

1 Etude du système binaire

1.1 Les nombres et leur représentation.

a) Définitions.

Dans le système décimal que nous connaissons, tout nombre est constitué de chiffres qui sont juxtaposés. Un système de numération va se caractériser par les chiffres qui sont utilisés pour représenter les nombres.

A chaque système de numération est associée une base qui correspond au nombre de chiffres que ce système utilise.

Système	Base	Chiffres
Décimal	10	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
Octal	8	{0, 1, 2, 3, 4, 5, 6, 7}
Binaire	2	{0, 1}
Hexadécimal	16	{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Dans le système hexadécimal, il n'est pas possible d'utiliser 10, 11, 12 puisque ce sont des nombres et pas des chiffres. Ces valeurs sont donc remplacées par des lettres qui peuvent être représentées en majuscule ou en minuscule.

b) Numération de position.

Pour représenter un nombre supérieur au plus grand des chiffres présent dans un système de numération, il faudra juxtaposer les chiffres, chacun des chiffres occupant un rang dans le nombre. Les rangs sont numérotés de la droite vers la gauche de 0 vers n (n+1 étant le nombre de rangs utilisés)

Exemple :

Rang	3	2	1	0
Nombre	1	5	3	4

A chaque rang est associé un facteur multiplicatif du chiffre (que l'on appelle le poids) dépendant du rang et de la base du système de numération. Dans le système décimal, on peut facilement s'en rendre compte. En effet la lecture du nombre repris en exemple se résume pas à lire individuellement chacun des chiffres le constituant mais à affecter ce facteur multiplicatif.

1534 se lit mil cinq cent trente quatre à savoir :

$1 \cdot 1000 + 5 \cdot 100 + 3 \cdot 10 + 4 \cdot 1$ ce qui correspond à
 $1 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0$

Si nous considérons donc le nombre $N_{10} = d_n d_{n-1} \dots d_3 d_2 d_1 d_0$

Nous pouvons alors écrire : $N_{10} = d_n \cdot 10^n + d_{n-1} \cdot 10^{n-1} + \dots + d_3 \cdot 10^3 + d_2 \cdot 10^2 + d_1 \cdot 10^1 + d_0 \cdot 10^0$

Et d'une manière générale, nous pouvons étendre ce concept à tout nombre exprimé dans une base B. Si nous considérons le nombre $N_B = C_n C_{n-1} \dots C_3 C_2 C_1 C_0$

Nous pouvons alors écrire que la représentation de ce nombre dans la base 10 serait alors :

$$N_{10} = C_n * B^n + C_{n-1} * B^{n-1} + \dots + C_3 * B^3 + C_2 * B^2 + C_1 * B^1 + C_0 * B^0$$

$C_n * B^n$ est le terme de rang n

B^n est le poids attaché au rang n

C_0 le chiffre de rang 0 est appelé le chiffre le moins significatif aussi appelé en anglais Least Significant Digit ou encore par ses abréviations : LSD

C_n le chiffre de rang n est appelé le chiffre le plus significatif aussi appelé en anglais Most Significant Digit ou encore par ses abréviations : MSD

c) Les ordinateurs et les nombres.

Du fait que dans les ordinateurs les chiffres sont représentés de façon électrique, il était beaucoup plus simple de travailler dans le système binaire. Le chiffre 0 peut être par exemple associé à une absence de tension tandis que le chiffre 1 sera associé à une présence de tension.

Dans un nombre binaire, tout chiffre occupant un rang s'appelle un bit qui est l'abréviation de Binary Digit.

Tout nombre composé de 8 bits s'appelle un octet ou encore en anglais Byte.

Tout nombre composé de 16 bits s'appelle un mot ou encore en anglais Word

Cette convention de nom est très utilisée en informatique puisque le stockage, le traitement ou le transfert des données s'effectue sur des données binaires.

Ex : pour un réseau Ethernet, on parlera d'un débit de 100Mb/sec (bits par seconde). Remarquez le b minuscule pour représenter le bit.

Pour le stockage sur un disque dur, on parlera de 500Go ou 500GB. Remarquez le B majuscule pour représenter le byte.

En nous basant sur un nombre purement positif, le nombre le plus petit que l'on pourra représenter est 0 tandis que le nombre le plus grand est $B^{(n)} - 1$ ou n est le nombre de rangs. Si l'on considère un nombre entier positif codé sur 8 bits, la valeur la plus grande sera donc $2^8 - 1 = 255$ en binaire représenté par 11111111

d) Les nombres fractionnaires.

Nous pouvons à nouveau nous référer au système décimal et à ce que nous connaissons dans la représentation d'un tel nombre dans cette base.

Exemple de nombre réel : 521,25

Si nous voulons lire le nombre, nous pouvons dire cinq cent vingt et un et vingt cinq centième. Ou encore pour la partie fractionnaire : 2 dixième et 5 centième.

Là aussi, on va associer à chaque rang de la partie fractionnaire un poids qui dépendra de la base et du rang sur lequel se trouve le chiffre

Rang	2	1	0		-1	-2
Nombre	5	2	1	,	2	5

$$5 * 100 + 2 * 10 + 1 * 1 + 2 * 1/10 + 5 * 5/100$$

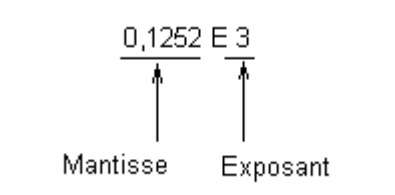
$$5 \cdot 10^2 + 2 \cdot 10^1 + 1 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

Une première approche dans le stockage d'un tel nombre serait de prévoir l'emplacement pour sauvegarder la partie entière et une autre partie indépendante pour sauvegarder la partie fractionnaire. Cette représentation s'appelle la représentation en virgule fixe. Cette représentation est utilisée dans certaines caisses enregistreuses de magasins pour la gestion de l'affichage. On retrouve la partie entière représentée par un certain nombre de chiffres fixe et de même pour la partie fractionnaire.

Si on prévoit le stockage de 4 chiffres répartis comme suit : 2 pour la partie entière et 2 pour la partie fractionnaire. Le nombre le plus petit sera donc 00,00 et le chiffre le plus grand 99,99. Si on envisage le stockage du plus grand nombre dont la partie fractionnaire est nul, rien ne change puisque l'on aura 99,00 et les deux chiffres de la partie fractionnaire, même s'ils sont nuls, ne peuvent être utilisés pour augmenter la valeur maximale de stockage du nombre. Nous pouvons porter également notre réflexion sur les nombres dont la partie entière est nulle.

L'idéal serait donc pour optimiser le stockage de travailler avec une représentation en virgule flottante. Nous envisageons à nouveau le stockage pour quatre chiffres et nous prenons la représentation du nombre suivant : 125,2

Nous pouvons transformer ce nombre en nombre fractionnaire pur (sans partie entière) en déplaçant la virgule et le nombre devient alors 0,1252. Il suffira alors de mémoriser le déplacement qu'il faudra donner à la virgule pour redonner à ce nombre la valeur d'origine. $0,1252 \cdot 10^3$. La forme normalisée de représentation sera : 0,1252 E 3



La question reste de savoir sur combien de chiffres seront stockés la mantisse et l'exposant. Pour le système binaire, cette représentation a été normalisée par l'Institut for Electrical and Electronic Engineers (IEEE) sous la référence IEEE 754. Les nombres réels sont représentés sur 32 bits (4 octets) ou sur 64 bits (8 octets).

	Encodage	Signe	Exposant	Mantisse	Valeur d'un nombre
Simple précision	32 bits	1 bit	8 bits	23 bits	$(-1)^S \times M \times 2^{(E-127)}$
Double précision	64 bits	1 bit	11 bits	52 bits	$(-1)^S \times M \times 2^{(E-1023)}$

Soit à coder la valeur 525,5.

- 525,5 est positif donc le 1er bit (bit de signe) sera 0.
- Sa représentation en base 2 est la suivante : 1000001101,1
- En normalisant, on trouve : $1,0000011011 \cdot 2^9$
- On ajoute 127 à l'exposant qui vaut 9 ce qui donne 136, soit en base 2 : 10001000
- La mantisse est composée de la partie décimale de 525,5 en base 2 normalisée, c'est-à-dire 0000011011.
- Comme la mantisse doit occuper 23 bits, il est nécessaire d'ajouter des zéros pour la compléter :
00000110110000000000000
- La représentation du nombre 525,5 en binaire avec la norme IEEE est donc :
0 1000 1000 00000110110000000000000
0100 0100 0000 0011 0110 0000 0000 0000

e) Les nombres négatifs.

