représentée par les expressions suivantes :

$$A.(A+B) = A$$

$$A+(A.B) = A$$

Ces propriétés sont vérifiées, car si $A = 0$, le résultat ne peut être que nul et si $A = 1$, il ne peut être égal qu'à 1, donc dans tous les cas, il est égal à A.

Axiome 5 (Elément neutre de la fonction ou)

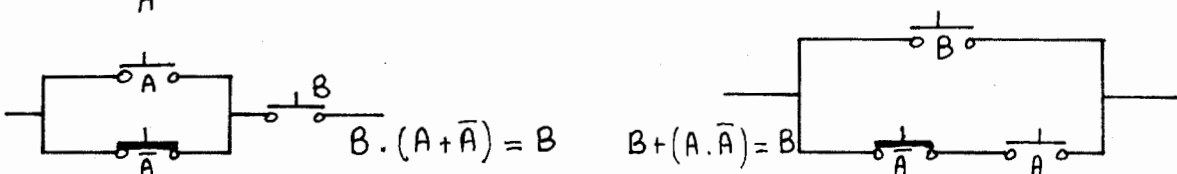
Il existe un élément noté 0 tel que : $A + 0 = 0 + A = A$

Axiome 6 (Elément neutre de la fonction ET)

Il existe un élément noté 1 tel que : $A.1 = 1.A = A$

Axiome 7 - (Propriétés du complément)

Pour tout A, il existe un élément et un seul noté $\bar{A}$ et appelé complément de A.



On voit que $A + \bar{A} = 1$ et $A.\bar{A} = 0$ sont les éléments neutres des circuits.

Relations fondamentales

Les axiomes précédents nous suffisent maintenant pour démontrer les divers théorèmes suivants.

Nota : Pour simplifier l'écriture, nous conviendrons d'écrire à droite de chaque identité ou équation le ou les axiomes ou théorèmes qui permettent d'y arriver.

Théorème 1 (idempotence)

$$\underline{A + A = A}$$

En effet, nous pouvons écrire :

$$\begin{aligned} A + A &= (A + A).1 && (A_6) \\ &= (A + A).(A + \bar{A}) && (A_7) \\ &= A + (A . \bar{A}) && (A_3) \\ &= A + 0 && (A_7) \\ &= A && (A_5) \end{aligned}$$

De manière plus générale, on peut démontrer que : $A + \underbrace{\dots\dots\dots}_{n \text{ termes}} + A = A$.

Théorème 2

$$\underline{A . A = A}$$

En effet,

$$\begin{aligned} A . A &= A . A + 0 && (A_5) \\ &= A . A + A . \bar{A} && (A_7) \\ &= A . (A + \bar{A}) && (A_3) \\ &= A . 1 && (A_7) \\ &= A && (A_6) \end{aligned}$$

Ces théorèmes (1 et 2) constituent les propriétés dites d'idempotence. Ils montrent que tous les termes répétés s'élimineront et qu'il n'y aura ni puissance, ni coefficient dans les expressions booléennes.

Conséquences : $A . (A . \bar{A}) = (A . A) . \bar{A} = A . \bar{A} = 0$ soit $A . 0 = 0$

de même : $A + (A + \bar{A}) = (A + A) + \bar{A} = A + \bar{A} = 1$ soit $A + 1 = 1$

Théorème 3

$$\underline{A + 1 = 1}$$

En effet :

$$\begin{aligned} A + 1 &= (A + 1).1 && (A_6) \\ &= (A + 1).(A + \bar{A}) && (A_7) \\ &= A + (1 . \bar{A}) && (A_3) \\ &= A + \bar{A} && (A_7) \\ &= 1 && (A_6) \end{aligned}$$

Théorème 4

$$\underline{A . 0 = 0}$$

En effet :

$$\begin{aligned} A . 0 &= (A . 0) + 0 && (A_5) \\ &= A . 0 + A . \bar{A} && (A_7) \\ &= A . (0 + \bar{A}) && (A_3) \\ &= A . \bar{A} && (A_5) \\ &= 0 && (A_7) \end{aligned}$$

Théorème 5

$$A + (\bar{A} \cdot B) = A + B \quad \text{et} \quad B + (\bar{B} \cdot A) = A + B$$

En effet :

$$\begin{aligned} A + (\bar{A} \cdot B) &= (A + \bar{A}) \cdot (A + B) && (A_3) \\ &= 1 \cdot (A + B) && (A_7) \\ &= A + B && (A_6) \end{aligned}$$

On peut démontrer de même que $B + (\bar{B} \cdot A) = A + B$.Théorème 6

$$A \cdot (\bar{A} + B) = A \cdot B \quad \text{et} \quad B \cdot (\bar{B} + A) = A \cdot B$$

En effet :

$$\begin{aligned} A \cdot (\bar{A} + B) &= A \cdot \bar{A} + A \cdot B && (A_3) \\ &= 0 + A \cdot B && (A_3) \\ &= A \cdot B && (A_5) \end{aligned}$$

On peut démontrer de même que $B + (\bar{B} \cdot A) = A \cdot B$.Théorème 7 (involution)

$$\bar{\bar{A}} = A$$

Nous devons avoir en vertu de l'axiome 7 :

$$\begin{cases} \bar{A} + \bar{\bar{A}} = 1 \\ \bar{A} \cdot \bar{\bar{A}} = 0 \end{cases} \quad \bar{\bar{A}} \text{ étant le complément de } \bar{A}.$$

d'après la définition du complément, nous avons d'autre part :

$$\begin{cases} A + \bar{A} = 1 = \bar{A} + A \\ A \cdot \bar{A} = 0 = \bar{A} \cdot A \end{cases}$$

Nous voyons donc que A et $\bar{\bar{A}}$ satisfont aux mêmes équations or d'après l'axiome 7 nous savons que le complément est unique, donc :

$$\bar{\bar{A}} = A.$$

Théorème 8

$$AB + BC + \bar{A}C = AB + \bar{A}C$$

En effet :

$$\begin{aligned} AB + BC + \bar{A}C &= (AB + \bar{A}C) + BC && (A_2) \\ &= (AB + \bar{A}C) + (\bar{A} + A) \cdot BC && (A_2) \\ &= (AB + \bar{A}C) + (\bar{A}BC + ABC) && (A_6) \\ &= AB + (\bar{A}C + \bar{A}BC) + ABC && (A_3) \\ &= AB + (\bar{A}C + \bar{A}C \cdot B) + ABC && (A_2) \\ &= AB + \bar{A}C + ABC && (A_2) \\ &= (AB + AB \cdot C) + \bar{A}C && (A_4) \\ &= AB + \bar{A}C && (A_2) \end{aligned}$$

Nota : Ce théorème est appelé parfois "Fonction de Lagrange". Elle se note :

$$\begin{cases} U(A, B, C) = AB + \bar{A}C \\ U(A, B, C) = AB + BC + \bar{A}C \end{cases}$$

On peut montrer de même qu'en permutant les opérateurs + et $\cdot$ dans les deux membres de l'égalité, on obtient une seconde égalité également vraie.

$$\text{Soit : } (A \cdot B) + (B \cdot C) + (\bar{A} \cdot C) = (A \cdot B) + (\bar{A} \cdot C)$$

$$\text{donne : } (A+B) \cdot (B+C) \cdot (\bar{A}+C) = (A+B) \cdot (\bar{A}+C)$$

Cette propriété particulière à l'algèbre de Boole est dite "principe de dualité". Ce principe dit que : "A tout théorème vrai dans une algèbre de Boole, correspond un théorème dual qui est vrai également".

Théorème 9 (1er théorème de De Morgan)

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Nous allons pour vérifier cette égalité chercher quel est le complément de $\overline{A} \cdot \overline{B}$ et montrer que c'est $A + B$.

Considérons en effet la quantité $(A + B) + \overline{A} \cdot \overline{B}$, elle doit être égale à 1 si $(A + B)$ est complément de $\overline{A} \cdot \overline{B}$ (axiome 7).

$$\begin{aligned} (A+B) + \overline{A} \cdot \overline{B} &= [(A+B)+\overline{A}] \cdot [(A+B)+\overline{B}] && (A_3) \\ &= [(A+\overline{A})+B] \cdot [A+(B+\overline{B})] && (A_2) \\ &= (1+B) \cdot (A+1) && (A_7) \\ &= 1 \cdot 1 = 1 && (Th_3-Th_2) \end{aligned}$$

De manière analogue, nous avons, toujours selon l'axiome 7, $(A+B) \cdot \overline{A} \cdot \overline{B} = 0$.

Si $(A+B)$ est complément de $\overline{A} \cdot \overline{B}$.

$$(A+B) \cdot \overline{A} \cdot \overline{B} = (\overline{A} \cdot \overline{B}) \cdot (A+B) \quad (A_1)$$

En effectuant le second membre, on obtient :

$$\begin{aligned} (A+B) \cdot \overline{A} \cdot \overline{B} &= A \cdot \overline{A} \cdot \overline{B} + \overline{A} \cdot \overline{B} \cdot B \\ &= (A \cdot \overline{A}) \cdot \overline{B} + \overline{A} \cdot (\overline{B} \cdot B) \\ &= 0 \cdot \overline{B} + \overline{A} \cdot 0 \\ &= 0 + 0 = 0 \end{aligned} \quad (Th_4)$$

Si $X = \overline{A} \cdot \overline{B}$

et $\overline{X} = A+B$, X et $\overline{X}$ vérifient bien les deux équations :

$$\begin{cases} X + \overline{X} = 1 \\ X \cdot \overline{X} = 0 \end{cases}$$

c'est à dire que $A + B$ est le complément (unique, rappelons-le) de $\overline{A} \cdot \overline{B}$, donc :

$$A + B = \overline{\overline{A} \cdot \overline{B}} \text{ et par suite : } \overline{A + B} = \overline{\overline{\overline{A} \cdot \overline{B}}} = \overline{A} \cdot \overline{B} \text{ donc } \underline{\overline{A+B} = \overline{A} \cdot \overline{B}}$$

Théorème 10 (2ème théorème de De Morgan)

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Tous les théorèmes précédemment définis peuvent se démontrer par la table de vérité. Nous allons montrer comment s'opère la démonstration pour ce dernier théorème.

Variables				Fonctions		
A	B	$\overline{A}$	$\overline{B}$	$A \cdot B$	$\overline{A \cdot B}$	$\overline{A} + \overline{B}$
0	0	1	1	0	1	1
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	1	0	0

On voit que les fonctions $\overline{A \cdot B}$ et $\overline{A} + \overline{B}$ prennent les mêmes valeurs pour des valeurs identiques des variables d'entrée, donc : $\overline{A \cdot B} = \overline{A} + \overline{B}$

Ces deux derniers théorèmes, très importants en algèbre logique, ne sont pas limités à deux variables, ils sont utilisables pour une infinité de variables, tel que :

1er théorème : $\overline{A + B + C + \dots + N} = \overline{A} \cdot \overline{B} \cdot \overline{C} \cdot \dots \cdot \overline{N}$

2e théorème : $\overline{A \cdot B \cdot C \cdot \dots \cdot N} = \overline{A} + \overline{B} + \overline{C} + \dots + \overline{N}$

RESUME DES AXIOMES ET THEOREMES DE L'ALGEBRE DE BOOLE

Symétrie : $A = B \longleftrightarrow B = A$ Réflexivité : $x = x$ Transitivité : $\left. \begin{array}{l} A = B \\ B = C \end{array} \right\} \Rightarrow A = C$ $\overline{\overline{A}} = A$ théorème 7

Forme de somme	Forme de produit	
$A+B = B+A$	$AB = BA$	axiome 1
$(A+B)+C = A+(B+C) = A+B+C$	$(AB)C = A(BC) = ABC$	axiome 2
$AB+AC = A(B+C)$	$(A+B).(A+C) = A+(BC)$	axiome 3
$A+(AB) = A$	$A.(A+B) = A$	axiome 4
$A+0 = A$	$A.1 = A$	axiome 5 et 6
$A+\overline{A} = 1$	$A.\overline{A} = 0$	axiome 7
$B+(A.\overline{A})=(A+B).(\overline{A}+B)=B$	$B.(A+\overline{A})=BA+B\overline{A}=B$	
$A+A = A$	$A.A. = A$	Théorèmes 1 & 2
$A+1 = 1$	$A.0 = 0$	Théorèmes 3 & 4
$A+\overline{AB} = B+\overline{BA} = A+B$	$A.(\overline{A}+B) = B.(\overline{B}+A) = A.B$	Théorèmes 5 & 6
$AB+BC+\overline{AC} = AB+\overline{AC}$	$(A+B).(B+C).(\overline{A}+C) = (A+B).(\overline{A}+C)$	Théorème 8
$\overline{A+B} = \overline{A}.\overline{B}$	$\overline{A.B} = \overline{A} + \overline{B}$	Théorèmes 9 & 10

On peut aussi démontrer que :

$$AB + A\overline{B} = A$$

$$\overline{A}B + A\overline{B} = B$$

$$A\overline{B} + \overline{A}B = (A+B).(\overline{A}+\overline{B})$$

FONCTIONS BOOLEENNES

Définition

Une fonction booléenne de n variables est une fonction qui à toute combinaison binaire à n digits, fait correspondre la valeur 0 ou la valeur 1.

S'il y a n variables, nous savons qu'il y a 2^n combinaisons possibles. L'ensemble de ces combinaisons constitue le "référentiel". Chaque combinaison incluse dans ce référentiel caractérise une fonction booléenne. Il y aura donc 2^{2^n} fonctions possibles pour n variables.

La tableau indique le nombre des fonctions possibles en raison du nombre de variables.

Nombre de variables	Nombre de combinaisons ou référentiel	Nombre de fonctions
1	$2^1 = 2$	$2^2 = 4$
2	$2^2 = 4$	$2^4 = 16$
3	$2^3 = 8$	$2^8 = 256$
4	$2^4 = 16$	$2^{16} = 65\ 536$

Nous allons montrer sous forme de tableau les 16 fonctions possibles des 4 combinaisons de 2 variables. Lorsque ces fonctions sont remarquables (fonctions fondamentales et secondaires), leur symbole est indiqué en haut de la colonne.

Variables		Fonctions															
A	B	0	$\overline{\text{ou}}$	/	$\overline{A}$	/	$\overline{B}$	$\oplus$	et	et	$A=B$ $\odot$	B	/	A	/	ou +	1
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Notre propos nous amène maintenant à nous intéresser aux fonctions logiques fondamentales concernant les opérateurs ET, OU, NON et aux fonctions logiques secondaires $\overline{\text{ET}}$, $\overline{\text{OU}}$, OU exclusif que l'on représente par $\oplus$ et identité symbolisée par $\odot$.

FONCTIONS LOGIQUES FONDAMENTALES

Fonction ET (AND) ou fonction "produit logique"

La variable binaire S de sortie est liée aux variables d'entrée par le tableau suivant (table de vérité). Cette règle peut être généralisée à un nombre de n variables.

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

On écrit $S = A \cdot B$ qui se lit A et B.

$S = 1$ si les variables A et B sont simultanément égales à 1.

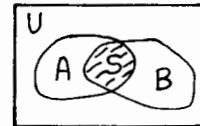
$S = 1$ si toutes les variables d'entrée sont simultanément égales à 1.

Diagrammes

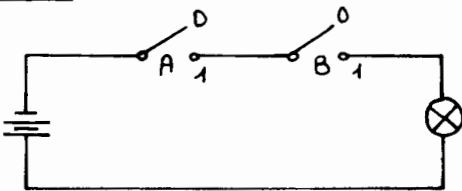
A	0	0	1	1
B	0	1	0	1
S	0	0	0	1

Cette opération est équivalente à l'intersection en théorie des ensembles où l'on note $S = A \cap B$.

S est la partie commune aux deux domaines A et B.



Exemple :



Pour que le courant passe et que la lampe soit allumée, il faut que les deux interrupteurs soient simultanément fermés. Avec les conventions définies précédemment, il faudra que les interrupteurs A et B soient sur 1 pour que le courant passe.

Symbole :



$$S = A \cdot B$$

Fonction OU (OR) ou fonction "somme logique"

La variable binaire S de sortie est liée aux variables d'entrée par le tableau suivant.

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

On écrit $S = A + B$ qui se lit A ou B.

$S = 1$ si une ou plusieurs ou toutes les variables sont égales à 1.

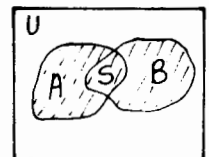
On appelle quelquefois cette fonction "OU inclusive".

Diagrammes

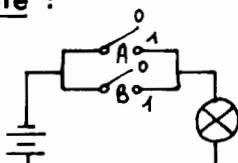
A	0	0	1	1
B	0	1	0	1
S	0	1	1	1

Cette opération est équivalente à la réunion en théorie des ensembles où l'on note $S = A \cup B$.

S est l'ensemble des domaines A et B. La somme logique ne fait pas de distinction entre un domaine couvert une seule fois et le même domaine couvert plusieurs fois.



Exemple :



Pour que la lampe soit allumée, il suffit que le courant passe par l'un ou l'autre des interrupteurs. Le fait que les deux interrupteurs soient simultanément fermés ne change rien.

Symbole :



$$S = A + B$$

Remarque importante

Il ne faut surtout pas confondre cette fonction "somme logique" avec l'addition binaire que nous avons étudiée précédemment.

En effet,

$$\begin{array}{ll} \text{fonction OU} & 1 \text{ ou } 1 = 1 \\ \text{addition binaire} & 1 \text{ plus } 1 = \textcircled{1} 0 \rightarrow \text{report} \end{array}$$

Fonction complément (NON)

On l'appelle aussi fonction NE PAS, PAS, INVERSE, NEGATION, NON.

La variable binaire S de sortie est liée à une variable d'entrée par le tableau suivant.

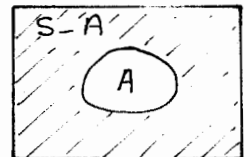
A	S
0	1
1	0

On écrit $S = \bar{A}$ qui se lit : A barre.

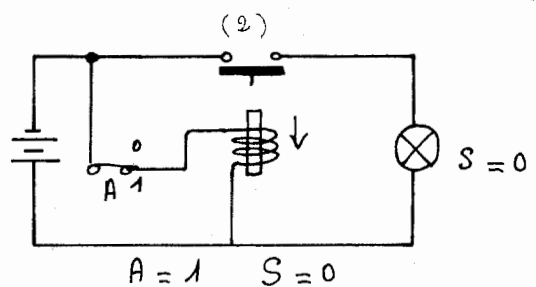
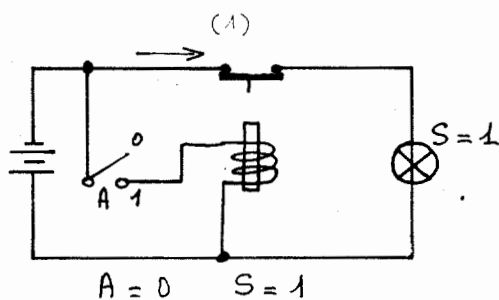
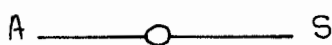
Diagrammes

A	0	1
S	1	0

Si l'on se reporte au diagramme de Venn, cette opération est équivalente en théorie des ensembles au complément où l'on note : $S = \bar{A}$ ou $C_S(A)$.

