ALGEBRE DE BOOLE
ET CIRCUITS LOGIQUES

Chapitre 1 : généralités.
Chapitre 2 : minimisation des fonctions.
Chapitre 3: réalisation des fonctions combinatoires
Chapitre 4: fonctions séquentielles**.**

Chapitre 1 : généralités : grandeurs et fonctions booléennes.

On rencontre fréquemment des grandeurs ayant deux états et deux seulement :

une relation est «VRAIE » ou «FAUSSE »
un interrupteur est «OUVERT» ou «FERME »
un transistor «CONDUIT» ou «NE CONDUIT PAS »
un support magnétique peut être aimanté de haut en bas ou de bas en haut,

etc..
Ces grandeurs sont dites «Booléennes» du nom de leur inventeur et formalisateur Boole.Pour unifier l'étude des fonctions booléennes, on unifie les notations en utilisant les deux symboles 0 et 1 avec par exemple les conventions

VRAI ,OUVERT, CONDUIT, etc... sont notés 1 par convention,
FAUX, FERME, NE-CONDUIT-PAS, etc... sont notés 0 par convention.

1 et 0 sont des symboles purement conventionnels (on pourrait les permuter, ou choisir 1 et 2, - et +, etc..) ; ce ne sont pas des chiffres ; les opérations arithmétiques doivent être oubliées dans cette étude.

Toujours par convention on définit la relation d'ordre **:** **0 < 1**

1.1. Fonctions booléennes, généralités

D'une façon générale on s'intéresse aux applications de $\{0,1\}^n$ dans $\{0,1\}$ou fonctions booléennes.

1.1.1. fonctions d'une variable.

Une fonction booléenne y = f(x) de la variable booléenne x peut être simplement définie « point par point », c'est à dire en donnant explicitement la valeur de la fonction pour chaque valeur de la variable. On constate qu'il y a 4 fonctions d'une variable données par le tableau suivant :

x	f0(x)	f1(x)	f2(x)	f3(x)
0	0	0	1	1
1	0	1	0	1

Il est facile de voir qu'il ne peut y avoir d'autres fonctions d'une seule variable.

Ainsi la fonction f0(x) est la fonction **dégénérée** constamment nulle ainsi que la fonction f3(x) constamment unité, la fonction f1(x) est identique à x,

la fonction f2(x) est le «contraire» de x, appelée son complément et notée $\overline{x}$ (parfois x').
on peut noter ses propriétés :
$0 = \overline{1}$, $1 = \overline{0}$, et surtout $(\overline{\overline{x}}) = x$.

1.2.2. fonctions de deux variables z = f(x,y) :

Comme précédemment elles peuvent être définies par la valeur de la fonction pour les 4 combinaisons de valeurs de x et y, c'est à dire par le tableau :

x	y	f0	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
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

parmi ces fonctions on retrouve

a) les fonctions constantes f0 et f15,
b) les fonctions respectivement identiques à x et à y (f3 et f5),
c) les fonctions compléments de x et y (f12 et f10).
d) Les autres sont des «vraies» fonctions de x et y.

Certaines ont des propriétés remarquables qui vont faire l'objet de la suite du chapitre.

Autres fonctions:
On peut s'intéresser aux fonctions de 3,4,… variables. Mais leur nombre devient vite gigantesque, bien que fini.

On montre facilement qu'il y a 2^{2^n} fonctions de n variables, par exemple pour n=3 il y en a 256, pour n=4 il y en a 32000 etc… On ne s'intéressera évidemment pas à une étude exhaustive de ces fonctions.

NOTA:
Les fonctions ci dessus sont dites "complètement spécifiées" c'est à dire que la valeur de la fonction pour une combinaison donnée des variables est soit 0 soit 1. Il existe cependant des cas ou cette valeur est indéterminée (notée ϕ) L'étude de ces fonctions est donnée en annexe 1 "fonctions incomplètement spécifiées". Dans un premier temps cette étude peut être ignorée.

1.3. fonctions ET et OU. (Treillis de Boole)

Les fonctions f1 et f7 ont des propriétés remarquables, bases de la plupart des calculs booléens :

1.3.1. fonction OU (f7)

X	y	f7
0	0	0
0	1	1
1	0	1
1	1	1

Cette fonction est égale à 1 si x OU y sont égaux à 1, d'où son nom. Elle est égale à 0 quand x et y sont nuls.

Notation : pour la commodité on note cette fonction **x + y**, bien que + ne désigne pas l'addition arithmétique !

On remarque également qu'avec la convention $0<1$, cette fonction est le maximum de x et de y. On en déduit une série de propriété classique du maximum :

(1) commutativité : max(x,y)=max(y,x) ou encore $x+y = y+x$

(2) associativité : max(x,max(y,z)) = max(max(x,y),z)
ou encore $x+(y+z)=(x+y)+z = x+y+z$

(3)idempotence : max(x,x)=x ou encore $x+x=x$ (et non 2x !)

(4) neutre 0 : max(x,0)=max(0,x)=x (puisque $0\leq x$) ou encore $x+0 = 0+x = x$

(5) 1 est absorbant : max(x,1)= max(1,x) = 1 (puisque $x\leq 1$) ou encore $x+1 = 1+x = 1$

(6) absence d'inverse pour la somme : en raison de (5), il n'existe pas quelque soit x un élément y tel que $x+y=0$.

Ces propriétés confèrent à l'ensemble des grandeurs booléennes une structure mathématique dite de «demi treillis distributif»

A ces propriétés fondamentales, on peut ajouter quelques remarques intéressantes pour la suite :
(7) **$x \leq x+y$** par définition d'un maximum
(8) **si $x \leq a$, alors $x+a=a$** (idem)

REMARQUE Cette opération a exactement les mêmes propriétés que l'union de la théorie des ensembles

1.3.2. fonction ET

x	y	F1
0	0	0
0	1	0
1	0	0
1	1	1

Cette fonction est égale à 1 si x ET y sont égaux à 1, d'où son nom. Elle est égale à 0 quand x ou y sont nuls.

Notation : pour la commodité on note cette fonction xy ou x.y

On remarque également qu'avec la convention $0<1$, cette fonction est le minimum de x et de y. comme ci dessus, on en déduit une série de propriété classique du minimum :

(10) commutativité : min(x,y)=min(y,x) ou encore $xy = yx$

(11) associativité : min(x,min(y,z)) = min(min(x,y),z)
ou encore $x(yz)=(xy)z = xyz$

(12) idempotence : min(x,x)=x ou encore $xx=x$ (et non x^2 !)

(13) neutre 1 : min(x,1)=min(1,x)=x (puisque $x\leq1$) ou encore $x.1 = 1.x = x$

(14) 0 est absorbant : min(x,0)= min(0,x) = 0 ou encore $x0 = 0x = 0$

NOTA: comme précédement très voisin de l'opérateur ensembliste UNION
(15) en général, absence d'inverse pour le produit: en raison de (14), il n'existe pas quelque soit x un élément y tel que xy=1. Mais on peut dire que tout élément non nul (il n'y en n'a qu'un!)a un inverse !. La situation est la même que pour les réels!

(16) Ces propriétés confèrent à l'ensemble des grandeurs booléennes une structure dite de «demi treillis distributif».

(17) Les deux fonctions ET et OU donnent à l'ensemble la structure de treillis distributif.

*Remarque : Il y a une symétrie parfaite entre les fonctions ET et OU due à la symétrie entre les symboles 0 et 1 (si on permute les symboles 0 et 1 dans le tableau définissant le OU, on obtient la définition du ET). Cette symétrie est appelée **dualité**. A toute propriété ou théorème correspond la propriété ou le théorème dual.*

(18) Distributivité de ET pour OU :

min(x,max(y,z)) = max(min(x,y),min(x,z)) ou encore $\boxed{x (y + z) = x y + x z}$

Cette propriété du max et du min, qui n'est peut-être pas évidente pour tous, peut se démontrer de la manière suivante :

pour démontrer que $x (y + z) = x y + x z$, on considère deux cas :
cas x=0 : $0.(x + y) = 0$ et $0 x + 0 y = 0 + 0 = 0$ puisque 0 est absorbant pour ET

cas x=1 : $1.(x + y) = x+y$ et $1.x + 1.y = x + y$ puisque 1 est neutre pour OU

cette propriété semblable à la distributivité du produit pour la somme en arithmétique simplifie la pratique des calculs, par exemple, on développera facilement des expressions telles que $(a+b+c)(d+e)(f+g+h)$ comme en algèbre classique.

A ces propriétés fondamentales, on peut ajouter comme précédemment les propriétés évidentes suivantes :

(19) **$x\,y \leq x$** par définition d'un minimum.

(20) on en déduit que **$x + a\,x = x$** (puisque $a.x \leq x$) c'est à dire **qu'on peut supprimer dans une somme tous les multiples d'un monôme**, exemple : $a\,b\,c + a\,c = a\,c$

(21) si $x \leq a$, alors $x\,a = x$.

(22) distributivité de OU pour ET

par dualité on déduit de (17) la propriété : $x+yz= (x+y)(x+z)$

sans utiliser la dualité, on pourrait développer :

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

$= x + x\,z + x\,y + y\,z$ d'après (12)

$= x + y\,z$ d'après la remarque (b) ci dessus, puisque $x\,z \leq x$ et $x\,y \leq x$

Nota : pratique des calculs :

Le développement d'expression donnera en général lieu à de nombreuses simplifications ; exemple :

$A= (a + b + c)(a + b + d) = a\,a + a\,b + a\,d + \cancel{a}\,b + b\,b + b\,d + a\,c + b\,c + c\,d$

expression qui se simplifie en supprimant les monômes répétés (idempotence de la somme):

$A = a + \cancel{a}\,b + \cancel{a}\,d + \cancel{a}\,b + b + \cancel{b}\,d + \cancel{a}\,c + \cancel{b}\,c + c\,d$

$= a + b + c\,d$ *(suppression des multiples de a et b)*

pour éviter d'écrire des termes à supprimer ensuite*, il faire la remarque suivante : posons $X= a + b$; l'expression à développer s'écrit $(X + c)(X + d)$ et d'après (21) ceci s'écrit $X + c\,d = a + b + c\,d$ (distributivité de OU pour ET) on évite ainsi des écritures inutiles.*

exemple développer $f(a,b,c)= (a + b + c)(c + d + e)(a + c + d)(c + e + f)$

on réorganise en $((a + c) + b)((a + c) + d)((c + e) + d)((c + e) + f)$

$= (a + c + b\,d)(c + e + d\,f)$

$= c + (a + b\,d)(e + d\,f)$

$= c + a\,e + a\,d\,f + b\,d\,e + b\,d\,f$

ce qui donne un calcul plus rapide que d'écrire les 81 monômes obtenus en développement bêtement !

1.3.3. relations avec le complément

(22) $a + \overline{a} = 1$ (évident)
application : simplification de certaines expressions, exemple : $a\,x + a\,\overline{x} = a\,(x + \overline{x}) = a$

(24) complément d'une somme : $\boxed{\overline{a + b} = \overline{a} \,.\, \overline{b}}$ (théoreme de De Morgan)
en effet on peut distinguer deux cas :

cas a=0 alors $\overline{a+b} = \overline{0+b} = \overline{b}$ et $\overline{a} \,.\, \overline{b} = \overline{0} \,.\, \overline{b} = 1 \,.\, \overline{b} = \overline{b}$

cas a=1 alors $\overline{a+b} = \overline{1+b} = \overline{1} = 0$ et $\overline{a} \,.\, \overline{b} = \overline{1} \,.\, \overline{b} = 0 \,.\, \overline{b} = 0$

(25) complément d'un produit : $\boxed{\overline{a.b} = \overline{a} + \overline{b}}$ (par dualité)

1.3.4. Forme canonique :

Toute fonction f(x) peut se mettre sous la forme dite canonique (unique):

(26) $\boxed{f(x) = x\,f(1) + \overline{x}\,f(0)}$ où f(0) et f(1) sont des constantes 0 ou 1

(démonstration évidente en distinguant les cas x=1 et x=0) On peut appliquer ceci pour une fonction de deux variables :

$f(x,y) = x\,f(1,y) + \overline{x}\,f(0,y)$ d'après (26)

$= x\,.(\,y\,.f(1,1) + \overline{y}\,f(1,0)) + \overline{x}\,.(\,y\,f(0,\,) + \overline{y}\,f(0,0)\,)$ toujours d'après (26)

on a donc :

(27) $f(x,y) = x\,y\,f(11) + x\,\overline{y}\,f(10) + \overline{x}\,y\,f(01) + \overline{x}\,\overline{y}\,f(00)$

(on notera la correspondance entre l'apparition de la variable accentuée et celle du 0 dans la fonction)
ceci se généralise à un nombre quelconque de variables :

(28) $f(w,x,y,z...) = w\,x\,y\,z\,f(1,1,1,1...) + w\,x\,y\,\overline{z}\,f(1,1,1,0...) + \ldots + \overline{w}\,\overline{x}\,\overline{y}\,\overline{z}f(0,0,0,0...)$

La forme canonique d'une fonction de n variables comporte donc 2^n termes appelés monômes canoniques. Cette forme est en général très loin d'être minimale, mais elle a l'avantage d'être unique.