Si nous nous référons au système décimal que nous connaissons, la représentation des nombres négatifs repose sur une représentation en grandeur et signe. Nous ajoutons un signe distinctif permettant de renseigner si le nombre est positif ou négatif mais en laissant la grandeur identique.
Ex : +125 et -125.

D'un point de vue informatique, les seules informations qu'un ordinateur puisse enregistrer dans le système binaire sont le chiffre 1 et le chiffre 0. Nous allons donc réserver le chiffre de point fort pour y coder le signe : la valeur 0 représentera le nombre positif tandis que la valeur 1 le nombre négatif. En règle générale, pour une base B, le bit de signe prendra les valeurs suivantes :

Nombre positif : $S = 0$

Nombre négatif : $S = B - 1$

Donc, si on se base uniquement sur l'information du signe en gardant la grandeur, nous aurions alors dans l'exemple suivant :

$N_2 = (01101110)_2$ Nombre binaire positif

$N_2 = (11101110)_2$ Nombre binaire négatif

Un désavantage du codage en grandeur et signe est qu'il faut coder de façon distincte le signe et la grandeur. Il faudrait trouver une représentation qui nous permettrait de pouvoir effectuer l'opération suivante

$(01101110)_2 + (11101110)_2$ et d'obtenir la valeur 0

Pour y parvenir, les nombres négatifs vont être représentés en complétant chacun des chiffres de chaque rang à savoir remplacer le chiffre C par $(B-1)-C$, B étant la base correspondant au système de numération dans lequel le nombre est représenté.

Exemple : $N_{16} = (0F1AC)_{16}$

Le nombre négatif correspondant sera donc $(F0E52)_{16}$ et il reste un nombre représenté en signe puisque le chiffre de poids fort correspond bien à B-1 et qui nous permettra d'effectuer l'opération d'addition suivante :

$(0F1AC)_{16} + (F0E52)_{16} = (FFFFFF)_{16}$ en admettant alors la double représentation du zéro +0 et -0

En binaire, nous parlerons de complément à un.

Une autre technique consiste à compléter le nombre dans son intégralité et non pas chiffre par chiffre. Nous parlerons du complément vrai ou complément à deux pour le système binaire. L'obtention de ce complément peut être simplifiée en complétant chacun des chiffres et en ajoutant la valeur 1. Exemple : soit le nombre binaire $(01101001)_2$.
La représentation négative de ce nombre sera

$$\begin{array}{l} (01101001) \\ \downarrow \text{ Complément bit à bit} \\ (10010110) \text{ Complément restreint} \\ \downarrow +1 \\ (10010111) \text{ Complément vrai} \end{array}$$

Si nous additionnons maintenant les deux nombres, nous obtiendrons le résultat suivant :

$$\begin{array}{r}
 (01101001) \\
 + (10010111) \\
 \hline
 (10000000) \\
 \uparrow \\
 \text{bit de report qui sera perdu}
 \end{array}$$

Si le stockage des données se limite à 8 bits et qu'une opération d'addition provoque un report sur le neuvième rang, celui-ci sera perdu et nous obtiendrons donc comme résultat 0. SI nous considérons le stockage possible d'un nombre entier signé sur 8 bits, nous aurons donc la valeur la plus grande qui sera 127, la valeur la plus petite négative qui sera -128 ce qui nous donne donc 256 combinaisons possibles en comptant la représentation du zéro.

1.2 Conversion d'entiers.

a) Conversion d'une base B vers la base 10.

Reprenons la formule générale vue précédemment limitée à quatre rangs :

$$\begin{aligned}
 N_B &= (C_3 C_2 C_1 C_0)_B \\
 N_{10} &= D_3 * B^3 + D_2 * B^2 + D_1 * B^1 + D_0 * B^0
 \end{aligned}$$

D_3 est la correspondance du chiffre C_3 dans la base 10

D_2 est la correspondance du chiffre C_2 dans la base 10

D_1 est la correspondance du chiffre C_1 dans la base 10

D_0 est la correspondance du chiffre C_0 dans la base 10

Adaptons cette formule de sorte à en optimiser la forme:

$$N_{10} = (((D_3 + 0 * B) + D_2) * B + D_1) * B + D_0$$

Cette dernière formule comprend 4 sommes et trois produits tandis que la première formule comprenait 4 produits, trois sommes et 4 puissances.

Exemples.

$$-(218)_8 = (\dots)_{10}$$

$N_{10} = ((2 + 0 * B) + 1) * B + 8$. Pour une question de facilité, nous poserons le calcul sous la forme :

$$\begin{array}{l}
 0 \times 8 + 2 = \textcircled{2} \\
 \swarrow \\
 2 \times 8 + 1 = \textcircled{17} \\
 \swarrow \\
 17 \times 8 + 7 = \boxed{143}
 \end{array}$$

$$- (1EF)_{16} = (\dots)_{10}$$

$N_{10} = ((1 + 0 * B) + 14) * B + 15$. Pour une question de facilité, nous poserons le calcul sous la forme :

$$\begin{array}{l} 0 \times 16 + 1 = \textcircled{1} \\ \swarrow \\ 1 \times 16 + E(14) = \textcircled{30} \\ \swarrow \\ 30 \times 16 + F(15) = \boxed{495} \end{array}$$

$$- (1100110)_2 = (\dots)_{10}$$

$$\begin{array}{l} 0 \times 2 + 1 = \textcircled{1} \\ \swarrow \\ 1 \times 2 + 1 = \textcircled{3} \\ \swarrow \\ 3 \times 2 + 0 = \textcircled{6} \\ \swarrow \\ 6 \times 2 + 0 = \textcircled{12} \\ \swarrow \\ 12 \times 2 + 1 = \textcircled{25} \\ \swarrow \\ 25 \times 2 + 1 = \textcircled{51} \\ \swarrow \\ 51 \times 2 + 0 = \boxed{102} \end{array}$$

b) Conversion d'une base 10 vers la base B.

Reprenons la formule générale vue précédemment limitée à quatre rangs :

$$N_{10} = D_3 * B^3 + D_2 * B^2 + D_1 * B^1 + D_0 * B^0$$

$$N_B = (C_3 C_2 C_1 C_0)_B$$

C_3 est la correspondance du chiffre D_3 dans la base B

C_2 est la correspondance du chiffre D_2 dans la base B

C_1 est la correspondance du chiffre D_1 dans la base B

C_0 est la correspondance du chiffre D_0 dans la base B

Si nous divisons N_{10} par la valeur correspondant à la base, nous obtiendrons :

$$(N_{10} / B) = (D_3 * B^3 + D_2 * B^2 + D_1 * B^1 + D_0 * B^0) / B$$

= $(D_3 * B^2 + D_2 * B^1 + D_1 * B^0)$ résultat de la division entière + D_0 qui est le reste de cette même division entière. Si nous considérons $(D_3 * B^2 + D_2 * B^1 + D_1 * B^0)$, nous avons un nouvel entier du système décimal.

$$N'_{10} = (D_3 * B^2 + D_2 * B^1 + D_1 * B^0)$$

Si nous divisons N'_{10} par la valeur correspondant à la base, nous obtiendrons :

$$(N'_{10} / B) = (D_3 * B^2 + D_2 * B^1 + D_1 * B^0) / B$$

= $(D_3 * B^1 + D_2 * B^0)$ résultat de la division entière + D_1 qui est le reste de cette même division entière.

Nous allons ainsi obtenir $D_3 D_2 D_1 D_0$ qu'il ne restera plus qu'à transformer en $C_3 C_2 C_1 C_0$

Exemples :

$$-(435)_{10} = (\dots)_8$$

$435 : 8 = 54$	reste	3	D0
$54 : 8 = 6$	reste	6	D1
$6 : 8$	reste	6	D2

Le résultat est donc $(663)_8$

$$-(176)_{10} = (\dots)_{16}$$

$176 : 16 = 11$	reste	0	D0
$11 : 16$	reste	11	D1 $\rightarrow B$

Le résultat est donc $(B0)_{16}$

$$-(35)_{10} = (\dots)_2$$

$35 : 2 = 17$	reste	1	D0
$17 : 2 = 8$	reste	1	D1
$8 : 2 = 4$	reste	0	D2
$4 : 2 = 2$	reste	0	D3
$2 : 2 = 1$	reste	0	D4
$1 : 2$	reste	1	D5

Le résultat est donc $(100011)_2$

1.3 Conversion de nombres fractionnaires purs.

a) Nombre de rangs à conserver.