Exemple :

L'interrupteur A (variable d'entrée) commande un relais dont le contact repos est fermé (passage du courant). Lorsque l'interrupteur A est fermé (1), il laisse passer le courant dans le relais, celui-ci ouvre alors son contact et coupe le courant dans le circuit. La variable de sortie S est représentée par la lampe. On voit que dans le circuit 1, l'interrupteur A est sur 0, le relais n'est pas alimenté, le courant circule dans son contact et allume la lampe $S = 1$. Dans le circuit 2, A est sur 1, le courant circule dans le relais, attirant son contact. Par suite, le courant ne circule plus : $S = 0$.

Symbole :

$$S = \bar{A}.$$

FONCTIONS LOGIQUES SECONDAIRES

Ces fonctions sont appelées secondaires car elles sont obtenues à partir des fonctions logiques fondamentales.

Fonction ET (NAND)

Cette fonction est appelée aussi ET NON. La variable binaire S de sortie est liée aux variables d'entrée par le tableau suivant.

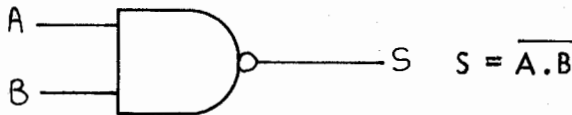
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

On écrit $S = \overline{A \cdot B}$ qui se lit : A et B barre.

$S = 0$ si toutes les variables d'entrée sont égales à 1.

Symbole :

C'est la combinaison des deux fonctions fondamentales ET, PAS.



Fonction OU (NOR)

Cette fonction est appelée quelquefois OU NON.

La variable binaire de sortie S est liée aux variables d'entrée par le tableau suivant.

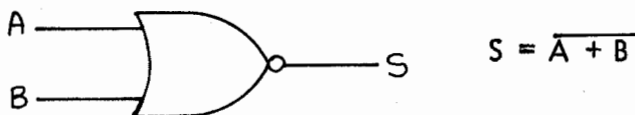
A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

On écrit $S = \overline{A + B}$ qui se lit : (A ou B) barre.

$S = 0$ si l'une quelconque ou plusieurs des variables d'entrée sont égales à 1.

Corollaire : $S = 1$ si toutes les variables d'entrée sont égales à 0.

Symbole :



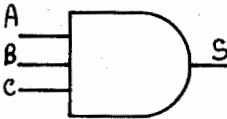
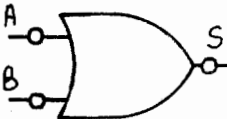
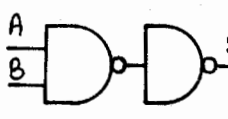
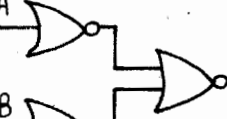
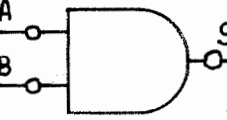
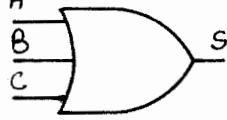
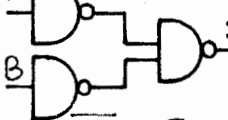
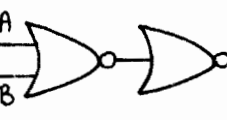
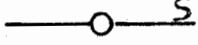
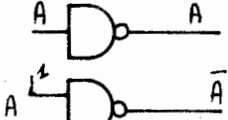
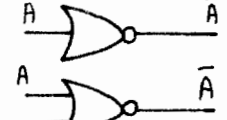
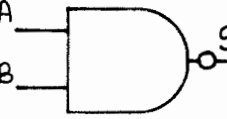
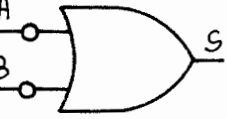
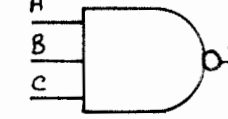
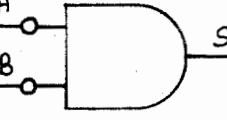
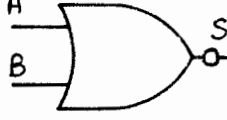
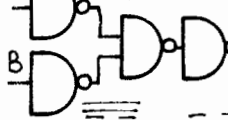
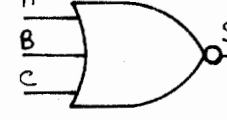
Obtention d'une fonction quelconque

Avant de poursuivre l'étude des fonctions logiques secondaires, il est intéressant de montrer comment, à partir des fonctions fondamentales et secondaires, il est possible d'obtenir des correspondances. En d'autres termes, nous allons à l'aide d'un tableau de correspondances, montrer comment, à partir d'une fonction ET par exemple, il est possible d'obtenir toutes les autres fonctions et inversement, à l'aide de toutes les autres fonctions, comment obtenir la fonction ET.

TABLEAU DES CORRESPONDANCES (Logique positive)

Fonction désirée

Fonction employée

	ET	OU	PAS	$\overline{\text{ET}}$	$\overline{\text{OU}}$
ET	 $S = A \cdot B \cdot C$	 $S = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$ $S = A \cdot B$		 $S = \overline{\overline{A} \cdot \overline{B}} = A \cdot B$	 $S = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}} = A \cdot B$
OU	 $S = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$ $S = A + B$	 $S = A + B + C$		 $S = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} + \overline{\overline{B}}$ $S = A + B$	 $S = \overline{\overline{A} + \overline{B}} = A + B$
PAS			 $S = \overline{A}$	 $S = \overline{A}$	 $S = \overline{A}$
$\overline{\text{ET}}$	 $S = \overline{A \cdot B}$	 $S = \overline{\overline{A} + \overline{B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$		 $S = \overline{A \cdot B \cdot C}$	 $S = \overline{A \cdot B}$
$\overline{\text{OU}}$	 $S = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} + \overline{\overline{B}}$	 $S = \overline{\overline{A} + \overline{B}}$		 $S = \overline{\overline{A} \cdot \overline{B}} = \overline{\overline{A}} + \overline{\overline{B}}$ $S = \overline{\overline{A} + \overline{B}}$	 $S = \overline{\overline{A} + \overline{B} + \overline{C}}$

Fonction "OU EXCLUSIF"

Cette fonction reçoit aussi les noms suivants : inéquivalence, dilemme, somme modulo 2. Dans cette fonction, on exclut la possibilité d'avoir simultanément les deux entrées à 1. La variable binaire S de sortie est liée aux variables d'entrée par le tableau suivant.

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

On écrit $S = A \oplus B$ qui se lit : A ou exclusif B.

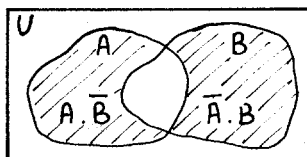
$S = 1$ si l'une ou l'autre des deux variables d'entrée est égale à 1, mais pas les deux à la fois.

On montre que : $S = A \cdot \bar{B} + \bar{A} \cdot B$

ou : $S = (A+B) \cdot (\bar{A} + \bar{B})$ (voir formes canoniques).

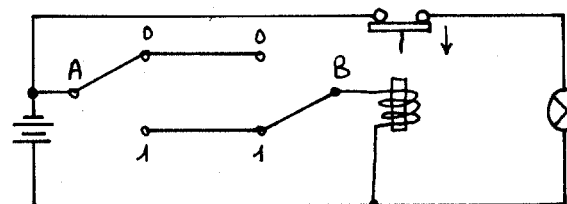
Diagrammes :

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

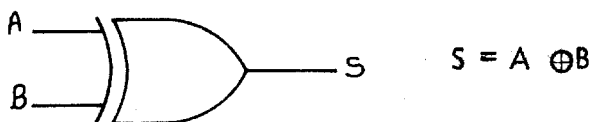


Exemple :

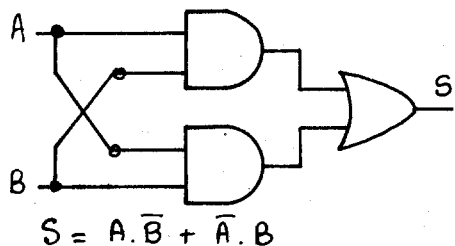
Dans cet exemple, on réutilise le relais à contact repos fermé (passage du courant). On voit facilement que $S = 1$ (lampe allumée) lorsqu'une seule des deux variables (interrupteur A ou B) est à 1.



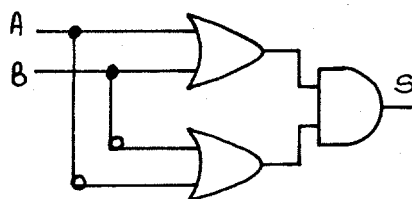
Symbole :



Obtention de la fonction "OU exclusif" avec les fonctions fondamentales

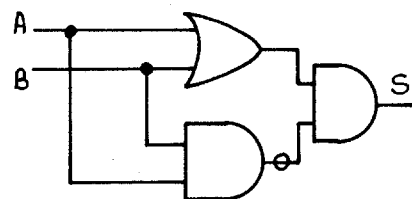


$$S = A \cdot \bar{B} + \bar{A} \cdot B$$



$$S = (A+B) \cdot (\bar{A} + \bar{B}) = A\bar{A} + \bar{A}B + A\bar{B} + B\bar{B}$$

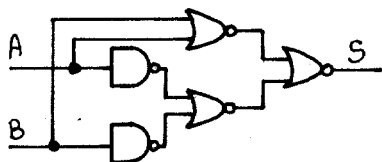
$$S = \bar{A}B + A\bar{B}$$



$$S = (A+B) \cdot (\bar{A} \cdot \bar{B})$$

$$= (A+B) \cdot (\bar{A} + \bar{B})$$

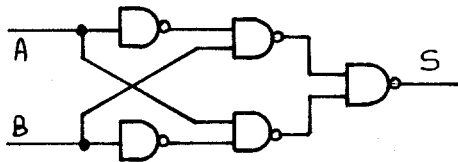
Obtention de la fonction "OU exclusif" avec les fonctions secondaires



$$S = \overline{(A+B)} + \overline{(\bar{A} + \bar{B})}$$

$$S = \overline{(A+B)} \cdot \overline{(\bar{A} + \bar{B})}$$

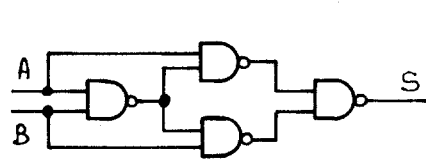
$$S = (A+B) \cdot (\bar{A} + \bar{B})$$



$$S = \overline{(\bar{A} \cdot B)} + \overline{(A \cdot \bar{B})}$$

$$S = \overline{(\bar{A} \cdot B)} \cdot \overline{(A \cdot \bar{B})}$$

$$S = (\bar{A} \cdot B) + (A \cdot \bar{B})$$



$$S = \overline{A \cdot (\bar{A} \cdot B)} + \overline{B \cdot (A \cdot \bar{B})}$$

$$S = \overline{A \cdot (\bar{A} \cdot B)} + \overline{B \cdot (A \cdot \bar{B})}$$

$$S = A \cdot (\bar{A} \cdot B) + B \cdot (A \cdot \bar{B})$$

$$S = A \cdot (\bar{A} + \bar{B}) + B \cdot (\bar{B} + \bar{A})$$

$$S = A \cdot \bar{B} + B \cdot \bar{A}$$

Fonction identité

Cette fonction quelquefois appelée "coïncidence" est corollaire de la fonction "OU exclusif" puisqu'elle en est le complément.

La variable binaire S de sortie est liée aux variables d'entrée par le tableau suivant.

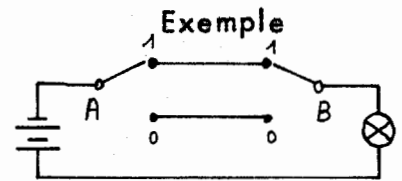
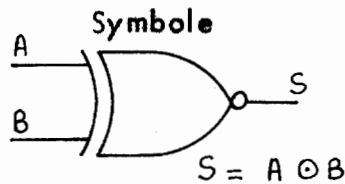
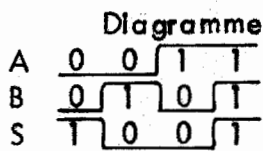
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Plusieurs méthodes d'écriture sont admises :

$$S = A \odot B \quad S = \overline{A \oplus B} \quad A = B$$

$$S = AB + \bar{A}.\bar{B}$$

$S = 1$ si les deux variables d'entrée ont la même valeur (0 ou 1).



Cette fonction n'est vérifiée que pour deux variables d'entrée. Au delà, elle devient l'inverse de la "somme modulo 2".

Somme modulo 2 ou "OU exclusif" à n variables.

Cette fonction est la fonction "OU exclusif" généralisée.

La variable binaire S de sortie est liée aux n variables d'entrée par le tableau suivant (exemple 3 variables).

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

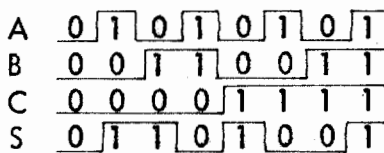
On écrit $S = A \oplus B \oplus C \oplus \dots \oplus n$

$S = 1$ pour un nombre impair de variables égales à 1.

On montre que :

$$S = A \oplus B \oplus C = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}BC + ABC \quad (1)$$

Diagramme :



Représentation

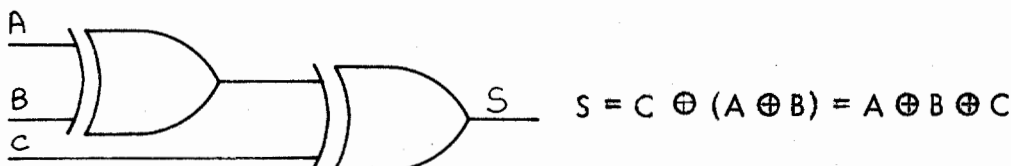
A partir de la relation précédente (1), nous pouvons écrire : $S = \bar{C}(\bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB)$

Soit $S = \bar{C}(A \oplus B) + C(\bar{A} \oplus \bar{B})$ c'est la forme $\oplus$

Posons : $(A \oplus B) = X$, on a : $S = (\bar{C}.X) + (C.\bar{X})$

d'où : $S = C \oplus A \oplus B$.

On a donc le schéma :



FONCTIONS PARTICULIERES

1/ Fonction de majorité ou de quorum

Il s'agit d'un type particulier de fonctions dites "fonctions de seuil". La fonction vaut 1 lorsque x variables au plus valent 1. Elle vaut 0 dans les autres cas.

x est un entier, $0 \leq x \leq n$, n est le nombre de variables d'entrée.

Si n est impair, on appelle fonction de majorité simple celle pour laquelle : $x = \frac{n+1}{2}$.

Exemple :

Fonction de majorité simple de 3 variables : la variable binaire S de sortie prendra la valeur 1 lorsque 2 variables d'entrée au moins sur les trois seront égales à 1.

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

On note en général : $S = \text{MAJ}(A, B, C)$

ou : $S = A \# B \# C$

On peut écrire :

$$\text{MAJ}(A, B, C) = AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC$$

$$= AB\bar{C} + A\bar{B}C + \bar{A}BC + ABC + (ABC) + (ABC) \quad (\text{Th}_1)$$

$$= (AB\bar{C} + ABC) + (A\bar{B}C + ABC) + (\bar{A}BC + ABC) \quad (\text{A}_2)$$

$$= AB(C + \bar{C}) + AC(B + \bar{B}) + BC(A + \bar{A})$$

$$= AB + AC + BC \quad (\text{A}_6 \text{ et } \text{A}_7)$$

$$\left\{ \begin{array}{l} \text{MAJ}(A, B, C) = AB + AC + BC \\ \text{MAJ}(A, B, C) = (A + B) \cdot (A + C) \cdot (B + C) \end{array} \right. \quad (\text{Th}_9 \text{ et } \text{Th}_{10})$$

On peut noter quelques propriétés :

$$\text{MAJ}(A, B, 0) = A \cdot B$$

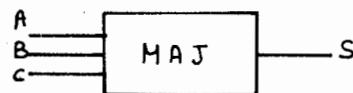
$$\text{MAJ}(A, B, 1) = A + B$$

$$\overline{\text{MAJ}(A, B, C)} = \text{MAJ}(\bar{A}, \bar{B}, \bar{C})$$

En effet :

$$\overline{AB + AC + BC} = \bar{A}\bar{B} \cdot \bar{A}\bar{C} \cdot \bar{B}\bar{C} = (\bar{A} + \bar{B}) \cdot (\bar{A} + \bar{C}) \cdot (\bar{B} + \bar{C})$$

Il n'y a pas de symbole particulier, mais on peut toujours dans les schémas utiliser une case rectangulaire portant les entrées et la sortie et munie de la désignation de la fonction.



2/ Fonction inhibition

Soit une fonction booléenne F de n variables binaires $x_1, x_2, \dots, x_n$ et une variable d'inhibition I . La table de vérité est alors la suivante.

$F(x_1, x_2, \dots, x_n)$	I	S
0	0	0
1	0	0
1	1	1
0	1	0
1	0	0

$$\text{d'où } S = I \cdot F(x_1, x_2, \dots, x_n).$$

Il n'y a transmission de F que si $I = 1$.

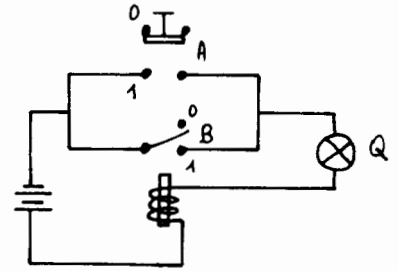
3/ Fonction mémoire

La variable binaire Q de sortie peut prendre la valeur 1 ou 0 pour une même configuration des variables d'entrée à des instants différents.

Mémoire simple

Elle peut être schématisée par le circuit ci-contre, utilisant un relais à contact "travail" fermé (passage du courant). Ce contact, lorsqu'il est fermé, autoalimente le relais et de ce fait, reste fermé, la lampe Q est allumée.

Pour exciter le relais (on dira "enclencher la mémoire"), il suffit d'agir sur le poussoir A et lui donner ainsi la valeur 1. D'après ce que nous connaissons, nous pouvons écrire : $Q = A + B$.



Voyons maintenant ce qui se passe lors des différentes phases de fonctionnement.

Etat initial :	t_0	$A = 0$	$B = 0$	$Q = 0 + 0 = 0$
Action sur A :	t_1	$A = 1$	$B = 1$	$Q = 1 + 1 = 1$
Retour de A à l'état initial :	t_2	$A = 0$	$B = 1$	$Q = 0 + 1 = 1$

C'est l'état final de la mémoire : B étant fermé et A ouvert. On voit que la fonction Q prend les mêmes valeurs que la variable B . Pour cette raison, on identifie Q et B , d'où :

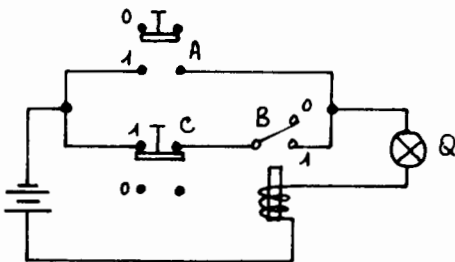
$$Q = A + Q$$

L'inconvénient majeur de cette mémoire est l'impossibilité de retour à zéro.