1.3.4.1. Application : détermination de l'expression algébrique d'une fonction donnée par points. exemple : soit la fonction f(x,y) définie par :

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

cette fonction s'écrit d'après (27) : $f(x,y)= x\,y\,0 + x\,\bar{y}\,1 + \bar{x}\,y\,1 + \bar{x}\,\bar{y}\,0 = x\,\bar{y} + \bar{x}\,y$

On en déduit au passage deux théorèmes:

Théorème 1: **toute fonction est un polynôme.**

Théoreme 2 avec les opérations ET, OU, COMPLEMENT **on peut exprimer toute fonction**..

1.3.4.2. Nota: l'expression polynomiale d'une fonction n'est pas toujours unique comme le montre l'exemple de la fonction OU : f(00)=0, f(11)=f(10)=f(01)=1 donc x+y peut également s'écrire: $x+y= x\,y + x\,\bar{y} + \bar{x}\,y$

On voit donc que la forme canonique ne donne pas forcément l'expression minimale de la fonction. D'une manière générale une fonction a <u>plusieurs écritures possibles</u> sous forme polynomiale ; il s'ensuit une réelle difficulté à décider si deux fonctions sont égales ou non ;

par exemple la fonction $x + \bar{x}\,y$ est elle égale ou non à $y + x\,\bar{y}$? Par contre sous forme canonique l'égalité est évidente

Le chapitre suivant sera consacré à la minimisation des expressions.

1.3.4.3. Passage d'une somme de produits quelconque à la forme canonique

lemme : toute expression composée de monômes canoniques (à n variables) est une forme canonique (trivial)

Il suffit donc de transformer chaque monôme de la somme en monôme canonique.

méthode : remplacer dans chaque monôme une lettre absente par la somme d'elle même et de son complément, puis développer en éliminant les redondances éventuelles.

Exemple 1:

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

Exemple 2 :

$$f(x,y,z)= x + y\,z = x\,(y + \bar{y})(z + \bar{z}) + (x + \bar{x})\,y\,z$$ développé en :

$$= x\,y\,z + x\,\bar{y}\,z + x\,y\,\bar{z} + x\,\bar{y}\,\bar{z} + x\,y\,z + \bar{x}\,y\,z$$ on élimine les redondances :

$$= x\,y\,z + x\,\bar{y}\,z + x\,y\,\bar{z} + x\,\bar{y}\,\bar{z} + \bar{x}\,y\,z$$ forme finale

1.4. Algèbre de Boole.

Cette structure beaucoup plus intéressante **en théorie** que la première fait appel à la fonction ET et à la fonction OU_EXCLUSIF :

1.4.1. Fonction OU EXCLUSIF.

X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Considérons la fonction f6 du paragraphe 1.2.2. :Cette fonction est égale à 1 quand x OU y (MAIS PAS LES DEUX) sont égaux à 1 ; on l'appelle donc le OU EXCLUSIF.

On peut aussi remarquer qu'elle est égale à 1 quand x et y sont différents on l'appelle alors également la DISJONCTION (son complément la fonction f9 égale à 1 quand x et y sont égaux est appelée la CONJONCTION).

On peut enfin remarquer que si on considère les symboles 0 et 1 comme des chiffres arithmétiques, cette fonction est alors la SOMME MODULO 2.

Notation : $z = x \oplus y$

Propriétés.

1.4.1.1. structure de groupe: de par la propriété d'être la somme modulo 2 cette fonction $\oplus$ confère à l'ensemble des grandeurs booléennes la propriété d'être un **groupe commutatif.** C'est une propriété mathématique classique de la somme modulo 2 qui peut cependant se démontrer de façon simple :

(29) commutativité : $x \oplus y = y \oplus x$ (évident en raison de la symétrie du tableau).

(30) 0 élément neutre : $0 \oplus x = x \oplus 0 = x$

(31) inverse : $x \oplus x = 0$ x est son propre inverse (évident)

(32) $x \oplus 1 = \overline{x}$ (évident)

(33) associativité : $x \oplus (y \oplus z) = (x \oplus y) \oplus z$ noté $x \oplus y \oplus z$

distinguons deux cas :

y=0 : $x \oplus (0 \oplus z) = x \oplus z$ et $(x \oplus 0) \oplus z = x \oplus z$ (d'après (30))

y=1 : $x \oplus (1 \oplus z) = x \oplus \overline{z}$ et $(x \oplus 1) \oplus z = \overline{x} \oplus z = x \oplus \overline{z}$ (si $\overline{x} \neq z$, alors $x \neq \overline{z}$)

on a bien les propriétés d'un **groupe commutatif** .

1.4.1.2. Structure de corps.

Le ET est **distributif à gauche et à droite pour le OU exclusif :**
(34) $x.(y \oplus z) = x.y \oplus x.z$ (évident en distinguant les cas x=0 et x=1)
(35) de plus, on peut remarquer que toute grandeur non nulle a un inverse pour le ET (il n'y en n'a qu'une c'est 1 !)
Les grandeurs booléennes forment donc un **corps commutatif.**

1.4.1.3 structure d'algèbre.

Sur l'ensemble des **fonctions** booléennes on peut donc définir deux opérations internes :

le **ou exclusif** (associative, commutative qui a un neutre 0 et un inverse)
le **et** (associatif, commutatif, distributif pour le ou exclusif, qui a un élément neutre 1 , mais pas d'inverse),

Les fonctions booléennes forment donc un **anneau commutatif.**

De plus on définit un produit extérieur par un scalaire pris sur le corps des booléens, ce qui donne à l'ensemble des fonctions la structure d'**algèbre.**

1.4.2. expression des fonctions.

Théorème toute fonction f(x) peut se mettre sous la forme :

(36) $\boxed{f(x) = (f(0) \oplus f(1))\, x \oplus f(0)} = A.x \oplus B$ (où A est indépendant de x)

en effet pour x=0 on a : $(f(0) \oplus f(1)).0 \oplus f(0) = f(0)$
et pour x=1: $(f(0) \oplus f(1))\oplus f(0) = f(1)$

(37) Théorème: toute fonction est **linéaire par rapport à chacune des variables.**

Nota : expression tout à fait comparable à celle de l'algèbre classique : on peut considérer que
$A= f(0) \oplus f(1)$ est la «dérivée booléenne» de la fonction (on pourrait abusivement écrire :

$$A = \frac{f(0) \oplus f(1)}{0 \oplus 1}$$

puisque $0 \oplus 1 = 1$ le seul élément inversible pour le ET.)
Le lecteur pourra vérifier lui même que cette « dérivée booléenne a les mêmes propriétés que la dérivée usuelle (dérivée d'une somme, d'un produit, d'une fonction de fonction, etc…)

en appliquant ce théorème à une fonction de deux ou plusieurs variables on obtient des formes telle que :

(38) $\boxed{f(x,y) = A\,x.y \oplus Bx \oplus Cy \oplus D}$

$f(x,y,z)= Axyz \oplus Bxy \oplus Cxz \oplus Dyz \oplus Ex \oplus Fy \oplus Gz \oplus H$

etc...

Théorème: avec les seules opérations ET et OU_EXCLUSIF **on peut réaliser toute fonction**

NOTA *La structure d'algèbre est beaucoup plus* <u>*riche*</u> *que celle de treillis, et il semblerait que la meilleur façon d'exprimer les fonctions booléennes soit plutôt avec ET et OU_EXCLUSIF qu'avec ET, OU, COMPLEMENT. Malheureusement le OU-EXCLUSIF n'est pas une fonction élémentaire en électronique, contrairement aux autres ;la structure de treillis est en fait la plus utilisée.*

Relations diverse avec ET et OU

(38) $\boxed{x \oplus y = x\,\bar{y} + x\,\bar{y}}$ (cf § 1.3.4.)

(39) $x + y = x\oplus y \oplus xy$

(40) Nota : si deux fonctions f et g sont **disjointes** (leur produit f.g est identiquement nul) alors **$f \oplus g = f + g$**

En effet, $f \oplus g$ et $f + g$ ne diffèrent que par leur valeur quand f et g valent 1 simultanément, ce qui est impossible puisque $f.g \equiv 0$.

Corollaire, à la forme canonique en ET et OU correspond une forme similaire en ET et OU_exclusif ; par exemple pour 2 variables :

$$f(x,y) = x\,y\,f(11) + x\,\bar{y}\,f(10) + \bar{x}\,y\,f(01) + \bar{x}\,\bar{y}\,f(00)$$

$$= x\,y\,f(11) \oplus x\,\bar{y}\,f(10) \oplus \bar{x}\,y\,f(01) \oplus \bar{x}\,\bar{y}\,f(00)$$

puisque tous les monômes canoniques sont disjoints 2 à 2

Ceci donne un moyen de passer de la forme ET/OU à la forme ET/OU_EXCLUSIF :

a) à partir d'une forme quelconque et ET/OU, mettre sous forme canonique ;
b) convertir tous les + en $\oplus$
c) remplacer toutes les lettres accentuées par exemple $\bar{x}$ en $(x \oplus 1)$
d) développer

Exemple

$a + b = a\,b + a\,\bar{b} + \bar{a}\,b$ (forme canonique)

$= a\,b \oplus a\,\bar{b} \oplus \bar{a}\,b$

$= a\,b \oplus a\,(b \oplus 1) \oplus (a \oplus 1)\,b$

$= ab \oplus ab \oplus a \oplus ab \oplus b$

$= a \oplus b \oplus ab$

Inversement : partant d'une forme ET/OU_EXCLUSIF , on procède comme au § 1.3.4.3. pour obtenir une forme canonique : exemple :

f(a,b) = $a \oplus b \oplus ab$ dans chaque monôme chaque lettre absente est remplacée :

$= a\,\mathbf{(b \oplus \bar{b})} \oplus \mathbf{(a \oplus \bar{a})}\,b \oplus ab$ on développe :

$= ab \oplus a\bar{b} \oplus ab \oplus \bar{a}b \oplus ab$ on simplifie :

$= ab \oplus a\bar{b} \oplus \bar{a}b$ on remplace $\oplus$ par + :

$= ab + a\bar{b} + \bar{a}b$ =forme canonique ET/OU.

1.4.3. équations

L'existence de l'inverse permet de faire sensiblement les mêmes calculs qu'en algèbre normale :

exemple , résoudre $a \oplus x = b$: on peut ajouter a aux deux membres de l'équation :

$a \oplus (a \oplus x) = a \oplus b$ ou encore :

$(a \oplus a) \oplus x = a \oplus b$ ou enfin :

ou encore $x = a \oplus b$

On constate qu'on peut faire passer un terme de gauche à droite du signe = ou inversement. Comme en algèbre classique (mais sans changement de signe!)

autres fonctions

Les autres fonctions de 2 variables n'ont pas de propriétés marquantes au point de vue mathématiques. On note cependant :

1.5.1. fonction NOR (fonction f8 du § 1.2.2.)

La fonction f7 est égale à 1 quand **ni** x **ni** y ne valent 1. On l'appelle la fonction NI en français, NOR en anglais elle s'écrit

$$\overline{x}\,\overline{y} = NOR(x,y)$$

elle est commutative (puisque le ET l'est)
elle n'est pas associative, n'a pas de neutre, n'est distributive par rapport à rien

néanmoins elle a deux propriétés fondamentales :

1.5.1.1. elle permet à elle seule de **<u>réaliser n'importe quelle fonction</u>**, puisqu'elle permet de réaliser :

le complément : $NOR(x,x)= \overline{x}$,

le ET: $NOR(\overline{x}, \overline{y})= x.y$,

le OU : $\overline{NOR(x,y)} = x + y$

comme ces trois opérateurs permettent de tout exprimer, il en va de même pour le NOR

1.5.1.2. c'est une des fonctions facilement réalisable en électronique
allié à la précédente, cette propriété fait du NOR une fonction des plus précieuse !

1.5.2. fonction NAND (fonction f14 du § 1.2.2.)

La fonction f14 est égale à 1 quand x ou y valent 0. On l'appelle la fonction NAND en anglais , contraction de NOT-AND elle s'écrit :

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

Ses propriétés sont analogues au NOR :elle est commutative (puisque le OU l'est), elle n'est pas associative, n'a pas de neutre, n'est distributive par rapport à rien. Néanmoins comme la précédente, elle a deux propriétés fondamentales :

1.5.2.1. elle permet à elle seule de **<u>réaliser n'importe quelle fonction</u>**, puisqu'elle permet de réaliser :

le complément : $NAND(x,x)= \overline{x}$,

le OU: $NAND(\overline{x}, \overline{y}) = x + y$, le ET : $\overline{NAND(x,y)} = x\,y$
comme ces trois opérateurs permettent de tout exprimer, il en va de même pour le NAND

1.5.2.2. C'est l'autre fonction facilement réalisable en électronique**. En fait la plus utilisée**.
On verra dans la deuxième partie du cours la réalisation de circuits logiques à l'aide de NOR ou de NAND.

1.5.3.fonction implication (fonction f13 du § 1.2.2.)

Cette fonction notée x—>y dont l'expression est $\overline{x} + y$ est très importante en logique : en revenant à la correspondance 0= faux et 1= vrai, elle signifie quand elle est vraie : « **soit x est faux soit y est vrai** » donc **x implique y.**

Elle n'a aucune propriété marquante.