Si pour la conversion des entiers la question ne se pose pas, nous allons nous apercevoir dans les mécanismes de conversion que nous devons nous limiter dans le nombre de rangs à conserver dans la réponse pour maintenir la précision de la donnée de départ.

Soit B_s la base source avec n le nombre de rangs occupés par le nombre source.
Soit B_c la base cible avec k le nombre de rangs que doit occuper le nombre cible.

Nous envisagerons que le plus petit chiffre occupant le rang n de la base source doit avoir le même poids que le plus petit chiffre occupant le rang k de la base cible.

$$(0,1)_{B_s}^n = (0,1)_{B_c}^k$$

Calculons le logarithme de part et d'autre de l'égalité et développons :

$$\log_{B_c} (0,1)_{B_s}^n = \log_{B_c} (0,1)_{B_c}^k$$

$$n * \log_{B_c} (0,1)_{B_s} = k * \log_{B_c} (0,1)_{B_c}$$

$$n * \log_{B_c} (10)_{B_s} = k * \log_{B_c} (10)_{B_c}$$

$$n * \log_{B_c} B_s = k * \log_{B_c} B_c$$

$$n * \log_{B_c} B_s = k$$

$$B = (10)_B$$

$$k = n * \frac{\log B_s}{\log B_c}$$

$$\log_A B = \frac{\log_{10} B}{\log_{10} A}$$

Nous pouvons résumer cette formule par le tableau suivant en fonction des bases les plus utilisées.

Base B	2	8	16
Base B vers base 10	n * 0,301	n * 0,903	n * 1,204
Base 10 vers base B	n * 3,321	n * 1,107	n * 0,830

b) Conversion d'une base B vers la base 10.

Reprenons la formule générale vue précédemment limitée à quatre rangs :

$$N_B = (.C_{-1} C_{-2} C_{-3} C_{-4})_B$$

$$N_{10} = C_{-1} * B^{-1} + C_{-2} * B^{-2} + C_{-3} * B^{-3} + C_{-4} * B^{-4}$$

Dans ce cas, n vaut 4. Nous allons multiplier de part et d'autre de l'égalité par B^4 , ce qui nous permettra d'obtenir l'égalité suivante :

$$(N_{10} * B^4) = (C_{-1} * B^{-1} + C_{-2} * B^{-2} + C_{-3} * B^{-3} + C_{-4} * B^{-4}) * B^4$$

$$(N_{10} * B^4) = (C_{-1} * B^3 + C_{-2} * B^2 + C_{-3} * B^1 + C_{-4} * B^0)$$

Nous obtenons donc maintenant un nombre entier qu'il nous reste à convertir vers la base 10 comme vu dans le paragraphe traitant des conversions d'entiers.

Une fois cette conversion effectuée, il restera à effectuer la division du nombre obtenu par B^4 en conservant dans la réponse k rangs suivant la formule

$$k = n * \log B_s$$

Exemples :

$$-(, 110101)_2 = (, \dots\dots\dots)_{10}$$

Nous retrouvons 6 rangs dans le nombre source. La valeur n de notre formule vaut donc 6.

Si nous calculons la valeur k, nous obtiendrons $k = 6 * \log 2 = 1,8$. Nous arrondissons à l'excès, ce qui nous donne donc 2 comme valeur de k.

La conversion du nombre fractionnaire en entier est très simple :

$(,110101)_2 * 2^6 = (110101)_2$ Il suffit simplement de supprimer la virgule située à la gauche du nombre.

Nous pouvons maintenant convertir ce nombre entier de la base 2 vers la base 10

$$\begin{aligned} 0 \times 2 + 1 &= 1 \\ 1 \times 2 + 1 &= 3 \\ 3 \times 2 + 0 &= 6 \\ 6 \times 2 + 1 &= 13 \\ 13 \times 2 + 0 &= 26 \\ 26 \times 2 + 1 &= 53 \end{aligned}$$

Il reste à diviser le résultat obtenu par 2^6 en conservant 2 rangs.

$53 : (2^6) = 0,82$. Comme nous avons arrondi à l'excès le nombre de rangs à conserver, il n'est plus nécessaire d'utiliser un quelconque arrondi dans le résultat de la division.

$$-(, 214)_8 = (, \dots\dots\dots)_{10}$$

Nous retrouvons 3 rangs dans le nombre source. La valeur n de notre formule vaut donc 3.

Si nous calculons la valeur k, nous obtiendrons $k = 3 * \log 8 = 2,7$. Nous arrondissons à l'excès, ce qui nous donne donc 3 comme valeur de k.

La conversion du nombre fractionnaire en entier est très simple :

$(,214)_8 * 8^3 = (214)_8$ Il suffit simplement de supprimer la virgule située à la gauche du nombre.

Nous pouvons maintenant convertir ce nombre entier de la base 8 vers la base 10

$$\begin{aligned} 0 \times 8 + 2 &= 2 \\ 2 \times 8 + 1 &= 17 \\ 17 \times 8 + 4 &= 140 \end{aligned}$$

Il reste à diviser le résultat obtenu par 8^3 en conservant 3 rangs..

$$140 : (8^3) = 0,273$$

$$-(, 1F)_{16} = (, \dots\dots\dots)_{10}$$

Nous retrouvons 2 rangs dans le nombre source. La valeur n de notre formule vaut donc 2.

Si nous calculons la valeur k, nous obtiendrons $k = 2 * \log 16 = 2,408$. Nous arrondissons à l'excès, ce qui nous donne donc 3 comme valeur de k.

La conversion du nombre fractionnaire en entier est très simple :

$(,1F)_{16} * 16^2 = (1F)_{16}$ Il suffit simplement de supprimer la virgule située à la gauche du nombre.

Nous pouvons maintenant convertir ce nombre entier de la base 16 vers la base 10

$$\begin{aligned} 0 \times 16 + 1 &= 1 \\ 1 \times 16 + F(15) &= 31 \end{aligned}$$

Il reste à diviser le résultat obtenu par 16^2 en conservant 3 rangs.
 $31 : (16^2) = 0,121$

c) Conversion d'une base 10 vers la base B.

Reprenons la formule générale vue précédemment limitée à quatre rangs :

$$N_{10} = C_{-1} * B^{-1} + C_{-2} * B^{-2} + C_{-3} * B^{-3} + C_{-4} * B^{-4}$$

$$N_B = (.C_{-1} C_{-2} C_{-3} C_{-4})_B$$

N_{10} est donné et nous souhaitons trouver $C_{-1} C_{-2} C_{-3} C_{-4}$.

Avant toute chose, nous devons déterminer le nombre de rangs que nous devons obtenir dans le nombre converti. Dans notre exemple, nous avons dans la base source 4 rangs et nous devons donc utiliser la formule suivante pour déterminer la valeur de k.

$$k = n * \frac{1}{\log B_c}$$

Nous allons multiplier de part et d'autre de l'égalité de la première formule par B, ce qui nous permettra d'obtenir alors :

$$(N_{10} * B) = (C_{-1} * B^{-1} + C_{-2} * B^{-2} + C_{-3} * B^{-3} + C_{-4} * B^{-4}) * B$$

Nous obtenons alors :

$(N_{10} * B) = (C_{-1} + C_{-2} * B^{-1} + C_{-3} * B^{-2} + C_{-4} * B^{-3})$ ce qui correspond donc à un nombre réel composé d'une partie entière et d'une partie fractionnaire. La partie entière correspond à C_{-1} .

Nous prenons la partie fractionnaire du résultat précédent que nous multiplions à nouveau par B, nous permettant alors d'obtenir:

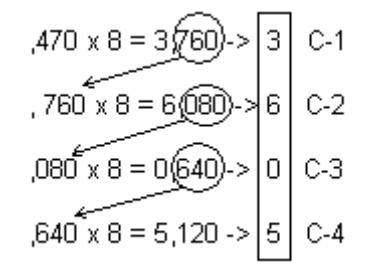
$$(C_{-2} * B^{-1} + C_{-3} * B^{-2} + C_{-4} * B^{-3}) * B = (C_{-2} + C_{-3} * B^{-1} + C_{-4} * B^{-2}).$$

Ce nombre correspond à nouveau à un nombre réel composé d'une partie entière et d'une partie fractionnaire, la partie entière étant C_{-2} . Nous pourrions poursuivre cette démarche jusqu'à l'obtention des k rangs dans la réponse finale.