Mémoire à déverrouillage

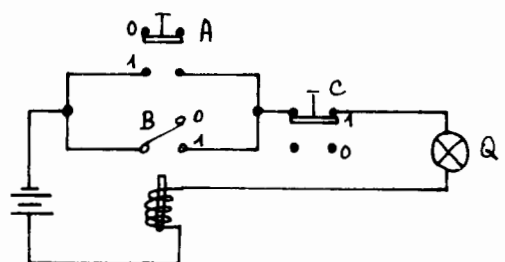
Il y a deux configurations possibles.

Circuit 1 : Mémoire prioritaire à l'enclenchement.



Si $A = 1$, C est sans action.
 Au départ, C est fermé : $C = 1$
 $Q = B$ et $Q = A.(B+C)$
 d'où : $Q = A.(Q+C)$

Circuit 2 : Mémoire prioritaire au déclenchement.



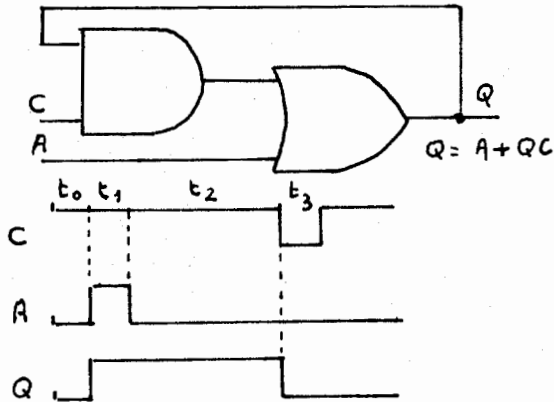
Si $C = 0$, A est sans action.
 Au départ, C est fermé : $C = 1$
 $Q = B$ et $Q = C.(A+B)$
 d'où : $Q = C.(A+Q)$

On voit d'après ces schémas que pour retourner à l'état initial, il a été nécessaire d'introduire une variable supplémentaire "C" dite variable de déclenchement.

Le fonctionnement est le suivant :

	C	A	B	Q
Etat initial	1	0	0	0 $\rightarrow t_0$
Action sur A	1	1	1	1 $\rightarrow t_1$
A revient à l'état initial	1	0	1	1 $\rightarrow t_2$
Retour à l'état initial	0	0	0	0 $\rightarrow t_3$
	1	0	0	0

Réalisation de mémoires à éléments de logique



A est la variable d'enclenchement.
C est la variable de déclenchement.
Au repos, $C = 1$, $A = 0$. Pour avoir les mêmes valeurs au repos, il faut ajouter un inverseur sur A ou sur C.

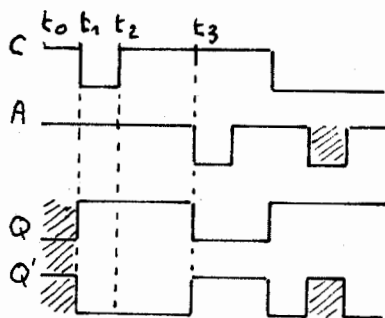
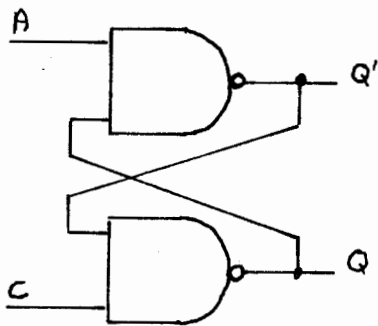
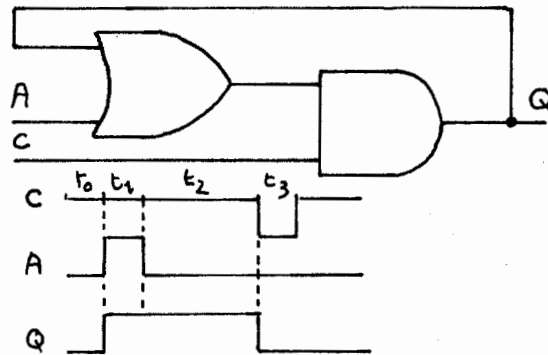
C'est une mémoire prioritaire à l'enclenchement. "A" est prioritaire.

C'est une mémoire prioritaire au déclenchement. "C" est prioritaire.

A étant la variable d'enclenchement.
C étant la variable de déclenchement.

Au repos, $C = 1$ et $A = 0$.

$$Q = C \cdot (A + Q)$$



Au repos, $C = A = 1$. Cette mémoire n'a pas de priorité d'enclenchement ou de déclenchement, les sorties et les entrées sont symétriques.

Au temps T_0 , l'état des sorties est indéterminé. Au temps T_1 , C est la variable d'enclenchement (dans cet exemple), donc A sera la variable de déclenchement. L'état $C = A = 0$ est un état anormal. Cette mémoire est du type $Q = A + (Q \cdot C)$.

En effet :

$$\begin{cases} Q' = \overline{A \cdot Q} = \overline{A} + \overline{Q} & (1) \\ Q = \overline{C \cdot Q'} = \overline{C} + Q' & (2) \end{cases}$$

En remplaçant dans (1) Q par sa valeur, on a : $Q' = \overline{A} + (\overline{C} + Q) = \overline{A} + \overline{C} + Q = \overline{A} + C \cdot Q'$ (Th9-10-7)

En remplaçant dans (2) Q' par sa valeur, on a : $Q = \overline{C} + \overline{(\overline{A} + C \cdot Q')} = \overline{C} + \overline{\overline{A} + C \cdot Q'} = \overline{C} + A + Q = \overline{C} + A \cdot Q$ (Th9-10-7)

A	C	Q	Q'
1	1	indéterminé $\rightarrow t_0$	
1	0	1	0 $\rightarrow t_1$
1	1	1	0 $\rightarrow t_2$
0	1	0	1 $\rightarrow t_3$
1	1	0	1

$$\begin{cases} Q = \bar{C} + (A \cdot Q) \\ Q' = \bar{A} + (C \cdot Q') \end{cases}$$

Ces deux égalités
sont bien de la forme :
 $Q = A + Q \cdot C$

Une mémoire similaire peut être réalisée à l'aide d'élément $\overline{OU}$, elle est du type $Q = C \cdot (A + Q)$.

On a :

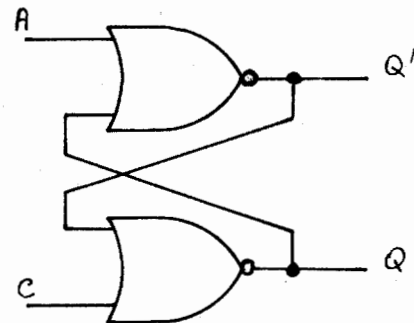
$$\begin{cases} Q' = \overline{A + Q} = \bar{A} \cdot \bar{Q} \\ Q = \overline{Q' + C} = \bar{Q}' \cdot \bar{C} \end{cases}$$

$$Q' = \overline{Q' + C + A} = \bar{A} \cdot (C + Q)$$

$$Q = \overline{A + Q + C} = \bar{C} \cdot (A + Q)$$

L'état $C = A = 0$ est l'état de repos.

L'état $C = A = 1$ est un état anormal.



Nota :

D'autres éléments plus complexes réalisant la fonction mémoire seront étudiés ultérieurement.

FORMES CANONIQUES DES FONCTIONS BOOLEENNES

Comme en algèbre classique, une grandeur S peut être fonction de plusieurs grandeurs A, B, C soit :

$$S = f(A, B, C)$$

Si on connaît les valeurs de A, B, C , on en déduit celle de S , étant entendu que ces grandeurs sont liées entre elles par l'une ou l'autre ou plusieurs des opérations de l'algèbre logique. Ainsi, toutes fonctions de la forme $f(x_1, x_2, \dots, x_n)$ de n variables binaires indépendantes $x_1, x_2, \dots, x_n$ est une fonction de commutation. La fonction f peut se représenter sous la forme d'une somme logique de produits logiques ou sous la forme d'un produit logique de sommes logiques.

1/ Forme canonique de somme de produits

Une expression est sous forme canonique de somme de produits, si toutes les variables (sous forme vraie ou inverse) figurent dans chaque terme et que, dans chacun de ces termes, elles sont toutes reliées entre elles par l'opération ET. Tous les termes étant reliés entre eux par l'opération OU.

Cette forme est appelée aussi "forme canonique disjonctive" ou "réunion d'intersections de base".

La fonction s'obtient à partir de la table de vérité.

Exemple :

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

$$\begin{aligned} 0 &= \bar{A} \cdot \bar{B} \\ 1 &= A \cdot \bar{B} \\ 1 &= \bar{A} \cdot B \\ 0 &= A \cdot B \end{aligned}$$

Puisque la fonction n'est vraie que lorsqu'elle vaut 1, on ne s'occupe que des termes qui donnent $S = 1$.

Nous avons $S = 1$ pour la combinaison $A \cdot \bar{B}$ ou $\bar{A} \cdot B$.

$$D'où : S = A \cdot \bar{B} + \bar{A} \cdot B \quad (1)$$

2/ Forme canonique de produit de sommes

Une expression est sous forme canonique de produit de sommes si toutes les variables (sous forme vraie ou inverse) figurent dans chaque facteur et si, dans chacun de ces facteurs, elles sont toutes reliées entre elles par l'opération OU, tous les facteurs étant reliés entre eux par l'opération ET.

Cette forme est appelée aussi "forme canonique conjonctive" ou "intersection de réunions de base".

Pour obtenir la forme canonique de produit de sommes, il faut avoir recours au théorème de De Morgan, car il s'agit de l'expression DUALE de la première forme canonique (inversion des opérateurs - voir théorème 8). On écrit d'abord la table de vérité de S et de son complément $\bar{S}$.

A	B	S	$\bar{S}$	
0	0	0	1	$\bar{A} \cdot \bar{B}$
1	0	1	0	$0 = A \cdot \bar{B}$
0	1	1	0	$0 = \bar{A} \cdot B$
1	1	0	1	$A \cdot B$

On cherche ensuite la forme canonique de somme de produits de $\bar{S}$ (1ère forme canonique). On trouve : $\bar{S} = \bar{A} \cdot \bar{B} + AB$

Ces termes sont ceux qui ont une transmission nulle dans la première forme.

Il suffit ensuite de complémentier le résultat pour obtenir S. Pour cela, on applique le théorème de De Morgan :

$$S = \overline{\overline{A} \cdot \overline{B} + A \cdot B} = \overline{\overline{A}} + \overline{\overline{B}} = A + B$$

$$S = (A + B) \cdot (\overline{A} + \overline{B}) \quad (2)$$

Cette expression est bien duale de la première forme $\overline{A}\overline{B} + \overline{A}B$ car on peut toujours écrire (théorème 8 - principe de dualité - inversion des opérateurs).

$$S(\text{forme 1}) = \overline{A}\overline{B} + \overline{A}B \xrightarrow{\text{duale}} A + \overline{B} = \overline{A} + B$$

en effectuant :

$$A + \overline{B} \cdot \overline{A} + B = \overline{A}\overline{A} + \overline{A}\overline{B} + AB + B\overline{B}$$

$$= \overline{A}\overline{B} + AB = \overline{S}$$

d'où :

$$S(\text{forme 2}) = \overline{\overline{A}\overline{B} + AB}$$

$$= (A + B) \cdot (\overline{A} + \overline{B})$$

Ces expressions (1) et (2) donnent la forme canonique de la fonction booléenne, c'est à dire la forme complète avant simplification.

On dit aussi : "Forme maximale disjonctive" (1)

et : "Forme maximale conjonctive" (2)

Simplification des formes canoniques

Lorsqu'on doit réaliser une certaine fonction logique, on cherche à simplifier la forme canonique pour minimiser le nombre des éléments à utiliser.

Il existe deux techniques de minimisation :

- soit utiliser directement les règles de l'algèbre de Boole ou des méthodes qui en dérivent.
- soit utiliser des méthodes graphiques.

Mais, quelle que soit la méthode utilisée, dans chaque cas, intervient l'ingéniosité de l'opérateur, car la simplification obtenue n'est pas toujours la meilleure, en ce sens que la forme finale n'est pas toujours une somme de produits logiques irréductibles.

Regroupement de termes et mise en facteurs communs

Nous avons déjà largement utilisé cette méthode lors des différentes démonstrations qui ont été nécessaires tout au long de cet exposé.

Des groupements de termes sont possibles chaque fois que ces termes présentent une forme similaire à celles des axiomes et des théorèmes de l'algèbre de Boole.

Exemple :

$$AB + A\overline{B} = A(B + \overline{B}) = A \quad (A_7)$$

$$A + \overline{A}B = A \quad (A_4)$$

$$A + \overline{A}B = A + B \quad (A_{5-6})$$

$$AB + BC + \overline{A}C = AB + \overline{A}C \quad (\text{Th}_8)$$

En outre, on peut ajouter à une expression un terme qui y figure déjà, un terme nul ($A\overline{A}$) ou multiplier par un terme égal à 1 ($A + \overline{A}$, $A + 1$, etc...).

Exemple : Soit une fonction $S(A,B,C)$. La forme canonique de somme de produits nous donne :

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

$$S = ABC\bar{C} + \bar{A}BC\bar{C} + AB\bar{C}\bar{C} + \bar{A}\bar{B}C\bar{C} + A\bar{B}C\bar{C} + \bar{A}BC\bar{C} + ABC\bar{C}$$

nous pouvons écrire (axiome 1) :

$$S = ABC + \bar{A}BC + A\bar{B}C + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + \bar{A}BC$$

une mise en facteurs partielle nous donnera (termes 2 à 2) :

$$S = BC(A + \bar{A}) + \bar{B}C(A + \bar{A}) + B\bar{C}(A + \bar{A}) + A\bar{B}\bar{C}$$

après l'application de l'axiome 7, on obtient :

$$S = BC + \bar{B}C + B\bar{C} + A\bar{B}\bar{C}$$

une mise en facteur des 2 premiers termes amène :

$$S = C(B + \bar{B}) + B\bar{C} + A\bar{B}\bar{C}$$

il reste, après une nouvelle application de l'axiome 7 :

$$S = C + B\bar{C} + A\bar{B}\bar{C}$$

ajoutons à cette expression le terme nul $C.\bar{C}$:

$$S = C + B\bar{C} + A\bar{B}\bar{C} + C\bar{C}$$

mettons en facteur commun à $\bar{C}$ les 3 derniers termes :

$$S = C + \bar{C}(B + A\bar{B} + C)$$

on applique l'axiome 5 à $B + A\bar{B} (= A + B)$

$$S = C + \bar{C}(A + B + C)$$

on a :

$$C + \bar{C} = 1$$

l'axiome 6 donne $A.1 = A$, soit :

$$S = A + B + C$$

Il existe d'autres méthodes algébriques (ex. méthode par suppression des termes superflus), mais elles sont peu sûres et il est conseillé d'utiliser de préférence les méthodes graphiques, plus rigoureuses et plus faciles en ce sens que la fonction entière est visualisée avec toutes les combinaisons possibles des termes et leur regroupement.

Une autre méthode souvent décrite dans la littérature spécialisée est la méthode dite des "consensus", regroupant les termes 2 à 2 pour supprimer chaque fois que cela est possible les variables biformes (directe et inverse) et ne conserver que les implicants premiers (termes irréductibles).

METHODES GRAPHIQUES

Il existe là aussi plusieurs méthodes plus ou moins utilisées. Nous ne pouvons les étudier toutes dans le cadre de cet exposé. Citons pour mémoire :

- les diagrammes d'Euler et de Venn ;
- les cubes à N dimensions ou hypercubes ;
- l'algorithme de Quine Mc Cluskey dans lequel on utilise une représentation graphique appelée : "grille des termes irréductibles" ;
- le diagramme de Karnaugh duquel dérive le diagramme de Veitch.

Le principe du diagramme de VEITCH est identique à celui du diagramme de KARNAUGH seule la forme de représentation est changée.

Par ailleurs, son utilisation est de plus en plus fréquente. C'est pourquoi, après avoir dit quelques mots du diagramme de Karnaugh, nous nous intéresserons plus particulièrement au diagramme de Veitch.

LE DIAGRAMME DE KARNAUGH

Une fonction booléenne peut être représentée graphiquement sur un diagramme de Karnaugh ; celui-ci traduit l'interdépendance des variables dépendantes et indépendantes. Son but principal est de conduire à une forme simplifiée des fonctions, il permet de voir facilement les éléments adjacents permettant un regroupement. Plus le regroupement est important, plus le terme résultant obtenu est concentré.

L'emploi de ce diagramme est commode lorsque le nombre de variables est ≤ 4 et le devient beaucoup moins dès que ce nombre croît, en raison de la complexité du tracé.

Les règles à suivre dans le cas des diagrammes sont les suivantes :

- Effectuer le regroupement des blocs d'éléments adjacents les plus importants possibles, de façon à obtenir des intersections d'ordre le plus bas ;
- Choisir des groupements qui aient au moins un terme non commun.

REPRESENTATION

A chaque ligne d'une table de vérité figure une valeur possible prise par la variable (s'il n'y en a qu'une) ou une combinaison $n^{\text{ième}}$ possible s'il y a n variables. Le diagramme de Karnaugh fait correspondre chaque ligne de la table de vérité à une surface du diagramme. Il y aura donc autant de surfaces élémentaires (cases) qu'il y aura de lignes à la table de vérité. Chaque ligne représente une combinaison possible des n variables. Il y a 2^n lignes, il y aura 2^n cases dans le graphique.

- Une surface élémentaire représente une valeur prise par la fonction booléenne dans sa forme canonique (0 ou 1).

On porte en abscisses et en ordonnées les variables et toutes les valeurs (0 ou 1) qu'elles peuvent prendre. L'ordre des lignes et des colonnes est choisi pour qu'une seule variable change de valeur quand on passe d'une ligne à la suivante ou d'une colonne à la suivante.

- Une fonction est complètement définie si, dans chaque case du diagramme, figure une valeur spécifiée. Cependant, pour gagner du temps, on n'inscrira que les valeurs 1, les cases vides ne pouvant renfermer que des zéros.

Le tableau suivant montre la construction du diagramme de Karnaugh pour $1 \leq n \leq 5$. Il montre qu'au delà de 4 variables son emploi devient assez complexe.