1.5.4. Fonction conjonction (fonction f9)

C'est le complément du ou exclusif ou disjonction on la note $x \otimes y$

Elle a les **<u>propriétés</u>** **<u>duales du ou-exclusif.</u>** En particulier avec le OU (dual du ET), elle donne également aux fonctions la structure d'algèbre.
C'est par pure habitude qu'on ne l'utilise pas et qu'on lui préfère le OU_EXCLUSIF.

ANNEXE 1
RELATION AVEC LA THEORIE DES ENSEMBLES

La théorie des ensemble utilise des opérations extrêmement voisines de l'algèbre de BOOLE.:
A chaque ensemble E on associe une "fonction caractéristique" C_E appelée appartenance: pour
Tout élément x de E C_E (x)=1; pour tout élément x n'appartenant pas à E, C_E (x)=0.

Nota: que la connaissance de C_E implique la connaissance de E et vice-versa.

LE COMPLEMENT

Soit A un sous ensemble d'un univers U le complément de A noté ¬A est caractérisé par $C_{\neg A}(x) = \overline{C_A(x)}$

L'UNION:

Soient A et B deux ensembles d'un univers U. L'union A∪B est définie comme l'ensemble des éléments appartenant ou à A ou à B. ces éléments peuvent être caractérisés par $C_A(x) + C_B(x)= C_{A\cup B}(x)$.

Il y a donc correspondance entre l'union et la somme booléenne toutes les propriétés de celle ci s'appliquent à l'union: associativité, commutativité, idempotence, neutre 0 (ou l'ensemble vide), absorbant 1 (ou l'univers tout entier).

L'INTERSECTION:

De la même manière, soient A et B deux ensembles d'un univers U. L'intersection A∩B est définie comme l'ensemble des éléments appartenant à la fois à A et B. ces éléments peuvent être caractérisés par $C_A(x) \,.\, C_B(x)= C_{A\cap B}(x)$.

Il y a donc ici aussi correspondance entre l'intersection et le produit booléen, toutes les propriétés de celle ci s'appliquent à l'intersection associativité, commutativité, idempotence, neutre 1 (ou l'univers tout entier)absorbant ∅ (ou l'ensemble vide).

Ces deux opérations sont doublement distributives l'une par rapport à l'autre:

$A\cap(B\cup X) = (A\cap B) \cup (A\cap C)$ puisque $a.(b+c) = a.b + a.c$

$A\cup(B\cap C) = (A\cup B)\cap(A\cup C)$ puisque $a+b.c= (a+b).(a+c)$

On procéderait de même pour toutes les autres propriété.

REMARQUE

Le formalisme de l'algèbre de Boole est plus lisible que celui des ensembles, on aura en général intérêt, quand se posent des problèmes de calcul ensemblistes, de les transposer en algèbre de Boole

ANNEXE 2

1. Fonctions incomplètement spécifiées.

Il arrive fréquemment qu'une fonction **ne soit pas définie** pour certaines combinaisons de variables. Ces valeurs indifférentes sont notées **Φ**.
Exemple de fonction "incomplète": f(0,0)=0, f(0,1)=f(1,0)=1, f(1,1)=**Φ** (la valeur de la fonction pour x=1 et y=1 n'est pas précisée)

Nota:: l'origine de cette non spécification peut être de deux natures

a) ou bien on se désintéresse réellement de la valeur de la fonction;
b) ou bien la combinaison des valeurs ne se produit jamais.

Dans les deux cas la valeur étant indéterminée, elle peut être précisée "à la demande" en fonction de critères divers, en particulier en fonction de critère de **simplicité de l'expression**.

Dans l'exemple précédent: si on dit que la valeur **Φ** peut arbitrairement être fixée à 0 la fonction devient $f= x\,\bar{y} + \bar{x}\,y$

par contre si cette valeur est fixée arbitrairement à 1 on obtient :

$f = x\,\bar{y} + x\,\bar{y} + x\,y = x + y$ on voit donc que le choix peut influer sur le coût final

Notation.

Une fonction incomplètement spécifiée (on dit plus simplement "incomplète") peut être définie par

- la fonction **g** caractérisant l'ensemble des valeurs pour lesquelles f est impérativement égale à 1. Cette fonction est appelée la **borne inférieure de f**

- la fonction **h** caractérisant l'ensemble des valeurs pour lesquelles f est impérativement égale à 0. cette fonction est le **complément de la borne supérieure de f** (la fonction est au maximum égale à 1 quand h=0)

pour qu'il n'y ait pas de contradiction, on doit avoir **g.h=0**

On note **f = {g,h}** et on remarque que quand la fonction est complètement spécifiée, $g=\bar{h}$

la fonction est égale à **Φ** sur $\overline{(g+h)}$

Remarque: $0 \leq \Phi \leq 1$ puisque $\Phi= 0$ ou 1
on en déduit des extensions des opérations OU et ET (déduites du min et du max)

$0+\Phi= \Phi+0=\Phi$, $1+\Phi=\Phi+1=1$, $\Phi+\Phi=\Phi$
et de même
$1.\Phi=\Phi.1=\Phi$, $0.\Phi=\Phi.0=0$ $\Phi.\Phi=\Phi$
et enfin $\overline{\Phi}=\Phi$

propriétés

a) si $f=\{g,h\}$ son complément est $\overline{f}=\{h,g\}$

b) soient $f1=\{g1,h1\}$ et $f2=\{g2,h2\}$

on démontre facilement les propriétés suivantes

$f+g=\{ (f1+f2) , (g1.g2)\}$ et de même $f.g=\{(f1.f2), (g1+g2)\}$

Définition:

Une fonction f1 est dite **compatible** avec f2 si on a les relations:

$$g2 \leq g1 \quad \text{et} \quad h2 \leq h1$$

une telle fonction a donc <u>moins d'indéterminations</u> que f2..

Réaliser une fonction incomplète f1 reviendra à chercher une fonction f2, **complète, compatible** avec f1, et de coût minimum pour un critère donné.

D'une manière générale, plus une fonction est incomplète, plus le nombre de fonctions compatible est grand et donc plus il y a des chance d'en trouver une de coût faible.
Exemple: $f=\{g=xyz,\ h= \overline{x}\ \overline{y}\ \overline{z}\ \}$ fonction très incomplète,

plusieurs fonctions sont une réalisation de f

- sa borne inférieure $f1= xyz$,
- sa borne supérieure $f2= x+y+z$,
- des solutions intermédiaires comme $f3= x+yz$, $f4=y+xz$, etc...
- mais une bonne solution est la fonction complète $f5= x$.

on a bien $x\,y\,z < x$ et $\overline{x}\ \overline{y}\ \overline{z} < \overline{x}$

Ces propriétés seront essentielles pour le calcul des circuits logiques réalisant les fonctions (chapitre 3).

CHAPITRE 2
MINIMISATION DES EXPRESSIONS.

2.1. introduction

On a vu au chapitre 1 que les formes polynomiales d'une même fonctions peuvent avoir des complexités très variées. Exemple:

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

Or la complexité de l'expression est liée directement au coût de réalisation, et il s'agit, sinon de trouver la forme minimale, au moins de trouver une forme <u>relativement optimale</u>. Les méthodes se répartissent en trois grandes classes:

méthodes empiriques (algébriques)
méthodes graphiques,
méthodes rigoureuses.

exemple: simplifier $x\,y + x\,\overline{y} + \overline{x}\,y$

$= x\,y + x\,y + x\,\overline{y} + \overline{x}\,y$ d'après (c)

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

$= x + y$ d'après (a)

Définitions:

A chaque monôme canonique est associé la valeur des variable pour lequel ce monôme vaut 1, c'est à dire un **point** dans l'espace $\{0,1\}^n$ .la valeur 1 du monôme "caractérise le point"

Deux monômes fusionnables (ne différant que par la complémentation d'une lettre sont dits "voisins" .
exemple $w\,x\,y\,z$ et $w\,\overline{x}\,y\,z$ sont fusionnables en un seul: $w\,y\,z$ moins "cher"

La fusion de deux monômes canoniques donne un monôme représentant **deux points** avec une lettre de moins. La fusion de deux tels monômes donne un monôme représentant 4 points (on dit "couvrant" 4 points), etc.... d'une manière générale un monôme de k lettres dans un espace de dimension n couvre 2^{n-k} points. On notera donc que <u>**plus un monôme est grand, moins il a de lettres.**</u>

Monômes maximaux: un monôme est dit "maximal" s'il ne peut être fusionné avec aucun autre, **c'est à dire s'il n'existe pas de monôme plus grand (ayant moins de lettres) et inférieur à la fonction.** Un tel monôme est également dit "premier" (il n'existe pas de diviseur de ce monôme inférieur à la fonction)

Exemple $f = x y z + x \bar{y} z + x y \bar{z} + \bar{x} y z$

le monôme $x y z$ n'est pas maximal puisqu'il existe un monôme $x y$ plus grand que lui et inférieur à f.

Un monôme m est dit "compatible" avec la fonction f si $f \geq m$ c'est à dire si $f + m = f$

Optimiser une fonction revient à la représenter sous forme de somme de <u>monômes maximaux</u> compatibles avec la fonction; par exemple l'expression minimale de la fonction ci dessus est f=xy+yz+zx

forme minimale: Certaines fonctions ont des écritures sous formes de somme de monômes maximaux mais qui ne sont pas minimales;

exemple $f = x \bar{y} + y z + x z$ n'est formée que de monômes maximaux mais n'est pas minimale. La forme minimale étant:

$f = y z + x \bar{y}$

en effet la forme canonique est:

$f = x y z + \bar{x} y z + x \bar{y} z + x \bar{y} \bar{z}$

une façon adroite est de regrouper les deux premiers et les deux derniers monômes:

$f = y z (x + \bar{x}) + x \bar{y} (z + \bar{z}) = y z + x \bar{y}$

mais on pouvait écrire (maladroitement!) en dédoublant les monômes 1 et 3:

$f = x y z + \bar{x} y z + x \bar{y} z + x \bar{y} \bar{z} + x y z + x \bar{y} z$

$f = y z (x + \bar{x}) + x \bar{y} (z + \bar{z}) + x z (y + \bar{y}) = y z + x \bar{y} + x z$

on dit que le monôme xz est "inutile" ou "redondant".

De manière plus générale si A et B sont des monômes quelconques, on a l'égalité

$$\boxed{A x + B \bar{x} = A x + B \bar{x} + A B}$$ où **AB est donc redondant**

En définitive, minimiser une fonction revient à la représenter avec un <u>nombre minimal de monômes maximaux.</u>

Ceci montre que ces méthodes ne sont pas toujours très faciles à manipuler s'il y a beaucoup de monômes; de plus elles nécessitent l'écriture initiale de la fonction sous une forme canonique lourde.

2.2.1. méthode "empirique": elle consiste à appliquer manuellement les règles ci dessus; mais la méthodes est lourde pour deux raisons

:

a) partant du tableau de valeurs de la fonction, on écrit tous les monômes canoniques, et il peut y en avoir beaucoup!

b) la fusion des monômes n'est pas évidente à voir.

2.2.2. méthode graphique: diagramme de Karnaugh.

L'idée est de partir du tableau (dit "tableau de vérité") donnant les valeurs de la fonction pour toutes les combinaisons des variables. Mais ce tableau est **réorganisé** de façon à mettre en évidence les points correspondant aux monômes fusionnables: Considérons par exemple une fonction de 4 variables f(a,b,c,d). A chaque case du tableau correspond un point donc un monôme canonique .

C,D \ A,B	00	01	11	10
00	$\bar a\,\bar b\,\bar c\,\bar d$	$\bar a\,b\,\bar c\,\bar d$	$a\,b\,\bar c\,\bar d$	$a\,\bar b\,\bar c\,\bar d$
01	$\bar a\,\bar b\,\bar c\,d$	$\bar a\,b\,\bar c\,d$	$a\,b\,\bar c\,d$	$a\,\bar b\,\bar c\,d$
11	$\bar a\,\bar b\,c\,d$	$\bar a\,b\,c\,d$	$a\,b\,c\,d$	$a\,\bar b\,c\,d$
10	$\bar a\,\bar b\,c\,\bar d$	$\bar a\,b\,c\,\bar d$	$a\,b\,c\,\bar d$	$a\,\bar b\,c\,\bar d$

$\bar a\,\bar b\,\bar c$

a.c

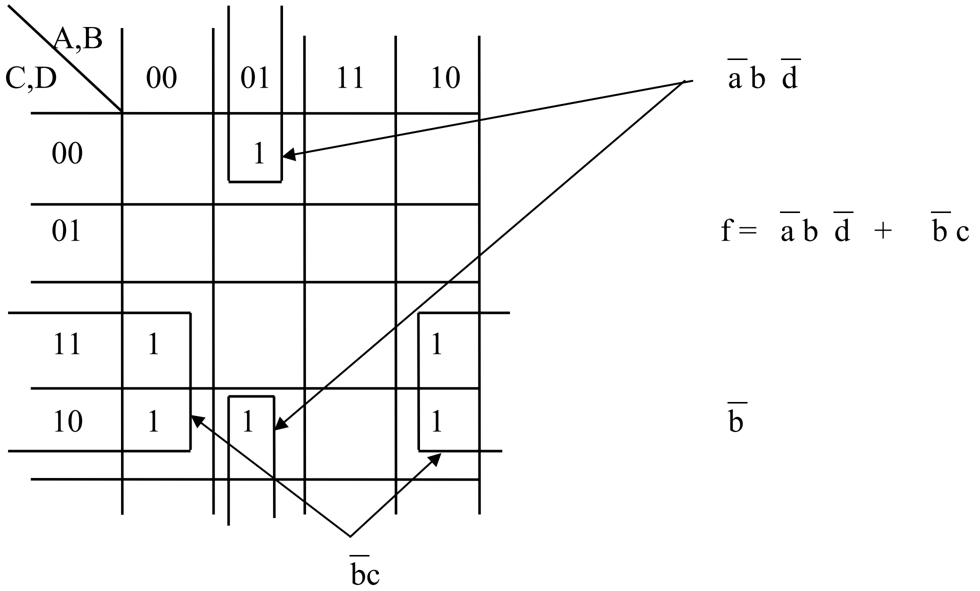
Le voisinage des monômes $\bar a\,\bar b\,\bar c\,d$ et $\bar a\,\bar b\,\bar c\,\bar d$ correspond au voisinage géographique. Donc le monôme fusionné $\bar b\,\bar c$ se matérialise par le rectangle en pointillé (petit) de même le monôme $a\,b\,c\,d + a\,\bar b\,c\,d + a\,b\,c\,\bar d + a\,\bar b\,c\,\bar d = a\,c\,d + a\,c\,\bar d$ =ac cette fusion est représentée par le grand rectangle.