Exemples :

$$-(, 470)_{10} = (, \dots\dots\dots)_8$$

Nous retrouvons 3 rangs dans le nombre source. La valeur n de notre formule vaut donc 3.
 Si nous calculons la valeur k, nous obtiendrons $k = 3 * 1/\log 8 = 3,321$. Nous arrondissons à l'excès, ce qui nous donne donc 4 comme valeur de k.



Nous obtenons alors comme résultat : $(,3605)_8$

$$-(,570)_{10} = (, \dots \dots \dots)_{16}$$

Nous retrouvons 3 rangs dans le nombre source. La valeur n de notre formule vaut donc 3.
Si nous calculons la valeur k, nous obtiendrons $k = 3 * 1/\log 16 = 2,491$. Nous arrondissons à l'excès, ce qui nous donne donc 3 comme valeur de k.

$$\begin{array}{l} ,570 \times 16 = 9,120 \rightarrow \boxed{9} \quad C-1 \\ \swarrow \\ ,120 \times 16 = 1,920 \rightarrow \boxed{1} \quad C-2 \\ \swarrow \\ ,920 \times 16 = 14,720 \rightarrow \boxed{14} \rightarrow E \quad C-3 \end{array}$$

Nous obtenons alors comme résultat : $(,91E)_{16}$

$$-(,27)_{10} = (, \dots \dots \dots)_2$$

Nous retrouvons 2 rangs dans le nombre source. La valeur n de notre formule vaut donc 3.
Si nous calculons la valeur k, nous obtiendrons $k = 2 * 1/\log 2 = 6,64$. Nous arrondissons à l'excès, ce qui nous donne donc 7 comme valeur de k.

$$\begin{array}{l} ,27 \times 2 = ,54 \rightarrow \boxed{0} \quad C-1 \\ ,54 \times 2 = 1,08 \rightarrow \boxed{1} \quad C-2 \\ ,08 \times 2 = 0,16 \rightarrow \boxed{0} \quad C-3 \\ ,16 \times 2 = ,32 \rightarrow \boxed{0} \quad C-4 \\ ,32 \times 2 = ,64 \rightarrow \boxed{0} \quad C-5 \\ ,64 \times 2 = 1,28 \rightarrow \boxed{1} \quad C-6 \\ ,28 \times 2 = 0,56 \rightarrow \boxed{0} \quad C-7 \end{array}$$

Nous obtenons alors comme résultat : $(,0100010)_2$

1.4 L'arithmétique binaire.

2 L'algèbre booléenne.

2.1 Définitions.

Une variable binaire ou variable de l'algèbre logique est une variable qui ne prend que les valeurs 0 et 1. Toutes les variables utilisées sont binaires

Une fonction booléenne est une fonction d'une ou plusieurs variables binaires, l'ensemble de variation de cette fonction étant $\{0,1\}$.

2.2 Application pratique des variables binaires.

La valeur 1 représente l'existence d'un phénomène physique, électrique. Le courant passe, un contact (interrupteur) est fermé, un contacteur est enclenché, un récepteur électrique est alimenté...

La valeur 0 représente l'absence d'un phénomène physique, électrique. Le courant ne passe pas, un contact (interrupteur) est ouvert, un contacteur est déclenché, un récepteur électrique n'est pas alimenté...

2.3 Les fonctions à une seule variable.

Les fonctions à une variable sont sous la forme $y=f(x)$.

Si nous envisageons l'ensemble des fonctions booléennes possibles, nous obtiendrons le tableau suivant :

x	f ₁	f ₂	f ₃	f ₄
0	0	0	1	1
1	0	1	0	1

f ₁	0	
f ₂	x	
f ₃	\bar{x}	Fonction NON (NOT)
f ₄	1	

Nous allons aborder la fonction la plus importante.


a) La fonction NON (NOT)

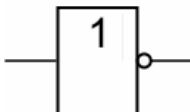
$$f_3(x) = \bar{x}$$

x	f ₃
0	1
1	0

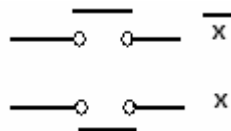
Représentation algébrique : $y = \bar{x}$

Sous forme de symbole logique, nous aurons

Symbole américain : 

Symbole européen : 

Représentation électrique, schéma à contact :



2.4 Les fonctions à deux variables.

Les fonctions à deux variables sont sous la forme $z = f(x,y)$.

Si nous envisageons l'ensemble des fonctions booléennes possibles, nous obtiendrons le tableau suivant :

x	y	f ₁	f ₂	f ₃	f ₄	f ₅	f ₆	f ₇	f ₈	f ₉	f ₁₀	f ₁₁	f ₁₂	f ₁₃	f ₁₄	f ₁₅	f ₁₆
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

f ₁	0	
f ₂	$x.y$	Fonction ET (AND)
f ₃	$x.\bar{y}$	
f ₄	x	
f ₅	$\bar{x}.y$	
f ₆	\bar{y}	
f ₇	$x \oplus y$	Fonction OU Exclusif (XOR)
f ₈	$x + y$	Fonction OU (OR)
f ₉	$\overline{x + y}$	Fonction NON OU (NOR)
f ₁₀	$\overline{x \oplus y}$	Fonction NON OU Exclusif (XNOR)
f ₁₁	\bar{y}	
f ₁₂	$x + \bar{y}$	
f ₁₃	\bar{x}	
f ₁₄	$\bar{x} + y$	
f ₁₅	$\overline{x.y}$	Fonction NON ET (NAND)
f ₁₆	1	

a) La fonction ET (AND)


$$f_2(x, y) = x.y$$

x	y	f ₂
0	0	0
0	1	0
1	0	0
1	1	1

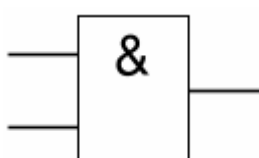
Représentation algébrique : $Z = x.y$

Sous forme de symbole logique, nous aurons

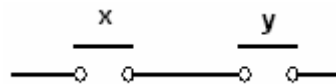
Symbole américain :



Symbole européen :



Représentation électrique, schéma à contact :
 Les contacts sont mis en série dans le montage.



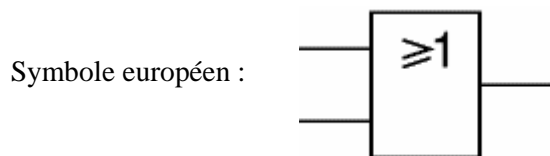
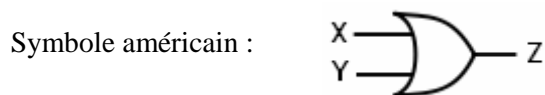
b) La fonction OU (OR)

$$f_2(x, y) = x + y$$

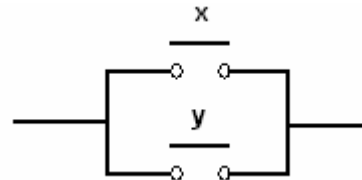
x	y	f _g
0	0	0
0	1	1
1	0	1
1	1	1

Représentation algébrique : $Z = x + y$

Sous forme de symbole logique, nous aurons



Représentation électrique, schéma à contact :
 Les contacts sont mis en parallèle dans le montage.



c) La fonction NON ET (NAND)


$$f_{15}(x, y) = \overline{x \cdot y}$$

x	y	f ₁₅
0	0	1
0	1	1
1	0	1
1	1	0

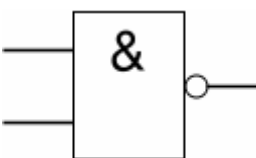
Représentation algébrique : $Z = \overline{x \cdot y}$

Sous forme de symbole logique, nous aurons

Symbole américain :



Symbole européen :



d) La fonction NON OU (NOR)

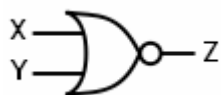
$$f_9(x, y) = \overline{x + y}$$

x	y	f ₉
0	0	1
0	1	0
1	0	0
1	1	0

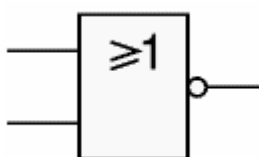
Représentation algébrique : $Z = \overline{x + y}$

Sous forme de symbole logique, nous aurons

Symbole américain :



Symbole européen :




e) La fonction OU Exclusif (XOR)

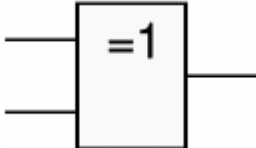
$$f_7(x, y) = x \oplus y$$

x	y	f ₇
0	0	0
0	1	1
1	0	1
1	1	0

Représentation algébrique : $Z = x \oplus y$

Sous forme de symbole logique, nous aurons

Symbole américain : 

Symbole européen : 


f) La fonction NON OU Exclusif (XNOR)

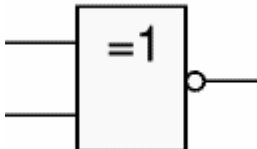
$$f_{10}(x, y) = x \oplus y$$

x	y	f_{10}
0	0	1
0	1	0
1	0	0
1	1	1

Représentation algébrique : $z = \overline{x \oplus y}$

Sous forme de symbole logique, nous aurons

Symbole américain : 

Symbole européen : 

2.5 Représentation des fonctions booléennes.

a) Représentation par tables de vérité.