Diagrammes de KARNAUGH

1 variable : A

A	S
0	s_0
1	s_1

A	0	1
s_0	s_0	s_1

3 variables : A, B, C

A	B	C	S
0	0	0	s_0
1	0	0	s_1
0	1	0	s_2
1	1	0	s_3
0	0	1	s_4
1	0	1	s_5
0	1	1	s_6
1	1	1	s_7

B	C	A	0	1
0	0		s_0	s_1
0	1		s_4	s_5
1	1		s_6	s_7
1	0		s_2	s_3

2 variables : A, B

A	B	S
0	0	s_0
1	0	s_1
0	1	s_2
1	1	s_3

		A	
		0	1
B	0	s_0	s_1
	1	s_2	s_3

4 variables : A, B, C, D

A	B	C	D	S
0	0	0	0	s_0
1	0	0	0	s_1
0	1	0	0	s_2
1	1	0	0	s_3
0	0	1	0	s_4
1	0	1	0	s_5
0	1	1	0	s_6
1	1	1	0	s_7
0	0	0	1	s_8
1	0	0	1	s_9
0	1	0	1	s_{10}
1	1	0	1	s_{11}
0	0	1	1	s_{12}
1	0	1	1	s_{13}
0	1	1	1	s_{14}
1	1	1	1	s_{15}

CD \ AB	00	01	11	10
00	s_0	s_2	s_3	s_1
01	s_8	s_{10}	s_{11}	s_9
11	s_{12}	s_{14}	s_{15}	s_{13}
10	s_4	s_6	s_7	s_5

5 variables : A, B, C, D, E

		0						1			
E	CD \ AB	00	01	11	10	E	CD \ AB	00	01	11	10
	00	s_0	s_2	s_3	s_1		00	s_{16}	s_{18}	s_{19}	s_{17}
	01	s_8	s_{10}	s_{11}	s_9		01	s_{24}	s_{26}	s_{27}	s_{25}
	11	s_{12}	s_{14}	s_{15}	s_{13}		11	s_{28}	s_{30}	s_{31}	s_{29}
	10	s_4	s_6	s_7	s_5		10	s_{20}	s_{22}	s_{23}	s_{21}

CD \ AB		AB			
		00	01	11	10
00	00	s_{16}	s_{18}	s_{19}	s_{17}
	01	s_{24}	s_{26}	s_{27}	s_{25}
11	11	s_{28}	s_{30}	s_{31}	s_{29}
	10	s_{20}	s_{22}	s_{23}	s_{21}

Propriétés fondamentales

- 1) Chaque carré du diagramme correspond à une combinaison de la forme canonique.
- 2) Entre 2 carrés adjacents (un côté commun), une seule variable change de valeur.
- 3) Entre 4 carrés adjacents 2 à 2, il n'y a que 2 variables qui changent de valeur.
- 4) Entre 8 carrés adjacents 2 à 2, il n'y a que 3 variables qui changent de valeur.

Le principe est identique à celui du diagramme de Karnaugh, seule la forme de représentation est changée. Chaque case est directement représentative des produits logiques de la forme canonique sous forme littérale.

Toutes les règles et propriétés du diagramme de Karnaugh sont applicables au diagramme de Veitch. Le tableau ci-dessous donne la construction du diagramme de Veitch pour $1 \leq n \leq 6$.

1 variable : A

A	S	$\bar{A}$	A
0	s_0	s_0	
1	s_1		s_1

3 variables : A, B, C

A	B	C	S	$\bar{A}$	A	
0	0	0	s_0			
1	0	0	s_1	s_4	s_5	C
0	1	0	s_2	s_0	s_1	$\bar{C}$
1	1	0	s_3	s_2	s_3	
0	0	1	s_4			
1	0	1	s_5	s_6	s_7	C
0	1	1	s_6			
1	1	1	s_7			

2 variables : A, B

A	B	S	$\bar{A}$	A
0	0	s_0		
1	0	s_1	s_0	s_1
0	1	s_2		
1	1	s_3	s_2	s_3

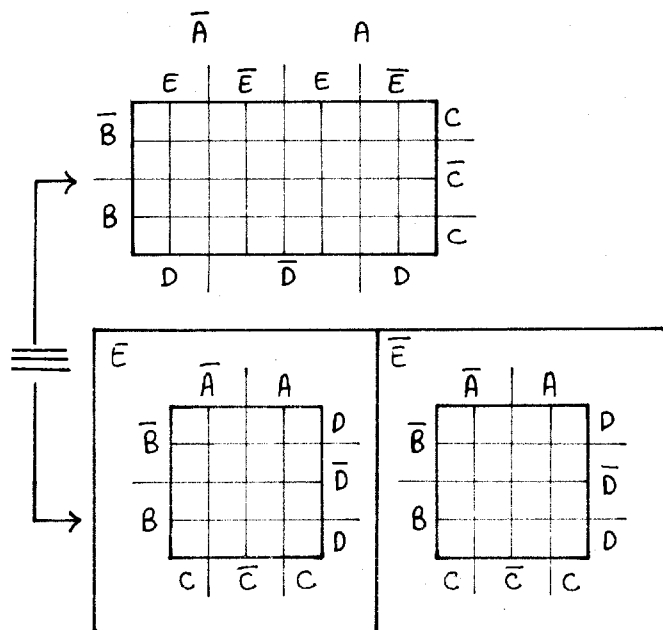
4 variables : A, B, C, D

A	B	C	D	S	$\bar{A}$	A	
0	0	0	0	s_0			
1	0	0	0	s_1	s_{12}	s_8	s_9
0	1	0	0	s_2	s_4	s_0	s_1
1	1	0	0	s_3	s_5	s_2	s_3
0	0	1	0	s_4			
1	0	1	0	s_5	s_6	s_2	s_3
0	1	1	0	s_6	s_{14}	s_{10}	s_{11}
1	1	1	0	s_7			
0	0	0	1	s_8			
1	0	0	1	s_9	s_{12}	s_8	s_9
0	1	0	1	s_{10}	s_4	s_0	s_1
1	1	0	1	s_{11}	s_5	s_2	s_3
0	0	1	1	s_{12}			
1	0	1	1	s_{13}	s_6	s_2	s_3
0	1	1	1	s_{14}	s_{14}	s_{10}	s_{11}
1	1	1	1	s_{15}			

6 variables A, B, C, D, E, F

		$\bar{A}$		A			
		E	$\bar{E}$	E	$\bar{E}$	E	
$\bar{B}$	F						D
	$\bar{F}$						
B	F						$\bar{D}$
	$\bar{F}$						
		C	$\bar{C}$	C			D

5 variables : A, B, C, D, E



Exemple de représentation d'une fonction dans le diagramme de VEITCH

Soit une fonction $f(A, B, C, D)$ dont la table de vérité est la suivante.

A	B	C	D	S
0	0	0	0	0 = 0
1	0	0	0	0 = 0
0	1	0	0	0 = 0
1	1	0	0	1 = $AB\bar{C}\bar{D}$
0	0	1	0	0 = 0
1	0	1	0	1 = $A\bar{B}C\bar{D}$
0	1	1	0	1 = $\bar{A}BC\bar{D}$
1	1	1	0	0 = 0
0	0	0	1	0 = 0
1	0	0	1	1 = $AB\bar{C}D$
0	1	0	1	1 = $\bar{A}B\bar{C}D$
1	1	0	1	0 = 0
0	0	1	1	1 = $\bar{A}\bar{B}CD$
1	0	1	1	0 = 0
0	1	1	1	0 = 0
1	1	1	1	0 = 0

Sa forme canonique aura pour expression :

$$S = AB\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D} + AB\bar{C}D + \bar{A}B\bar{C}D + \bar{A}\bar{B}CD$$

Et l'on pourra la représenter dans le diagramme de VEITCH de la façon suivante :

	$\bar{A}$	A	
$\bar{B}$	1 $\bar{A}BC\bar{D}$	0	1 $AB\bar{C}\bar{D}$
	0	0	1 $AB\bar{C}D$
B	1 $\bar{A}BCD$	0	1 $AB\bar{C}\bar{D}$
	0	1 $\bar{A}B\bar{C}D$	0
	C	$\bar{C}$	C

Nota :

Les termes inscrits dans les cases (sous le 1, valeur de la fonction) sont là pour illustrer l'exemple. On ne les y met pas habituellement. A noter que cette expression n'est pas réductible, car aucune case marquée 1 n'est adjacente à une autre.

Représentation des formes incomplètes

Soit une expression de la forme $f(A, B, C, D)$ exprimée de façon incomplète.

Exemple :

$$S = ABC + \bar{A}C$$

Pour représenter cette fonction dans le diagramme de Veitch, on dira :

1/ Pour ABC , la fonction est vraie, pour $D = 1$ et $D = 0$; on note deux fois la fonction ABC , une fois dans la case $ABCD$, une fois dans la case $AB\bar{C}D$.

2/ Pour $\bar{A}C$, la fonction est vraie pour $B = 1$ et $B = 0$
 $D = 1$ et $D = 0$
 On note 4 fois la fonction $\bar{A}C$ sous les formes : $\bar{A}CB\bar{D}$, $\bar{A}CB\bar{D}$, $\bar{A}CB\bar{D}$ et $\bar{A}CB\bar{D}$.

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

L'expression initiale : $S = ABC + \bar{A}C$

a pris la forme : $S = ABCD + AB\bar{C}D + \bar{A}BCD + \bar{A}\bar{B}CD + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}\bar{D}$

En pratique, lorsque l'opérateur est très habitué à l'utilisation du diagramme, il notera directement les fonctions dans la grille, deux fois chaque fois qu'il manque une variable.

Exemple : $ABC = 1$ dans D et $\bar{D}$
 $\bar{A}C = 1$ dans B et $\bar{B}$ et 1 dans D et $\bar{D}$.

Il arrive que deux termes différents donnent naissance à des 1 dans la même case. Dans ce cas, on n'écrit qu'un seul 1. Ceci revient à faire l'opération : $1 + 1 = 1$.

Utilisation du diagramme

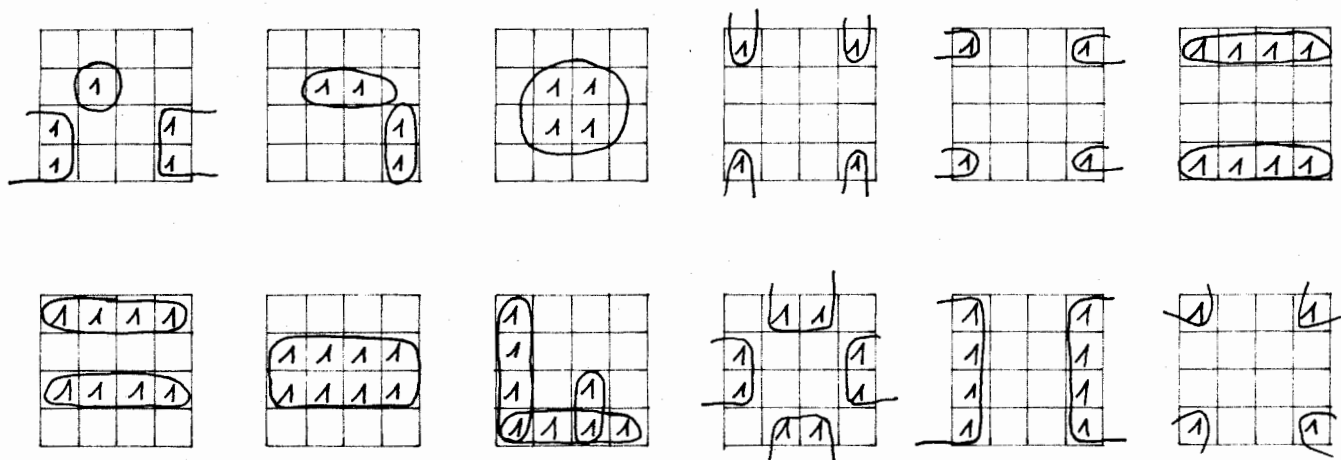
Pour minimiser les expressions, on groupe les cases contenant des 1 lorsqu'elles sont adjacentes par groupe de 2, 4, 8.

RAPPEL : Il faut effectuer des regroupements de blocs d'éléments adjacents les plus importants possible de façon à obtenir des intersections d'ordre le plus bas.

Règles de groupement

- Une même case peut appartenir à plusieurs groupements ;
- Toutes les cases contenant des 1 doivent être utilisées ;
- On ne peut faire de groupement que par puissance de 2 (1-2-4-8).

Types de groupement

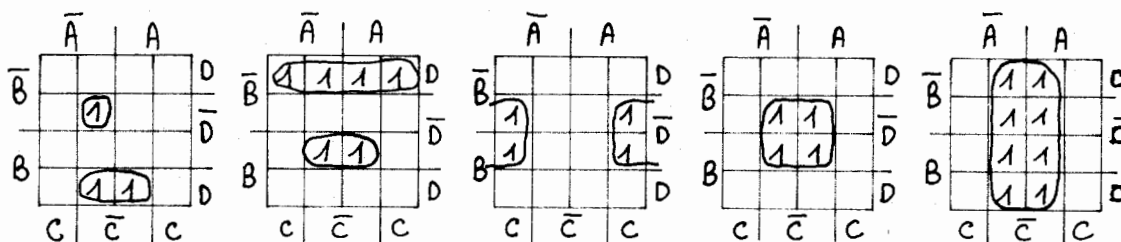


Lecture du diagramme - Minimisation

Lorsque l'opérateur pense avoir effectué les meilleurs groupements, il peut passer à la lecture du diagramme et écrire l'expression sous sa forme minimale. On dit que l'on est passé de la "forme maximale disjonctive" à la "forme minimale disjonctive" (dans le cas d'une forme de somme de produits).

Comme il se doit, la lecture s'opère à l'inverse de l'écriture.

Exemples :



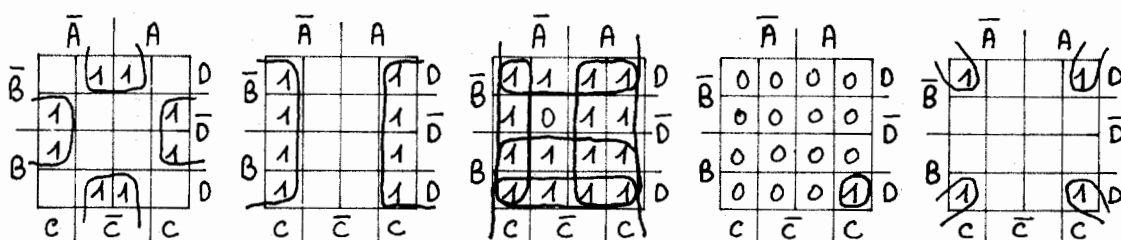
$$S = \bar{A}\bar{B}\bar{C}\bar{D} + B\bar{C}D$$

$$S = \bar{B}D + B\bar{D}\bar{C}$$

$$S = C\bar{D}$$

$$S = \bar{C}\bar{D}$$

$$S = \bar{C}$$



$$S = C\bar{D} + D\bar{C}$$

$$S = C$$

$$S = A + B + C + D$$

$$S = A.B.C.D$$

$$S = C.D$$

- Lors de l'écriture, on inscrit la valeur prise par la fonction pour le terme considéré dans la case correspondante à l'intersection des coordonnées.

- A la lecture, on prend les coordonnées du groupement en négligeant les variables qui s'annulent lorsque le bloc de variables regroupées est à la fois sur la variable vraie et inverse (puisque $A.\bar{A} = 0$). Il est donc évident que plus le groupement comporte de variables, plus le terme résultant est faible.

Pour un groupement de 1, le terme résultant comporte 4 variables.

Pour un groupement de 2, le terme résultant comporte 3 variables.

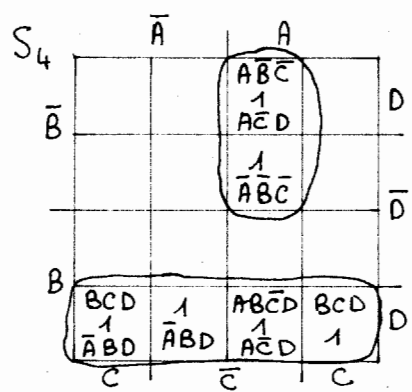
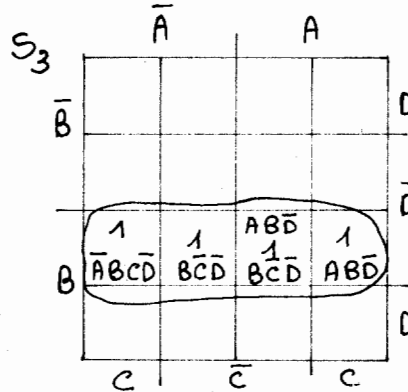
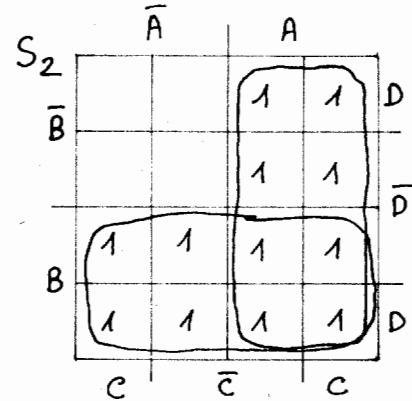
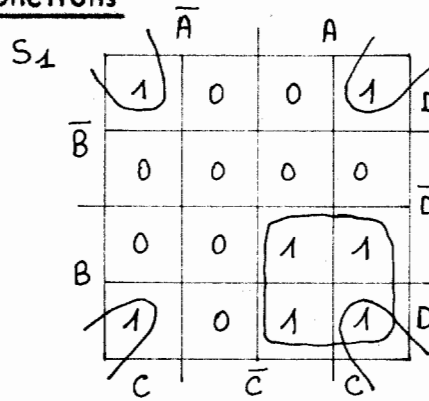
Pour un groupement de 4, le terme résultant comporte 2 variables.

Pour un groupement de 8, le terme résultant comporte 1 variable.

(Un groupement de 16 signifierait que la fonction est toujours égale à 1).

Exemples de simplification de fonctions

A	B	C	D	S ₁	S ₂	
0	0	0	0	0	0	
1	0	0	0	0	1	$\overline{A}\overline{B}\overline{C}\overline{D}$
0	1	0	0	0	1	$\overline{A}B\overline{C}\overline{D}$
1	1	0	0	1	1	$A\overline{B}\overline{C}\overline{D}$
0	0	1	0	0	0	
1	0	1	0	0	1	$\overline{A}\overline{B}C\overline{D}$
0	1	1	0	0	1	$\overline{A}BC\overline{D}$
1	1	1	0	1	1	$ABC\overline{D}$
0	0	0	1	0	0	
1	0	0	1	0	1	$\overline{A}\overline{B}CD$
0	1	0	1	0	1	$\overline{A}BCD$
1	1	0	1	1	1	$AB\overline{C}D$
0	0	1	1	1	0	$\overline{A}\overline{B}CD$
1	0	1	1	1	1	$\overline{A}BCD$
0	1	1	1	1	1	$\overline{A}BCD$
1	1	1	1	1	1	$ABCD$

FORMES COMPLETES
(maximales disjonctives)

$$S_1 = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + A\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D} + \overline{A}\overline{B}CD + \overline{A}BCD$$

$$S_2 = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}BC\overline{D} + \overline{A}\overline{B}CD + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + ABC\overline{D} + ABCD$$

FORMES INCOMPLETES

$$S_3 = \overline{B}\overline{C}\overline{D} + AB\overline{D} + \overline{A}BC\overline{D}$$

$$S_4 = A\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C} + \overline{A}BD + BCD$$

APRES SIMPLIFICATION
(minimales disjonctives)

$$S_1 = AB + CD$$

$$S_2 = A + B$$

APRES SIMPLIFICATION

$$S_3 = B\overline{D}$$

$$S_4 = \overline{A}\overline{B}C + BD$$

Disposition du diagramme

La disposition des variables autour du diagramme n'intervient pas, le tableau suivant où sont regroupées les 16 combinaisons de sens possibles avec 4 variables, le montre. Cependant, trois impératifs sont à respecter :

- 1/ Une variable ne peut recouper son inverse, elles sont du même côté du tableau ;
- 2/ Une variable et son inverse doivent couvrir tout le tableau, soit horizontalement, soit verticalement ;
- 3/ Toutes les variables et leurs inverses doivent être présentes.