On remarque que ces groupements peuvent se faire directement sur le tableau des valeurs de la fonction, <u>sans écrire les monômes canoniques</u>:

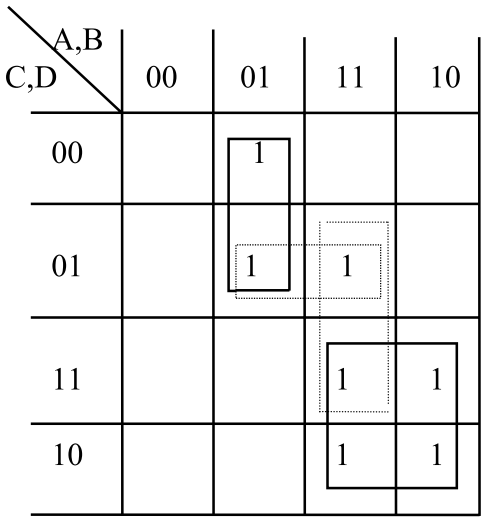
C,D \ A,B	00	01	11	10
00	1			
01	1			
11			1	1
10			1	1

NOTA: a) inversement connaissant la "forme" du monôme on peut trouver son expression; par exemple le "grand" monôme ci dessus est d'une part indépendant de b et de d, et égal à 1 quand a et c valent 1 c'est donc le monôme ac.

b) l'adjacence des cases doit se voir comme si le diagramme était <u>dessiné sur une sphère</u>: les cases à l'extrême gauche sont adjacentes à l'extrême droite, et l'extrême haut à l'extrême bas. Exemple



Dans certains cas le choix des monômes peut être compliqué; exemple:



des deux monômes pointillés, un seul est nécessaire. Donc la fonction a deux expressions possibles.

2.2.3. Méthodes algébriques

La méthode ci-dessus devient inextricable des que le nombre de variable dépasse 4. On doit avoir alors recours à des méthodes plus formalisées et de ce fait programmables

2.2.3.1. Méthode par consensus

Cette méthode est destinée à donner la liste de **tous** les monômes maximaux (bien que tous ne soient pas forcément utiles).

rappel:

$A x + B \bar{x} = A x + B \bar{x} + A B$ Par définition, le monôme AB est le "consensus" des deux premiers.

Théorème

en rajoutant à une forme quelconque **tous les consensus** déterminés de toutes les manières possibles, et en **éliminant les multiples éventuels**, on obtient **tous** les monômes maximaux.

La démonstration en est complexe et sans grand intérêt; mais un exemple permet de fixer les idées:

$$f = a + \bar{a} b + \bar{a} \bar{b} c$$

le consensus des deux premiers est b qui élimine le deuxième monôme : $f = a + \bar{a} \bar{b} c + b$

Dans cette nouvelle forme, le consensus de a et de $\bar{a} \bar{b} c$ donne $\bar{b}.c$ qui élimine $\bar{a} \bar{b} c$ donc $f=a+b+\bar{b} c$

le consensus de b et de $b \bar{c}$ donne c qui élimine $b \bar{c}$; donc la fonction s'écrit $f = a + b + c$ il n'y a plus de consensus possible, la fonction est irréductible.

Exemple 2:

$f = a \bar{b} + \bar{a} b + b \bar{c} + \bar{b} c$

en faisant les consensus de toutes les manières possibles, on obtient

$f = a \bar{b} + \bar{a} b + b \bar{c} + \bar{b} c + a \bar{c} + \bar{a} c$

Cette méthode est facilement programmable, mais difficile à faire à la main

2.2.3.2. Méthode par complémentation

théoreme

si f1 et f2 sont des fonctions toutes les deux sous forme **de somme de tous leurs monômes maximaux** leur produit donne par développement (et élimination des multiples) une **somme de <u>tous</u> les monômes maximaux**

exemple: $f1 = a + \bar{b} c$ $f2 = b + \bar{a} c$ (ces deux fonctions sont sous forme de somme de tous les monômes maximaux, puisque dans chacune aucun consensus n'est possible)

$$f1.f2 = (a + \bar{b}\, c).(b + \bar{a}\, c) = a\,b + \cancel{a\,\bar{a}}\, c + \cancel{b\,\bar{b}}\, c + \bar{a}\,\bar{b}\, c\, c = a\,b + \bar{a}\,\bar{b}\, c$$

corollaire

Si on part du complément de f et qu'on calcule f par la formule donnée au chapitre 1 on obtient tous les monômes maximaux. En effet chaque monôme de $\bar{f}$ donne par complémentation une somme de variables donc des monômes maximaux

Exemple

$$f = \bar{a}\, b\, c + a\, \bar{b}\, c + a\, b\, \bar{c} + \bar{a}\, \bar{b}\, c + \bar{a}\, b\, \bar{c} + a\, \bar{b}\, \bar{c}$$

Son complément est de toutes évidence $\bar{f} = a\, b\, c + \bar{a}\, \bar{b}\, \bar{c}$ (les deux monômes canoniques qui manquent) donc $f = (\bar{a} + \bar{b} + \bar{c}).(a + b + c)$ en développant on obtient:

$$f = \cancel{\bar{a}\, a} + \bar{a}\, b + \bar{a}\, c + \bar{b}\, a + \cancel{\bar{b}\, b} + \bar{b}\, c + \bar{c}\, a + \bar{c}\, b + \cancel{\bar{c}\, c}$$

La méthode donne les mêmes résultats que les consensus, mais beaucoup plus naturelle, elle est plus facile à la main.

2.2.3.3. Choix optimum des monômes: méthode de "couverture"

Il s'agit avec un minimum de monômes de "couvir" les points de la fonction. Dans un premier temps, on recense les points (ou monômes canoniques) de la fonction et pour chaque, la liste des monômes qui le couvrent

Méthode générale

on rencontre fréquemment des problèmes de choix complexes faisant intervenir divers critères. Exemple: choisir une équipe de personnes pour répondre à une mission. Par exemple on a besoin d'un informaticien, d'un angliciste, d'un commercial, d'un matheux. 5 personnes se présentent : Jean(J) informaticien et angliciste, Marie (M) informaticienne et commerciale, Paul (P) informaticien matheux et commercial, Colette (C) matheuse et commerciale et angliciste. Quel est le meilleur choix (le max de compétence pour le moindre coût) ?

on associe à chacun une variable booléenne = 1 si on la recrute.

Critère "informaticien": il faut prendre Jean ou Marie ou Paul : J+M+P=1
critère "matheux" il faut prendre Paul ou Colette : P+C=1
critère "commercial" il faut prendre Paul ou Marie P+M=1
critère "anglais" il faut prendre Jean ou Colette J+C=1

comme il faut répondre aux quatre critères, il faut avoir :

J+P+M= P+C=P+M=J+C=1
ou encore f=(J+P+M)(P+C)(P+M)(J+C)=1
pour que cette fonction f soit égale à 1 il suffit qu'un de ses monôme soit égale à 1

développons donc:

f=(P+CM)(J+PC+MC)= PJ+PC+PMC+JCM+PCM+CM= PJ+PC+CM *donc trois choix optimaux: Pierre et Jean (PJ), Pierre et Colette(PC), Colette et Marie(CM), les autres choix étant redondants (par exemple JMC multiple de MC) .*

Application

Le choix des monômes relève de cette méthode il faut satisfaire aux critères "chaque point doit être couvert", chacun des monômes maximaux couvrant un ou plusieurs points.

Exemple

$$f = \underset{a}{\bar{x}yz} + \underset{b}{x\bar{y}z} + \underset{c}{xy\bar{z}} + \underset{d}{x\bar{y}\bar{z}} + \underset{e}{\bar{x}y\bar{z}} + \underset{f}{\bar{x}\bar{y}z}$$

$$= \underset{A}{x\bar{y}} + \underset{B}{\bar{x}y} + \underset{C}{y\bar{z}} + \underset{D}{\bar{y}z} + \underset{E}{x\bar{z}} + \underset{F}{\bar{x}z}$$

appelons a,b,c,d,e,f les monomes initiaux et A,B,C,D,E,F les variables associées aux monômes maximaux. On raisonnera comme ci dessus:

pour couvrir a= $\bar{x}yz$, il faut prendre B ou F ($\bar{x}y$ ou $\bar{x}z$) donc B+F=1
pour couvrir b= $x\bar{y}z$, il faut prendre A ou D ($x\bar{y}$ ou $\bar{y}z$) donc A+D=1
etc...
on obtient : (B+F)(A+D)(E+C)(A+E)(B+C)(F+D)=1

on réarrange: (B+F)(B+C)(A+D)(F+D)(E+C)(E+A)
= (B+FC)(D+AF)(E+AC)

et en développant, on trouve **toutes** les combinaisons de monômes donnant une forme non redondante. Mais plutôt que de développer bêtement pour trouver toutes les formes, ce qui n'a pas d'intérêt (il suffit d'en trouver une!), on remarque que le degré du polynôme développé étant certainement supérieur à trois, un de ses monômes de plus bas degré est BDE (ce n'est pas le seul, il y a par exemple AFC) donc **une** solution minimale est

$$f= \bar{x}y + \bar{y}z + \bar{z}x$$

CHAPITRE 3
REALISATION DES FONCTIONS COMBINATOIRES

3.1. Technologie de fabrication

Réaliser une fonction revient à câbler des opérateurs physiques. Ceci peut se faire dans différentes technologies

3.1.1. technologie à composants discrets:

On trouve sur le marché des composants électroniques correspondant aux fonctions les plus usuelles (NOR, NAND, éventuellement ET, OU, NON). Ces composants seront disposés sur une "plaque" et reliés entre eux par des connexions. Ces connexions peuvent être faites de diverses manières:

3.1.1.1. à l'aide de **fils** soit soudés à la main, soit réalisés par "wrapping" (le fil est entortillé par une machines aux broches du composant. Cette deuxième technique est beaucoup plus fiable.

3.1.1.2. par la technique des circuits imprimés: la plaque comporte sur ses deux faces des connexions "imprimées" (en fait obtenues par photogravure). Il suffit de souder les broches du composant aux connections . (soudure manuelle ou "à la vague", en trempant le tout dans un bain de soudure.)

3.1.2. technologie des circuits intégrés.

Les composants ainsi que les connexions sont réalisés directement sur une "puce" de silicium par des techniques de photogravure. Cette technique permet un miniaturisation extrême des composants; on arrive actuellement à des puces comportant un million de transistors.

Cette technique est évidemment préférable à la première, mais n'est applicable que pour des circuits définis **une fois pour toutes** (pas question de faire des retouches!) et de grande diffusion (plusieurs milliers !).

3.1.3. technologies intermédiaires.

On trouve sur le marché des circuits intégrés avec une organisation prédéfinie permettant de réaliser des fonctions quelconques, par une sorte de spécialisation..

Dans les trois cas, la diminution du coût du circuit passe par une minimisation appropriée de la fonction à réaliser.