Comme abordé dans le paragraphe précédent, toute fonction booléenne peut être représentée par une table dans laquelle on retrouvera un groupe de colonnes pour les entrées et un groupe de colonnes pour la sortie. Chaque ligne comprendra une seule configuration des entrées et l'on renseignera dans les colonnes correspondant aux sorties, les valeurs de sortie de la fonction. On parlera de table complète lorsque toutes les configurations des entrées sont présentes.

Ce type de représentation est très pratique lorsque l'on souhaite tester le circuit physique associé à l'équation logique correspondante.

Exemple :

Entrées		Sortie
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

b) Représentation par équation (SDPC).

Toute fonction peut être obtenue sur base d'une somme de produits canonique ou d'un produit de sommes canoniques. Nous nous contenterons de la première représentation appelée aussi de part ses initiales SDPC.

Un produit canonique comprend dans un produit l'ensemble des variables d'entrée de façon unique, qu'elles soient normale ou complétement. Si nous nous basons sur les fonctions à une variable, toute équation peut être sous la forme :

$$f(x) = a_0.\bar{x} + a_1.x$$

Nous pouvons constater qu'en fonction des valeurs de a_0 et a_1 , nous pouvons obtenir l'ensemble des équations abordées dans un des paragraphes précédents.

a_0	a_1	$f(x)$	$f(x)$
0	0	$0.\bar{x} + 0.x$	0
0	1	$0.\bar{x} + 1.x$	x
1	0	$1.\bar{x} + 0.x$	\bar{x}
1	1	$1.\bar{x} + 1.x$	1

Nous pouvons très facilement déterminer les différentes valeurs d' a_0 et a_1 en pratiquant comme suit : $a_0 = f(0)$ et $a_1 = f(1)$.

Le passage d'une table de vérité à une équation logique est maintenant assez évident. Reprenons la table de vérité associée à la fonction logique NON (NOT).

x	f_3	coefficient
0	1	a_0 correspond à la valeur de la fonction c.-à-d. 1
1	0	a_1 correspond à la valeur de la fonction c.-à-d. 0

$$f(x) = 1.\bar{x} + 0.x \qquad f(x) = \bar{x}$$

Nous pouvons étendre notre démarche aux fonctions à deux variables et nous pouvons donc écrire que toute fonction à deux variables aura la forme suivante :

$$f(x, y) = a_0.\bar{x}.\bar{y} + a_1.\bar{x}.y + a_2.x.\bar{y} + a_3.x.y$$

Prenons le cas de la fonction logique NON OU et partons de la table de vérité pour en obtenir l'équation :

x	y	f_9	coefficients
0	0	1	a0
0	1	0	a1
1	0	0	a2
1	1	0	a3

$$f(x, y) = 1.\bar{x}.\bar{y} + 0.\bar{x}.y + 0.x.\bar{y} + 0.x.y$$

$f(x, y) = \bar{x}.\bar{y}$ Qui comme le démontrera le théorème de De Morgan, correspond à l'équation

$$f(x, y) = \overline{x + y}$$

c) Représentation par tables de Karnaugh (grilles Karnaugh Mahoney).

Le principe de ces grilles est d'associer chaque case à un produit canonique et d'y renseigner la valeur du coefficient 'a' correspondant.

Les cases sont organisées de telle façon à ce que deux cases voisines (ayant un côté commun) soient associées à deux produits canoniques adjacents c.-à-d. qui ne diffèrent que d'une seule variable dans le produit.

Les grilles sont carrées pour un nombre pair de variables et rectangulaires pour un nombre impair de variables.

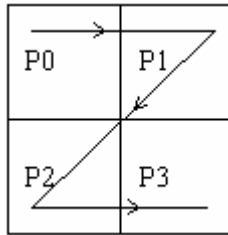
➤ Grilles avec deux variables.

$\bar{X}.\bar{Y}$	$\bar{X}.Y$
$X.\bar{Y}$	$X.Y$

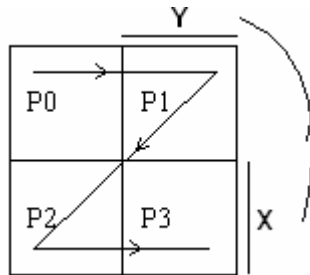
Pour une question de facilité, les produits canoniques ne sont pas toujours représentés et on se contente de renseigner l'indice du coefficient associé à ce dernier. Pour indiquer le fait que c'est un produit canonique, on associera à cet indice la lettre P sous la forme Px. La grille alors modifiée donnera :

P0	P1
P2	P3

Pour retrouver facilement l'emplacement du bon produit canonique, il existe une méthode assez simple :

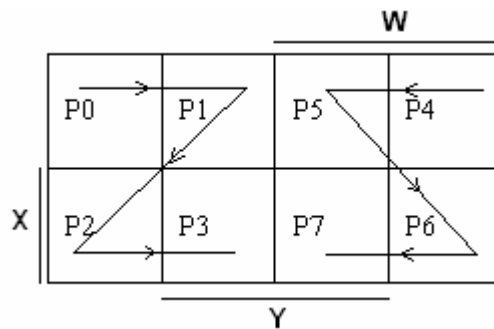


Comme nous pouvons l'observer dans la grille, la rangée du bas correspond, pour chacune des cases, à des produits canoniques contenant la variable X simple (non complémentée). Dans cette grille, la colonne de droite correspond également, pour chacune des cases, à des produits canoniques contenant la variable Y simple (non complémentée). Nous pouvons le renseigner dans la grille de Karnaugh comme suit :



Il est là aussi aisé de retrouver les différents emplacements. Pour une grille d'allure carrée, nous commencerons par renseigner la variable de poids fort dans le produit canonique en commençant à droite, en bas. Il faudra ensuite tourner dans le sens anti horlogique pour mettre les autres variables sur les autres côtés en suivant l'ordre poids fort vers poids faible.

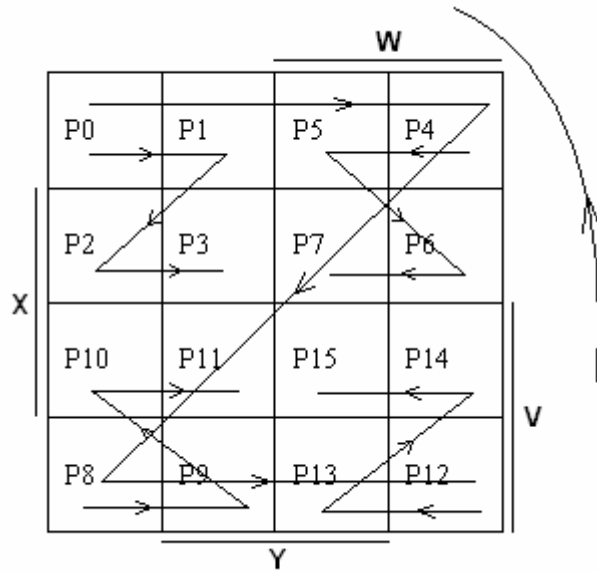
➤ **Grilles avec trois variables.**



Dans cette grille, nous sommes partis du principe que les variables sont dans l'ordre WXY avec W la variable de poids fort et Y la variable de poids faible. Dans une table d'allure rectangulaire, les variables se complètent autour de la table en commençant en haut à droite

➤ **Grilles avec quatre variables.**

Dans cette grille, nous sommes partis du principe que les variables sont dans l'ordre VWXY avec V comme variable de poids fort et Y comme variable de poids faible.