Soit la fonction précédente : $S_4 = A\bar{C}D + AB\bar{C}D + A\bar{B}\bar{C} + \bar{A}BD + \bar{B}CD = A\bar{B}\bar{C} + BD$ représentée dans les 16 configurations possibles.

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

Ce tableau montre que, malgré leur place dans le diagramme, les groupements possibles sont toujours les mêmes et les résultats identiques.

Redondances (ou cas indifférents)

Il arrive que dans une table de vérité définissant une fonction S de plusieurs variables indépendantes, certaines combinaisons de ces variables ne soient jamais réalisées de par la nature même du processus défini par S .

Par exemple, si on veut représenter les nombres 0 à 9 par les combinaisons binaires de 4 variables, 6 de ces combinaisons ne se présenteront jamais.

Ces combinaisons interdites appelées "redondances" sont utilisées pour simplifier les expressions booléennes. On peut, en effet, considérer qu'elles rendent la fonction S égale à 0 ou 1, selon les possibilités de simplification sans pour autant modifier S puisqu'on est assuré qu'elles ne se produiront jamais.

C'est sur des diagrammes tels ceux de Karnaugh ou de Veitch que ces redondances sont le mieux interprétées.

Exemple :

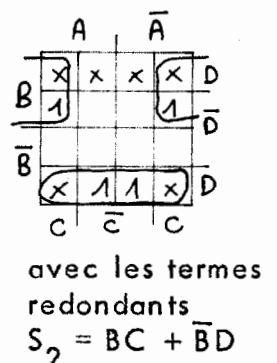
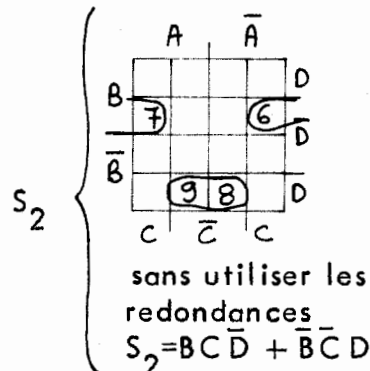
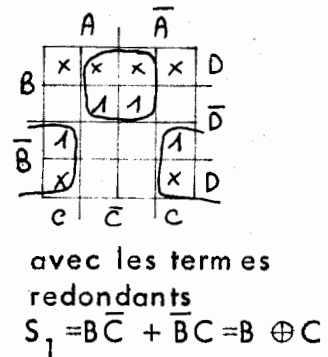
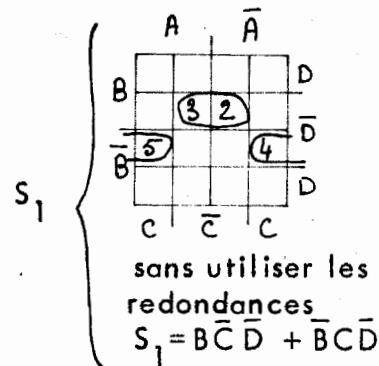
Soit $0 \leq N \leq 9$ exprimé en binaire dans le code 1,2,4,8. Trouver les fonctions booléennes S telles que :

$$\begin{aligned} S_1 &= 1 \text{ si } 2 \leq N \leq 5 \\ S_2 &= 1 \text{ si } 6 \leq N \leq 9 \end{aligned}$$

On écrit la table de vérité.

A	B	C	D	N	S_1	S_2
0	0	0	0	0	0	0
1	0	0	0	1	0	0
0	1	0	0	2	1	0
1	1	0	0	3	1	0
0	0	1	0	4	1	0
1	0	1	0	5	1	0
0	1	1	0	6	0	1
1	1	1	0	7	0	1
0	0	0	1	8	0	1
1	0	0	1	9	0	1
0	1	0	1	10	x	
1	1	0	1	11	x	
0	0	1	1	12	x	
1	0	1	1	13	x	
0	1	1	1	14	x	
1	1	1	1	15	x	

combinaisons interdites, "redondantes"



On a représenté les combinaisons interdites par des x pour montrer que leur valeur est indifférente et que l'on a le droit de les considérer comme des "0" ou des "1" de façon à pousser la simplification au maximum.

2^e PARTIELES CIRCUITS LOGIQUESLES CIRCUITS COMBINATOIRESDEFINITION

Dans un circuit combinatoire, l'état de la (ou des) sortie est uniquement fonction de l'état des variables d'entrée. Cet état reste stable tant que l'état des variables d'entrée n'est pas modifié.

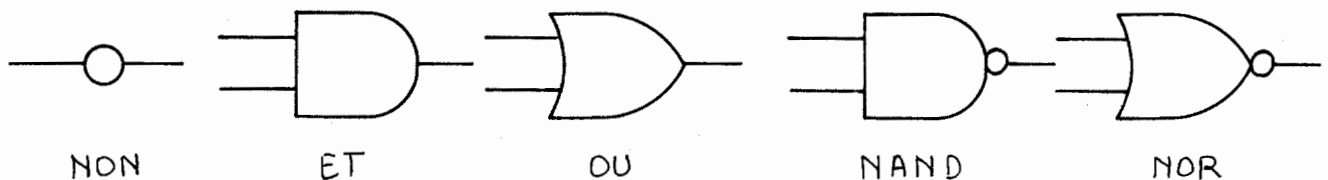
La sortie d'un circuit combinatoire est une fonction booléenne de ses entrées.

Une fonction de sortie déterminée peut être réalisée à l'aide de circuits de structures différentes.

I - CONSTRUCTION D'ENSEMBLES DE LOGIQUE COMBINATOIRE

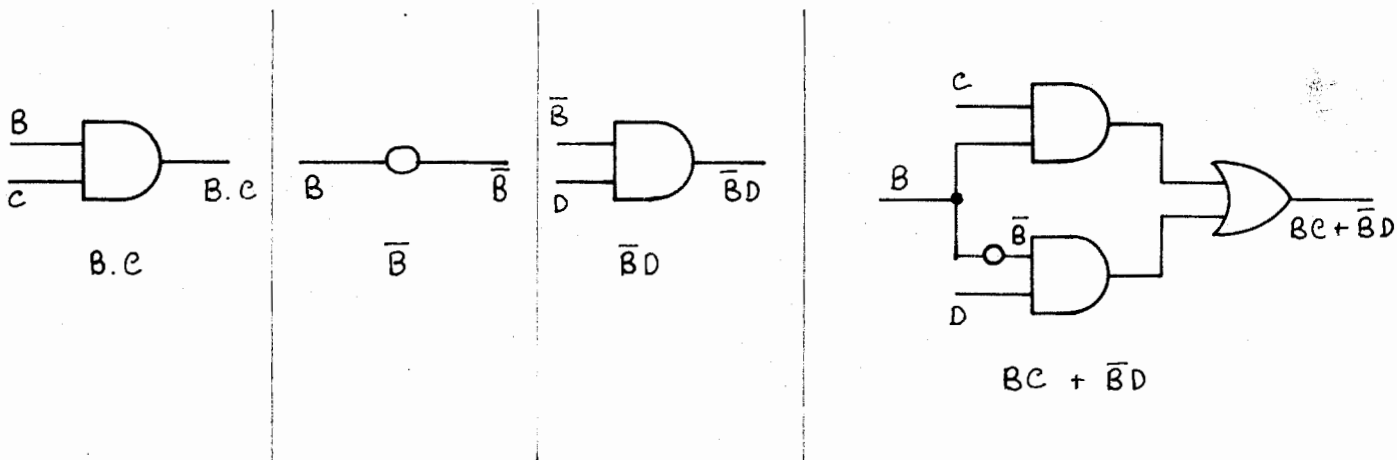
Dès lors que nous connaissons les règles fondamentales des fonctions de commutation et les techniques de minimisation, nous sommes en mesure de concevoir des circuits de logique combinatoire répondant à des fonctions complexes. Cette technique est au demeurant fort simple puisqu'elle consiste à transposer l'écriture d'une fonction booléenne en assemblage de blocs logiques fondamentaux ou secondaires, assemblage conduisant par opérations successives à la valeur désirée de la fonction.

Nous allons illustrer cette technique par quelques exemples dans lesquels nous utiliserons les opérateurs dont nous disposons, soit :

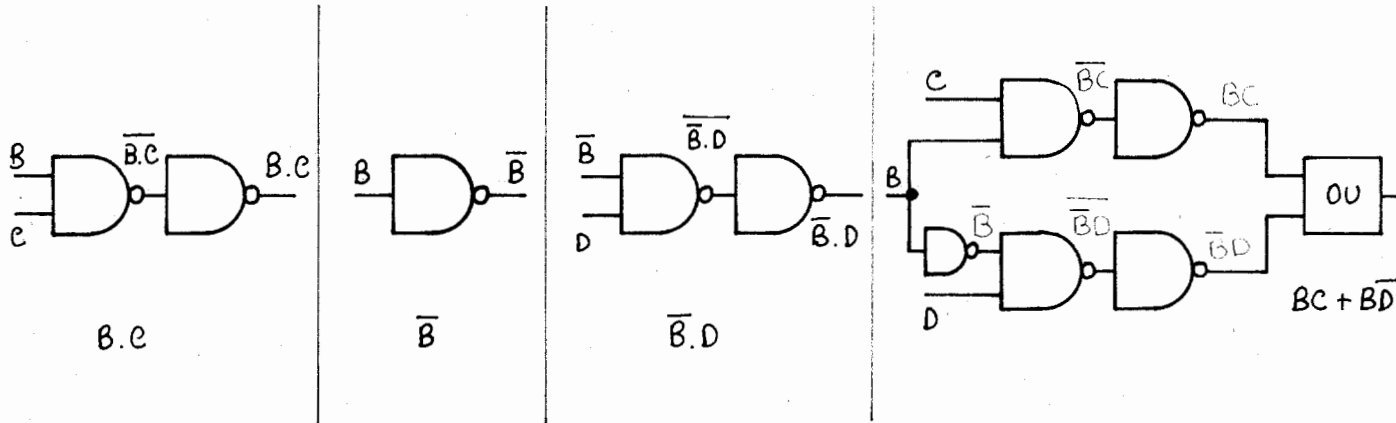


I-1 - Soit à construire la fonction : $S = BC + \bar{B}D$.

On peut décomposer les opérations de la façon suivante :



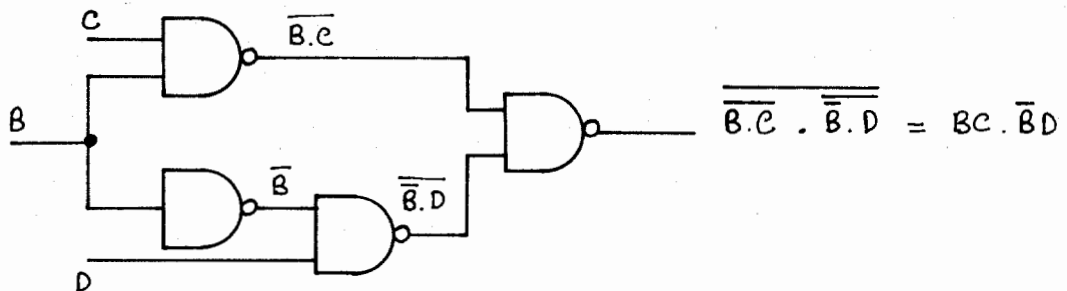
Si nous ne disposons que de fonctions "NAND", le schéma sera le suivant :



Comme nous n'utilisons que des "NAND", la fonction "OU" doit être réalisée avec ce circuit. Appliquons le théorème de De Morgan. Aux entrées du "OU", nous avons $\overline{B}D$, BC . Si nous prenons les inverses, nous aurions $\overline{B}C$, $\overline{B}D$ aux entrées d'une fonction "NAND". En sortie, nous aurions :

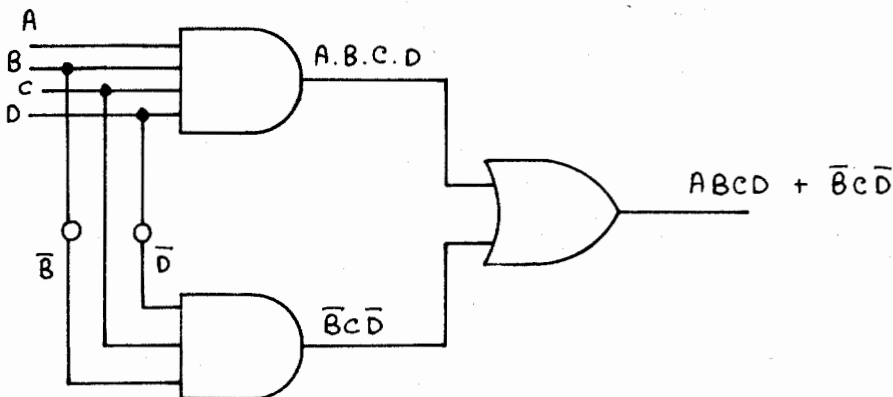
$$\overline{\overline{B}C} \cdot \overline{\overline{B}D} = \overline{\overline{B}C + \overline{B}D} = BC + \overline{B}D$$

Il ya donc 2 fonctions "NAND" en trop et le schéma devient :

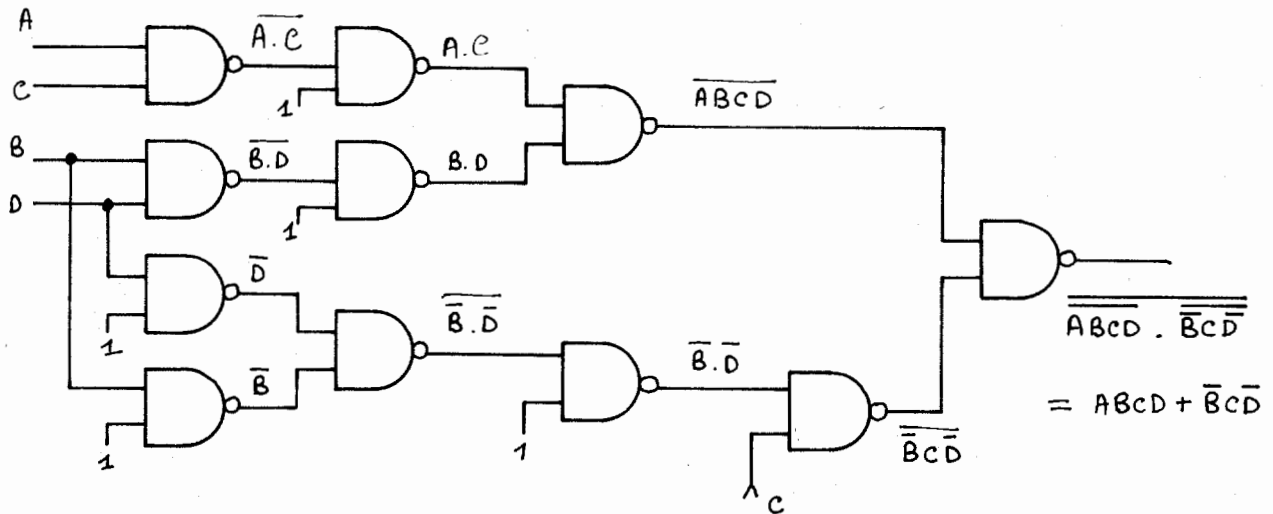


1 - 2 - Soit à construire la fonction : $S = ABCD + \overline{B}c\overline{D}$

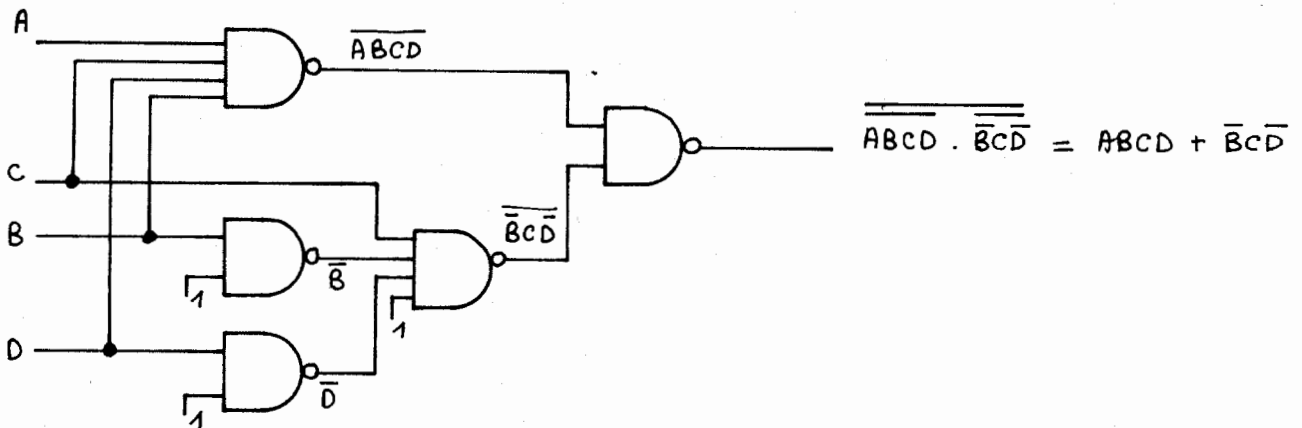
Le schéma utilisant les opérateurs fondamentaux donnera :



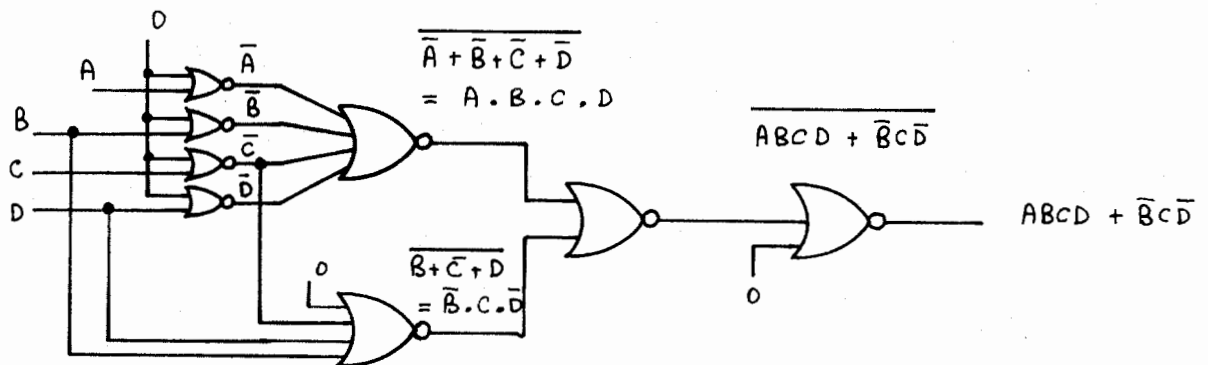
a) CONTRAINTE : Nous ne disposons que de circuits "NAND" à 2 entrées.



b) CONTRAINTE : Nous ne disposons que de circuits "NAND" à 4 et à 2 entrées.



c) CONTRAINTE : Nous ne disposons que de circuits "NOR" à 4 et à 2 entrées.