3.2. Portes logiques

Les composants disponibles dans la technique discrète se présentent comme un boitier réalisant une ou plusieurs fonctions simples appelées "portes" par exemple

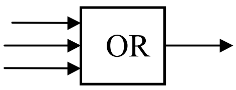
le ET de k variables (k fixé)
le OU de k variables (idem)

plus fréquemment:

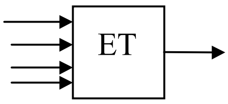
le NOR de k variables (idem)
le NAND de k variables (idem)
(on utilise suivant les technologie, soit l'un soit l'autre, mais en général pas les deux)

plus rarement le OU EXCLUSIF..

exemple



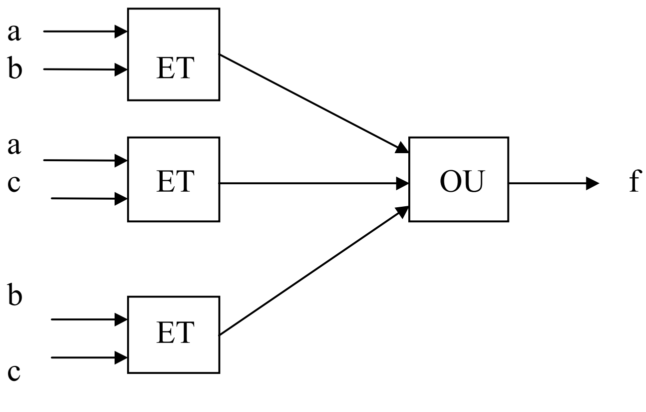
porte OU a trois entrées



porte ET à 4 entrées

On notera que seules les entrées et sorties logiques sont dessinées. (les alimentations en énergie ne figurent pas).

Ces portes peuvent être connectées pour réaliser des fonctions
par exemple la fonction f=ab+bc+ca est réalisée par:



Remarques:

Si le problème semble simple et ET et OU il est plus complexe en NOR ou en NAND, surtout du fait de la limitation du nombre d'entrées: exemple, réaliser la fonction précédente avec des NOR à 2 entrées.

3.3. Méthodes de calcul.

3.3.1. méthode algébrique, cas du NOR à k entrées.

Soit f la fonction à réaliser

a) on calcule $\overline{f}=\Sigma\, m_i$

b) on partage cette somme en k sommes partielles

$\overline{f}= \Sigma 1 + \Sigma 2 + \;....\; +\Sigma k$ (on a bien f = NOR(Σ1, Σ2,... Σk))

c) pour chaque somme partielle fi= Σi on recommence le calcul en (a) jusqu'à trouver des variables

exemple f= ab+bc+ca à réaliser en NOR a deux entrées au plus

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

on coupe en deux: $f1= \overline{a}\,\overline{b}$ et $f2= \overline{b}\,\overline{c} + \overline{c}\,\overline{a}$

on complémente chacune:

$\overline{f1} = a+b$ décomposée en f11= a et f12=b

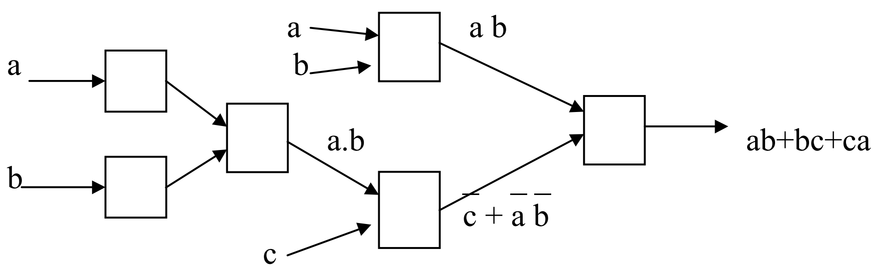
$\overline{f2} = c + a\,b$

décomposée en

f21= a et f22 = a b

$\overline{f22} = \overline{a} + \overline{b}$ décomposée en $f221= \overline{a}$ et $f222= \overline{b}$

on obtient donc le circuit suivant:



On notera que la méthode nécessite de nombreuses complémentations et peut devenir fastidieuse, d'où la méthode qui suit:

3.3.2. méthode graphique

3.2.2.1 cas des fonctions incomplètement spécifiées.

On a vu que le complément d'une fonction incomplète f={g,h} est $\overline{f}$={h,g}, le complément est donc facile à calculer.

Soit alors f={g=Σmi , h=Σmj} son complément est $\overline{f}$={h=Σmj , g=Σmi} cette fonction est à partager en k parties f1, f2, fk (le plus incomplètes possibles pour améliorer la "probabilité qu'elles soient simples à réaliser) telles que

$$f1 + f2 + ... + fk = \overline{f}$$

c'est à dire telles que

$$g1 + g2 + ... + gk = h \text{ et } \quad h1 = h2 = . . . = hk = g$$

(on segmente la **borne inférieur** et on conserve la **borne supérieur**).
En effet la somme de ces fonctions est (CF chapitre 1 , annexe)

{(g1+g2+...+gk), (h1.h2....hk)}={h,g}=f

et par ailleurs chacune des fonctions est non spécifiée au maximum, et donc en principe facile à réaliser.

Théorème La méthode converge: chaque fonction obtenue par fractionnement est **strictement** moins spécifiée que la fonction initiale (dans une des bornes, il y a strictement moins de monômes). Donc au bout d'un nombre **<u>fini</u>** d'itération, on obtiendra des fonctions fi ={gi,hi} gi et hi n'ayant qu'un monôme canonique et pour lesquelles une **variable** est solution.

Représentation graphique: soit f={g=Σmi , h=Σmj}

on considère un tableau rectangulaire dont les colonnes sont indexées par les monômes mi et les lignes par les monômes mj. Dans la case de coordonnée [i,j] on écrit les lettres qui différencient les monômes mi et mj (sous la forme où elles apparaissent dans mi)

exemple: f={a b + b c + c a , $\overline{a}\,\overline{b} + \overline{b}\,\overline{c} + \overline{c}\,\overline{a}$ }

on construit le tableau suivant, en remarquant que complémenter la fonction revient à **transposer** le tableau (voir aussi la remarque ci dessous). Les monômes de la borne inférieure du complément sont donc les mj.

Donc fractionner la borne inférieure de $\overline{f}$ revient à partager le tableau <u>horizontalement.</u>

mi / mj	a b	b c	c a
$\bar{a}\,\bar{b}$	a,b	b	c
$\bar{b}\,\bar{c}$	b	b,c	c
$\bar{c}\,\bar{a}$	a	c	a,c

Le fractionnement de ce tableau matérialisé par le trait noir correspond au partage en deux fonctions:

$f1=\{\ \bar{a}\ \bar{b}\ , a\,b + b\,c + c\,a\}$ et $f2=\{\ \bar{b}\ \bar{c} + \bar{c}\ \bar{a}\ , a\,b + b\,c + c\,a\ \}$

et on a donc NOR(f1,f2)=f

on pourrait itérer les fractionnement en séparant en deux tableaux, mais il est plus simple de continue sur le même: chacune des deux parties sera complémentée puis fractionnée, cette fois donc verticalement (puisque la complémentation se fait en transposant le tableau).

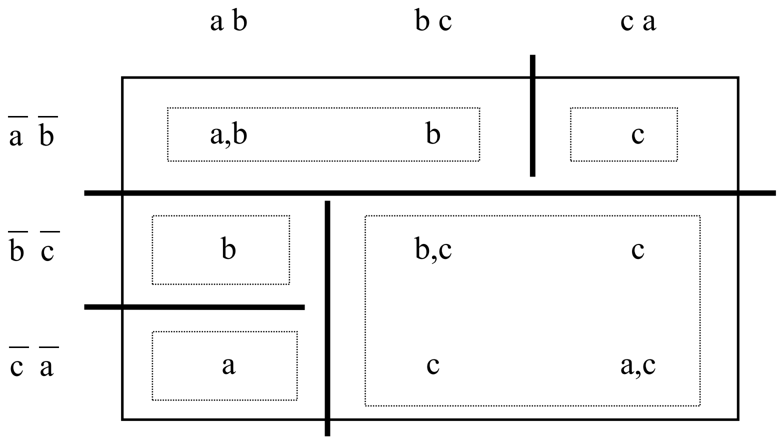
	a b	b c	c a
$\bar{a}\,\bar{b}$	a,b	b	a
$\bar{b}\,\bar{c}$	b	b,c	c
$\bar{c}\,\bar{a}$	a	c	a,c

interprétons chaque quart:

la fonction f11 du quart supérieur gauche a pour borne inférieure ab+bc et pour complément de sa borne supérieure $\bar{a}\ \bar{b}$. La fonction b est une solution (puisque
$a\,b + b\,c < b < a + b$) ceci est matérialisé par la <u>présence de la lettre b dans chaque case.</u>

De même le quart supérieur droit a pour solution a (car $a\,c < a < a + c$)

De même le quart inférieur droit a pour solution c matérialisé par le fait que c est présent dans chaque case (on a bien $b\,c + c\,a < c < c + a\,b$).
Par contre le quart inférieur gauche doit encore être fractionné, (cette fois horizontalement)



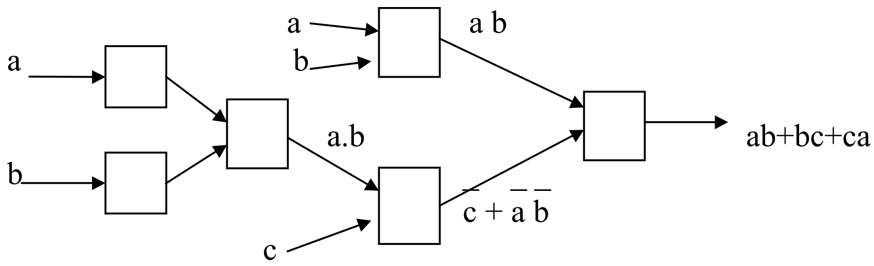
les deux fonctions incomplètes obtenues ont cette fois pour solution respectivement $\bar{b}$ et $\bar{c}$ en haut et en bas.

En définitive:

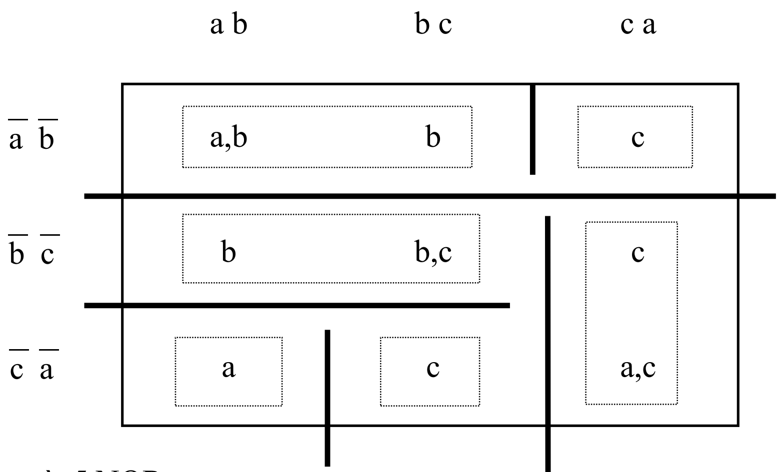
chaque rectangle pointillé représente une entrée du réseau d'opérateurs,

chaque trait gras représente une porte NOR

le résultat obtenu est le même que par la méthode précédente:



Tout "l'art" du découpage va alors consister à trouver des grandes zones "résolues" par une variables. En effet des mauvaises solutions existent, par exemple:



solution à 5 NOR

remarque. Nous avons dit plus haut que complémenter la fonction revient à transposer le tableau. Ceci est incomplet: considérons par exemple la fonction f1 ci-dessus:

f1={ $\bar{a}\ \bar{b}$, a b + b c + c a} son tableau est:

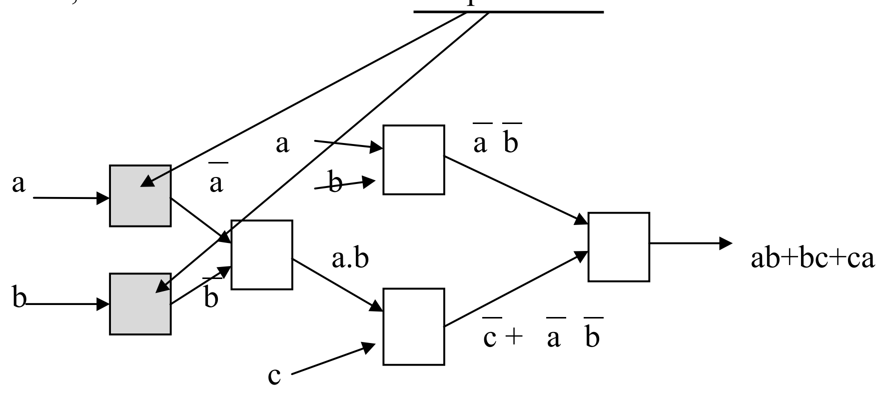
	$\bar{a}\ \bar{b}$
ab	$\bar{a}\ \bar{b}$
bc	$\bar{b}$
ca	$\bar{a}$

Ce n'est donc pas exactement le transposé du haut du tableau précédent, il faut en plus <u>complémenter</u> les variables dans les cases Cependant on peut simplement se rappeler qu'à chaque complémen- -tation, il faut inverser lettres complémentées et non complémentées). si donc on est par exemple à k° complémentation,, si k est pair, on ne change rien, si k est impair , on complémente.