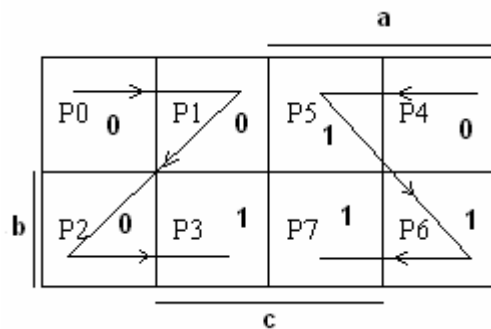
Nous nous limiterons dans ce paragraphe à quatre variables bien que dans certains exercices résolus nous aborderons les tables de Karnaugh avec cinq variables.

Le passage d'une table de vérité à une grille de Karnaugh est assez simple puisqu'il suffira de renseigner dans les cases la valeur des différents coefficients correspondant.

Exemple.

Entrées			Sortie
a	b	c	L
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Nous considérons les entrées abc en respectant l'ordre suivant : a variable de poids fort et c variable de poids faible.



Comme nous avons pu le voir dans la conversion d'une table de vérité vers une équation, nous pouvons très rapidement retrouver les coefficients en nous basant sur les valeurs que prend la fonction pour les différentes configurations des entrées.

Attention : nous retrouvons une table de Karnaugh par fonction (par sortie).
La représentation par table n'est qu'utilisée lorsque nous envisageons de simplifier une équation logique. Nous en reparlerons dans un paragraphe ultérieur.

2.6 Simplification des fonctions booléennes.

a) Simplification par les postulats et théorèmes.

Seuls les postulats et théorème pouvant avoir une utilité dans les mécanismes de simplification seront abordés.

- **Postulat 2** : à chaque opérateur est associé un élément neutre.

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

- **Postulat 3** : commutativité des opérateurs + et *.

$$x + y = y + x \quad x \cdot y = y \cdot x$$

- **Postulat 4** : les opérateurs + et * sont mutuellement distributifs.

$$x + (y \cdot z) = (x + y) \cdot (x + z) \quad x \cdot (y + z) = (x \cdot y) + (x \cdot z)$$

- **Postulat 5** : pour chaque variable x , il existe son complément noté \bar{x} .

$$(x + \bar{x}) = 1 \quad (x \cdot \bar{x}) = 0$$

- **Théorème 4** : loi de 0 et 1.

$$(x + 1) = 1 \quad (x \cdot 0) = 0$$

- **Théorème 5** : loi d'impotence.

$$(x + x) = x \quad (x \cdot x) = x$$

- **Théorème 6** : première loi d'absorption.

$$x + (x \cdot y) = x \quad x \cdot (x + y) = x$$

- **Théorème 7** : deuxième loi d'absorption.

$$x + (\bar{x} \cdot y) = x + y \quad x \cdot (\bar{x} + y) = x \cdot y$$

- **Théorème 10** : loi d'involution.

$$\bar{\bar{x}} = x$$

➤ **Théorème 11** : les opérateurs + et * sont associatifs.

$$(x + y) + z = x + (y + z) \quad (x.y).z = x.(y.z)$$

➤ **Théorème 12** : lois de De Morgan.

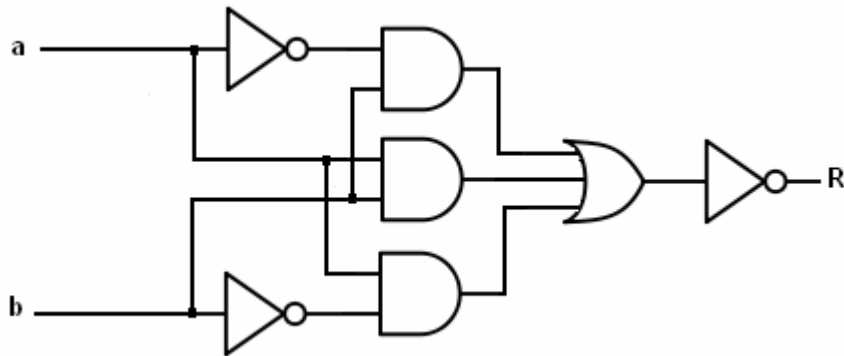
$$\overline{x + y} = \bar{x}.\bar{y} \quad \overline{x.y} = \bar{x} + \bar{y}$$

➤ **Le OU exclusif.**

$$x \oplus y = \bar{x}.y + x.\bar{y}$$

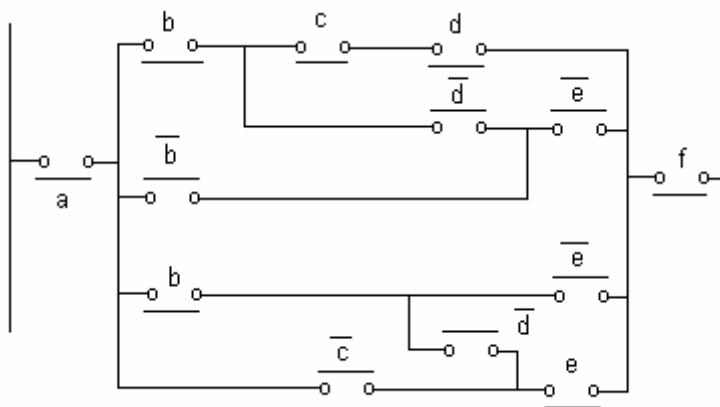
b) Exercices.

I- Soit le schéma logique suivant :



- 1- Donner la représentation algébrique du schéma logique
- 2- Simplifier l'équation en vous basant sur les postulats et théorèmes
- 3- Fournir le schéma logique correspondant à l'équation simplifiée.

II- Soit le schéma électrique suivant :



- 1- Donner la représentation algébrique du schéma électrique
- 2- Simplifier l'équation en vous basant sur les postulats et théorèmes
- 3- Fournir le schéma électrique correspondant à l'équation simplifiée.

III- Simplifier les équations suivantes :

1) $(a \oplus b).a$

2) $\overline{(a+b)}.a$

3) $(a+b).(\bar{a}+\bar{b})$

4) $(\bar{a}+b) \oplus \bar{a}.\bar{c}$

IV- Donner le schéma à base de portes logiques correspondant aux équations suivantes :

1) $\overline{(\bar{a}+\bar{b})}.(\bar{a}+\bar{b})$

2) $(\bar{a}+b).(c+d)$

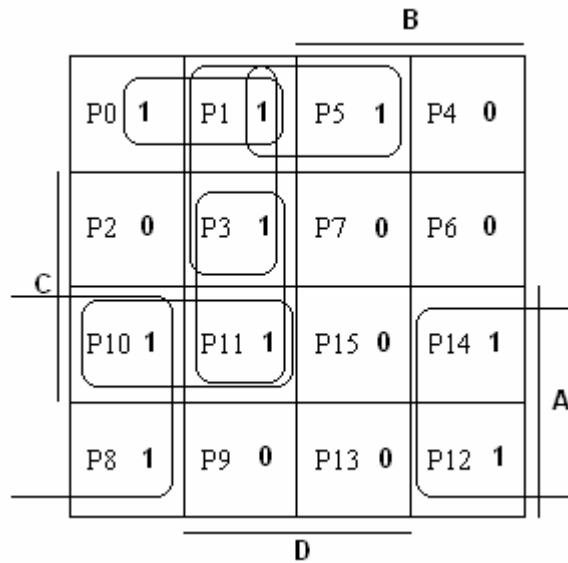
3) $(\bar{a}.\bar{b}) \oplus \bar{c}.\bar{d}$

c) Simplification par les tables de Karnaugh.

Soit la table de Karnaugh suivante :

		B			
		<hr/>		<hr/>	
		P0 1	P1 1	P5 1	P4 0
		P2 0	P3 1	P7 0	P6 0
C		P10 1	P11 1	P15 0	P14 1
		P8 1	P9 0	P13 0	P12 1
		<hr/>			
		D			
		<hr/>			
		A			

La première étape consiste à trouver les regroupements les plus grands possibles de cases voisines par un côté contenant des uns. Ces groupes doivent comprendre 2, 4 8 ou 16 cases. Il est inutile de reprendre des groupes qui seraient eux-mêmes inclus complètement dans des groupes plus grands bien qu'une ou plusieurs cases peuvent très bien se retrouver dans différents groupes. Il faut également tenir compte du fait que le bord droit est voisin avec celui de gauche et le bord supérieur est voisin avec le bord inférieur.