Ces quelques exemples montrent la grande variété de solutions qu'il est possible d'envisager pour parvenir au résultat. Un technicien habile parviendra facilement à réaliser de tels circuits au moindre coût par l'application judicieuse des fonctions booléennes et une bonne connaissance des types de circuits disponibles sur le marché.

II - CIRCUITS ARITHMETIQUES

Dans la catégorie des circuits combinatoires, on trouve les circuits arithmétiques, ensembles particuliers de logique combinatoire.

Nous avons vu dans le chapitre "opérations dans le système binaire" qu'il n'existe que deux opérations fondamentales en binaire : l'addition et la soustraction, la multiplication se ramenant à une suite d'additions successives, la division à une suite de soustractions.

Nous verrons plus loin que la soustraction elle-même en binaire se ramène à une addition.

II-1 - Additionneur binaire

Il s'agit de réaliser un circuit capable de faire la somme de 2 digits binaires et de fournir éventuellement le digit de report (ou retenue) comme le montre le chapitre relatif à l'addition dans le système binaire. La table de vérité de l'addition binaire est la suivante.

A	B	A+B	S	R
0	0	0	0	0
1	0	1	1	0
0	1	1	1	0
1	1	10	0	1

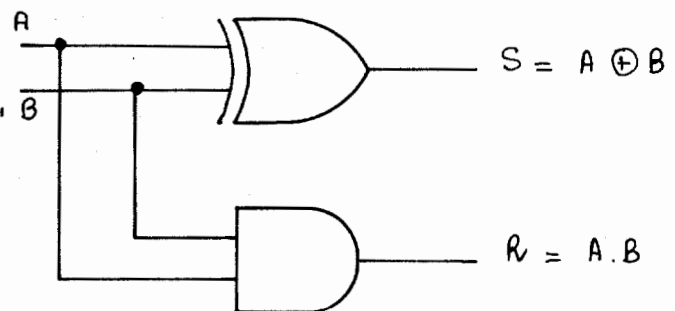
On voit immédiatement que la somme S est la somme modulo 2 de A et de B et le report R est l'intersection de A et B.

D'où l'on peut écrire :

$$\begin{cases} S = A \oplus B = A\bar{B} + \bar{A}B = (A+B) \cdot (\bar{A} + \bar{B}) \\ R = A \cdot B \end{cases}$$

Le schéma logique de l'additionneur élémentaire ou "semi-additionneur" sera donc réalisé comme il est indiqué ci-contre. On l'appelle "semi-additionneur" car il ne permet d'obtenir qu'une partie de l'addition, l'addition complète devant tenir compte de la retenue de l'addition précédente, retenue due à l'addition de tous les digits de rangs précédents.

En d'autres termes, un semi-additionneur ne permet de faire l'opération que sur 2 digits.



1/ Additionneur complet

Il doit réaliser la somme de 3 digits binaires et fournir le digit de report.

Soit A_n et B_n les nombres de rang n à additionner.

R_{n-1} le report dû à l'addition de tous les digits de rangs précédents.

S_n et R_n l'addition et le report de $A_n + B_n$.

s_n, r_n, r'_n la somme partielle et les restes partiels de l'addition.

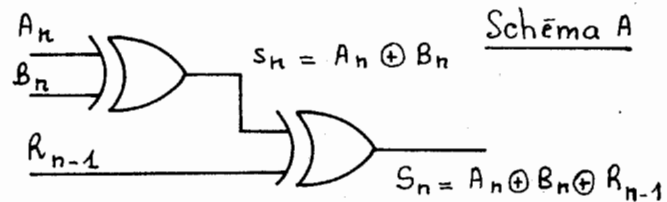
On a vu dans le chapitre des fonctions logiques secondaires comment réaliser un "OU exclusif" sur 3 variables.

On a le schéma suivant (A).

Sur ce schéma, on voit qu'il faut tenir compte d'une retenue par circuit $\oplus$,

soit :

$$\begin{cases} r_n = A_n \cdot B_n \\ r'_n = R_{n-1} \cdot (A_n \oplus B_n) = R_{n-1} \cdot s_n \end{cases}$$



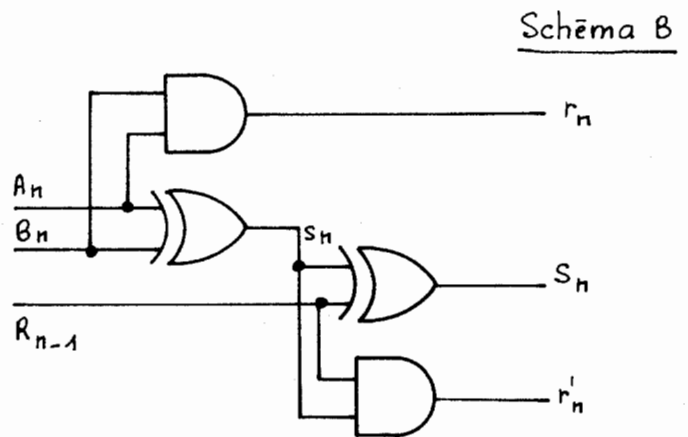
On obtient donc le schéma suivant (B) où

l'on voit qu'il reste deux retenues partielles r_n et r'_n . Or en examinant la table de vérité précédente, on voit que s'il y a un report dans la première addition (s_n, r_n), le digit de somme s_n est un zéro.

Si $s_n = 0$, R_{n-1} peut être égal à 1 ou à zéro.

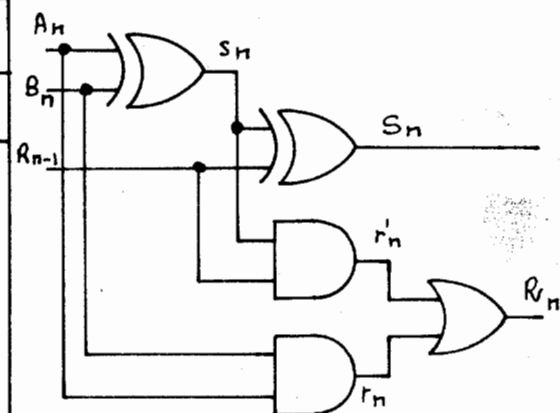
Il n'y aura pas de retenue et r'_n sera égal à zéro.

On aura à ce moment $r_n = 1$ et $r'_n = 0$.



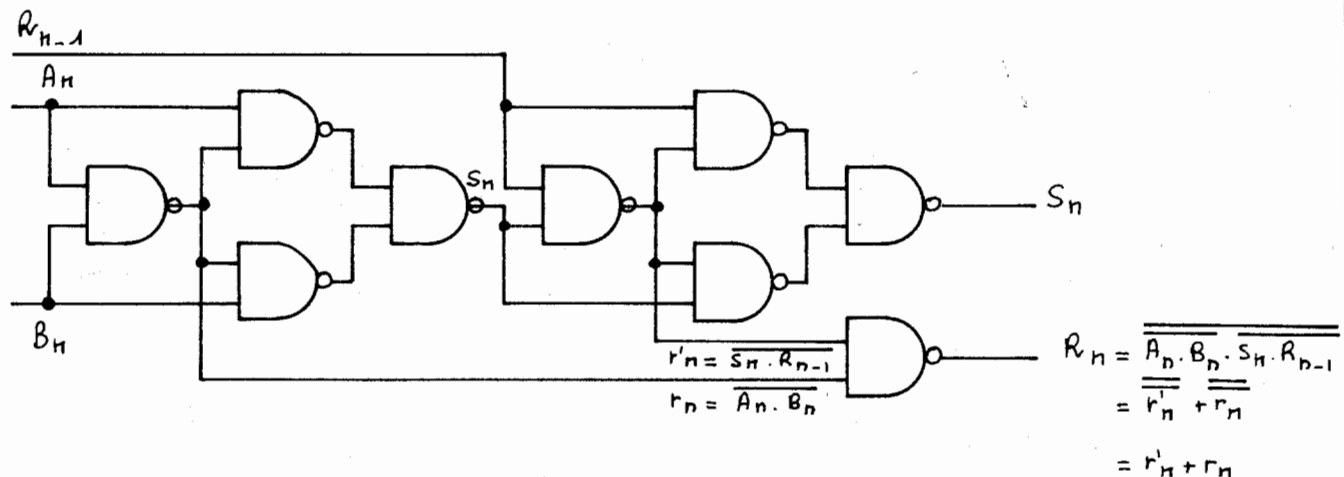
Le même raisonnement pouvant être appliqué lorsque $r_n = 0$, on constate que les restes partiels r_n et r'_n ne sont jamais en même temps égaux à 1. Il suffit donc de prendre l'un ou l'autre pour connaître le reste global de l'addition d'où le schéma et le table de vérité de l'additionneur binaire complet.

Variables d'entrée			Variables intermédiaires			Variables de sorties	
A_n	B_n	R_{n-1}	$S_n = A_n \oplus B_n$	$r_n = A_n \cdot B_n$	$r'_n = s_n \cdot R_{n-1}$	$S_n \oplus R_{n-1}$	$R_n = r_n + r'_n$
0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
0	1	0	1	0	0	1	0
1	1	0	0	1	0	0	1
0	0	1	0	0	0	1	0
1	0	1	1	0	1	0	1
0	1	1	1	0	1	0	1
1	1	1	0	1	0	1	1



L'additionneur est constitué de 2 semi-additionneurs et d'une fonction "OU".

A titre d'exemple, il est intéressant de montrer un additionneur réalisé avec des circuits "NAND".



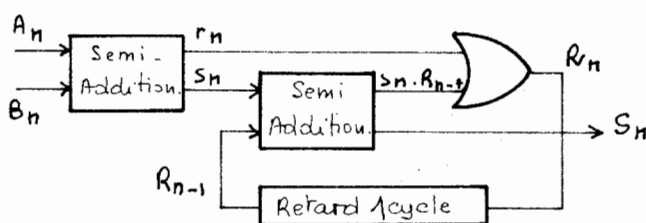
On a utilisé pour cette fonction 9 circuits "NAND". Deux fois quatre circuits réalisant 2 fonctions $\oplus$ et un circuit réalisant la fonction "OU". Le fonctionnement du $\oplus$ réalisé avec des circuits "NAND" est décrit au chapitre "Fonctions logiques secondaires" dans la paragraphe "Ou exclusif".

2/ Additionneurs série et parallèle

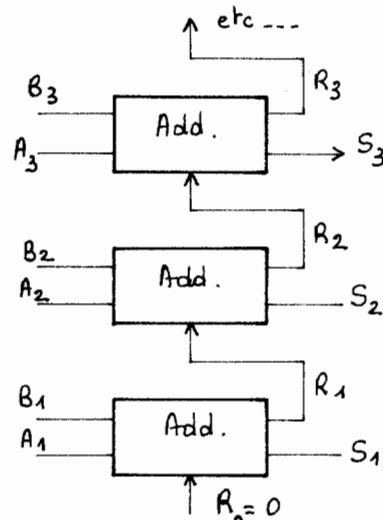
Il existe deux types principaux d'additionneurs binaires : l'additionneur série et l'additionneur parallèle. L'additionneur série effectue l'addition des deux nombres binaires poids à poids, les poids faibles arrivant les premiers. Le résultat S_n est donc obtenu pas à pas, le reste R est réintégré au calcul par l'intermédiaire d'une ligne à retard d'une durée égale à un cycle de calcul. La durée totale du calcul sera donc proportionnelle au nombre de digit constituant les nombres à additionner.

Dans l'additionneur parallèle, tous les digits constituant les nombres à additionner sont présents au même instant. Les reports sont transmis des poids faibles vers les poids forts de la même façon qu'en arithmétique. Ce circuit a l'avantage d'être rapide puisque, par rapport à l'additionneur série, quel que soit le nombre de digits à additionner, il ne représente qu'un cycle de calcul du système série.

Par contre, il présente l'inconvénient majeur de nécessiter autant d'additionneurs complets que les nombres à additionner comportent de digits.



Structure d'un additionneur série



Structure d'un additionneur parallèle

Il existe d'autres types d'additionneurs. Citons pour mémoire, car nous ne pouvons pas tous les étudier sans sortir du cadre de cet exposé :

- l'addition en excédent de 3 ;
- l'additionneur synchrone ;
- l'additionneur asynchrone ;
- l'additionneur à prévision de report ;
- l'additionneur à sommes conditionnelles ;
- l'additionneur à registres.

Dans presque tous les cas, le but poursuivi est l'addition du plus grand nombre possible de digits dans le délai le plus bref possible.

II-2 - Soustracteur binaire

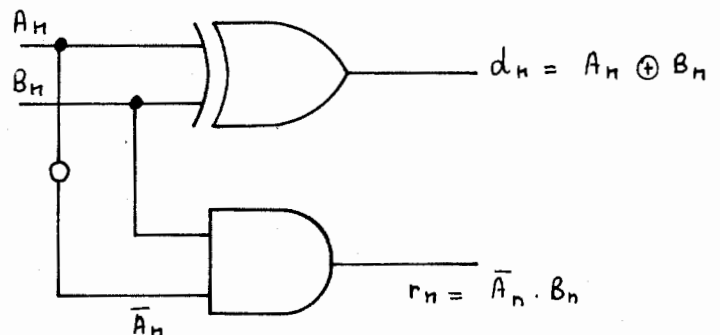
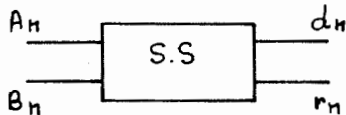
De même que l'additionneur, il s'agit de réaliser un circuit capable de faire la soustraction de 2 nombres binaires et de fournir le digit de report éventuel. Dans le chapitre relatif à la soustraction dans le système binaire, on donne la table de vérité suivante :

A_n	B_n	$d_n = A_n - B_n$	r_n
0	0	0	0
1	0	1	0
0	1	1	1
1	1	0	0

d'où l'on tire les relations suivantes :

$$\begin{cases} d_n = \bar{A}_n \cdot B_n + A_n \cdot \bar{B}_n = (A_n + B_n) \cdot (\bar{A}_n + \bar{B}_n) = A_n \oplus B_n \\ r_n = \bar{A}_n \cdot B_n \end{cases}$$

On appelle semi-soustracteur l'organe logique capable de réaliser ces fonctions. On le représente de la façon suivante :

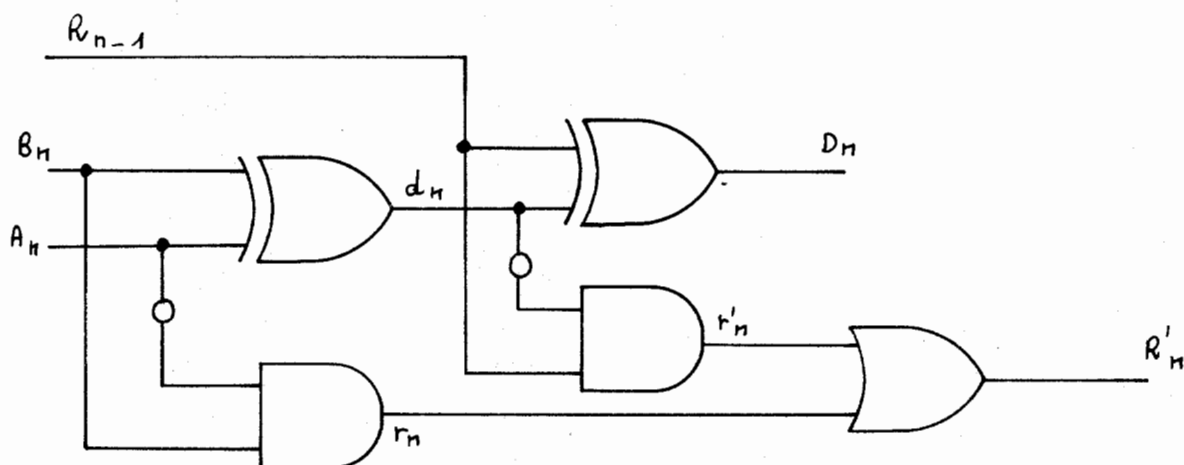


Mais, de même que dans l'additionneur binaire, on est obligé de tenir compte de la retenue de rang précédent R_{n-1} due à la soustraction successive de tous les digits précédents.

La structure d'un soustracteur complet est semblable à celle d'un additionneur et il satisfait au même raisonnement sur les restes partiels que l'additionneur.

Il va de soi de que l'on peut mettre indifféremment le premier inverseur sur A_n ou B_n sans modifier le résultat. Par contre, le report des opérations précédentes ne doit pas être modifié, le second inverseur sera donc impérativement sur d_n .

Variables d'entrée			Variables intermédiaires			Variables de sortie	
A_n	B_n	R_{n-1}	$d_n = A_n \oplus B_n$	$r_n = \bar{A}_n \cdot B_n$	$r'_n = \bar{d}_n \cdot R_{n-1}$	$D_n = d_n \oplus R_{n-1}$	$R'_n = r'_n + r_n$
0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
0	1	0	1	1	0	1	1
1	1	0	0	0	0	0	0
0	0	1	0	0	1	1	1
1	0	1	1	0	0	0	0
0	1	1	1	1	0	0	1
1	1	1	0	0	1	1	0



Il est bien entendu possible de réaliser des soustracteurs série et parallèle ayant la même constitution que les additionneurs de même type.

NOTA : Une autre méthode de soustraction de nombres binaires, consistant à ramener la soustraction à une simple addition est celle décrite dans le chapitre relatif aux opérations dans le système binaire et utilisant le complément restreint, c'est à dire le complément à $2^n - 1$ compte tenu du digit de signe.

II-3 Additionneur/Soustracteur

Les tables de vérité et les schémas logiques montrent que les structures de l'additionneur et du soustracteur sont semblables. Nous pouvons, pour récapituler, donner les expressions booléennes complètes des 2 types d'opérateurs.

Addition

$$S_n = A_n \oplus B_n \oplus R_{n-1}$$

$$R_n = (A_n \cdot B_n) + (A_n \oplus B_n \cdot R_{n-1})$$

Soustraction

$$D_n = A_n \oplus B_n \oplus R_{n-1}$$

$$R'_n = (A_n \cdot B_n) + (A_n \oplus B_n \cdot R_{n-1})$$

Ces relations montrent qu'un même circuit peut réaliser soit l'une, soit l'autre opération, à condition de disposer d'une variable auxiliaire validant ou non les 2 inverseurs. On appelle ce circuit "additionneur/soustracteur binaire".

Pour conclure, disons que ce type d'opérateur n'est pratiquement plus réalisé selon les formes décrites dans ce chapitre car il existe sur le marché depuis quelques années des additionneurs complets sous forme intégrée (du type SN 7483). Il est néanmoins utile de connaître le fonctionnement détaillé de ces opérateurs, cette expérience permettant une meilleure compréhension de circuits plus complexes.