Ainsi .dans le tableau suivant:

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

ces cases doivent être complémentées puisqu'on a fait 3 fractionnement successif de f pour les obtenir; leurs solutions sont donc complémentées. Le réseau final est donc



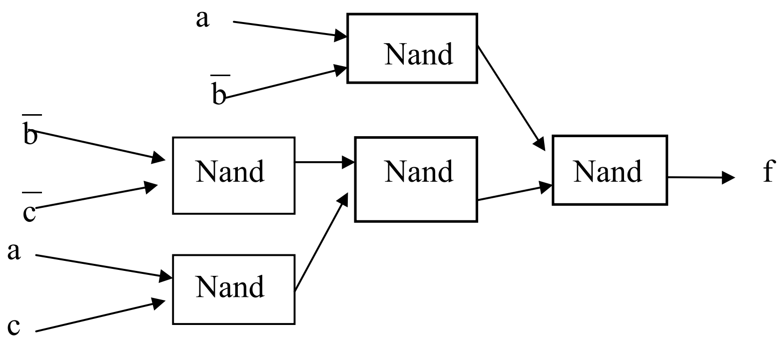
Cas des portes "NAND":

Les opérations seront exactement les mêmes, sauf qu'on commence par fractionner le tableau **verticalement.**

Exemple: $f = a\,\bar{b} + b\,\bar{c} + c\,\bar{a}$ $\qquad f = a\,b\,c + \bar{a}\,\bar{b}\,\bar{c}$

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

on obtient donc



<u>portes à k entrées</u>

La méthode serait la même pour des fractionnement en k sous fonctions, et il est clair qu'on arrivera plus vite à la solution avec k grand.

Exemple k=3 dans l'exemple précédent:

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

le circuit comporte 4 nand

CHAPITRE 4

FONCTIONS A MEMOIRE OU SEQUENTIELLES

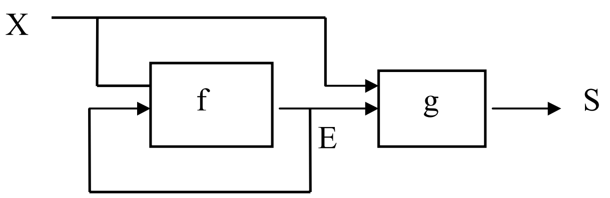
Au chapitre 3 on a vu le cas de fonctions dépendant uniquement d'un ensemble de variables:

$$y=f(x1,x2,...xn)$$

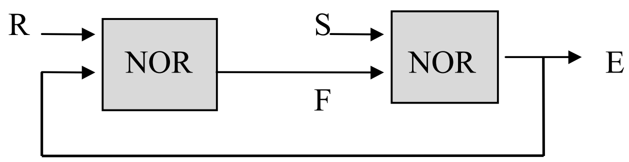
En fait on rencontrera souvent des fonctions gardant une mémoire du passé des évolutions des entrées. Ces fonctions ont un **état interne E**, et on a deux fonctions en fait:

une fonction d'état:	E=f(E,x1,x2,.....xn)
une fonction de sortie:	S=g(E,x1,x2,...xn)

:
ce qui correspond à un schéma bouclé



4.1. Etude d'un exemple:



Ce circuit est caractérisé par l' équation: $\boxed{E = \overline{S}\,(E + R)}$

Il comporte <u>une mémoire du passé,</u> en effet:

a) considérons la situation R=0 S=1 — entrée X= {0,1}
   puisque S=1 quelque soit F, on a E=0;
   puisque E=0 et R=0 on a F=1 .
b) supposons que S devienne nul, R inchangé. — Entrée X={0,0}
   Puisque F=1, E reste nul, donc F reste 1
   la situation est dite "stable"
c) supposons maintenant que c'est R qui passe à 1 — Entrée X={1,0}
   puisque R=0 et E=0, on a F=1 donc E passe à 1,
   l'état du système a changé.
d) si enfin R repasse à 0 — entrée X={0,0}

puisque E=1 on a F=0, puisque S=0 et F=0 on a E=1

la situation reste stable

Mais on remarque que pour la <u>même valeur de l'entrée X={0,0} on peut avoir deux états différents :</u>

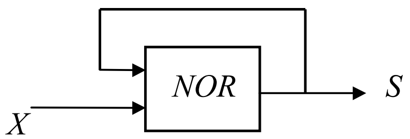
si avant cette entrée {00} on avait {01} on a E=0

si avant cette valeur {0,0} on avait {1,0}, on a E=1

en bref E a <u>mémorisé</u> la dernière valeur de R

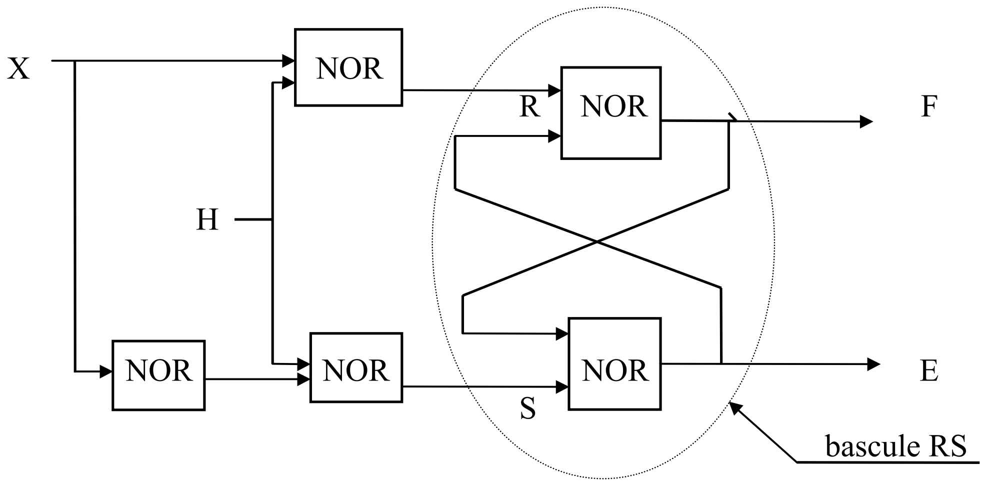
<u>Ce circuit appelé "bascule RS" est la base des éléments de mémoire de l'ordinateur</u>
(terminologie anglaise: bascule = "flipflop")

Remarque; *tout circuit "bouclé" n'est pas une fonction mémoire; exemple:*



on voit facilement que quant X=0, S va <u>osciller</u> entre les deux valeurs 0 et 1, sans atteindre un <u>état stable.</u>

4.2. Bascules D rajoutons à la bascule précédente deux entrées supplémentaires H et X



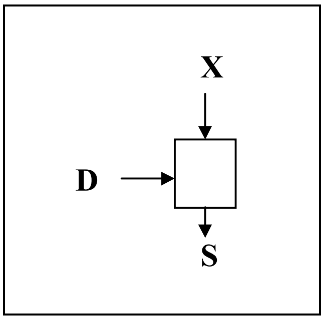
On constate que quand H=1 on a R=S=0 donc l'état de mémorisation:
quand H=0 on a R=X et S=X donc la bascule va mémoriser la dernière valeur de X

➔ mode de mémorisation:

a) enregistrer un 1: faire X=1 **puis** H=0: la bascule enregistre l'entrée, puis H=1: la bascule mémorise cette valeur 1 pendant tout le temps que H=1

b) de même enregistrer un 0: faire x=0 puis H=0, (enregistre X) puis H=1 (mémorisation de X)

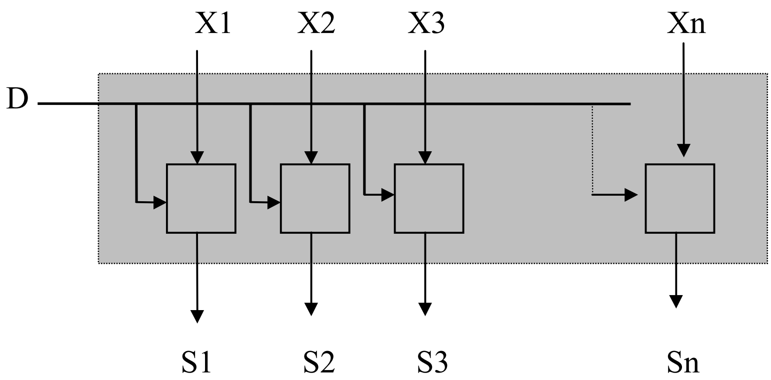
Cette bascule mémorise, pendant tout le temps que H=1, la valeur qu'avait X quand H=0.
Représentation simplifiée:



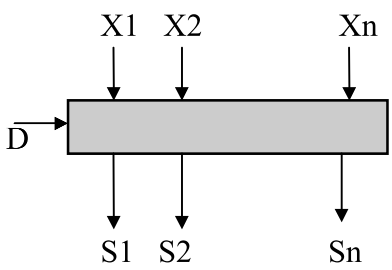
On a donc un moyen d'enregistrer un bit .

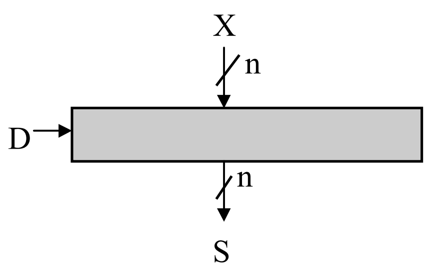
4.3. Registres

La répétition de ce circuit va permettre de mémoriser des **mots** de n bits.



Ce circuit à n bascules D est appelé **"registre"** et sera représenté ci dessous sous une des deux formes simplifiées suivantes:



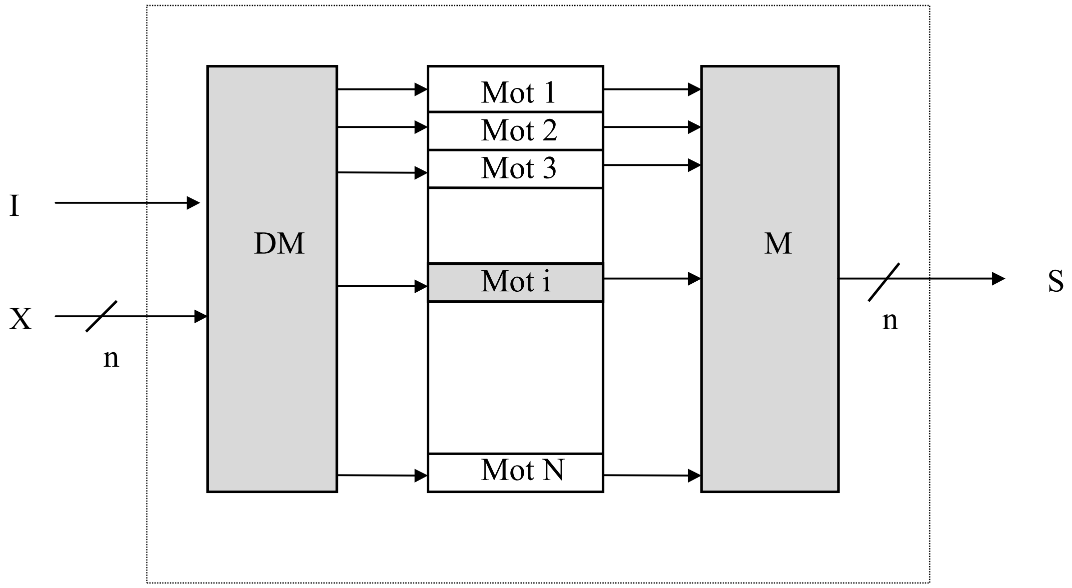


1

Ces registres peuvent apparaître comme un des constituants de base de l'unité centrale de l'ordinateur, ou comme un des constituant de la mémoire.

4.4. Mémoire RAM (Random Access Memory)

N tels registres peuvent être groupés dans un seul composant intégré (par exemple N=1 million)**. Mais il n'est évidemment pas possible d'avoir N.n connexions entre ces registres et l'extérieur.** La structure de la mémoire sera alors la suivante;



Mémoire de N registres

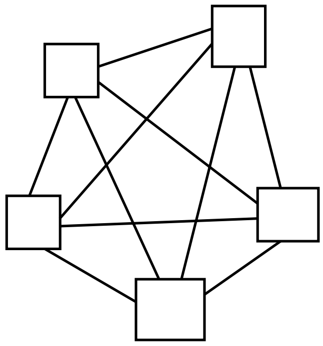
Où DM est appelé Démultiplexeur: capable à partir d'une adresse I **d'aiguiller** l'information en entrée X vers le registre n° I.
M est un circuit appelé Multiplexeur capable de concentrer sur une seule sortie S les sorties des divers registres

En général, les mots sont de 8 bits (octets) Si on utilise en pratique des mots de 16, 32 ou 64 bits , ils sont formés d'octets consécutifs.