Si nous voulons mettre en évidence chaque groupe en reprenant les produits canoniques qu'ils contiennent, nous aurons pour les 7 groupes :

- 1) P8 P10 P12 P14
- 2) P10 P11
- 3) P0 P8
- 4) P0 P1
- 5) P1 P3
- 6) P3 P11
- 7) P1 P5

Une fois ces groupes trouvés, il convient de déterminer ceux que l'on peut appeler impliquant premier essentiel et impliquant premier non essentiel. Un impliquant premier essentiel comprend au moins un produit canonique (une case) qui n'est repris dans aucun autre groupe. Nous noterons IPE pour Impliquant Premier Essentiel et IPNE pour Impliquant Premier Non Essentiel.

P8 P10 P12 P14	IPE	A cause de P12 et P14
P10 P11	IPNE	
P0 P8	IPNE	
P0 P1	IPNE	
P1 P3	IPNE	
P3 P11	IPNE	
P1 P5	IPE	A cause de P5

La solution finale comprendra tous les impliquants premiers essentiels et un choix arbitraire d'impliquants premiers non essentiels de sortes que l'ensemble des produits canoniques associés à un coefficient de valeur 1 soient repris dans l'équation finale.

Somme des IPE : P8 P10 P12 P14 + P1 P5 (il manque alors P11 P0 P3). Deux solutions optimales existent alors :

$$P8 P10 P12 P14 + P1 P5 + P3 P11 + P0 P1 \text{ ou } P8 P10 P12 P14 + P1 P5 + P3 P11 + P0 P8$$

Il nous reste maintenant à déterminer les équations de chacun des différents groupes en essayant de les voir comme l'intersection d'une ou plusieurs zones verticales et d'une ou plusieurs zones

horizontales, chaque zone étant associée à une variable simple ou à son complément. Pour déterminer les zones et de ce fait les variables, il suffira de procéder au calcul suivant :

Un groupe de 2 cases en final permet de supprimer 1 variable de celles de départ

Un groupe de 4 cases en final permet de supprimer 2 variables de celles de départ

Un groupe de 8 cases en final permet de supprimer 3 variables de celles de départ

Si nous prenons le groupe P0 P1 constitué de deux produits canoniques, nous pourrions supprimer une variable sur les quatre de départ et donc il nous faudra trouver trois zones dont l'intersection donnera P0 P1.

	B			
	P0 1	P1 1	P5 1	P4 0
	P2 0	P3 1	P7 0	P6 0
	P10 1	P11 1	P15 0	P14 1
	P8 1	P9 0	P13 0	P12 1
	D			
C	A			

La zone de couleur bleue correspond à la variable \bar{A}

La zone de couleur verte correspond à la variable \bar{C}

La zone de couleur rouge correspond à la variable \bar{B}

L'équation logique de la zone P0 P1 sera donc $\bar{A}.\bar{B}.\bar{C}$

Reprenons un autre groupe, soit P8 P10 P12 P14. Ce groupe contient 4 produits canoniques, ce qui nous permet de supprimer 2 variables sur les quatre de départ et donc, il nous faudra trouver deux zones dont l'intersection donnera P8 P10 P12 P14.

	B			
	P0 1	P1 1	P5 1	P4 0
	P2 0	P3 1	P7 0	P6 0
	P10 1	P11 1	P15 0	P14 1
	P8 1	P9 0	P13 0	P12 1
	D			
C	A			

La zone de couleur bleue correspond à la variable \bar{D}

La zone de couleur rouge correspond à la variable A

L'équation logique de la zone P8 P10 P12 P14 sera donc $A.\overline{D}$

Exercices :

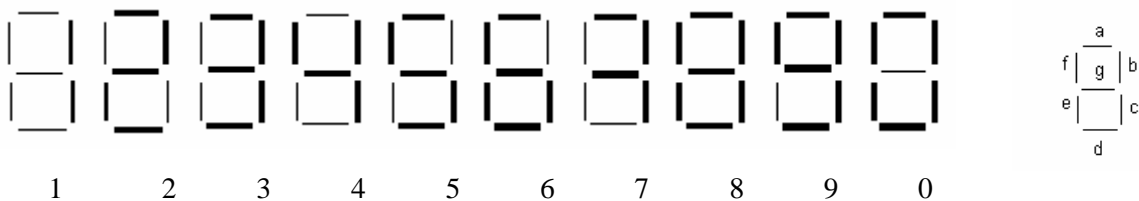
- 1- Etablir l'équation du dernier groupe P3 P11
- 2- En reprenant l'équation obtenue au point 2.6 b II, utiliser la grille de Karnaugh pour en obtenir la forme simplifiée.

3 Les fonctions combinatoires.

3.1 Définitions.

La logique combinatoire comprend les fonctions logiques pour lesquelles une configuration des variables d'entrées donne une et une seule configuration de la ou des variables de sortie.

3.2 Le décodeur 7 segments.



Le décodeur 7 segments permet, à partir d'une donnée binaire codée sur 4 bits, de pouvoir commander les différents segments de l'afficheur de sorte d'afficher la valeur décimale de cette donnée.

Un segment est allumé si la commande correspondante est à la valeur logique 0

Un segment est allumé si la commande correspondante est à la valeur logique 1

Nous pouvons établir la table de vérité suivante :

Entrées				Sorties						
w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

Nous allons déterminer la fonction qui permet la commande de chacun des segments en nous basant sur les tables de Karnaugh. Avec 4 variables d'entrée, nous avons pour chaque sortie une table de Karnaugh d'allure carrée comprenant 16 cases (2^4).

Dans notre table de vérité, nous retrouvons uniquement la sortie définie pour dix configurations d'entrées alors que nous devrions en avoir 16. Il suffit de compléter la grille en indiquant les

valeurs 1 ou 0 dans les cases correspondant aux produits canoniques traités dans la table de vérité. Les cases vides seront alors remplies arbitrairement pour nous permettre des regroupements les plus grands possibles.

a) Segment a

		X			
		P0 1	P1 0	P5 1	P4 0
		P2 1	P3 1	P7 1	P6 1
y		P10	P11	P15	P14
		P8 1	P9 1	P13	P12
		Z			
				w	

		X			
		P0 1	P1 0	P5 1	P4 0
		P2 1	P3 1	P7 1	P6 1
y		P10 1	P11 1	P15 1	P14 1
		P8 1	P9 1	P13 1	P12 1
		Z			
				w	

Les différents regroupements auront donc comme équations :

$$P8 P9 P10 P11 P12 P13 P14 P15 : W$$

$$P2 P3 P6 P7 P10 P11 P14 P15 : Y$$

$$P0 P2 P10 P8 : \overline{Z}\overline{X}$$

$$P5 P7 P15 P13 : XZ$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$W + Y + \overline{Z}\overline{X} + XZ$$

b) Segment b

	x			
	P0 1	P1 1	P5 0	P4 1
y	P2 1	P3 1	P7 1	P6 0
	P10 1	P11 1	P15 1	P14
	P8 1	P9 1	P13	P12
	z			
	w			

	x			
	P0 1	P1 1	P5 0	P4 1
y	P2 1	P3 1	P7 1	P6 0
	P10 1	P11 1	P15 1	P14
	P8 1	P9 1	P13	P12 1
	z			
	w			