III - COMPAREUR BINAIRE

Outre la fonction secondaire identité indiquant : $A = B$ ou $A \neq B$, il existe des comparateurs binaires donnant le sens de la différence entre deux nombres binaires, soit :

$$A = B \quad A > B \quad A < B$$

Ces comparateurs sont appelés "comparateurs d'amplitude binaire" ou, en anglais "Bit Magnitude Comparator".

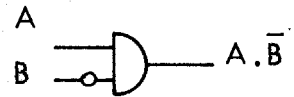
COMPARAISON DE DEUX POIDS BINAIRES

Soient 2 poids binaires A et B dont on veut connaître le sens de la différence.

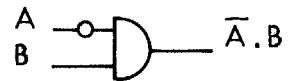
On peut écrire une table de vérité comportant 5 variables de sortie. Dans la pratique, 2 ou 3 de ces variables seront utilisées.

		x	y	s	x'	y'
A	B	$A > B$	$A < B$	$A = B$	$A \geq B$	$A \leq B$
0	0	0	0	1	1	1
0	1	0	1	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	1	1

$$x = A > B = A \cdot \bar{B}$$

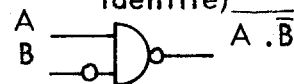


$$y = A < B = \bar{A} \cdot B$$

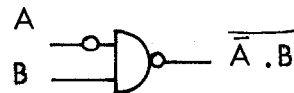


$$S = A = B = \bar{A}\bar{B} + AB = A \oplus B \text{ (fonction identité)}$$

$$x' = A \geq B = \overline{A \cdot \bar{B}}$$

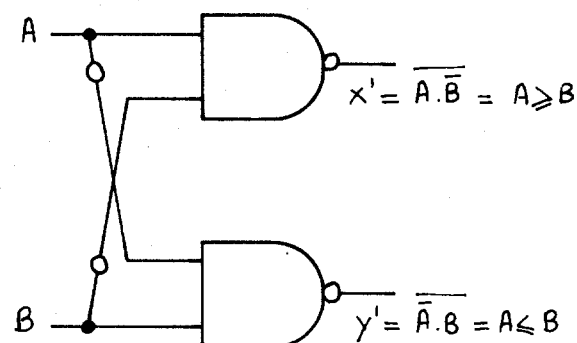


$$y' = A \leq B = \overline{\bar{A} \cdot B}$$

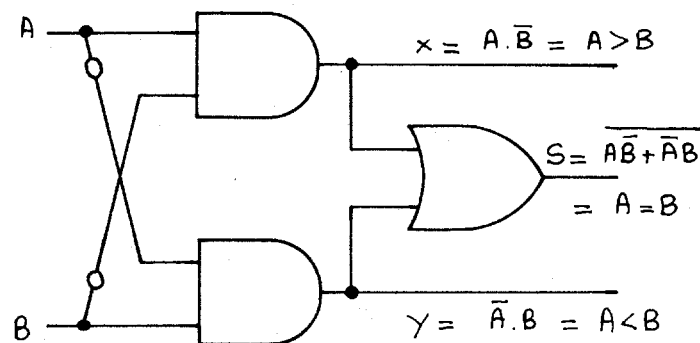


On peut tirer de ces expressions deux schémas logiques d'un comparateur binaire.

1/ Comparateur binaire à 2 sorties (x' , y')



2/ Comparateur binaire à 3 sorties (x , y , s)



COMPARATEUR D'AMPLITUDE BINAIRE A ENTREES MULTIPLES

Lorsque l'on doit comparer 2 nombres, la manière la plus rapide est de commencer la comparaison par les digits les plus significatifs.

Si les digits de plus fort poids sont différents, l'information est suffisante et les sorties du comparateur indiqueront le sens de cette différence. Si les digits sont égaux, le comparateur de rang inférieur doit être validé pour effectuer la comparaison sur ses digits et ainsi de suite.

Exemple : Soient à comparer 2 nombres binaires : $\begin{cases} A^n + A^{n-1} + A^{n-2} + \dots \\ B^n + B^{n-1} + B^{n-2} + \dots \end{cases}$

Si $A^n \neq B^n \rightarrow S = 0 \rightarrow x \text{ ou } y = 1 \rightarrow$ les sorties indiquent le sens de la différence.

Si $A^n = B^n \rightarrow S = 1 \rightarrow x = y = 0$

$\begin{cases} A^{n-1} \neq B^{n-1} \rightarrow S = 0 \rightarrow x \text{ ou } y = 1 \rightarrow \text{les sorties indiquent le sens de la différence.} \\ A^{n-1} = B^{n-1} \rightarrow S = 1 \rightarrow x = y = 0 \end{cases}$

$\begin{cases} A^{n-2} \neq B^{n-2} \rightarrow S = 0 \rightarrow x \text{ ou } y = 1 \rightarrow \text{les sorties indiquent le sens de la différence.} \\ A^{n-2} = B^{n-2} \rightarrow S = 1 \rightarrow x = y = 0 \end{cases}$

	n	n-1	n-2
A	1	0	0
B	0	1	0

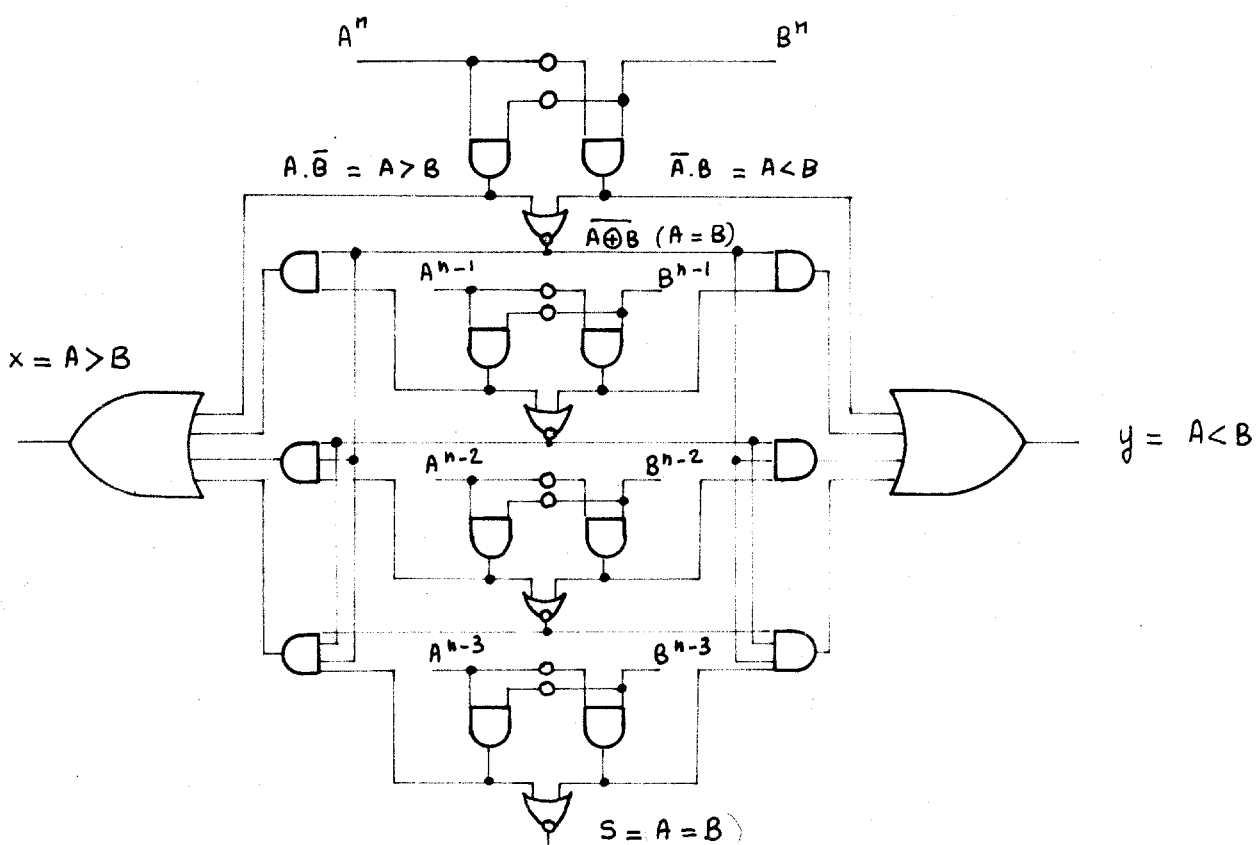
$A^n > B^n$
 $A > B$

	n	n-1	n-2
A	1	0	0
B	1	1	0

$A^{n-1} < B^{n-1} = A < B$
 $S=1$
 $A^n = B^n$

Réalisation

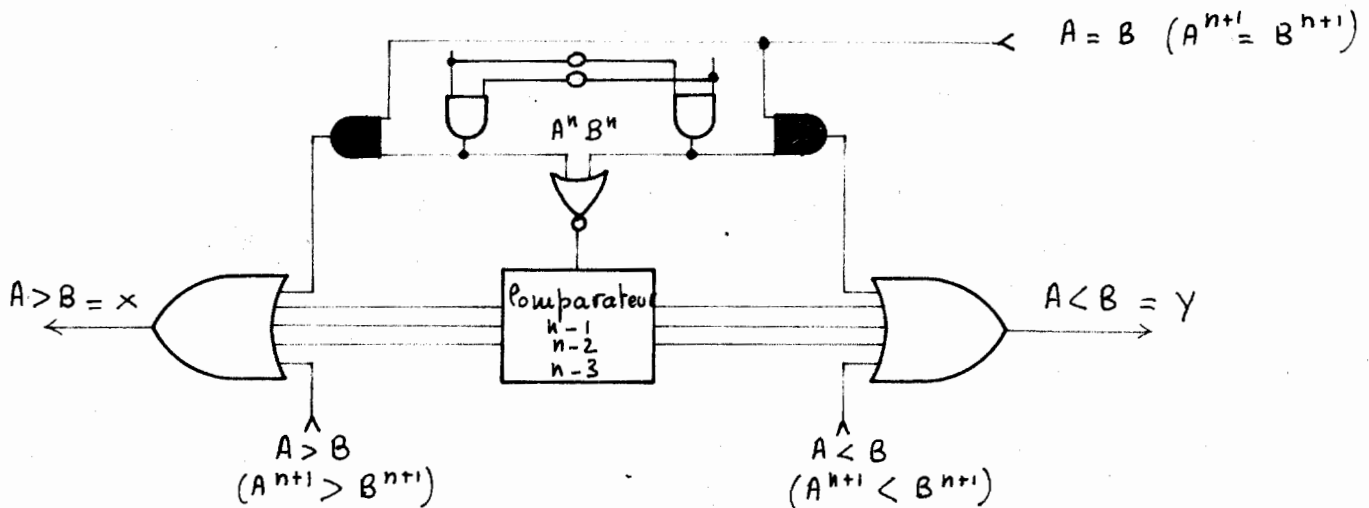
En prenant comme élément de base le comparateur binaire à 3 sorties, on obtient la structure ci-dessous dans laquelle on remarque la validation de tous les comparateurs de rangs inférieurs à n par les égalités de rangs supérieurs.



La complexité du circuit s'accroît très vite avec le nombre de variables. Aussi, dans la pratique, les constructeurs se limitent (actuellement) à la fabrication de comparateurs à 4 variables (DM 8200, SN 7485, etc...) ayant quelquefois la possibilité d'association en cascade (SN 7485).

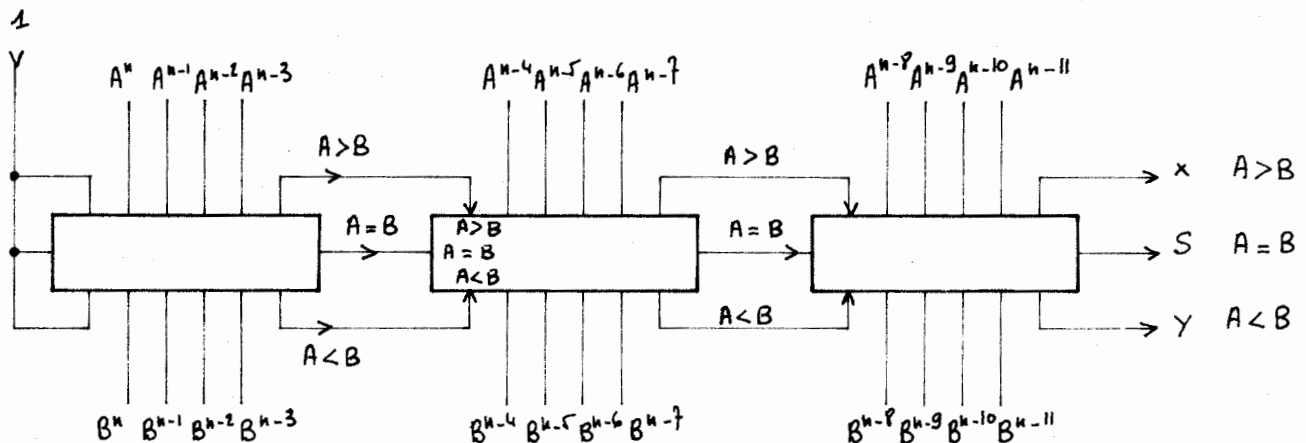
Cette possibilité est réalisée par 3 entrées supplémentaires :

$A > B$ $A < B$ $A = B$ disposées de la façon suivante :



ASSOCIATION DE COMPAREURS BINAIRES EN CASCADE

Les comparateurs utilisés sont du type SN 7485. Les entrées $A = B$, $A > B$, $A < B$ du premier comparateur (chiffres les plus significatifs) sont mises au niveau logique "1".



IV - LES CONVERTISSEURS DE CODES - LE TRANSCODAGE

La terminologie des ouvrages spécialisés regroupe habituellement sous le terme de "convertisseurs" : les codeurs, les décodeurs, les transcodeurs.

Les codeurs ont généralement pour objet la conversion d'un signal analogique en code binaire ainsi que la conversion d'un nombre décimal en binaire.

Les décodeurs effectuent l'opération inverse.

Quant aux transcodeurs, ils réalisent les conversions de codes binaires en d'autres codes binaires. Un tableau donnera plus précisément ces définitions.

OPERATION	conversion
CODAGE	décimal ou analogique $\longrightarrow$ code binaire
DECODAGE	code binaire $\longrightarrow$ décimal ou analogique
TRANSCODAGE	code binaire A $\longrightarrow$ code binaire B

Nous ne traiterons dans ce chapitre, relatif aux circuits combinatoires, que des convertisseurs numérique/numérique, les convertisseurs analogique/numérique faisant l'objet d'un autre chapitre.

Il est évidemment impossible de passer en revue tous les convertisseurs ; leur nombre étant lié à toutes les possibilités d'arrangement des codes binaires, on dénombrera 2^{n^2} convertisseurs (décodeurs-transcodeurs) n étant le nombre de codes possibles. La méthode générale de calcul d'un convertisseur étant identique à celle d'un circuit combinatoire quelconque. Nous étudierons quelques exemples classiques des convertisseurs les plus courants.

CODAGE-DECODAGE

Il est bien entendu toujours possible de convertir un nombre décimal en un quelconque code binaire. Toutefois, la conversion la plus utilisée (des circuits intégrés réalisant ces fonctions existent sur le marché) est la conversion décimal/DCB (décimal codé binaire) et son inverse DCB/décimal.

IV-1 - Codeur décimal/DCB

Ecrivons la table de vérité :

	B1	B2	B3	B4
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1

D'où l'on tire la forme canonique des fonctions de sorties :

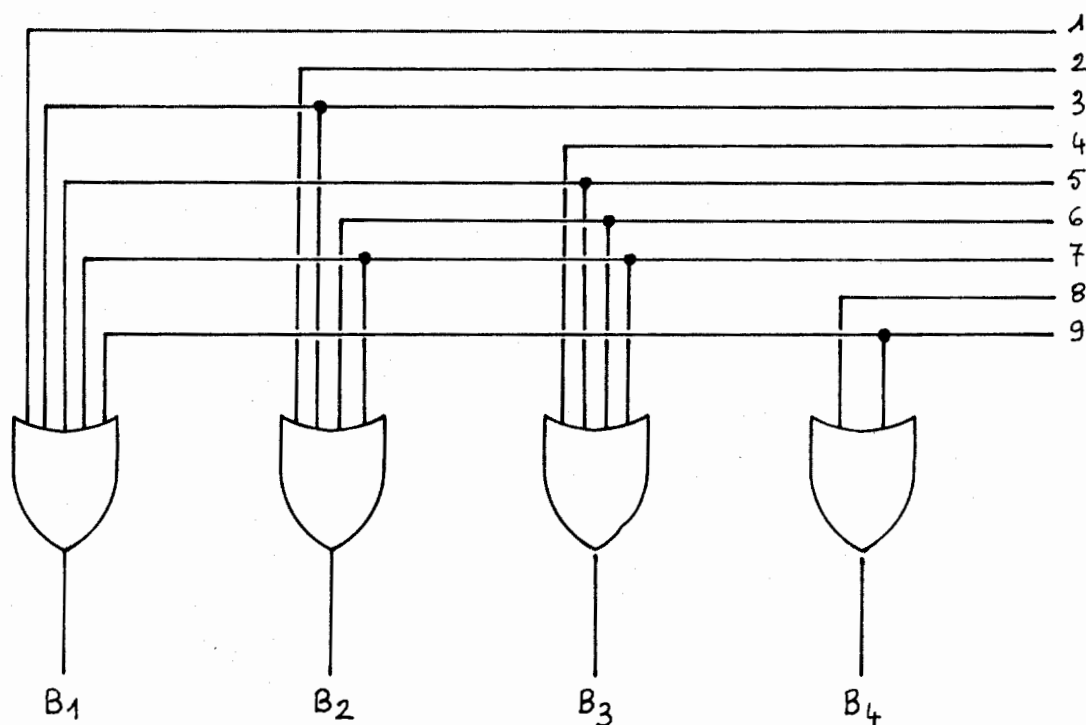
$$B1 = 1 + 3 + 5 + 7 + 9$$

$$B2 = 2 + 3 + 6 + 7$$

$$B3 = 4 + 5 + 6 + 7$$

$$B4 = 8 + 9$$

Le schéma logique n'est constitué que de circuits "OU".