> ***NOTA il n'existe pas de convention unique: certains constructeurs définissent par exemple un mot de 32 bits comme 4 octets rangés en mémoire dans l'ordre "poids forts en tête", d'autre "poids faibles en tête", ce qui peut poser des problèmes de portabilité!***

4.4.1. le démultiplexeur Notion de BUS

Quand on veut faire communiquer K éléments quelconques, il n'est pas judicieux de réaliser les k(k-1)/2 connexions (schéma a). On préfère faire passer l'information par une sorte d'autoroute ou "bus" (schéma b)



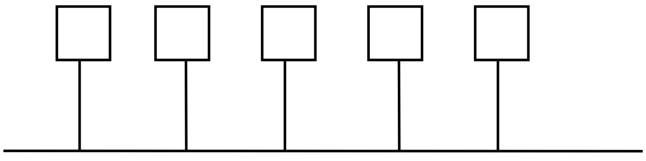
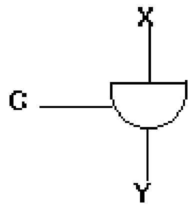


schéma (a) "toile d'araignée" schéma (b) à bus

Dans le schéma à bus il faut évidemment prévoir **un mécanisme d'<u>isolement</u>** des n-2 éléments quand les éléments i et j communiquent. Ceci se fait au moyen de "portes 3 états"

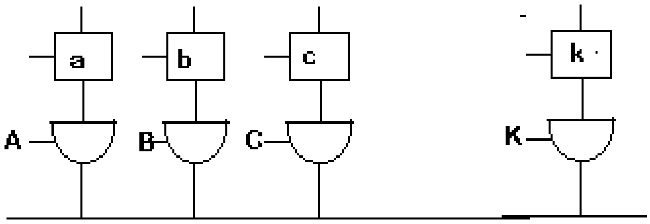


cette porte a une propriété similaire à un interrupteur:

Quand C=0 X et Y sont **isolés (état Y indéterminé)**

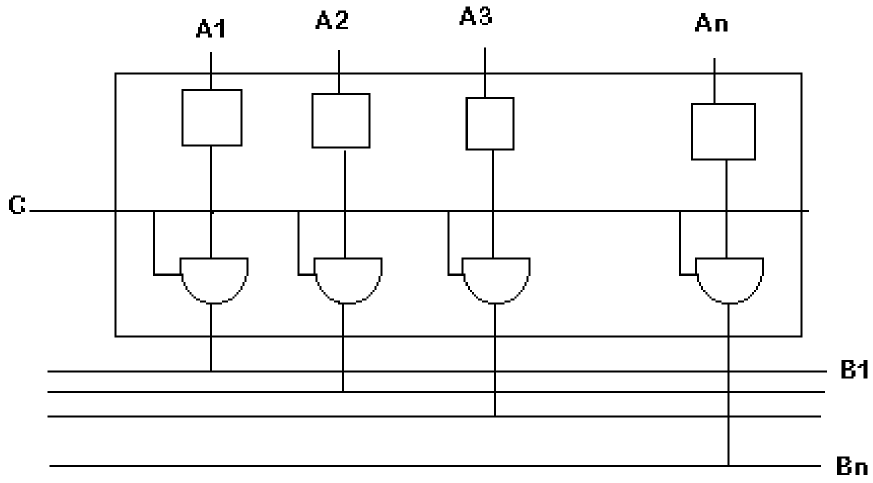
Quand C=1 Y est égal à X **(états 0 ou 1)**

On peut alors "connecter" à la demande une sortie de bascule D à un bus:

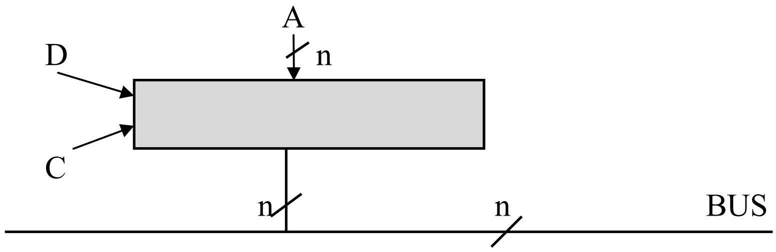


Ainsi en activant la seule commande A par exemple, on connecte la sortie de la bascule a au bus, toutes les autres étant isolées.

On peut évidemment répéter ce schéma pour des registres à n bits:

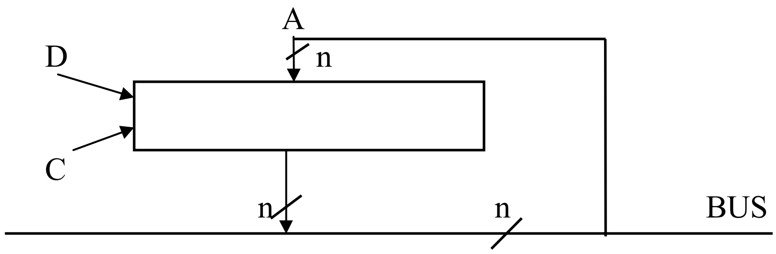


Une même commande C entraîne la connexion des n sorties de bascules vers n bus En représentation simplifiée ceci se notera:

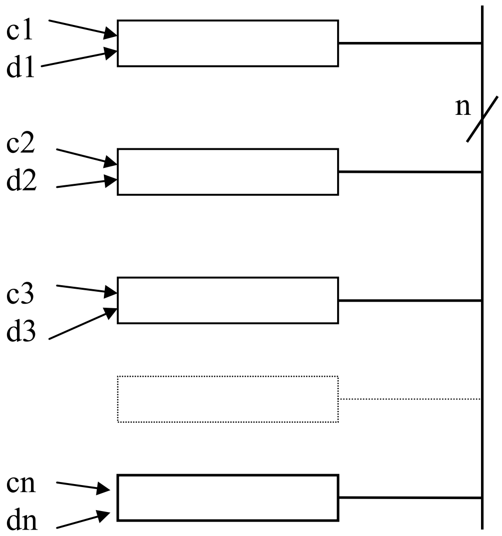


4.4.2. démultiplexage:

La commande D du registre permettant l'entrée dans le registre, on peut également connecter A au bus: D=1 et C=0 c'est une entrée du bus vers le registre, D=0 et C=1 c'est une sortie.

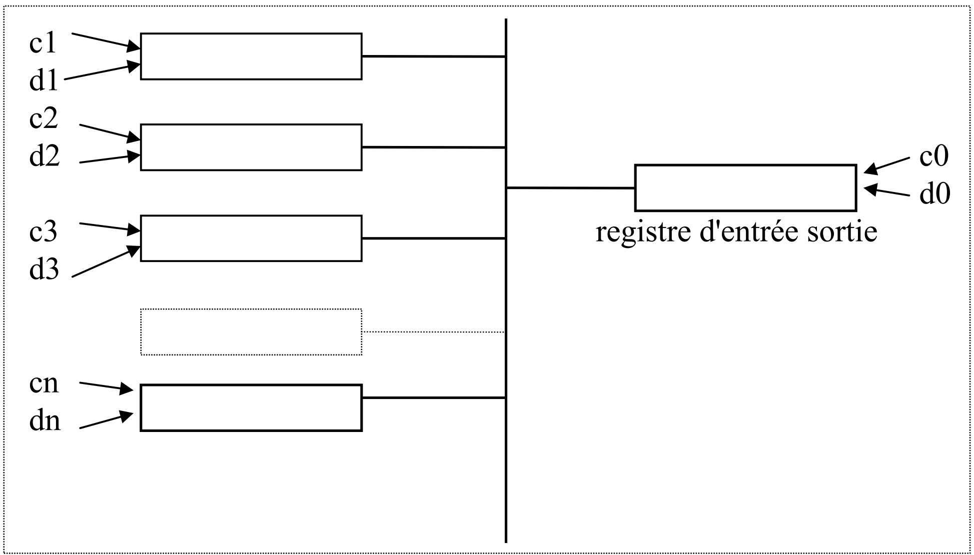


Si plusieurs registres sont connectés au bus, on notera plus simplement:



Ainsi pour transférer le contenu de R1 vers R3 il suffirait de faire C1=D3=1, tous les autres étant nuls.

Nota: en général la mémoire comportera un registre spécial dit "registre d'entrée sortie (seul communicant avec l'extérieur de la mémoire



4.4.3 circuit de sélection.

Ce dispositif doit être complété par un circuit permettant de générer des commandes Ci et Di en fonction de :

a) l'adresse i d'un registre Ri de la mémoire

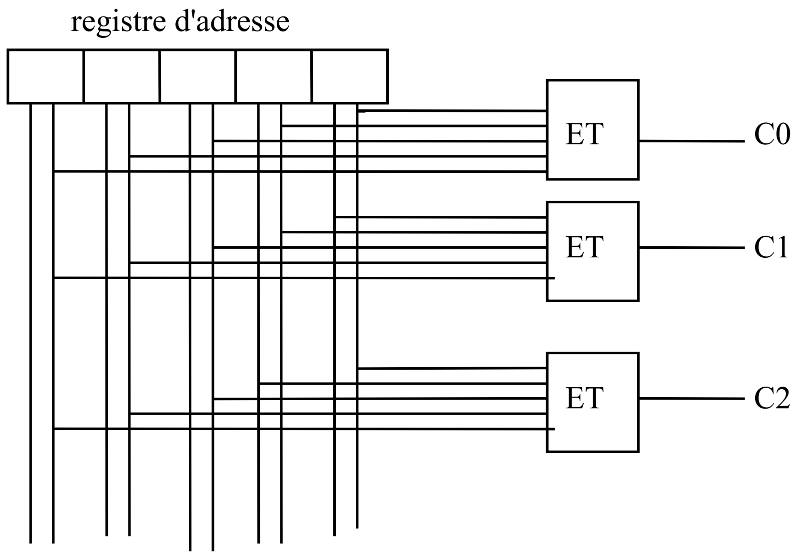
b) du sens registre d'entrée sortie➔ Ri (écriture en mémoire) ou du sens Ri➔ registre d'entrée sortie (lecture de la mémoire)

dans le cas de l'écriture, ce circuit devra générer Di=1 et C0=1
dans le cas de la lecture, ce circuit devra générer Ci=1 et D0=1

Il s'agit donc à partir d'une information i contenue dans un registre dit "d'adresse" à n bits de générer 2^n commandes, de la manière suivante

i= 00000 en binaire → c0=1
i= 00001 → c1=1
i= 00010 → c2=1
etc...

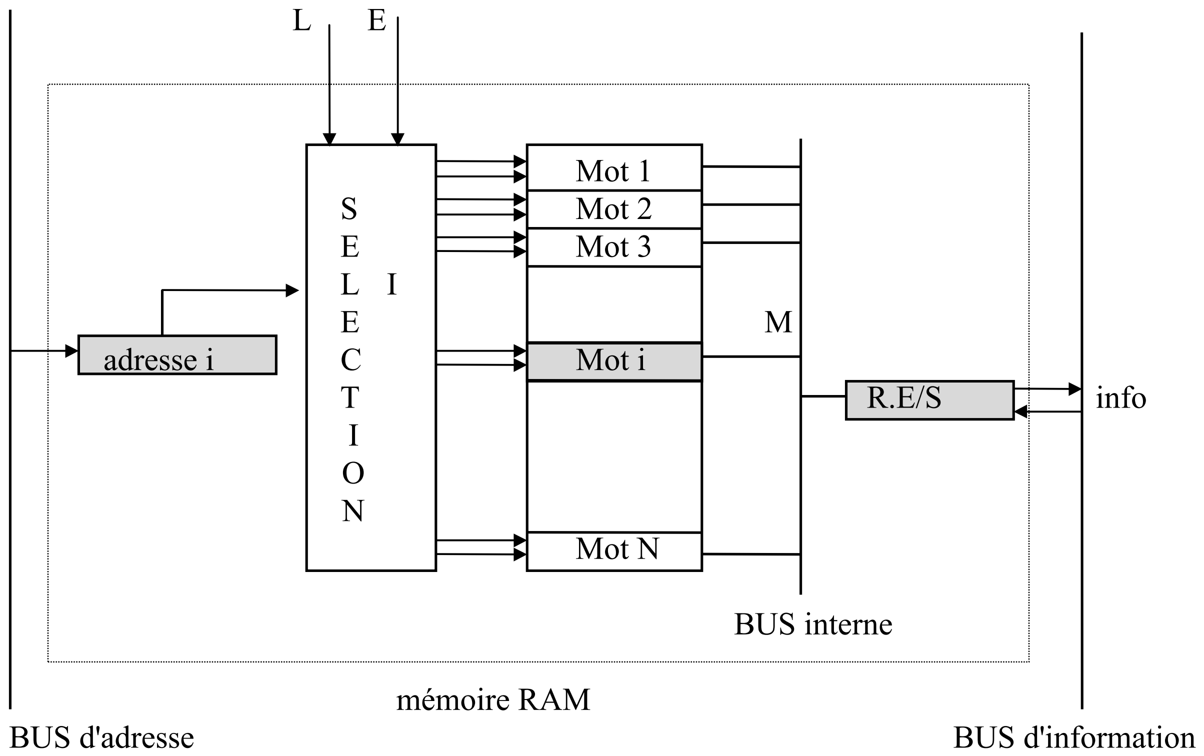
En remarquant que dans une bascule RS les sories des deux NOR sont le complément l'une de l'autre, on a alors le schéma suivant:



avec 2^n portes, on fait donc les 2^n commandes Ci; Il en est de même pour les Di

Nota: il existe des circuits plus optimaux qu'il est inutile de décrire ici.