Les différents regroupements auront donc comme équations :

$$P0 P1 P2 P3 P8 P9 P10 P11 : \overline{X}$$

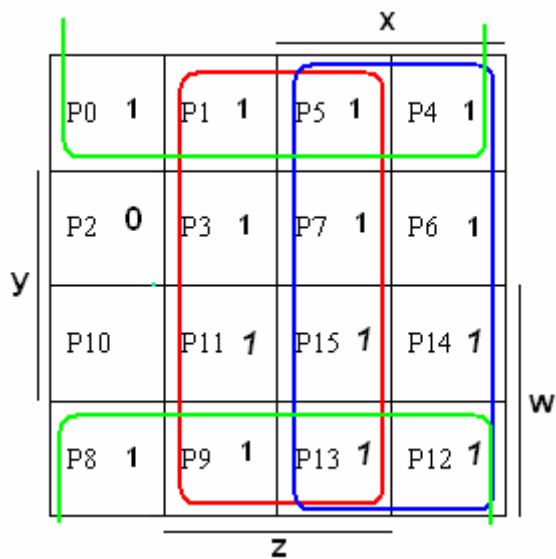
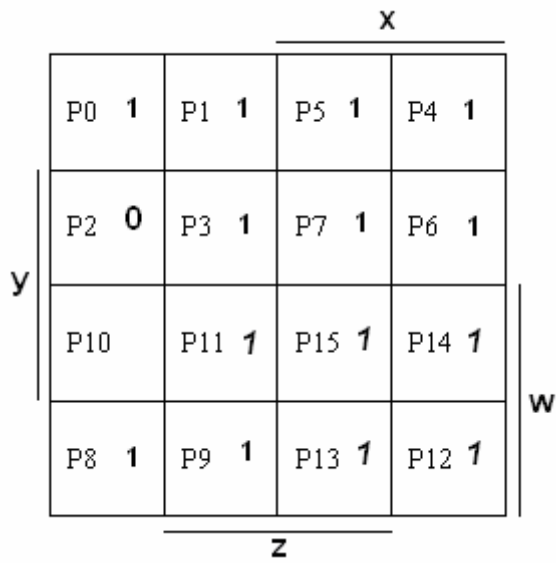
$$P3 P7 P11 P15 : YZ$$

$$P0 P4 P8 P12 : \overline{Y}Z$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$\overline{X} + YZ + \overline{Y}Z$$

c) Segment c



Les différents regroupements auront donc comme équations :

$$P1 P3 P5 P7 P11 P15 P9 P13 : Z$$

$$P0 P1 P4 P5 P8 P9 P12 P13 : \bar{Y}$$

$$P4 P5 P6 P7 P14 P15 P12 P13 : X$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$\bar{Y} + Y + X$$

d) Segment d

	x			
	P0 1	P1 0	P5 1	P4 0
y	P2 1	P3 1	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
	z			
	w			

	x			
	P0 1	P1 0	P5 1	P4 0
y	P2 1	P3 1	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
	z			
	w			

Les différents regroupements auront donc comme équations :

$$P0 P2 P10 P8: \overline{Z} \overline{X}$$

$$P10 P11 P15 P14 P8 P9 P13 P12: W$$

$$P2 P3 P10 P11: Y \overline{X}$$

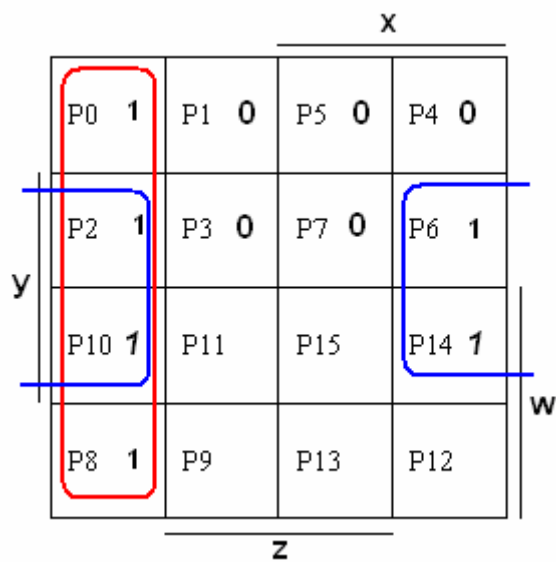
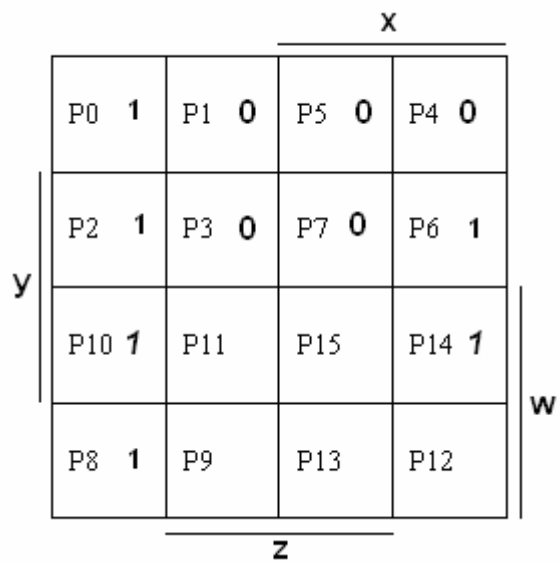
$$P2 P10 P6 P14: Y \overline{Z}$$

$$P5 P13: X Z \overline{Y}$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$\overline{Z} \overline{X} + W + Y \overline{X} + Y \overline{Z} + X Z \overline{Y}$$

e) Segment e



Les différents regroupements auront donc comme équations :

$$P0 P2 P10 P8: \overline{Z}\overline{X}$$

$$P2 P10 P6 P14: \overline{Z}Y$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$\overline{Z}\overline{X} + \overline{Z}Y$$

f) Segment f

		x		
	P0 1	P1 0	P5 1	P4 1
y	P2 1	P3 0	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
		z		w

		x		
	P0 1	P1 0	P5 1	P4 1
y	P2 0	P3 0	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
		z		w

Les différents regroupements auront donc comme équations :

P10 P11 P15 P14 P8 P9 P13 P12: W

P4 P6 P14 P12: $X\bar{Z}$

P5 P4 P13 P12: $\bar{Y}X$

P0 P4 P8 P12 : $\bar{Y}\bar{Z}$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$W + X\bar{Z} + \bar{Y}X + \bar{Y}\bar{Z}$$

g) Segment g

	x			
	P0 0	P1 0	P5 1	P4 1
y	P2 1	P3 1	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
	z			
	w			

	x			
	P0 0	P1 0	P5 1	P4 1
y	P2 1	P3 1	P7 0	P6 1
	P10 1	P11 1	P15 1	P14 1
	P8 1	P9 1	P13 1	P12 1
	z			
	w			

Les différents regroupements auront donc comme équations :

$$P10 P11 P15 P14 P8 P9 P13 P12: W$$

$$P4 P6 P14 P12: X\bar{Z}$$

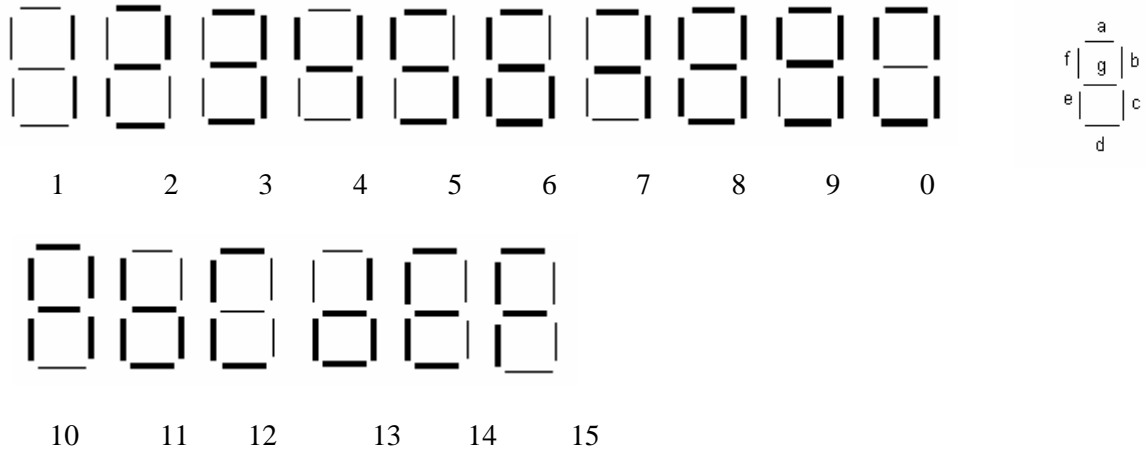
$$P2 P3 P10 P11: Y\bar{X}$$

$$P5 P4 P13 P12: \bar{Y}X$$

Avec comme équation finale la somme des équations des différents groupes obtenues précédemment :

$$W + X\bar{Z} + Y\bar{X} + \bar{Y}X$$

3.3 Exercice : le décodeur 7 segments (variante hexadécimale).



3.4 Le code gray.

Entrées				Sorties			
w	x	y	z	a	b	c	d
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1