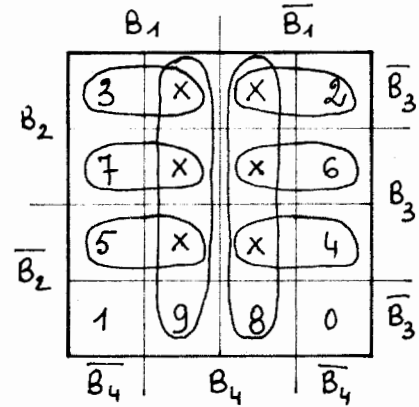
IV-2 - Décodeur DCB/décimal

De la table de vérité précédente, on tire les 10 fonctions pour les chiffres de 0 à 9.

$$\begin{aligned}
 0 &= \overline{B_1}.\overline{B_2}.\overline{B_3}.\overline{B_4} & 5 &= B_1.\overline{B_2}.B_3.\overline{B_4} \\
 1 &= B_1.\overline{B_2}.\overline{B_3}.\overline{B_4} & 6 &= \overline{B_1}.B_2.B_3.\overline{B_4} \\
 2 &= \overline{B_1}.B_2.\overline{B_3}.\overline{B_4} & 7 &= B_1.B_2.B_3.\overline{B_4} \\
 3 &= B_1.B_2.\overline{B_3}.\overline{B_4} & 8 &= \overline{B_1}.\overline{B_2}.\overline{B_3}.B_4 \\
 4 &= \overline{B_1}.\overline{B_2}.B_3.\overline{B_4} & 9 &= B_1.\overline{B_2}.\overline{B_3}.B_4
 \end{aligned}$$

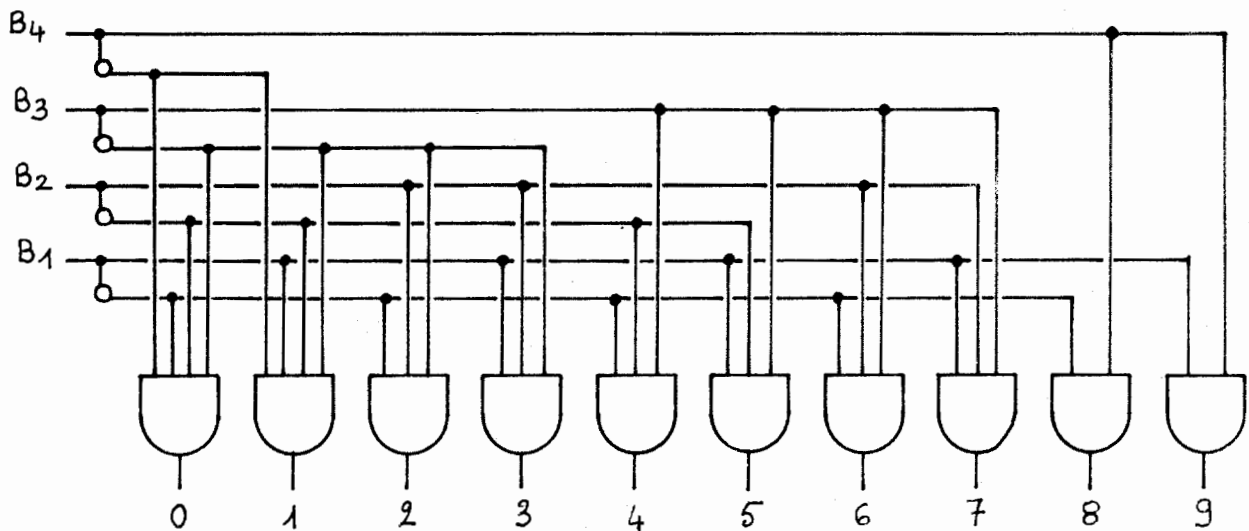
En utilisant les termes redondants (10 à 15) dans le diagramme de Veitch, on obtient après simplification :

$$\begin{aligned}
 0 &= \overline{B_1}.\overline{B_2}.\overline{B_3}.\overline{B_4} & 5 &= B_1.\overline{B_2}.B_3 \\
 1 &= B_1.\overline{B_2}.\overline{B_3}.\overline{B_4} & 6 &= \overline{B_1}.B_2.B_3 \\
 2 &= \overline{B_1}.B_2.\overline{B_3} & 7 &= B_1.B_2.B_3 \\
 3 &= B_1.B_2.\overline{B_3} & 8 &= \overline{B_1}.B_4 \\
 4 &= \overline{B_1}.\overline{B_2}.B_3 & 9 &= B_1.B_4
 \end{aligned}$$

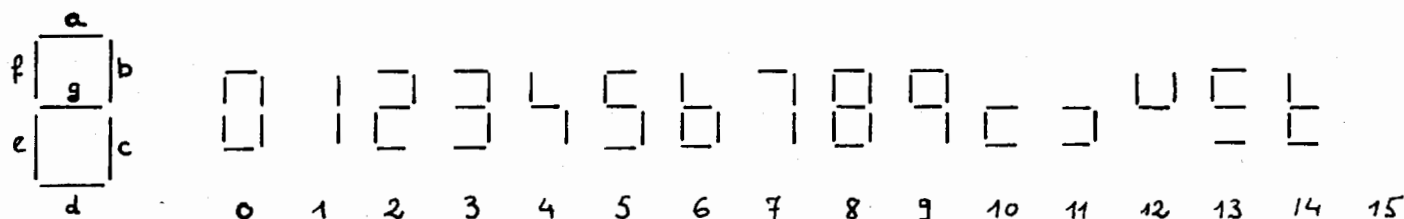


Le circuit n'est constitué que de portes ET et d'inverseurs.

Il existe sur le marché des circuits de ce type (SN 7442, SN 7445, SN 74141, SN 74145, DM 8301, etc...).



Il s'agit, à partir du code binaire naturel, d'afficher le chiffre correspondant en décimal par l'intermédiaire d'un dispositif comportant 7 segments indépendants qui, lorsqu'ils sont excités, s'illuminent (diodes électro-luminescentes, nixies, spuries) ou diffractent la lumière (cristaux liquides). Ils sont disposés de la façon suivante :



Etablissons une correspondance entre la table binaire et l'affichage

B_1	B_2	B_3	B_4	Décimal	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
0	1	0	0	2	1	1	0	1	1	0	1
1	1	0	0	3	1	1	1	1	0	0	1
0	0	1	0	4	0	0	1	0	0	1	1
1	0	1	0	5	1	0	1	1	0	1	1
0	1	1	0	6	0	0	1	1	1	1	1
1	1	1	0	7	1	1	1	0	0	0	0
0	0	0	1	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	1	1	0	1	1
0	1	0	1	10	0	0	0	1	1	0	1
1	1	0	1	11	0	0	1	1	0	0	1
0	0	1	1	12	0	1	0	0	0	1	1
1	0	1	1	13	1	0	0	1	0	1	1
0	1	1	1	14	0	0	0	1	1	1	1
1	1	1	1	15	0	0	0	0	0	0	0

Les expressions, sous forme canonique de sommes de produits, seront pour chaque segment :

$$a = 0+2+3+5+7+8+9+13 \text{ soit en binaire}$$

$$a = \overline{B_1}\overline{B_2}\overline{B_3}\overline{B_4} + \overline{B_1}B_2\overline{B_3}\overline{B_4} + \overline{B_1}B_2\overline{B_3}B_4 + \overline{B_1}B_2B_3\overline{B_4} + \overline{B_1}B_2B_3B_4 + B_1\overline{B_2}\overline{B_3}\overline{B_4} + B_1\overline{B_2}\overline{B_3}B_4 + B_1\overline{B_2}B_3\overline{B_4} + B_1\overline{B_2}B_3B_4$$

$$b = 0+1+2+3+7+8+12$$

$$c = 0+1+3+4+5+6+7+8+9+11$$

$$d = 0+2+3+5+6+8+9+10+11+13+14$$

$$e = 0+2+6+8+10+14$$

$$f = 0+4+5+6+8+9+12+13+14$$

$$g = 2+3+4+5+6+8+9+10+11+12+13+14$$

Après simplification par le diagramme de Veitch de chacune de ces expressions, nous obtenons les formes minimales suivantes :

$$a = \overline{B_1}B_2\overline{B_4} + \overline{B_1}\overline{B_2}B_3 + \overline{B_2}\overline{B_3}B_4 + \overline{B_3}\overline{B_3}B_4$$

$$b = \overline{B_2}\overline{B_3} + \overline{B_1}B_2\overline{B_4} + \overline{B_1}\overline{B_2}B_4 + \overline{B_1}\overline{B_3}\overline{B_4}$$

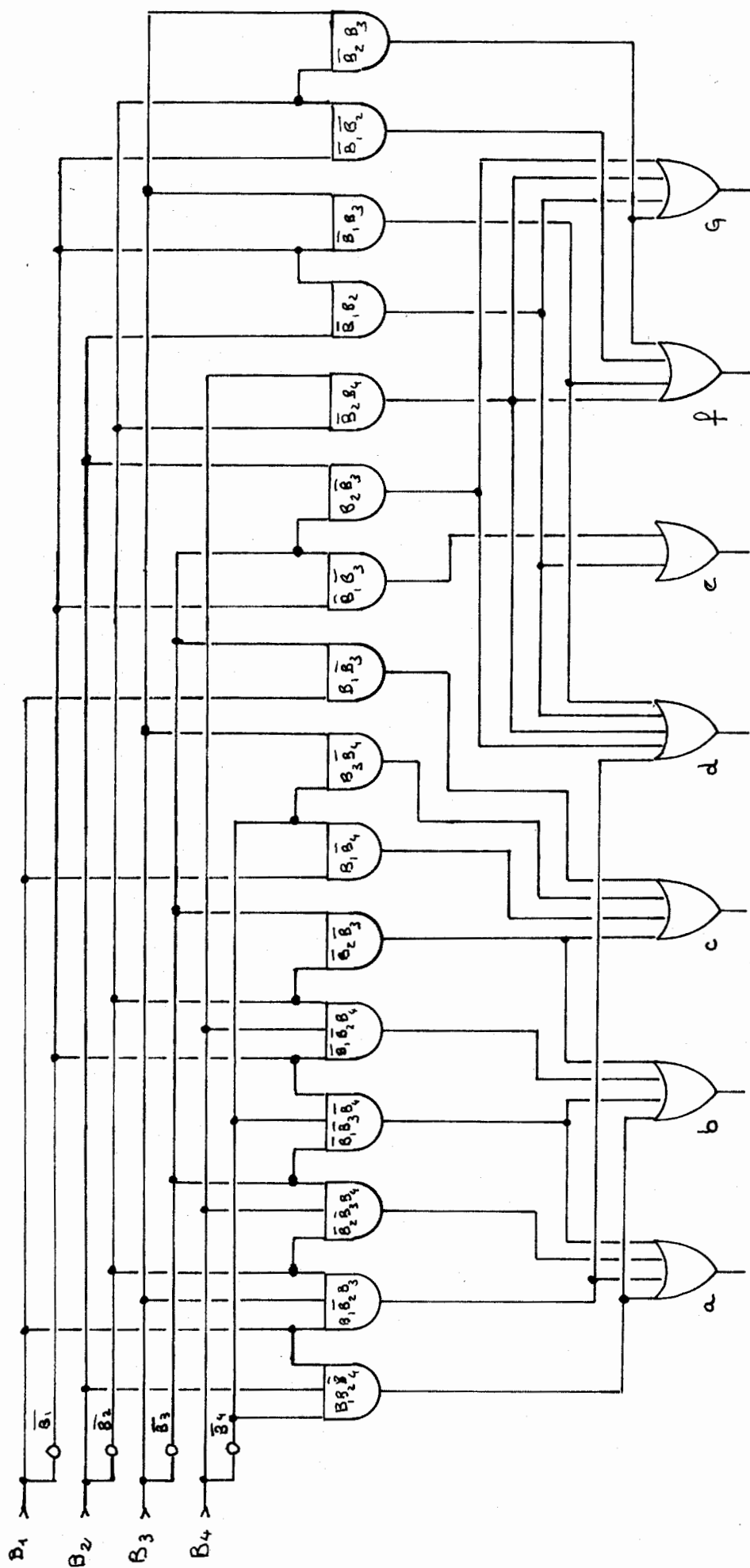
$$c = \overline{B_2}\overline{B_3} + \overline{B_1}\overline{B_4} + \overline{B_3}\overline{B_4} + \overline{B_3}\overline{B_3}$$

$$d = \overline{B_2}\overline{B_3} + \overline{B_2}B_4 + \overline{B_1}B_2 + \overline{B_1}B_3 + \overline{B_1}\overline{B_2}B_3$$

$$e = \overline{B_1}B_2 + \overline{B_1}\overline{B_3}$$

$$f = \overline{B_1}B_3 + \overline{B_1}\overline{B_2} + \overline{B_2}B_4 + \overline{B_2}B_3$$

$$g = \overline{B_2}\overline{B_3} + \overline{B_1}B_2 + \overline{B_2}B_3 + \overline{B_2}B_4$$



DECODEUR BCD / 7 SEGMENTS.

IV-3 - Transcodeurs

Lorsqu'il s'agit de convertir un nombre représenté dans un code en son expression dans un autre code, on utilise la méthode générale des circuits combinatoires (table de vérité et minimisation des termes). Dans certains cas, pour des codes particuliers, les circuits de transcodage peuvent être considérablement simplifiés par l'utilisation de circuits additionneurs existant sur le marché. Par exemple, le passage du code BCD au code excédent 3 (cf. chapitre "Codes") se réalise en utilisant un additionneur 2 bits (SN 7482) ajoutant systématiquement +3 au nombre BCD. Le passage inverse excédent 3/BCD s'obtient en retranchant 3 au code excédent 3, ou bien en ajoutant le complément à 2 (cf. opérations dans le système binaire) en utilisant un additionneur 4 bits (SN 7483).

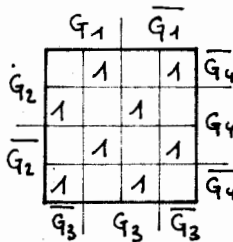
On pourrait citer d'autres exemples pour montrer qu'en techniques binaires, il existe souvent plusieurs solutions pour un même problème, l'intérêt d'utiliser une solution plutôt qu'une autre étant toujours commandé par l'emploi du plus petit nombre de circuits ou par des contraintes techniques, telles : le temps de propagation de l'information, une faible consommation sur les alimentations ou un encombrement réduit.

Nous allons montrer à titre d'exemple le calcul d'un circuit de conversion souvent utilisé : le Gray-binaire.

Conversion code Gray/code binaire

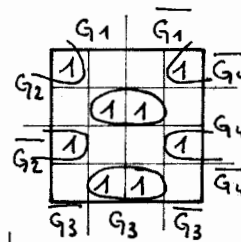
Le code Gray ou binaire réfléchi est très utilisé dans les capteurs de positions. En effet, un seul digit changeant entre deux positions successives, les risques de fausses informations durant les transitions sont éliminés.

	BINAIRE				GRAY			
	B1	B2	B3	B4	G1	G2	G3	G4
0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	0	0
2	0	1	0	0	1	1	0	0
3	1	1	0	0	0	1	0	0
4	0	0	1	0	0	1	1	0
5	1	0	1	0	1	1	1	0
6	0	1	1	0	1	0	1	0
7	1	1	1	0	0	0	1	0
8	0	0	0	1	0	0	1	1
9	1	0	0	1	1	0	1	1
10	0	1	0	1	1	1	1	1
11	1	1	0	1	0	1	1	1
12	0	0	1	1	0	1	0	1
13	1	0	1	1	1	1	0	1
14	0	1	1	1	1	0	0	1
15	1	1	1	1	0	0	0	1



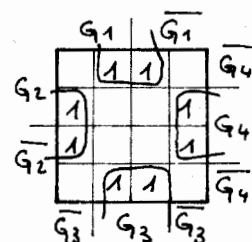
$$\begin{aligned}
 B1 = & G1 \cdot G2 \cdot G3 \cdot \overline{G4} \\
 & + G1 \cdot G2 \cdot \overline{G3} \cdot G4 \\
 & + G1 \cdot \overline{G2} \cdot G3 \cdot G4 \\
 & + \overline{G1} \cdot G2 \cdot G3 \cdot G4 \\
 & + G1 \cdot \overline{G2} \cdot \overline{G3} \cdot \overline{G4} \\
 & + \overline{G1} \cdot G2 \cdot \overline{G3} \cdot \overline{G4} \\
 & + \overline{G1} \cdot \overline{G2} \cdot G3 \cdot \overline{G4} \\
 & + \overline{G1} \cdot \overline{G2} \cdot \overline{G3} \cdot G4
 \end{aligned}$$

$$\begin{aligned}
 B1 = & G1 \oplus G2 \\
 & \oplus G3 \oplus G4
 \end{aligned}$$



$$\begin{aligned}
 B2 = & G2 \cdot G3 \cdot G4 \\
 & + G2 \cdot \overline{G3} \cdot \overline{G4} \\
 & + \overline{G2} \cdot G3 \cdot \overline{G4} \\
 & + \overline{G2} \cdot \overline{G3} \cdot G4
 \end{aligned}$$

$$\begin{aligned}
 B2 = & G2 \oplus G3 \\
 & \oplus G4
 \end{aligned}$$



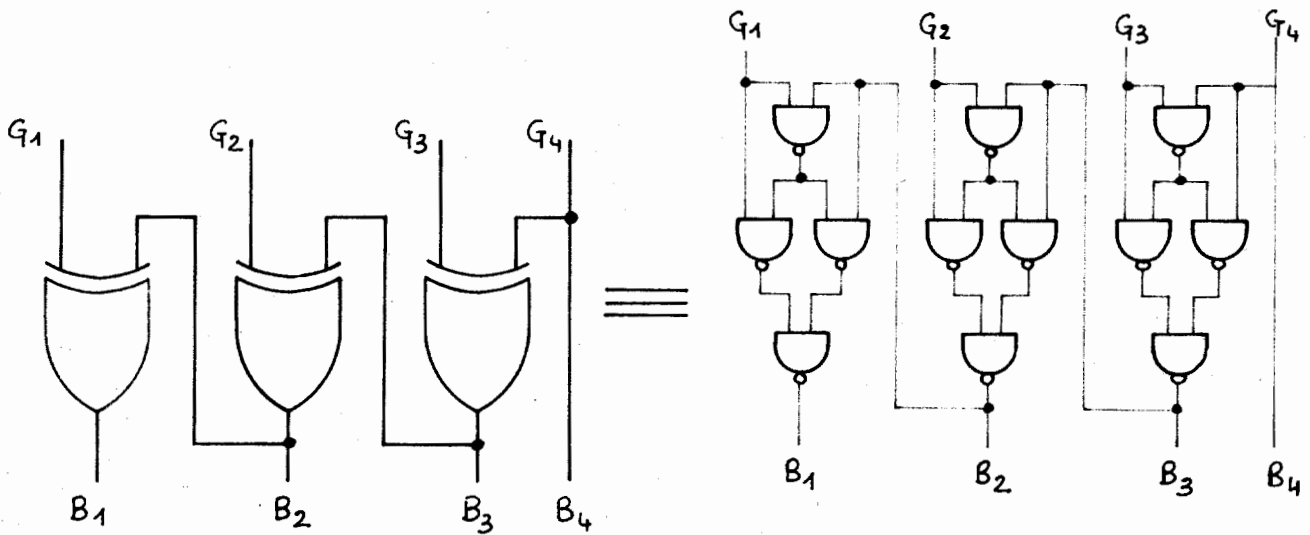
$$\begin{aligned}
 B3 = & G3 \cdot \overline{G4} \\
 & + \overline{G3} \cdot G4
 \end{aligned}$$

$$B3 = G3 \oplus G4$$

En lisant directement la table de vérité, on constate que :

$$B4 = G4$$

Le schéma logique sera donc le suivant :



Cette conversion peut se généraliser de la manière suivante :

$$B_{n-1} = G_{n-1} \oplus G_n \quad \text{et} \quad B_n = G_n$$