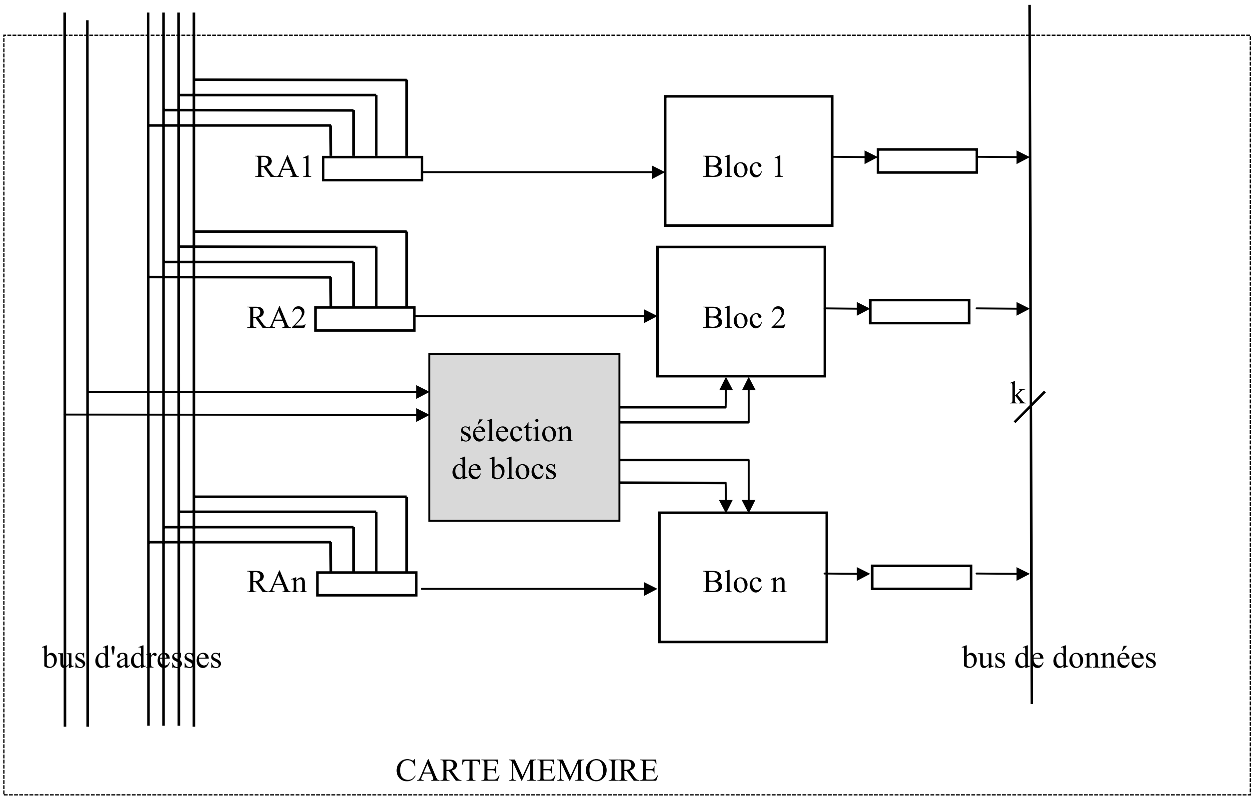
Schéma définitif de la mémoire



Les entrées L (lecture) et E (écriture) étant des commande "d'opérations": L=1 et E=0 commandent la génération des Di, L=0 et E=1 commandent la génération des Ci.; si L=D=0 , la mémoire est inerte (on dit qu'elle n'est pas **sélectionnée**)

4.4.4. Cas de plusieurs mémoires.

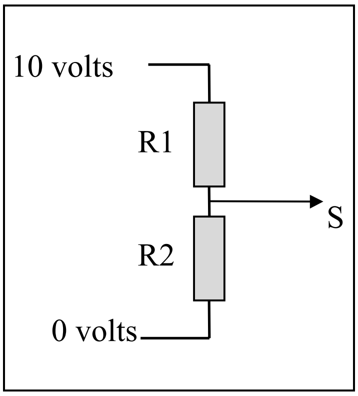
Il arrive qu'on ait besoin de plus de mémoire qu'il y en a dans les composants disponibles. On est amené à mettre plusieurs blocs de mémoire sur un même support, reliés entre eux et vers l'extérieur par un "bus de données" et un "bus d'adresse". Un circuit de "sélection de mémoire" utilise les bits poids forts de l'adresse pour n'activer que les commandes Li et Ei du bloc n°i sélectionné:



On notera que la carte mémoire peut être achetée <u>incomplète</u>, certains blocs manquants étant achetés ultérieurement.

4.4.6 Mémoire ROM (Read only memory)

Les mémoires RAM ont le défaut d'être **<u>évanescentes</u>** (elles perdent leur information en cas de coupure de courant) Pour certaines applications, on leur préfère des mémoires statiques, enregistrées une fois pour toutes. Un exemple de telle technique pour la réalisation **d'un bit** serait la suivante:



à la fabrication, on "grille" (par laser) une des deux résistances R1 ou R2
Si R1 est grillée, la sortie est V=10 volts (bit 1 par définition)
Si R2 est grillée, la sortie est V=0 volts (bit 0 par définition)

La réalisation d'un bit est très simple, bien que la complexité des circuits d'adressage soit exactement la même
avantage: rigoureusement insensible aux coupures de courant
inconvénient: non réinscriptible

Nota: il existe des mémoires intermédiaires entre la RAM et la ROM: les mémoires PROM (programmable read only mémory) qui peuvent être difficilement réécrites et facilement lues

4.5. Circuits séquentiels quelconques

Outre les mémoires, on rencontre des circuits réalisant des fonctions quelconques de la forme

équation d'état: $Y=F(X,Y)$
équation de sortie: $Z=G(X,Y)$

ou y représente l'état interne. Sil circuit comporte plus de deux états, Y n'est pas booléen

exemple de fonction séquentielle:
pour démarrer le moteur (Z) appuyer deux fois sur le bouton marche (X)
développement:

Y= état initial: on n'a appuyé sur rien, Z=0
Y= état intermédiaire on appuie une fois: Z=0
Y= état intermédiaire 2 on relache Z=0
Y= état final on a appuyé deux fois: Z=1

on peut représenter ceci par un "tableau d'états" donnant l'évolution des états

	ENTREES		
ETATS	**relaché**	**appuyé**	***SORTIE***
initial	initial	intermédiaire1	arrêt
intermédiaire 1	intermédiaire 2	intermédiaire 1	arrêt
intermédiaire 2	intermédiaire 2	final	marche
final	final	final	marche

Ce tableau qui se lit ainsi: (1[ere] ligne) quant on est dans **l'état initial**, si **X="appuyé"** on passe à l'état intermédiaire 1, sinon on reste dans l'état initial; dans les deux cas la sortie est nulle etc...

4.5.1. Méthode de Huffmann

on peut faire un "codage" des états, des entrées et des sorties: pour obtenir un tableau de Karnaugh

entrées: une variable booléenne x: relâché ➔x=0 appuyé➔x=1

états: deux variables booléennes y1 et y2:

- initial: ➔ y1=0, y2=0
- intermédiaire 1: ➔ y1=1, y2=0
- intermédiaire 2: ➔ y1=1, y2=1
- final ➔ y1=0, y2=1

sortie: une variable booléenne z: marche➔z=1 arrêt ➔z=0

on obtient:

	0	1	Z
0 0	0 0	0 1	0
0 1	1 1	0 1	0
1 1	1 1	1 0	1
1 0	1 0	1 0	1

donc, en fait, trois diagrammes de Karnaugh:

Pour y1:

	0	1
0 0	0	0
0 1	1	0
1 1	1	1
1 0	1	1

donc $y1 = \overline{x}\, y2 + y1$

pour y2:

	0	1
0 0	0	1
0 1	1	1
1 1	1	0
1 0	0	0

donc $y2 = \overline{x}\, y2 + x\, \overline{y1}$

pour z : fonction triviale: z=y1

On a donc un moyen de réaliser la fonction séquentielle à l'aide de circuits logiques classiques. Mais la méthode est lourde (le coût des fonctions dépend très fortement du codage des états) et la réalisation en composants discrets n'est guère optimale.

4.5.2. micro-programmation

La méthode usuellement employée fait appel à un **enregistrement en mémoire** du tableau d'états. Reprenons l'exemple précédent

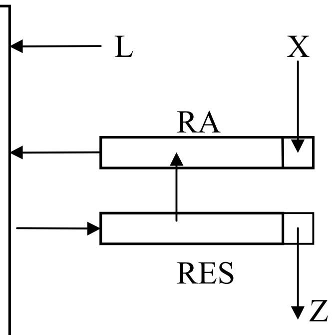
y1,y2 \ x	0	1	Z
0 0	0 0	0 1	0
0 1	1 1	0 1	0
1 1	1 1	1 0	1
1 0	1 0	1 0	1

tableau qui peut être réorganisé en:

y1,y2,x	état suivant	sortie
0 0 0	0 0	0
0 0 1	0 1	0
0 1 0	1 1	0
0 1 1	0 1	0
1 0 0	1 0	1
1 0 1	1 0	1
1 1 0	1 1	1
1 1 1	1 0	1

Le contenu du tableau peut alors être rangé dans une mémoire. Chaque ligne est un "état global" (état interne et entrées). On note que l'état global est également l'adresse de la ligne en mémoire; cette ligne contient en partie gauche l'état suivant, en parti droite la sortie.

adresse	état suivant	sortie
0 0 0	0 0	0
0 0 1	0 1	0
0 1 0	1 1	0
0 1 1	0 1	0
1 0 0	1 0	1
1 0 1	1 0	1
1 1 0	1 1	1
1 1 1	1 0	1



La sortie du registre RES est "cablée" sur le registre d'adresse partie poids forts, l'entrée X sur RA poids faible. Ainsi si on est dans l'état i, l'adresse 2.i contient ce qu'il faut faire si X=0, et l'adresse 2.i+1 ce qu'il faut faire si X=1

exemple i= 1 (0 1):
la ligne 2 (adresse 0 1 0) contient l'état suivant (1 1) quand x=0
la ligne 3 (adresse 0 1 1) contient l'état suivant (0 1) quand x=1

La commande L est activée périodiquement par une horloge haute fréquence, l'état et les sorties sont donc remis à jour périodiquement.

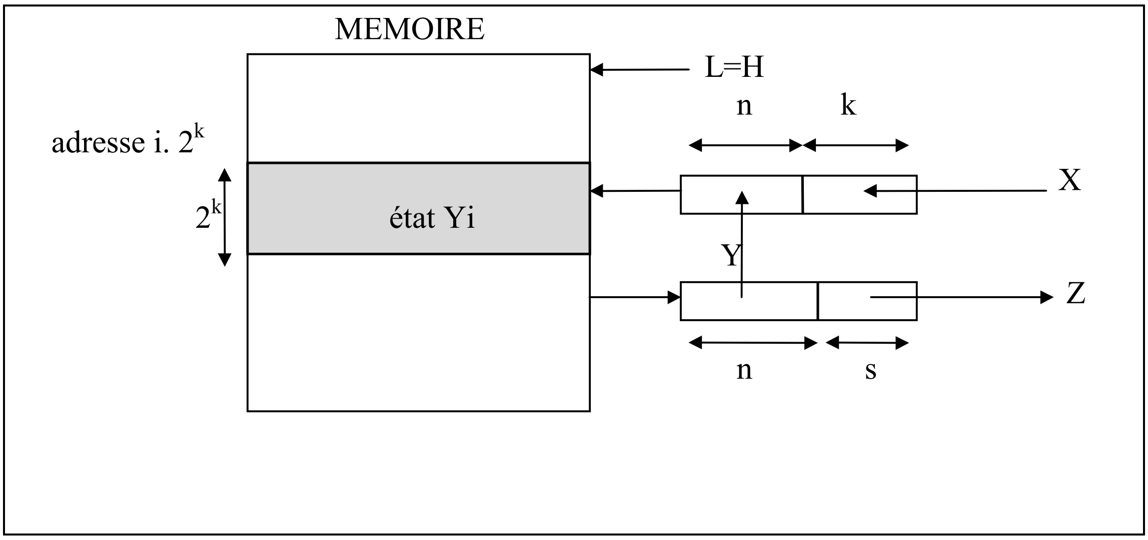
Généralisation

Un système séquentiel quelconque décrit par son tableau d'état peut être réalisé sous cette forme:

S'il y a N états E1,E2,.. EN il faut n variables de codage y1,y2,...yn, avec n=LOG2(N)

S'il y a k entrées X1, X2,..... Xk, chaque ligne du tableau se subdivise en 2^k lignes

S'il y a s sorties, dans chaque ligne, il, y outre les n valeurs de l'état suivant, les s valeurs des sorties



Le fonctionnement du système est le même que précédemment:

L'état global {Y,X} contenu dans RA forme l'adresse de la ligne (on dit la "micro_instruction"). Cette micro instruction va être transférée à chaque top de l'horloge H dans le registre RES. La partie adresse de RES sera "recyclée" dans RA, la partie sortie utilisée pour une commande.
Ainsi avec un seul composant mémoire, on peut réaliser un automate complet.

Définition :

Cette technique est appelée "micro-programmation" (à ne pas confondre avec "programmation de micro" !).

L'ensemble des micro-instructions forme le micro-programme.

C'est cette technique qui permet de réaliser le cœur (appelé "séquenceur") de l'unité centrale de l'ordinateur.

Nota:

Diverses techniques, (qu'il n'est pas indispensable de développer ici) permettent une optimisation de la mémoire. En particulier, il existe des langages de programmation permettant de générer automatiquement le micro-programme.